基于简化的二进制差别矩阵的快速属性约简算法

徐章艳[1,2] 杨炳儒[2] 宋 威[2]

(广西师范大学计算机系 桂林 541004)[1] (北京科技大学信息工程学院 北京 100083)[2]

摘 要 目前,基于二进制差别矩阵的属性约简算法有如下不足:算法的时间和空间复杂度不理想;所得到的属性约简与由基于正区域的属性约简的定义得到的属性约简不一致。本文给出一个简化的二进制差别矩阵和相应的属性约简的定义,证明了该定义与基于正区域的属性约简的定义是一致的。由于在简化的二进制的差别矩阵中,要先求出 $IND(C)$,故设计了一个较好的求 $IND(C)$ 的算法,其复杂度被降低为 $O(|C\|U|)$。在此基础上设计了一个快速属性约简算法,其时间复杂度和空间复杂度分别被降为 $\max\{O(|C|^2(|U'_{pos}\|U/C|)),O(|C\|U|)\}$ 和 $\max\{O(|U|),O(|C|(|U'_{pos}\|U/C|))\}$。

关键词 粗糙集,二进制差别矩阵,简化的二进制差别矩阵,核,复杂度

Quick Attribution Reduction Algorithm Based on Simple Binary Discernibility Matrix

XU Zhang-Yan[1,2] YANG Bing-Ru[2] SONG Wei[2]

(Department of Computer, Guangxi Normal University, Guilin 541004)[1]

(College of Information Engineering, Science and Technology University of Beijing, Beijng 100083)[2]

Abstract At present, the attribution reduction algorithm based on binary discernibility matrix has the following shortcomings: it's time complexity and space complexity are not good; the attribution reduction acquired from this algorithm is not the one acquired from definition of attribution reduction based on positive region. In this paper, a simple binary discernibility matrix and the corresponding definition of attribution reduction are provided. At the same time, it is proved that the above definition of attribution reduction is the same as the definition of attribution reduction based on positive region. For first computing $IND(C)$ in the simple binary discernibility matrix, a good algorithm for computing $IND(C)$ is designed, it's time complexity is cut down to $O(|C\|U|)$. On this condition, a quick attribution reduction algorithm is designed, time complexity and space complexity of the new algorithm are cut down to $\max\{O(|C|^2(|U'_{pos}\|U/C|)),O(|C\|U|)\}$ and $\max\{O(|U|),O(|C|(|U'_{pos}\|U/C|))\}$ respectively.

Keywords Rough set, Binary dicernibility matrix, Simple binary discernibility matrix, Core, Complexity

1 引言

在粗糙集理论[1,2]中,属性约简算法是重要的研究内容之一,其中在基于正区域的属性约简算法[1,2]和基于差别矩阵的属性约简算法[3,4]是两种常用的属性约简算法。在文[5,6]中又给出一种基于二进制差别矩阵的属性约简算法。

本文是在文[5,6]的基础上进行研究的。在文[6]中给出的属性约简算法,其时间复杂度为 $O(|C\|U|^4)$,空间复杂度为 $O(|C\|U|^2)$,这个复杂度并不理想。文[7]中虽然对文[6]进行了改进,使其算法的效率有所提高,但复杂度并未变。文[8]指出,基于二进制差别矩阵的核的定义与基于正区域的核的定义是不一致的。经研究发现,文[6]中属性约简的定义也与基于正区域的属性约简的定义不一致。为降低算法的复杂度,本文给出一个简化的二进制差别矩阵和相应的属性约简定义,并证明了该定义与基于正区域的属性约简的定义是等价的。由于在简化的二进制差别矩阵中,首先要求出 $IND(C)$,而目前求 $IND(C)$ 的最好方法是文[9]方法,该方法的复杂度为 $O(|C\|U|\log|U|)$,并不理想,因此本文利用基数排序的思想使得求 $IND(C)$ 的算法的复杂度被降为 $O(|C\|U|)$。在此基础上,设计了一个快速属性约简算法,其时间复杂度和空间复杂度分别被降为 $\max\{O(|C|^2(|U'_{pos}\|U/C|)),O(|C\|U|)\}$ 和 $\max\{O(|U|),O(|C|(|U'_{pos}\|U/C|))\}$。

