控制与决策
Control and Decision

利用浓缩布尔矩阵重排技术求所有约简

冯琴荣， 胡競丹

引用本文:
冯琴荣, 胡丹. 利用浓缩布尔矩阵重排技术求所有约简[J]. 控制与决策, 2021, 36(5): 1157−1164.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.1307

利用浓缩布尔矩阵重排技术求所有约简

冯琴荣†, 胡競丹

(山西师范大学 数学与计算机科学学院，山西 临汾 041000)

摘 要: 针对当前求所有约简的算法其结果中存在较多冗余(约简的超集)的现状,对矩阵重排技术进行改进,设计一个多次运用改进矩阵重排技术求所有约简的算法,从而能够更高效地在属性集的幂集上进行剪枝,删除所有非约简和大部分超约简,同时给出一种快速判断属性子集是否为超约简的方法. 与已有方法相比,所提出算法结果中超约简的数量更少,算法效率更高.

关键词: 粗糙集；辨识矩阵；浓缩辨识矩阵；浓缩布尔矩阵；矩阵重排技术；属性约简

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.1307 **开放科学(资源服务)标识码(OSID):**

引用格式: 冯琴荣,胡競丹. 利用浓缩布尔矩阵重排技术求所有约简[J]. 控制与决策, 2021, 36(5): 1157-1164.

Finding all reductions through the technique of rearranging concentration Boolean matrix

FENG Qin-rong†, HU Jing-dan

(School of Mathematics and Computer Science，Shanxi Normal University，Linfen 041000，China)

Abstract: Due to more super-reduction output by most existing algorithms in finding all reductions, the technology of matrix rearrangement is improved in this paper, and an efficient algorithm for finding all reductions is designed, which uses improved matrix rearrangement many times to prune the power set of the attribute set more efficiently, and all of non-reduction and a majority of super-reduction are deleted. And a method for judging whether a subset of attributes is a super-reduction or not is also presented. Compared with existing algorithms, the proposed algorithm is more effective and can output fewer super-reduction.

Keywords: rough set；discernibility matrix；concentration discernibility matrix；concentration Boolean matrix；matrix rearrangement technology；attribute reduction

0 引 言

粗糙集理论是Pawlak[1]于1982年提出的处理不确定、不精确和不完备知识的数学理论,目前已成功运用于决策分析、模式识别、数据挖掘等领域. 属性约简是粗糙集理论研究的一个重要问题,它以保持知识库分类能力不变为前提,删除不重要或不相关的属性,使得在不丢失基本信息的基础上简化知识表示. 文献[2]证明了找到所有约简或最小约简是一个NP-hard问题,导致NP-hard的主要原因是属性的组合爆炸问题. 为了解决这一问题,有学者提出了利用启发式算法寻找最小属性约简[3-4],但算法在有些情况下并不能得到预期的结果,于是提出了求所有约简[5-3]的算法. 文献[5]提出了直接用辨识函数求所有约简的算法. 文献[6]提出了浓缩辨识矩阵的概念,并在简化的辨识矩阵上利用辨识函数求所有约简. 文献[7]提出了浓缩布尔矩阵概念,并提出一种新的用于直接生成辨识函数最小析取范式的算法. 文献[8]提出了一种基于辨识矩阵的约简树构造方法,以实现合取范式到析取范式的等价转换. 文献[9]充分利用合取运算和析取运算的吸收率,借助队列结构提出了一种面向辨识函数的范式转换算法. 文献[10]通过研究发现合取范式与析取范式之间存在的内在规律,提出了相应的范式增量转换算法求所有约简. 文献[11]将样本对与幂图相结合,提出了基于幂图的搜索算法,将样本对集的简化问题转化为幂图的搜索问题,从而得到所有约简. 文献[12]给出了浓缩布尔矩阵重排的方法,利用矩阵重排技术对属性的幂集进行剪枝,从中删去所有非约简和部分超约简,最终得到所有约简. 文献[13]提出了一种基于“间隙”消除和属性贡献度的剪枝技术得到所有约简. 以上

收稿日期: 2019-09-15；**修回日期:** 2019-12-19.
责任编委: 刘宝碇.
†通讯作者. E-mail: fengqr72@163.com.

文献在求所有约简时要么算法复杂度较高,要么所得结果中超约简(约简的超集)的数量较多.

本文通过在浓缩布尔矩阵上寻找规律,在文献[12]的基础上,对矩阵重排技术进行改进,多次利用改进矩阵重排技术,在找到所有约简的基础上使得冗余更少,即超约简数量更少. 由于本文算法所得结果仍有少数超约简,最后给出一种快速判断超约简的方法.

1 基本知识

约简的目标有很多种[14],本文选择在保持正域不变的情况下作约简.

定义1[6] 给定决策表$\mathrm{DT}=(U,C\bigcup D,V,f)$,辨识矩阵$M=\{M(x_m,x_n)\}$为

$$M(x_m,x_n)=\begin{cases}\{a\in C:f(x_m,a)\neq f(x_n,a)\},\\ \quad f(x_m,D)\neq f(x_n,D),\ x_m,x_n\in U_1;\\ \{a\in C:f(x_m,a)\neq f(x_n,a)\},\\ \quad x_m\in U_1,x_n\in U_2';\\ \varnothing,\ \text{otherwise.}\end{cases}\tag{1}$$

其中:$U_1=\mathrm{POS}_C(D),\mathrm{POS}_C(D)$为$D$的$C$正域;$U_2=U-\mathrm{POS}_C(D),U_2'=\mathrm{delrep}(U_2)$. 对于$U_2'$的解释:$\forall x_m\in U_2',U_2'$中不存在$x_n$,使得$\forall a\in C,f(x_m,a)=f(x_n,a)$,且$f(x_m,D)\neq f(x_n,D)$. $\mathrm{delrep}(U_2)$的详细解释见文献[6].

