Fast Algorithm for Attribute Reduction Based on Rough Set Theory Using Binary Discernibility Matrix

Changsheng Zhang1, Jing Ruan2,*

1 College of Physics & Electronic Information Engineering, Wenzhou University, Zhejiang, China
2 Wenzhou Vocational & Technical College, Wenzhou, Zhejiang, China
Email: 156732998@qq.com, ruanjing1979@126.com

***Abstract* -- Rough set is a valid mathematical tool that deals with uncertain, vague and incomplete information of decision systems. Attribute reduction is one of the issues in rough set theory, and core attributes are indispensable in the process of attribute reduction. Recently, some researchers proposed the method of binary discernibility matrix to compute the results of attribute reduction, which can not only save space, but also exploit the benefit of bit operations to improve the classification performance. In this paper, a novel definition of binary discernibility matrix is introduced, and then a new algorithm of computing the matrix is provided, whose time complexity is $\max\{O(|C||U'_{pos}||IND(C)|),\ O(|U'_{pos}|^2|IND(C)|^2)\}$. Finally, a fast algorithm for computing the result of attribute reduction based on rough set theory is proposed, and an example is used to illustrate the effectiveness of the proposed algorithm.**

Keywords* -- *Rough sets; binary discernibility matrix; attribute reduction; complexity

I. INTRODUCTION

Rough set theory [1-3] is a valid mathematical approach to deal with the vagueness information, and has attracted a lot of attention. It has been applied successfully in the fields of data mining, knowledge discovery and machine learning, signal processing and image processing, and so on. Attribute reduction is one of the most significant applications in rough sets, which aims to select the indispensable attribute subset from candidate attributes to express the target concepts of a decision system. In the past decade years, many rough set-based attribute reduction algorithms were presented. Pawlak proposed the attribute reduction definitions based on algebra view and designed the corresponding algorithms [3]. So for, some researchers proposed many attribute reduction algorithms based on algebra view in the literature [4-6], some researchers introduced attribute reduction algorithms based on discernibility matrix [7-9] and attribute reduction algorithms based on Shannon information entropy [10-12]. However, the temporal complexity of this kind of algorithms is not good.

At present, in rough set theory, there are a lot of different attribute reduction algorithms based on rough set theory have been developed, and it is proved that finding all attribute reduction is NP hard[13,14], so there are few algorithms to discover all attribute reduction. A new algorithm to find all attribute reduction based on a generalized decision is proposed [15], which is based on the Apriori algorithm. In this algorithm, it is first to compute the complement of all the elements of the discernibility matrix (each element is a subset of the condition attributes), then to seek for the maximum element of all these obtained elements (the maximum element refers to no super-sets of all the elements of the discernibility matrix). In this process, there are a lot of time-consuming operations on determining whether an element is the subset of other elements, whose time and space complexity of this process are both .This process is relatively more detrimental to the whole algorithm when the longest attribute reduction is so short. Meanwhile, the follow-up process of the algorithm is also used the above process: there are a lot of time-consuming operations on determining whether an element is the subset of other elements.

At present, some researchers use the binary discernibility matrix to design some efficient algorithms, for which the bit binary (1 or 0) represents the elements of the discernibility matrix is easy to storage, so as to reduce a great deal of memory space. Meanwhile, the bit binary operation is so easy that the running speed of the algorithm is improved. But some researchers pointed out that the attribute reduction based on Skowron discernibility is equivalent to the attribute reduction based on positive region. Actually, it is wrong. In the paper [16], a new binary discernibility matrix based on positive region is given. In order to reduce the time complexity of computing the binary discernibility matrix, the definition of binary discernibility matrix is introduced at first. However, the binary discernibility matrix does not need to compute until the simplified decision system is computed. Therefore, a new algorithm for computing the simplified decision system is designed, whose time complexity is $O(|U||C|)$. Based on the simplified decision system, and a new algorithm for computing the simplified binary discernibility matrix is presented. On basis of all the above, in this paper, the simplified binary discernibility matrix is constructed, this matrix can delete a number of

DOI 10.5013/IJSSST.a.17.27.21

ISSN: 1473-804x online, 1473-8031 print

repeated elements in decision system, and then a fast attribute reduction algorithm based on the simplified binary discernibility matrix is designed. The proposed algorithm is better than the previous algorithms in the large decision systems. Finally, an example is used to illustrate the effectiveness of the proposed algorithm.

II. PRELIMINARIES

In this section, we introduce some basic concepts and related definitions about rough set theory.