2 基本概念及文[6]中属性约简算法的不足

定义 1 一个决策表定义为:$S=(U,C,D,V,f,d)$,其中 $U=\{x_1,x_2,\cdots,x_n\}$ 是论域;$C=\{c_1,c_2,\cdots,c_r\}$ 为条件属性集;D 为决策属性集;$f:U\times\rightarrow V$ 和 $d:U\times D\rightarrow V$ 是信息函数,其中 $F=C\cup D$,$C\cap D=\varnothing$,$V=\cup V_a$,$a\in F$,V_a 表示属性 a 的值域。

定义 2 在一个决策表 $S=(U,C,D,V,f,d)$ 中,$\forall R\subseteq C\cup D$,$X\subseteq U$,记 $U/R=\{R_1,R_2,\cdots,R_l\}$,则称 $R_-(X)=\cup\{R_i|R_i\in U/R,R_i\subseteq X\}$ 为 X 关于 R 的下近似集。

定义 3 在一个决策表 $S=(U,C,D,V,f,d)$ 中,设 $U/D=\{D_1,D_2,\cdots,D_k\}$ 表示由决策属性集 D 对论域 U 的划分,$U/C=\{C_1,C_2,\cdots,C_m\}$ 表示由条件属性集 C 对论域 U 的划分,称 $POS_C(D)=\bigcup_{D_i\in U/D}C_-(D_i)$ 为 C 关于 D 的正区域。

定义 4 设在一个决策表 $S=(U,C,D,V,f,d)$ 中,若 $P\subseteq C$,$POS_P(D)=POS_C(D)$,且 $\forall a\in P\Rightarrow POS_{P-\{a\}}(D)\neq POS_P(D)$,则称 P 是 C 关于 D 的一个属性约简(基于正区域的属性约简的定义)。

在属性约简的定义中,一直以来多数学者引用上述定义作为属性约简的基本定义。后来,HU 等提出一种基于差别矩阵的属性约简算法,一些学者[10]认为两种方法是等价的。

徐章艳 博士研究生,研究方向为粗糙集理论及其应用与数据挖掘;杨炳儒 教授,博士生导师,研究方向为人工智能、数据挖掘和柔性建模;宋威 博士研究生,研究方向为粗糙集理论及其应用与数据挖掘。

由于基于差别矩阵的方法易求出核属性集，从而在基于正区域的属性约简算法中常引用由基于差别矩阵方法求出的核属性集。直到文[8]指出两种方法求出的核属性集并不等价后，才引起人们的注意。文[6,7]中的属性约简算法实际上也与基于正区域的属性约简算法不等价。文[6]中例子可以说明这一点。

在文[6]中，用其属性约简算法求出决策表1得到的属性约简[6]为：$\{a,b\},\{b,c\},\{b,d\},\{a,c,d\}$，其中$\{b,d\}$就不是由定义5得到的属性约简。因为

$POS_{\{b,d\}}(D)=\{X1,X2,X3,X7,X10\}\neq POS_{\{a,b,c,d\}}(D)=\{X1,X2,X3,X7,X8,X10\}$。