由于辨识矩阵中的元素之间可能存在包含关系,杨明等[6]提出了浓缩辨识矩阵的概念.

定义2[6] 对于给定的辨识矩阵M,其浓缩辨识矩阵定义如下:

$$\begin{aligned}&M^*=\\&\{M(x_m,x_n):M(x_m,x_n)(M(x_m,x_n)\neq\varnothing)\in M,\\&\text{且不存在}M(x_m',x_n')(M(x_m',x_n')\neq\varnothing)\in M,\\&\text{使得}M(x_m',x_n')\subseteq M(x_m,x_n)\}.\end{aligned}$$

在浓缩辨识矩阵的基础上,约简定义如下.

定义3 对于给定的辨识矩阵M,M^*是由辨识矩阵M生成的浓缩辨识矩阵,若:1)$\forall M(x_m,x_n)\in M^*$,均有$R\bigcap M(x_m,x_n)\neq\varnothing$;2)$\forall a\in R,\exists M(x_m,x_n)\in M^*$,使得$(R-\{a\})\bigcap M(x_m,x_n)=\varnothing$. 则称$R$是一个约简.

2 浓缩布尔矩阵及其求所有约简方法

由于判断辨识矩阵的浓缩需要进行大量字符匹配、查询、删除等操作,利用布尔矩阵相对于辨识矩阵更为简单. 文献[7]给出了浓缩布尔矩阵的定义,将辨识矩阵转化为布尔矩阵,再进行浓缩. 简要阐述如下.

定义4[7] 给定决策表$\mathrm{DT}=(U,C\bigcup D,V,f)$,$\forall x_m\in U_1,x_n\in U_1\bigcup U_2'$,布尔矩阵中的行(row)为$(x_m,x_n)$集合,$(x_m,x_n)$表示该行(row)为对象$x_m$与$x_n$进行属性比较后的结果,列为条件属性集合$C=\{C_1,C_2,\ldots,C_l\}$,布尔矩阵BM中元素值定义如下:

$$M[(x_m,x_n),C_k]=\begin{cases}1,\ f(x_m,D)\neq f(x_n,D),\ f(x_m,C_k)\neq f(x_n,C_k),\\ \quad x_m,x_n\in U_1;\\ 1,\ f(x_m,C_k)\neq f(x_n,C_k),\ x_m\in U_1,x_n\in U_2';\\ \varnothing,\ \text{otherwise.}\end{cases}\tag{2}$$

其中:$k=1,2,\ldots,l;U_1=\mathrm{POS}_C(D);U_2=U-\mathrm{POS}_C(D);U_2'=\mathrm{delrep}(U_2)$,同定义1. $M[(x_m,x_n),C_k]=1$表示属性C_k可以区分对象对(x_m,x_n),同理$M[(x_m,x_n),C_k]=0$表示属性C_k不可以区分对象对(x_m,x_n).

下面给出浓缩布尔矩阵的定义.

定义5[7] 给定决策表$\mathrm{DT}=(U,C\bigcup D,V,f)$,浓缩布尔矩阵$\mathrm{BM}^*=\{m:m\in\mathrm{BM}(m\neq0)$,且不存在$m'\in\mathrm{BM}(m'\neq0)$使得$m$与$m'$各位相或的结果与$m$的各位相同$\}$,其中$m$、$m'$为其集合的布尔表达形式.

文献[7]给出了具体求浓缩布尔矩阵的算法,本文在此不再赘述. 对于给定的属性子集R,在浓缩布尔矩阵BM^*中提取R中所有属性所对应的列构成的矩阵,称为由属性子集R构成的浓缩布尔矩阵. 结合约简的定义及浓缩布尔矩阵的结构给出如下定理.

定理1(浓缩布尔矩阵约简) 给定决策表$\mathrm{DT}=(U,C\bigcup D,V,f),R\subseteq C,R$为一个约简,当且仅当$R$的浓缩布尔矩阵具有以下两个特征:

1) 矩阵中每行至少有一个值为1;

2) 若对于任意的属性$a\in R$,删除属性a所在的列,则剩余部分一定有至少一行值全为0.

若只满足定理1的1),则表明属性子集R是一个超约简;若不满足定理1的1),则表明R是一个非约简,且R的任意子集也是非约简.

命题1[7] 在浓缩布尔矩阵中,若某行只有一位元素是1,则该元素所对应的列属性为核属性.

Lazo-Cortés等[12]在浓缩布尔矩阵上通过矩阵重排技术对属性集的幂集进行剪枝,从而删除所有非约简和部分超约简. 下面给出矩阵重排技术[12]:

1) 对矩阵的每行求行和,若最小行和唯一,则将最小行和所对应的行放置在矩阵的第1行;若最小行和不唯一,则需要对这几行求行密度(即行中值为1对应的列中值为1的数量之和),将行密度最大的行

放置在矩阵的第1行.

2) 将第1行值为1对应的列全部调整至矩阵左侧,0对应的列调整至矩阵右侧.