Definition 1 A decision system is defined as:
$S=(U,C,D,V,f)$,
where $U=\{x_1,x_2,\cdots,x_n\}$ is the set of objects,
$C=\{c_1,c_2,\cdots,c_r\}$ is the set of condition attributes, D is the set of decision attributes, and $C\cap D=\varnothing$;
$V=\bigcup_{a\in C\cup D} V_a$,
where V_a is the value range of the attribute a. $f:U\times C\cup D\rightarrow V$ is an information function, which is the information value for each attribute of each object, that is $\forall a\in C\cup D,\ x\in U,\ f(x,a)\in V_a$ holds.
For the attribute subset $P\subseteq(C\cup D)$, it determines a binary discernibility relation $IND(P)$:
$IND(P)=\{(x,y)\in U\times U\mid \forall a\in P, f(x,a)=f(y,a)\}$

Definition 2 For a decision system $S=(U,C,D,V,f)$, $\forall R\subseteq C\cup D,\ X\subseteq U$, denote $U/R=\{R_1,R_2,\cdots,R_l\}$, then
$R\_(X)=\cup\{R_i\mid R_i\in U/R, R_i\subseteq X\}$, is called lower approximation of X for R.

Definition 3 For a decision system $S=(U,C,D,V,f)$, Let $U/D=\{D_1,D_2,\cdots,D_k\}$ be the partition of D to U, and $U/C=\{C_1,C_2,\cdots,C_m\}$ be the partition of C to U, then $POS_C(D)=\bigcup_{D_i\in U/D} C\_(D_i)$ is called the positive region of C on D.

Definition 4 For a decision system $S=(U,C,D,V,f)$, for $\forall P\subseteq C$, if $POS_P(D)=POS_C(D)$,
and for $\forall a\in P$ there is $POS_{P-\{a\}}(D)\neq POS_C(D)$,
then P is called as an attribute reduction of C on D. (This is the definition of attribute reduction based on the positive region).

Definition 5 For a decision system $S=(U,C,D,V,f)$, let $\mathrm{Re}\,d(C)$ be the attribute reduction of C on D,
$PCore(C)=\bigcap_{B\in \mathrm{Re}\,d(C)} B$ is called the core based on positive region of the decision system.

Definition 6 For a decision system $S=(U,C,D,V,f)$, denote $U/C=\{[x'_1]_C,\ [x'_2]_C,\cdots,[x'_m]_C\}$,
$U'=\{x'_1,x'_2,\cdots,x'_m\}$,
Assume
$POS_C(D)=[x'_{i_1}]_C\cup[x'_{i_2}]_C\cup\cdots\cup[x'_{i_t}]_C$,
where $\{x'_{i_1},x'_{i_2},\cdots,x'_{i_t}\}\subseteq U'$
and $|[x'_{i_s}]_C/D|=1\ (s=1,2,\cdots,t)$,
denote $U'_{pos}=\{x'_{i_1},x'_{i_2},\cdots,x'_{i_t}\}$, $U'_{neg}=U'-U'_{pos}$,
then $S'=(U',C,D,V,f)$ is called the simplified decision system.

Definition 7 For a decision system $S=(U,C,D,V,f)$, $S'=(U'=U'_{pos}\cup U'_{neg},\ C,D,V,f)$ is the simplified decision system, M' is defined as a matrix of the simplified decision system with the (i',j')th entry $m((i',j'),k)$ given by:

$$m((i',j'),k)=\begin{cases}1 & c_k\in C, f(x'_i,c_k)\neq f(x'_j,c_k), f(x'_i,D)\neq f(x'_j,D)\wedge x'_i,\ x'_j\in U'_{pos}\\ 1 & c_k\in C, f(x'_i,c_k)\neq f(x'_j,c_k)\ \wedge x'_i\in U'_{pos}\wedge x'_j\in U'_{neg}\\ 1 & c_k\in C, f(x'_i,c_k)\neq f(x'_j,c_k)\ \wedge x'_j\in U'_{pos}\wedge x'_i\in U'_{neg}\\ 0 & else\end{cases}$$

Where $k=1,2,\cdots,r$.

Definition 8 For a decision system $S=(U,C,D,V,f)$, let $M'=(m((i',j'),k))$ be the binary discernibility matrix, $\forall P\subseteq C$, if P satisfies:(1) the sub-array of M' composed of corresponding columns of all attributes in P, the number of all lines of sub-array not all are 0, which is equal to the number of lines of M' not all are 0;(2) $\forall B' \subset B$ is not satisfied(1), then P is called the attribute reduction of C for D base on binary discernibility matrix.