上述例子说明两种方法不等价。基于这一点，本文提出一个简化的二进制差别矩阵和相应的属性约简的定义，该定义被证明是与基于正区域的属性约简的定义是等价的。

3 简化的二进制差别矩阵及其属性约简的定义

定义5 设在一个决策表中$S=(U,C,D,V,f,d)$，记$U/C=\{[x'_1]_C,[x'_2]_C,\cdots,[x'_m]_C\}$，记$U'=\{x'_1,x'_2,\cdots,x'_m\}$。由正区域的定义可设$POS_C(D)=[x'_{i_1}]_C\cup[x'_{i_2}]_C\cup\cdots\cup[x'_{i_t}]_C$，其中$\{x'_{i_1},x'_{i_2},\cdots,x'_{i_t}\}\subseteq U'$，且$[x'_{i_s}]_C(s=1,2,\cdots,t)$中任意两个不同的对象在决策属性上的取值均相同；记$U'_{pos}=\{x'_{i_1},x'_{i_2},\cdots,x'_{i_t}\}$，$U'_{neg}=U'-U'_{pos}$；称$S'=(U',C,D,V,f,d)$为简化的决策表。

定义6 在决策表$S=(U,C,D,V,f,d)$中，$S'=(U'=U'_{pos}\cup U'_{neg},C,D,V,f,d)$为简化的决策表，定义简化的二进制差别矩阵为：$M'=(m((i',j'),k))$，其元素定义如下：

$$m((i',j'),k)=\begin{cases}1 & c_k\in C,f(x'_i,c_k)\neq f(x'_j,c_k),d(x'_i,D)\neq d(x'_j,D)\text{且}x'_i,x'_j\text{在}U'_{pos}\text{中}\\ 1 & c_k\in C,f(x'_i,c_k)\neq f(x'_j,c_k)\text{且}x'_i\text{和}x'_j\text{一个在}U'_{pos}\text{中，一个在}U'_{neg}\text{中，}\\ 0 & \text{否则}\end{cases}$$

其中$k=1,2,\cdots,r$。

定义7 设$M'=(m((i',j'),k))$为决策表$S=(U,C,D,V,f,d)$的简化的二进制差别矩阵，$P\subseteq C$，若P满足：(1)由P中所有属性对应的各列所构成的M'的子阵中，不全为0的行数等于M'中不全为0的行数；(2)$\forall B'\subset B$均不满足(1)；则称P是C关于D的一个属性约简(基于简化的二进制差别矩阵的属性约简的定义)。

定义8 在决策表$S=(U,C,D,V,f,d)$中，令$P\subseteq C,Q\subseteq C$，记$U/P=\{P_1,P_2,\cdots,P_t\}$和$U/Q=\{Q_1,Q_2,\cdots,Q_s\}$，若$\forall P_i\in U/P\Rightarrow\exists Q_j\in U/Q$使$P_i\subseteq Q_j$，则称$U/P$为$U/Q$的加细，记为$U/P\preccurlyeq U/Q$。

定理1 设为决策表$S=(U,C,D,V,f,d)$中，$\forall Q\subseteq P\subseteq C$，则有$U/P\preccurlyeq U/Q$。

证明：记$U/P=\{P_1,P_2,\cdots,P_t\}$，$U/Q=\{Q_1,Q_2,\cdots,Q_s\}$，任取$P_i=[x]_P\in U/P$，由于$Q\subseteq P$，则有$P_i=[x]_P=\{y\mid\forall a\in P,f(y,a)=f(x,a)\}\subseteq Q_j=[x]_Q=\{y\mid\forall a\in Q,f(y,a)=f(x,a)\}$。由$P_i$的任意性知：$U/P\preccurlyeq U/Q$。

定理2 设$M'=(m((i',j'),k))$为决策表$S=(U,C,D,V,f,d)$的简化的二进制差别矩阵，$\forall P\subseteq C$，若P满足：由P中所有属性对应的各列所构成的M'的子阵中，不全为0的行数等于M'中不全为0的行数，则有$POS_P(D)=POS_C(D)$。