通过计算发现文献[12]算法中,重排矩阵的第1行值为1对应的列的排列顺序与求解复杂性密切相关,即矩阵中第1行值为1对应的列的排列顺序不同会导致所得结果中超约简数量不同,从而导致计算量不同. 所以针对该问题,本文将文献[12]的矩阵重排技术改进如下:

1)、2)与上述步骤相同,此略.

3) 对第1行值为1的列求列和,列和大的列调整至矩阵左侧,列和小的列调整至矩阵右侧.

注意到,矩阵重排时对应的属性也要进行列交换,保持列与属性之间的对应关系. 改进后的矩阵重排技术大大减少了运算量,并且会减少产生超约简的可能性,后续将给出解释.

例1　给定决策表(表1),论域$U=\{x_1,x_2,x_3,x_4,x_5,x_6\}$,属性集$C=\{a_1,a_2,a_3,a_4,a_5,a_6\}$,$D=\{d\}$为决策属性,根据得出的浓缩布尔矩阵给出改进后的重排矩阵.

表1　决策表

	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	1	0	2	2	2	1	0
x_2	2	1	0	2	0	1	2
x_3	1	2	0	2	0	0	1
x_4	0	1	2	1	0	0	2
x_5	2	1	0	1	1	1	2
x_6	1	0	2	1	0	2	1

根据文献[7]算法得出浓缩布尔矩阵

$$\mathrm{BM}^*=\begin{array}{c} \\ (x_1,x_2)\\(x_1,x_3)\\(x_1,x_6)\\(x_2,x_3)\\(x_3,x_4)\end{array}\begin{array}{c}\begin{array}{cccccc}a_1&a_2&a_3&a_4&a_5&a_6\end{array}\\\left[\begin{array}{cccccc}1&1&1&0&1&0\\0&1&1&0&1&1\\0&0&0&1&1&1\\1&1&0&0&0&1\\1&1&1&1&0&0\end{array}\right]\end{array}.$$

根据改进矩阵重排技术1)对上述矩阵中每行求行和,可以发现矩阵中第3行、第4行的行和最小,并且均为3,因为最小行和不唯一,需要计算行密度. 首先计算第3行的行密度(用下划线表示),有

$$\mathrm{BM}^*=\begin{array}{c}\\1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{cccccc}a_1&a_2&a_3&a_4&a_5&a_6\end{array}\\\left[\begin{array}{cccccc}1&1&1&0&\underline{1}&0\\0&1&1&0&\underline{1}&\underline{1}\\0&0&0&\underline{1}&\underline{1}&\underline{1}\\1&1&0&0&0&\underline{1}\\1&1&1&\underline{1}&0&0\end{array}\right]\end{array}.$$

由上述矩阵可以看出,第3行行密度是:$2+3+3=8$,同理可以得出第4行行密度是$3+4+3=10$,应将第4行放置在矩阵的第1行,其余各行不考虑排序,矩阵变为

$$\mathrm{BM}^*=\begin{array}{c}\\4\\1\\2\\3\\5\end{array}\begin{array}{c}\begin{array}{cccccc}a_1&a_2&a_3&a_4&a_5&a_6\end{array}\\\left[\begin{array}{cccccc}1&1&0&0&0&1\\1&1&1&0&1&0\\0&1&1&0&1&1\\0&0&0&1&1&1\\1&1&1&1&0&0\end{array}\right]\end{array}.$$

根据改进矩阵重排技术2),将上述矩阵中第1行值为1的列全部调整至左侧,得到

$$\mathrm{BM}^*=\begin{array}{c}\\4\\1\\2\\3\\5\end{array}\begin{array}{c}\begin{array}{cccccc}a_1&a_2&a_6&a_3&a_4&a_5\end{array}\\\left[\begin{array}{cccccc}1&1&1&0&0&0\\1&1&0&1&0&1\\0&1&1&1&0&1\\0&0&1&0&1&1\\1&1&0&1&1&0\end{array}\right]\end{array}.$$

根据改进矩阵重排技术3),对上述矩阵中第1行值为1的列求列和,即分别对矩阵中对应属性a_1、a_2、a_6所在的列求列和. 属性a_1对应列的列和为3,属性a_2对应列的列和为4,属性a_6对应列的列和为3. 矩阵最终的重排结果为

$$\mathrm{BM}^*=\begin{array}{c}\\4\\1\\2\\3\\5\end{array}\begin{array}{c}\begin{array}{cccccc}a_2&a_1&a_6&a_3&a_4&a_5\end{array}\\\left[\begin{array}{cccccc}1&1&1&0&0&0\\1&1&0&1&0&1\\1&0&1&1&0&1\\0&0&1&0&1&1\\1&1&0&1&1&0\end{array}\right]\end{array}.$$

根据上述方法得到重新排列的矩阵,可以更有利于求所有约简. 由改进矩阵重排技术可以得出以下结论.

定理2　给定决策表DT,对其浓缩布尔矩阵重排后,第1行从右至左值为0对应的所有属性构成的集合记为N,则N的任意属性子集均不为约简.

证明　对于重排矩阵第1行从右至左值为0对应属性构成的集合,其任意属性子集在重排矩阵中第1行值一定全为0,不满足定理1的1),故此属性子集不为约简. □

根据矩阵重排技术和浓缩布尔矩阵的定义可知,在重排矩阵中,由第1行值为0的列构成的矩阵只有第1行值全为0,其余各行值均不全为0.