Definition 9 For a decision system $S=(U,C,D,V,f)$, $S'=(U'=U'_{pos}\cup U'_{neg},\ C,D,V,f)$ is the simplified decision system, $DBCore(C)$ is called the core based on discernibility matrix, which is defined as follows:

$$DBCore(C)=\{c_k \mid c_k\in C, m((i',j'),k)=1,$$
$$m((i',j'),h)=0, h=1,2,\cdots,k-1,k+1,\cdots,m\}$$

Theorem 1 For a decision system $S=(U,C,D,V,f)$, let $M'=(m((i',j'),k))$ be its is the binary discernibility matrix, $\forall P\subseteq C$, if P satisfies that the sub-array of M' composed of corresponding columns of all attributes in P, the number of all lines of sub-array not all are 0 is equal to the number of lines of M' not all are 0, then $POS_P(D)=POS_C(D)$.

Lemma 1 For a decision system $S=(U,C,D,V,f)$, there is $PCore(C)=\{a \mid a\in C, POS_{C-\{a\}}(D)\neq POS_C(D)\}$.

Theorem 2 For a decision system $S=(U,C,D,V,f)$, there is $DBCore(C)=PCore(C)$.

Proof: For $\forall c_k\in DBCore(C)$, there exists x'_i, x'_j such that $m((i',j'),k)=1$ and $m((i',j'),h)=0$ (where $h=1,2,\cdots,k-1,k+1,\cdots,r$).

When $x'_i\in U'_{pos}\wedge x'_j\in U'_{pos}$, then $[x'_i]_C\cup[x'_j]_C\subseteq[x'_i]_{C-\{c_k\}}$ (because the only attribute c_k can distinguish between two equivalent classes $[x'_i]_C$ and $[x'_j]_C$), but $d(x'_i,D)\neq d(x'_j,D)$, so $([x'_i]_C\cup[x'_j]_C)\not\subset POS_{C-\{c_k\}}(D)$. On the other hand, there is $([x'_i]_C\cup[x'_j]_C)\subseteq POS_C(D)$, therefore $POS_{C-\{c_k\}}(D)\neq POS_C(D)$. According to Lemma 1, there is $c_k\in PCore(C)$.

When $x'_i\in U'_{pos}\wedge x'_j\in U'_{neg}$ or $x'_j\in U'_{pos}\wedge x'_i\in U'_{neg}$, we can suppose $x'_i\in U'_{pos}\wedge x'_j\in U'_{neg}$, for the same reason, there is $[x'_i]_C\cup[x'_j]_C\subseteq[x'_i]_{C-\{c_k\}}$. Due to $x'_j\in U'_{neg}$, $([x'_i]_C\cup[x'_j]_C)\not\subset POS_{C-\{c_k\}}(D)$ holds. On the other hand, there is $[x'_i]_C\subseteq POS_C(D)$, so $POS_{C-\{c_k\}}(D)\neq POS_C(D)$, according to Lemma 1, there is $c_k\in PCore(C)$. Since the condition attribute c_k is randomly selected, we have $DBCore(C)\subseteq PCore(C)$.

For $\forall c_k\in PCore(C)$, there is $POS_{C-\{c_k\}}(D)\neq POS_C(D)$, then there exists $x'_i\in U'_{pos}$ such that $[x'_i]_{C-\{c_k\}}\not\subset POS_{C-\{c_k\}}(D)$, so there at least exists $x'_j\in[x'_i]_{C-\{c_k\}}\wedge x'_j\notin[x'_i]_C\ (x'_j\in U')$ such that $d(x'_i,D)\neq d(x_j,D)$. Since $x'_j\notin[x'_i]_C$, then $f(x'_i,a)\neq f(x'_j,a)$, according to Definition 7, there is $m((i',j'),k)=1$ and $m((i',j'),h)=0$ (where $h=1,2,\cdots,k-1,k+1,\cdots,r$). Therefore, there is $c_k\in DBCore(C)$. Since the condition attribute c_k is randomly selected, we have $DBCore(C)\supseteq PCore(C)$, thus we have $DBCore(C)=PCore(C)$.

Definition 10 For a decision system $S=(U,C,D,V,f)$, for $\forall P\subseteq C$, if $POS_P(D)\neq POS_C(D)$ and then *P* is not an attribute reduction of *C* on *D*. But if $\forall a\in P$ there is $POS_P(D)=POS_C(D)$, then P is called as an attribute reduction of C on D.

Note: The attribute reduction is computed by most of the algorithms, but the result of the attribute reduction is not complete.

III. COMPUTING THE SIMPLIFIED DECISION SYSTEM

In this section, we will calculate the simplified decision system. That is to say, calculating the simplified decision system is in fact to calculate the *IND*(*C*). To our best knowledge, the familiar algorithm for computing *IND*(*C*) is not ideal with the time complexity $O(|U|^2|C|)$. The radix sorting algorithm is used to calculate the simplified decision system [16]. Its time complexity is $O(|C||U|)$.