证明：假设有$POS_P(D)\neq POS_C(D)$，则有$U/P\neq U/C$。由定理2知$U/P\succcurlyeq U/C=\{[x'_1]_C,[x'_2]_C,\cdots,[x'_m]_C\}$，故至少存在$x'_i,x'_j\in U'$使得$[x'_i]_C\cup[x'_j]_C\subseteq[x'_i]_P\in U/P$且$x'_i$和$x'_j$中至少有一个在$U'_{pos}$中(若只存在$x'_i,x'_j\in U'$使得$[x'_i]_C\cup[x'_j]_C\subseteq[x'_i]_P\in U/P$且$x'_i$和$x'_j$都在$U'_{neg}$中，则有$POS_P(D)=POS_C(D)$。故当$x'_i$和$x'_j$至少有一个在$U'_{pos}$中，一定存在$c_k\in C-P$，使得$m((i',j'),k)=1$(这是因为$[x'_i]_C\neq[x'_j]_C$)；另一方面，由于$[x'_i]_C\cup[x'_j]_C\subseteq[x'_i]_P$，故$\forall c_h\in P$，有$m((i',j'),h)=0$。即在$M'$中存在元素不全为0的一行而该行对应在由$P$中所有属性对应的各列所构成的$M'$的子阵中的那一行的元素全为0，这与条件矛盾，故假设不成立。从而有$POS_P(D)=POS_C(D)$。

定理3 设$M'=(m((i',j'),k))$为决策表$S=(U,C,D,V,f,d)$的简化的二进制差别矩阵，$\forall P\subseteq C$，若$POS_P(D)=POS_C(D)$，则有由P中所有属性对应的各列所构成的M'的子阵中，不全为0的行数等于M'中不全为0的行数。

证明：假设由$P=\{c_{i_1},c_{i_2},\cdots,c_{i_t}\}$中所有属性对应的各列所构成的$M'$的子阵$M'(P)$中，不全为0的行数不等于$M'$中不全为0的行数，则在$M'(P)$中一定存在一行$(m((i,j),i_1),m((i,j),i_2,\cdots,m((i,j),i_t))=0$，而$(m((i,j),1),m((i,j),2),\cdots,m((i,j),r))\neq 0$。即存在属性$c_h\in C\wedge c_h\notin P$，使得$m((i,j),h)=1$。由定义6可知，$[x_i]_P=[x_j]_P$，且$x_i$和$x_j$至少有一个在$U'_{pos}$中，从而有$[x_i]_C\cup[x_j]_C\subseteq[x_i]_P$。当$x_i,x_j\in U'_{pos}$时，由于$d(x_i,D)\neq d(x_j,D)$，则$[x_i]_P\not\subset POS_P(D)$，从而$[x_i]_C\cup[x_j]_C\not\subset POS_P(D)$，但$[x_i]_C\cup[x_j]_C\subseteq POS_C(D)$，故$POS_P(D)\neq POS_C(D)$，这与条件矛盾，故假设不成立；当$x_i$和$x_j$只有一个在$U'_{pos}$中，无妨设$x_i\in U'_{pos}$而$x_j\in U'_{neg}$，由于$x_j\in U'_{neg}$，则一定存在$x'_j\in[x_j]_C$使得$d(x_i,D)\neq d(x'_j,D)$，从而有$[x_i]_P\not\subset POS_P(D)$，故$[x_i]_C\not\subset POS_P(D)$，但$[x_i]_C\subseteq POS_C(D)$，故$POS_P(D)\neq POS_C(D)$，这与条件矛盾，故假设不成立。综上所述，故命题成立。

定理4 基于正区域的属性约简定义与基于简化的二进制差别矩阵的属性约简定义是等价的。

证明：在决策表$S=(U,C,D,V,f,d)$中，设基于正区域的所有属性约简的集合为$PosReduc$，基于二进制差别矩阵的所有属性约简的集合为$BDReduc$。任取$P\in BDReduc$，则由P中所有属性对应的各列所构成的M'的子阵中，不全为0的行数等于M'中不全为0的行数，由定理2知，$POS_P(D)=POS_C(D)$；由于$P\in BDReduc$，则$\forall c_h\in P$有：由$P-\{c_h\}$中所有属性对应的各列所构成的M'的子阵中，不全为0的行数不等于M'中不全为0的行数，由定理3的逆否命题知，$\forall c_h\in P$有$POS_{P-\{c_h\}}(D)\neq POS_C(D)$。故有$P\in PosReduc$。由$P$的任意性知$BDReduce\subseteq PosReduc$。