例2 (续例1)　根据例中重排矩阵观察得知,取第1行值为0的列构成如下矩阵:

$$\begin{array}{c} \begin{matrix} a_3 & a_4 & a_5 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}.$$

可见,上述矩阵不满足定理1的1),故属性子集$\{a_3, a_4, a_5\}$的任意子集均不为约简.

定理3 给定决策表DT,对其浓缩布尔矩阵重排后,第1行从右至左第1个值为1的属性与其右侧的所有属性构成的集合记为S,则S一定为超约简.

由定理1和定理2易证,此略.

例3 (续例1) 根据定理3,有如下矩阵:

$$\begin{array}{c} \begin{matrix} a_6 & a_3 & a_4 & a_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

根据定理1的1)可知,上述矩阵中每行都至少有一个值为1,则属性子集构成超约简,所以属性子集$\{a_6, a_3, a_4, a_5\}$是一个超约简. 至于属性子集$\{a_6, a_3, a_4, a_5\}$是否为约简,需根据定理1的2)进一步判断.

定义6 给定决策表DT,对其浓缩布尔矩阵重排后,第1行值为1的对应属性称为主要属性.

例4 (续例1) 主要属性有a_2、a_1、a_6.

定理4 在重排矩阵中,选择不同的主要属性与其右侧属性子集的一些子集构成的约简,这些约简之间无重复.

因为选择不同主要属性时,其主要属性与其右侧属性子集的一些子集构成的约简是含有主要属性的一些集合, 所以若主要属性不同, 则产生的约简也不相同.

例5 (续例4) 属性a_2、a_1、a_6 是主要属性, 根据重排后的矩阵可知:考虑以a_6为主要属性与其右侧的属性子集$\{a_3, a_4, a_5\}$的子集构成的所有约简$\{\{a_4, a_5, a_6\}, \{a_3, a_6\}\}$,与考虑以$a_1$为主要属性与其右侧的属性子集$\{a_6, a_3, a_4, a_5\}$的子集构成的所有约简$\{\{a_1, a_3, a_4\}, \{a_1, a_5\}, \{a_1, a_6\}\}$彼此互不相同,所以它们之间是无重复的.

根据定理1可知,在求约简时,首先需要满足定理1的1),即保证矩阵中每行都至少有一个值为1,所以当在重排矩阵中找到第1行值为1对应的属性子集时,从中选择一个属性,只需要考虑此属性对应列中值为0的行的右侧行元素所构成的矩阵(矩阵中不再含此属性对应的列)即可;然后对所提取的矩阵进行重排得到重排矩阵,重复上述步骤,直到矩阵为空. 将过程中所选择的属性依次添加到属性子集A中,属性子集A构成的浓缩布尔矩阵满足定理1的1). 为了找到满足定理1的1)的所有属性子集,规定第1次重排矩阵中第1行值为1对应的属性为主要属性,后续重排矩阵中第1行值为1对应的属性为当前主要属性,在此基础上给出如下定理.

定理5 针对每次的重排矩阵,若选择一个主要属性(当前主要属性)a_i添加到属性子集A(属性子集A为在每次重排矩阵中选取一个当前主要属性共同构成的集合,下同)中,提取a_i对应列中值为0的行的右侧行元素构成矩阵,则对此矩阵重排后矩阵第1行值为0所对应的属性子集的任意子集与添加属性a_i后的属性子集A均不构成约简.

根据上述定理可知,找到的属性子集构成的浓缩布尔矩阵不满足定理1的1).

例6 由例5可知,当主要属性取a_6时,将其添加到属性子集A,即$A = \{a_6\}$,提取属性a_6对应列中值为0的行的右侧行元素构成的矩阵,即

$$\mathrm{BM}_1^* = \begin{array}{c} \begin{matrix} a_3 & a_4 & a_5 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}.$$

重排后得到矩阵

$$\mathrm{BM}_1^* = \begin{array}{c} \begin{matrix} a_3 & a_4 & a_5 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{array}.$$

其中a_3、a_4为当前主要属性. 观察上述矩阵知,第1行值为0对应的属性子集为$\{a_5\}$,由定理5可知,属性子集$\{a_6\}$与属性子集$\{a_5\}$构成的集合不为约简. 因为以属性子集$\{a_6, a_5\}$构成的浓缩布尔矩阵

$$\begin{array}{c} \begin{matrix} a_5 & a_6 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \end{array},$$

存在一行值全为0,不满足定理1的1).

定理6 针对每次的重排矩阵,选择一个主要属性(当前主要属性)a_i添加到属性子集A中,提取a_i对应列中值为0的行的右侧行元素构成矩阵,若矩阵为空,则在添加属性a_i前的重排矩阵中,属性a_i右侧的属性子集的任意子集与添加属性a_i后的属性子集A构成的集合为超约简.

将某个属性添加到属性子集A时,会使得矩阵为空,若此时该属性子集A构成的浓缩布尔矩阵已经满足定理1的1),则不需再向该集合添加任何属性,否则

会导致在求所有约简时出现更多冗余.

例7 (续例6)　同样以属性a_6为主要属性, $A=\{a_6\}$, 根据例6的重排矩阵BM_1^*可知, 若在属性子集A中添加属性a_3, 提取矩阵中属性a_3对应列中值为0的行的右侧行元素构成的矩阵, 矩阵为空, 属性子集$A=\{a_3,a_6\}$, 则属性子集A构成的浓缩布尔矩阵已满足定理1的1). 在重排矩阵BM_1^*中, a_3右侧的属性子集为$\{a_4,a_5\}$, 属性子集$\{a_4,a_5\}$的任意子集与属性子集$A=\{a_3,a_6\}$构成的集合均为超约简, 故不需要考虑.