任取$P\in PosReduc$，则有$POS_P(D)=POS_C(D)$。由定理3有：由P中所有属性对应的各列所构成的M'的子阵中，不全为0的行数等于M'中不全为0的行数。由于$P\in PosReduc$，则有$\forall c_h\in P$，有$POS_{P-\{c_h\}}(D)\neq POS_C(D)$，故对$\forall P'\subset P$有$POS_{P'}(D)\neq POS_C(D)$。由定理2的逆否命题知：由$P'$中所有属性对应的各列所构成的$M'$的子阵中，不全为0的行数不等于$M'$中不全为0的行数，从而有$P\in BDReduc$。由$P$的任意性知，$BDReduce\supseteq PosReduc$。

综上所述，命题成立。

由于计算简化的二进制差别矩阵，首先要计算$IND(C)$。文[9]虽然给出了一个较好的计算$IND(C)$的算法，但并不理想。因此，下面我们给出一个更好的计算$IND(C)$的算法。

4 求$IND(C)$的快速算法

为求出简化的二进制差别矩阵，首先要求出不可区分关系$IND(C)=U/C$。求$IND(C)$的一般方法的时间复杂度为

$O(|U|^2|C|)$。这个复杂度并不理想，文[9]利用快速排序的方法给出了一个计算 $IND(C)$ 的时间复杂度为 $O(|C\|U|\log|U|)$ 的算法。我们对计算 $IND(C)$ 的方法进行深入研究后，利用基数排序的思想，给出一个计算 $IND(C)$ 的时间复杂度为 $O(|C\|U|)$ 的算法。

算法 1 计算 $IND(C)$

输入：决策表 $S=(U,C,D,V,f,d)$，$U=\{x_1,x_2,\cdots,x_n\}$，$C=\{c_1,c_2,\cdots,c_r\}$

输出：$IND(C)$，U'_{pos}，U'_{neg}。

1. 对每一个 $c_i(i=1,2,\cdots,r)$ 求出 $f(x_j,c_i)(j=1,2,\cdots,n)$ 的最大值和最小值，分别记为 M_i 和 m_i；
2. 以静态链表依次存储对象 $x_1,x_2,\cdots,x_n$；令表头指针指向 x_1；
3. *for* $(i=1;i<r+1;i++)$
   - 3.1 第 i 趟“分配”：建立 M_i-m_i+1 空对列，令 $front_k$ 和 $end_k(k=0,1,2,\cdots,M_i-m_i)$ 分别为第 k 个对列的头指针和尾指针。将链表中的对象 $x\in U$ 按链表中的次序分配到第 $f(x,c_i)-m_i$ 个对列中去。
   - 3.2 第 i 趟“收集”：表头指针指向第一个非空对列的头指针，修改每一个非空对列的尾指针，令其指向下一个非空对列的对头对象，这样将 M_i-m_i+1 个对列重新组成一个链表；
4. 设由第 3 步得到链表中的对象序列为 $x'_1,x'_2,\cdots,x'_n$；
   $t=1;B_t=\{x'_1\}$；
   for $(j=2;j<n+1;j++)$
   若任一 $c_i\in C(i=1,2,\cdots,r)$ 均有 $f(x'_j,c_i)=f(x'_{j-1},c_i)$，则 $B_t=B_t\cup\{x'_j\}$；
   否则$\{t=t+1;B_t=\{x'_j\}$；
5. $U'_{pos}=\varnothing;U'_{neg}=\varnothing$；
   for $(i=1;i<t+1;i++)$
   若 B_i 中所有对象在决策属性上取值均相同，则取出 B_i 中的第一个对象并入 U'_{pos}；否则将 B_i 中的第一个对象并入 U'_{neg}；