根据上述定理, 多次运用矩阵重排技术, 得出所有约简, 具体算法描述如下.

算法1　基于浓缩布尔矩阵求所有约简算法.

输入: 决策表$\mathrm{DT}=(U,C\bigcup D,V,f)$, $A=\varnothing$;

输出: $\mathfrak{R}$ (所有约简构成的集族).

step 1: 根据文献[7]得到浓缩布尔矩阵BM^*.

step 2: 对浓缩布尔矩阵重排得BM^*.

step 3: 将BM^*第1行从右至左第1个值为1的属性添加到属性子集A中, 并且提取在重排矩阵BM^*中此属性对应列中值为0的行的右侧行元素, 得到新矩阵.

step 4: 对新矩阵重排后, 选择矩阵中第1行从右至左第1个值为1对应的属性添加到A中, 并且提取在重排矩阵中此属性对应列中值为0的行的右侧行元素构成新矩阵.

step 5: 重复step 4.

step 6: 直到矩阵为空, 得到一个属性子集A, 将其添加到$\mathfrak{R}$.

step 7: 返回上一层, 在上一层矩阵中选择第1行从右至左下1个值为1对应的属性, 添加到形成此矩阵前的属性子集A中, 再提取此矩阵中所选属性对应列中值为0的行的右侧行元素构成矩阵.

step 8: 重复step 4 ~ step 7.

step 9: 直到取遍BM^*的第1行值为1的属性为止.

step 10: 输出$\mathfrak{R}$.

最坏情况下算法的复杂度是关于样本数的一个指数函数, 然而通过矩阵重排技术求所有约简的算法具有并行的特点, 所以大多数情况下算法的复杂度远低于指数级别.

上述求所有约简的算法(算法1)中不考虑一定不是约简的所有属性子集, 并且不考虑大部分超约简.

定理7　算法1找到了决策表DT的所有约简.

证明　决策表DT中条件属性集的幂集由两部分构成, 一部分为重排矩阵中第1行值为0对应的属性构成集合的所有子集, 另一部分为选择不同主要属性与其右侧属性子集的所有子集构成的集合, 因此算法1在寻找约简时并没有漏掉任何属性子集. 本算法中不考虑所有非约简和大部分超约简.

1) 由定理2, 重排矩阵中第1行值为 0 对应的属性构成集合的所有子集, 这些子集构成的浓缩布尔矩阵均不满足定理1的1), 所得属性子集为非约简, 故不予考虑;

2) 由定理5, 当找到一个或若干个当前主要属性添加到属性子集A中, 剩余矩阵重排后第1行值为0对应的属性子集的任意子集与属性子集A构成的集合是非约简, 因为这些集合构成的浓缩布尔矩阵满足至少一行值全为0, 故不予考虑;

3) 若剩余重排矩阵不为空, 则属性子集A为非约简, 故不予考虑;

4) 由定理6, 在主要属性与其右侧属性子集构成的集合中, 若找到一个属性子集A使得剩余重排矩阵为空, 则不再向属性子集A中添加任何属性, 否则一定为超约简, 故不予考虑.

由算法1可知, 本算法对所有非约简和大部分超约简都不予考虑, 算法可求得所有约简. □

利用改进后的矩阵重排技术大大减少了运算量, 当选择一个主要属性(当前主要属性)时, 考虑此属性对应列中值为0的行的右侧行元素构成的矩阵, 如果没有改进矩阵重排技术3), 则会出现列和较小的主要属性位置相对靠左时, 其属性对应的列中值为0的行较多, 选择此属性对应列中值为0的行的右侧行元素构成矩阵规模较大, 从而复杂度较高.

下面通过例8说明算法1.

例8 (续例1)　求所有约简. 重排矩阵为

$$\mathrm{BM}^*=\begin{array}{c}\begin{matrix}a_2 & a_1 & a_6 & a_3 & a_4 & a_5\end{matrix}\\ \begin{bmatrix}1&1&1&0&0&0\\1&1&0&1&0&1\\1&0&1&1&0&1\\0&0&1&0&1&1\\1&1&0&1&1&0\end{bmatrix}\end{array}.$$

由上述重排矩阵可知, 属性子集$\{a_3,a_4,a_5\}$的任意子集均不能构成约简, 由于矩阵中主要属性是a_2、a_1、a_6, 需要3次循环.

第1次循环: 将重排矩阵BM^*第1行从右至左第1个值为1的属性a_6添加到属性子集A中, 得到$A=\{a_6\}$, 提取此属性的列中值为0的行的右侧行元素构成新矩阵(不含a_6的列), 对新矩阵进行重排, 得到

$$\mathrm{BM}_1^*=\begin{matrix}\\1\\1\end{matrix}\begin{matrix}\begin{matrix}a_3&a_4&a_5\end{matrix}\\\left[\begin{matrix}1&1&0\\1&0&1\end{matrix}\right]\end{matrix}.$$

将重排矩阵BM_1^*第1行从右至左第1个值为1对应的属性a_4添加到属性子集$A=\{a_6\}$中,得到$A=\{a_4,a_6\}$,提取属性a_4的列中值为0的行的右侧行元素构成新矩阵(不含a_4的列),并对新矩阵进行重排,得到

$$\mathrm{BM}_{11}^*=\begin{matrix}\\1\end{matrix}\begin{matrix}\begin{matrix}a_5\end{matrix}\\\left[\begin{matrix}1\end{matrix}\right]\end{matrix}.$$

将重排矩阵BM_{11}^*第1行从右至左第1个值为1对应的属性a_5添加到属性子集$A=\{a_4,a_6\}$中,得到$A=\{a_4,a_5,a_6\}$. 对矩阵BM_{11}^*提取属性a_5的列中值为0的行的右侧行元素构成新矩阵,若得到矩阵BM_{111}^*为空,则将属性子集A添加到$\mathfrak{R}$中. 由于重排矩阵BM_{11}^*没有第2个可取属性,返回上一层,即矩阵BM_1^*,将重排矩阵BM_1^*第1行从右至左第2个值为1对应的属性a_3添加到$A=\{a_6\}$中得到$A=\{a_3,a_6\}$,并在矩阵BM_1^*中提取属性a_3的列中值为0的行的右侧行元素构成新矩阵,得到矩阵BM_{12}^*为空. 将属性子集$A=\{a_3,a_6\}$添加到$\mathfrak{R}$中,得到$\mathfrak{R}=\{\{a_4,a_5,a_6\},\{a_3,a_6\}\}$,矩阵$\mathrm{BM}_1^*$循环结束.

第2次循环: 将重排矩阵BM^*第1行从右至左第2个值为1的对应属性a_1添加到属性子集A中,得到$A=\{a_1\}$,并在矩阵BM^*中提取属性a_1对应列中值为0的行的右侧行元素构成矩阵(不含a_1的列),重排得到矩阵

$$\mathrm{BM}_2^*=\begin{matrix}\\1\\1\end{matrix}\begin{matrix}\begin{matrix}a_6&a_5&a_3&a_4\end{matrix}\\\left[\begin{matrix}1&1&1&0\\1&1&0&1\end{matrix}\right]\end{matrix}.$$

将重排矩阵BM_2^*第1行从右至左第1个值为1对应的属性a_3添加到属性子集$A=\{a_1\}$,此时$A=\{a_1,a_3\}$,对矩阵提取后重排得到矩阵

$$\mathrm{BM}_{21}^*=\begin{matrix}\\1\end{matrix}\begin{matrix}\begin{matrix}a_4\end{matrix}\\\left[\begin{matrix}1\end{matrix}\right]\end{matrix}.$$

同理,将重排矩阵BM_{21}^*第1行从右至左第1个值为1对应的属性a_4添加到属性子集$A=\{a_1,a_3\}$,对矩阵提取后若BM_{211}^*为空,则将属性子集$A=\{a_1,a_3,a_4\}$添加到$\mathfrak{R}$中,得到

$$\mathfrak{R}=\{\{a_4,a_5,a_6\},\{a_3,a_6\},\{a_1,a_3,a_4\}\}.$$

接下来将重排矩阵BM_2^*第1行从右至左第2个值为1对应的属性a_5添加到$A=\{a_1\}$,对矩阵提取后若BM_{22}^*为空,则将属性子集$A=\{a_1,a_5\}$添加到$\mathfrak{R}$中. 再将重排矩阵BM_2^*第1行从右至左第3个值为1对应的属性a_6添加到$A=\{a_1\}$,对矩阵提取后若BM_{23}^*为空,则将属性子集$A=\{a_1,a_6\}$添加到$\mathfrak{R}$中,得到

$$\mathfrak{R}=\{\{a_4,a_5,a_6\},\{a_3,a_6\},\{a_1,a_3,a_4\},\{a_1,a_5\},\{a_1,a_6\}\},$$

矩阵BM_2^*循环结束.

第3次循环: 将重排矩阵BM^*第1行从右至左第3个值为1的属性a_2添加到A中得到$A=\{a_2\}$,提取此属性列中值为0的行的右侧行元素构成新的矩阵后,重排得到矩阵

$$\mathrm{BM}_3^*=\begin{matrix}\\1\end{matrix}\begin{matrix}\begin{matrix}a_4&a_5&a_6&a_1&a_3\end{matrix}\\\left[\begin{matrix}1&1&1&0&0\end{matrix}\right]\end{matrix}.$$

将重排矩阵BM_3^*第1行从右至左第1个值为1对应的属性a_6添加到$A=\{a_2\}$中,提取后若矩阵BM_{31}^*为空,则将属性子集$A=\{a_2,a_6\}$添加到$\mathfrak{R}$中. 同理,将重排矩阵BM_3^*第1行从右至左第2个值为1对应的属性a_5添加到$A=\{a_2\}$中,提取后若矩阵BM_{32}^*为空,则将属性子集$A=\{a_2,a_5\}$添加到$\mathfrak{R}$中. 将重排矩阵BM_3^*第1行从右至左第3个值为1对应的属性a_4添加到$A=\{a_2\}$中,提取后若矩阵BM_{33}^*为空,则将属性子集$A=\{a_2,a_4\}$添加到$\mathfrak{R}$中,最终得到所有约简

$$\mathfrak{R}=\{\{a_4,a_5,a_6\},\{a_3,a_6\},\{a_1,a_3,a_4\},\{a_1,a_5\},\{a_1,a_6\},\{a_2,a_6\},\{a_2,a_5\},\{a_2,a_4\}\}.$$

由例8可以看出,分别从各主要属性求约简所得到的结果之间相互独立,且互不相同,因此本文算法具有并行的特点,算法效率更高,后续将进一步进行研究.