算法 1 的复杂度分析：算法 1 的第 1 步的时间复杂度为 $O(|C\|U|)$；算法的第 2 步的时间复杂度为 $O(|U|)$；算法的第 3.1 步的时间复杂度为 $O(|U|+M_i-m_i+1)$，算法的第 3.2 步的时间复杂度为 $O(M_i-m_i+1)$，故第 3 步总的时间复杂度为 $O(|C\|U|+\sum_{i=1}^{r}(M_i-m_i+1))$；第 4 步的时间复杂度为 $O(|C\|U|)$；第 5 步的时间复杂度为 $O(|D\|U|)$（决策属性通常是一个）；从而算法 1 的时间复杂度为 $O(|C\|U|+\sum_{i=1}^{r}(M_i-m_i+1))$。大多数情况下，特别是对大型决策表而言（属性 $c\in C$ 的取值分布不是特别分散），常有 $\max_{1\leqslant i\leqslant s}(M_i-m_i+1)<|U|$（例如 UCI 中的 mushroom，共有 8000 多个对象和 22 个属性，但属性值均为单个的字母，因此每一个属性的不同取值最多为 26 种，而 26 是比 8000 小很多的），故 $(|C\|U|+\sum_{i=1}^{r}(M_i-m_i+1))\leqslant|C\|U|+|C\|U|$，从而算法 1 的时间复杂度为 $O(|C\|U|)$。易知空间复杂度为 $O(|U|)$。

5 快速属性约简算法

由定理 4 和算法 1 则可设如下的快速属性算法。

算法 2 快速属性约简算法

输入：决策表 $S=(U,C,D,V,f,d)$，$U=\{x_1,x_2,\cdots,x_n\}$，$C=\{c_1,c_2,\cdots,c_r\}$，

输出：决策表的属性约简 $reduce(C)$

1. 由算法 1 求出：$U'_{pos}=\{y_1,y_2,\cdots,y_s,\}$，$U'_{neg}=\{z_1,z_2,\cdots,z_t,\}$；
2. $reduce(C)=\varnothing$；
3. for $(i=1;i<s;i++)$
   for $(j=i+1;j<s+1;j++)$
   if $(d(y_i,D)!=d(y_j,D))$
   for $(k=1;k<r+1;k++)$
   if $(f(y_i,c_k)!=f(y_j,c_k))\quad m((i,j),k)=1$；
   else $m((i,j),k)=0$；
4. for $(i=1;i<s+1;i++)$
   for $(j=1;j<t+1;j++)$
   for $(k=1;k<r+1;k++)$
   if $(f(y_i,c_k)!=f(z_j,c_k)\quad m((i,j),k)=1$；
   else $m((i,j),k)=0$；
5. 对二进制差别矩阵 $M=(m((i,j),k)$ 的每一行做如下处理：若该行的元素全为 0 或 1，则删除该行；
6. while$(M=(m((i,j),k))!=\varnothing)$
   { $R=\varnothing;flag=0$；
   对矩阵的每一行判断，若只有一个元素的值为 1
   将该元素所在列对应的属性 a 并入 $reduce(C)$ 和 R 并将属性 a 对应列上值为 1 的元素所在的行去掉；
   if $R!=\varnothing$ {在矩阵中去掉 R 中的每一属性所对应的列；$flag=1$；}
   if $(flag!=1)$
   { 将矩阵的各行纵向相加，结果存入相应 $col[1],col[2],\cdots,col[|C|-|reduce(C)|]$ 中，求出 $col_{max}=\max\{col[1],col[2],\cdots,col[|C|-|reduce(C)|]\}$，将 col_{max} 所对应的属性 a（若有多个，则择其一）并入 $reduce(C)$，并将属性 a 对应列上值为 1 的元素所在的行去掉，在矩阵中去掉该属性所对应的列；
   }
   }

表 1 决策表 1

	a	b	c	d	D
X_1	1	2	0	1	1
X_2	1	2	0	1	1
X_3	2	0	0	1	0
X_4	0	0	1	2	1
X_5	2	1	0	2	1
X_6	0	0	1	2	2
X_7	2	0	0	1	0
X_8	0	1	2	2	1
X_9	2	1	0	2	2
X_{10}	2	0	0	1	0

算法的复杂度分析：算法 2 的第 1 步的时间复杂度由算法 1 知为 $O(|C\|U|)$，第 3 步和第 4 步的最坏时间复杂度为 $O(|C\|U'_{pos}|(|U'_{pos}|+|U'_{neg}|))=O(|C\|U'_{pos}\|IND(C)|)$。第 5 步的时间复杂度为 $O(|C\|U'_{pos}\|IDN(C)|)$；第 6 步每一次循环的最坏时间复杂度为 $O(|C\|U'_{pos}\|IND(C)|)$，最多循环 $|C|-1$ 次，故第 6 步最坏的时间复杂度为 $O(|C|^2|U'_{pos}\|IND(C)|)$。故新属性约简算法的最坏时间复杂度为 $\max\{O(|C|^2|U'_{pos}\|IND(C)|),O(|C\|U|)\}$。算法 2 的第 1 步的空间复杂度由算法 1 易得 $O(|U|)$，第 3,4,5,6 步的最坏空间复杂度为 $O(|C\|U'_{pos}\|IND(C)|)$。故新求算法的最坏空间复杂度为 $\max\{O(|C\|U'_{pos}\|IND(C)|),O(|U|)\}$。由于文[6,7]的算法的时间和空间复杂度分别为 $O(|C|^2|U|^4)$ 和 $O(|C\|U|^2)$，故新属性约简算法无论是时间复

杂度还是空间复杂度都较以前的算法好。

为更好地说明新算法的快速性，下面以文[6]中例子(决策表1)说明。

6 实例

对决策表1(a,b,c,d为条件属性，D为决策属性)的10个对象U，用算法1计算$U/\{a,b,c,d\}$过程如下($c_1=a,c_2=b,c_3=c,c_4=d$)：

由算法1的第1步计算出

$M_1=2,m_1=0;M_2=2,m_2=0;M_3=2,m_3=0;M_4=2,m_4=1$。

由算法的第2步得到：

→X1→X2→X3→X4→X5

→X6→X7→X8→X9→X10

第1趟"分配"结果为：

front[0]→X4→X6→X8←*end*[0] *front*[1]→X1→X2←*end*[1]

front[2]→X3→X5→X7→X9→X10←*end*[2]

第1趟"收集"结果为：

→X4→X6→X8→X1→X2→X3→X5→X7→X9→X10

第2趟"分配"结果为：

front[0]→X4→X6→X3→X7→X10←*end*[0]

front[1]→X8→X5→X9←*end*[1]

front[2]→X1→X2←*end*[2]

第2趟"收集"结果为：

→X4→X6→X3→X7→X10→X8→X5→X9→X1→X2

第3趟"分配"结果为：

front[0]→X3→X7→X10→X5→X9→X1→X2←*end*[0]

front[1]→X4→X6←*end*[1]

front[2]→X8←*end*[2]

第3趟"收集"结果为：

→X3→X7→X10→X5→X9→X1→X2→X4→X6→X8

第4趟"分配"结果为：

front[0]→X3→X7→X10→X1→X2←*end*[0]

front[1]→X5→X9→X4→X6→X8←*end*[1]