由于算法1得到的结果可能会存在少数超约简,下面给出判断所得结果是否为超约简的简便方法. 根据定理1的2)可知,若属性子集A是一个约简,由属性子集A构成的浓缩布尔矩阵删除任意一个属性所在列时,则矩阵中一定会出现至少一行其值全为0,所以若一个属性子集构成的浓缩布尔矩阵变形后,可以形成一个单位阵,则构成该矩阵的属性子集一定是约简,否则是超约简,由此得到算法2.

算法2 一种快速判断超约简的算法.

输入: 算法1求得的属性子集A;

输出: 属性子集A是约简或超约简.

step 1: 在浓缩布尔矩阵中找到属性子集A构成浓缩布尔矩阵.

step 2: 对该矩阵求行和,将行和不为1的行删去.

step 3: 再对剩余矩阵求列和,若列和中存在值为0,则属性子集A一定是超约简,否则属性子集A一定

是约简.

step 4: 输出属性子集 A 是约简或超约简.

算法中若属性子集 A 构成的浓缩布尔矩阵为 $(m\times n)$ 阶,则step 2算法复杂度为 $O(mn)$, step 3算法复杂度为 $O(mn)$,文献[15]算法中最后判断约简的方法其算法复杂度为 $O(n\times m(n-1))$,故本文方法更为简单.

3　比较分析

下面通过两个例子对本文算法与文献[8-9, 11-12]算法所得结果进行比较分析,例9选用本文例8,其中例8的条件属性有6个,例10选择有7个条件属性的属性集. 通过对比发现虽然条件属性多了一个,但文献中算法求所有约简的运算量会显著增加,而本文的运算量并没有增加过多. 下面给出具体对比结果.

例9 (续例8)　用文献[8-9, 11-12]中算法求所有约简.

由文献[8]可知,若根节点顺序选择不同,则会导致最终所得约简个数不同,依次以属性 a_2、a_3、a_1 为根节点,根据文献算法生成树后考虑9个属性子集,未去冗余前的属性子集个数为9. 由于文献算法并没有给出去除冗余的方法,该9个属性子集即为最终所得结果. 经判断,比本文所得结果多出的属性子集 $\{a_1,a_3,a_5\}$ 为超约简,故本文算法与文献[8]算法相比在求所有约简过程中产生的超约简数量更少.

由文献[9]可知,其中合取矩阵即为本文浓缩布尔矩阵 BM^* (未重排),根据文献[9]算法可将合取矩阵转化为析取矩阵,其行对应属性构成的集合即为需要考虑的属性子集,若此析取矩阵不化简(去除冗余),则得到32个属性子集,化简后得到8个约简,与本文结果相同. 因此,文献[9]算法需对矩阵化简才能得到所有约简,否则结果中会存在较多超约简,故本文算法与文献[9]算法相比在求所有约简过程中产生的超约简数量更少,算法效率更高.

由文献[11]可知,采用幂图搜索算法需判断63个节点的属性重要度,而属性集 C 的幂集有64个节点,只有个别节点不需要判断,并没有减少很多节点,最终得到8个满足条件的约简. 因此,本文算法与文献[11]算法相比在求所有约简的过程中需要考虑的属性子集更少,算法效率更高.

由文献[12]可知,删除所有非约简和部分超约简后,需要考虑的属性子集有16个,这16个集合为未去冗余前的属性子集,去冗余后得到8个约简,与本文最终所得结果相同,然而本文算法在求所有约简过程中并没有产生很多超约简,故本文算法可以有效去除更多超约简.

例10　给定决策表(表2),求所有约简.

表2　决策表

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	d
x_1	1	1	0	0	1	0	1	0
x_2	0	0	0	0	1	0	1	1
x_3	2	1	1	2	1	1	1	1
x_4	0	0	2	1	0	0	2	0
x_5	0	1	1	0	2	1	0	1
x_6	1	1	0	1	0	1	0	1

由表2可以得到,辨识矩阵浓缩后的辨识元素为 $\{a_1,a_2\}$, $\{a_1,a_3,a_4,a_6\}$, $\{a_1,a_3,a_5,a_6,a_7\}$, $\{a_4,a_5,a_6,a_7\}$, $\{a_3,a_4,a_5,a_7\}$. 根据文献[8]算法,先后以 a_3、a_6、a_4、a_1 为根节点,需要考虑的属性子集有19个,未去冗余前的属性子集为19个,最终所得结果为19个属性子集;根据文献[9]算法,将合取矩阵转化成析取矩阵后需要考虑的属性子集有40个,未去冗余前属性子集为40个,最终所得结果为13个约简;根据文献[11]算法,需要考虑的属性子集有124个,最终得到约简有13个,由文献[11]算法的特性可知,不需要对所得结果判断是否存在超约简;根据文献[12]算法,需要考虑的属性子集为27个,未去冗余前属性子集为27个,删除集合中的超约简后最终所得结果为13个约简;本文算法需要考虑的属性子集有13个,未去冗余前属性子集为13个,最终所得结果为13个约简.

由此可知,文献[11]是对属性集的幂集有规律地寻找所有约简,但是在寻找所有约简时,需要考虑大量属性子集;文献[8-9, 12]在求所有约简过程中存在大量超约简,需对所得结果去冗余后最终得到所有约简,上述两例对比结果见表3和表4.

表3　约简算法性能对比(例9)

	文献[8]	文献[9]	文献[11]	文献[12]	本文算法
a	9	32	63	16	8
b	9	32	8	16	8
c	9	8	8	8	8

表4　约简算法性能对比(例10)

	文献[8]	文献[9]	文献[11]	文献[12]	本文算法
a	19	40	124	27	13
b	19	40	13	27	13
c	19	13	13	13	13

表中: a 为求约简时需要考虑的属性子集个数; b 为未去冗余前所得到的属性子集个数,由于本文只考虑算法1,不排除有少数超约简,故与其他文献仅对比没有去除冗余前的属性子集; c 为最终所得约简个数. 注意到,由于文献[8]的算法中并没有去除冗余,

其最终所得结果与其他文献均不相同.

由表3和表4可以看出,本文算法在求所有约简时需考虑的属性子集数更少,产生的超约简数更少,算法效率更高.

4 结 论

本文首先在浓缩布尔矩阵上提出了一种改进的矩阵重排技术;然后多次利用改进矩阵重排技术寻找所有约简,并给出一种快速判断超约简的方法;最后通过实例表明所提出的算法是高效的,相比其他文献产生超约简的数量更少. 由于重排矩阵中不同主要属性与其右侧属性子集构成的集合所产生的约简互不相同,所提出算法具有并行的特点,后续将对其进行研究.

参考文献(References)

[1] Pawlak Z. Rough sets[J]. International Journal of Computer & Information Sciences, 1982, 11(5): 341-356.

[2] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of the Polish Academy of Sciences Mathematics, 1985, 33(12): 693-696.

[3] 苗夺谦, 胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展, 1999, 36(6): 681-684.
(Miao D Q, Hu G R. A heuristic algorithm for reduction of knowledge[J]. Journal of Computer Research and Development, 1999, 36(6): 681-684.)

[4] 蒋瑜. 基于改进差别信息树的粗糙集属性约简算法[J]. 控制与决策, 2019, 34(6): 1253-1258.
(Jiang Y. Attribute reduction with rough set based on improved discernibility information tree[J]. Control and Decision, 2019, 34(6): 1253-1258.)

[5] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[C]. Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publishers, 1992: 331-362.

[6] 杨明, 杨萍. 差别矩阵浓缩及其属性约简求解方法[J]. 计算机科学, 2006, 33(9): 181-183.
(Yang M, Yang P. Discernibility matrix enriching and computation for attributes reduction[J]. Computer Science, 2006, 33(9): 181-183.)

[7] 李丹. 基于粗糙集的数据挖掘属性约简算法的研究[D]. 哈尔滨:哈尔滨工程大学计算机科学与技术学院, 2008: 20-25.
(Li D. Research on attribute reduction algorithms for data mining based on rough set[D]. Harbin: School of Computer Science and Technology, Harbin Engineering University, 2008: 20-25.)

[8] 黄治国, 孙伟, 吴海涛. 基于差别矩阵的约简树构造方法[J]. 计算机应用, 2008, 28(6): 1457-1459.
(Huang Z G, Sun W, Wu H T. Reduction tree algorithm based on discernibility matrix[J]. Journal of Computer Applications, 2008, 28(6): 1457-1459.)

[9] 张德栋, 李仁璞, 赵永升. 一种高效的分辨函数范式转换算法[J]. 计算机应用研究, 2010, 27(3): 879-882.
(Zhang D D, Li R P, Zhao Y S. High-efficient algorithm for normal form conversion of discernibility function[J]. Application Research of Computers, 2010, 27(3): 879-882.)

[10] 俞雪平, 胡云安. 属性约简中的范式转换算法研究[J]. 计算机应用与软件, 2015, 32(1): 271-274.
(Yu X P, Hu Y A. On normal form conversion algorithm in attribute reduction[J]. Computer Applications and Software, 2015, 32(1): 271-274.)

[11] 苏跃斌, 郭进, 郭瑞. 基于幂图的属性约简[J]. 控制与决策, 2014, 29(4): 743-747.
(Su Y B, Guo J, Guo R. Attribute reduction based on power graph[J]. Control and Decision, 2014, 29(4): 743-747.)

[12] Lazo-Cortés M S, Martínez-Trinidad J F, Carrasco-Ochoa J A, et al. A new algorithm for computing reducts based on the binary discernibility matrix[J]. Intelligent Data Analysis, 2016, 20(2): 317-337.

[13] Rodríguez-Diez V, Martínez-Trinidad J F, Carrasco-Ochoa J A, et al. A new algorithm for reduct computation based on gap elimination and attribute contribution[J]. Information Sciences, 2018, 435: 111-123.

[14] Miao D Q, Zhao Y, Yao Y Y, et al. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model[J]. Information Sciences, 2009, 179(24): 4140-4150.

[15] 冯卫兵, 张梅. 基于核与改进的条件区分能力的反向删除属性约简算法[J]. 计算机应用与软件, 2016, 33(5): 252-255.
(Feng W B, Zhang M. Converse delete attribute reduction algorithm based on core and improved condition distinguishing ability[J]. Computer Applications and Software, 2016, 33(5): 252-255.)

作者简介

冯琴荣(1972–), 女, 教授, 博士, 从事粗糙集理论、数据挖掘等研究, E-mail: fengqr72@163.com;

胡競丹(1994–), 女, 硕士生, 从事粗糙集理论的研究, E-mail: 136174834@qq.com.

(责任编辑: 郑晓蕾)