第4趟"收集"结果为：

→X3→X7→X10→X1→X2→X5→X9→X4→X6→X8

由第4步得到$U/\{a,b,c,d\}$为：

$\{X3,X7,X10\},\{X1,X2\},\{X5,X9\},\{X4,X6\},\{X8\}$。

由第5步得到：

$U'_{pos}=\{X3,X1,X8\};U'_{neg}=\{X5,X4\}$。

由算法2的第3和4步得到简化的二进制差别矩阵如表2所示。

表2 简化的二进制差别矩阵

m(i,j)	a	b	c	d
m(3,1)	1	1	0	0
m(3,5)	0	1	0	1
m(3,4)	1	0	1	1
m(1,5)	1	1	0	1
m(8,5)	1	0	1	0
m(8,4)	0	1	1	0

表3 第一次次循环后决策表

m(i,j)	b	c	d
m(3,5)	1	0	1
m(8,4)	1	1	0

由算法2的第5步可知：第一次循环得到的$col_{max}=6$，所对应的属性为$\{a,b\}$；取出属性a，则有$reduc(C)=\{a\}$，二进制差别矩阵变为表3所示。

第二次循环得到的$col_{max}=2$，所对应的属性为$\{b\}$；取出属性b，则有$reduc(C)=\{a,b\}$，二进制差别矩阵变空，算法结束，得到的属性约简为$reduc(C)=\{a,b\}$。

若用文[6]中的算法所求得的二进制差别矩阵是45行，再用其算法中的变换，其计算量要多得多。而用新算法所求得的二进制差别矩阵是6行。可见新算法不仅空间存贮量小，而且计算量也大大减少，这是因为在生成简化的二进制差别矩阵时减少了大量的计算时间，在后续的计算中又减少了大量的计算时间。因而新算法是一个快速属性约简算法。

结论 由于在原来基于二进制差别矩阵的属性约简算法中，对属性约简的定义与基于正区域的属性约简的定义是不一致的，且算法的时间和空间复杂度都不理想。经深入研究后，我们给出一个简化的二进制差别矩阵，并给出了相应的属性约简的定义，证明了该定义与基于正区域的属性约简的定义是等价的。为快速求出简化的二进制差别矩阵，设计了一个快速求$IND(C)$的算法，在此基础上，设计了一个新的基于简化的二进制差别矩阵的属性约简算法，并分析了新的属性约简算法的时间和空间复杂度，分别为$\max\{O(|C|(|U'_{pos}\|U/C|)),O(C\|U|)\}$和$\max\{O(|U|),O(|C|(|U'_{pos}\|U/C|))\}$。新算法的时间和空间复杂度都比文[6,7]中的算法的时间和空间复杂度要好。

参考文献

1 Pawlak Z. Rough Sets. International Journal of Computer and information Science [J], 1982,11(5): 341～356

2 Pawlak Z, Wong S K M, Ziarko W. Rough sets: probabilistic versus deterministic approach [J]. Int J Man-Machine Studies, 1988,29:81～95

3 Skowron A, Rauszer C. The Discernibility Matrices and Functions in Information Systems [A]. In: Slowinski R, ed. intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory, 1992. 331～362

4 Hu Xiao Hua, Cercone N. Learning in relational databases: a rough set approach [J]. Computational Intelligence,1995,11(2): 323～337

5 Fleix R, Ushio T. Rough Sets-based Machine Learning Using a Binary Discernibinity Matrix [J]. IPMM'99 published,1999. 299～305

6 支天去，苗夺谦. 二进制可辨别矩阵的变换及高效属性约简算法的构造[J]. 计算机科学，2002,29(2):140～142

7 周海岩，杨汀. 基于二进制可辨识矩阵属性约简算法的改进[J]. 计算机工程与设计，2003,24(12):35～42

8 叶东毅，陈昭炯. 一个新的二进制可辨识矩阵及其核的计算[J]. 小型微型计算机系统，2004,25(6):965～967

9 刘少辉，盛秋戬，等. Rough集高效算法的研究[J]. 计算机学报，2003,26(5):524～529

10 叶东毅. Jelonek属性约简算法的一个改进[J]. 电子学报，2000,28(12):81～82