Algorithm 1. Computing the simplified decision system

Input: Decision system: $S=(U,C,D,V,f)$, $U=\{x_1,x_2,\cdots,x_n\}$, $C=\{c_1,c_2,\cdots,c_r\}$

Output: $U'_{pos}\cup U'_{neg}$.

Step 1: To each $c_i\ (i=1,2,\cdots,r)$, calculate the maximum and minimum of $f(x_j,c_i)\ \ (j=1,2,\cdots,n)$ and denote M_i and m_i respectively;

Step 2: use state list to store the objects $x_1, x_2, \cdots, x_n$ in turn, and let the head pointer of the list point to x_1 ;

Step 3: $for(i=1; i<r+1; i++)$

Step 3.1: the ith "distribution": construct M_i-m_i+1 empty queues, let $front_k$ and end_k $(k=0,1,\cdots, M_i - m_i)$ be the head pointer and tail pointer of the kth queue respectively. Distribute the object x of the list U to the $f(x, c_i) - m_i$ th queue according to the elements order of list U.

Step 3.2: the ith "collection": the head pointer of the list points to the head pointer of the first nonempty queue, modify the tail pointer of each nonempty and let it point to the head object of the next nonempty queue;

Step 4: Let the sequence of list from Step 3 be $x_1', x_2', \cdots, x_n'$;

$t = 1$; $B_t = \{x_1'\}$;

$for\ (j=2; j<n+1; j++)$

if any $c_i \in C(i=1,2,\cdots,r)$ there is $f(x_j', c_i) = f(x_{j-1}', c_i)$,

then $B_t = B_t \cup \{x_j'\}$;

else $\{t=t+1;\ B_t = \{x_j'\}\ ;\}$

Step 5: $U'_{pos} = \varnothing$; $U'_{neg} = \varnothing$;

for (i=1;i<t+1;i++)

if any $x,y \in B_i$ there is $f(x,D) = f(y,D)$, then we take out the first object of B_i to U'_{pos} ; *else* we take out the first object of B_i to U'_{neg} ;

IV. THE ATTRIBUTE REDUCTION ALGORITHM USING BINARY DISCERNIBILITY MATRIX

According to Theorem 1 and Theorem 2, we will propose the algorithm 2 for computing the result of the attribute reduction using the binary discernibility matrix.

Algorithm 2: The attribute reduction algorithm using the binary discernibility matrix.

Input: Decision system $S = (U,C,D,V,f)$, $U = \{x_1,x_2,\cdots,x_n\}$, $C = \{c_1,c_2,\cdots,c_r\}$

Output: the binary discernibility matrix, $core(C)$

Step1: According to Algorithm 1, calculate $U'_{pos} = \{y_1, y_2, \cdots, y_s,\}$, $U'_{neg} = \{z_1, z_2, \cdots, z_t,\}$; $core(C) = \varnothing$;

Step2: $for(\ i=1;\ i<s;\ i++)$

$for(\ j=i+1;\ j<s+1;\ j++)$

$if\ (d(y_i, D)! = d(y_j, D))$

$for\ (\ k=1;\ k<r+1;\ k++)$

$if\ (f(y_i, c_k)! = f(y_j, c_k))$

$m((i,j),k) = 1;$

$else\ \ m((i,j),k) = 0;$

Step3: $for\ (i=1; i<s+1; i++)$

$for\ (j=1; j<t+1; j++)$

$for\ (k=1; k<r+1; k++)$

$if\ (f(y_i, c_k)! = f(z_j, c_k))$

$m((i,j),k) = 1;$

$else\ \ m((i,j),k) = 0;$

Step 4: Add each column of matrix $M = (m(i,j),k)$, then the results are stored in corresponding row arrays $row[1], row[2], \cdots, row[z]$.

To each $row[h](h=1,2,\cdots,z)$, do the following process:

if $row[h] = |C|$ or zero, then delete the row;

if $row[h] = 1$, then store attribute r to $Core(C)$.

Step 5: While ($M = (m(i,j),k) \neq \varnothing$)

Update $M = (m(i,j),k)$ for $\forall x \in U$;

If $a_i \in m((i,j),k)$ then

$Red = Red \cup \{a_i\}$;

Step 6: Return Red

The time complexity Analysis of Algorithm 2: The time complexity of Step 1 is $O(|C||U|)$. The time complexities of Step 2 and Step 3 are $O(|C|(|U'_{pos}|(|U'_{pos}|+|U'_{neg}|))$ $= O(|C||U'_{pos}||IND(C)|)$. The worst time complexity of Step 4 is $O(|C||U'_{pos}||IND(C)|)$. The time complexity of Step 5 is $O(|C\text{-}Core||U'_{pos}||IND(C)|)$, according to Algorithm 1, the time complexity is $O(|C||U|)$.

Therefore, the time complexity of the algorithm 2 is $\max\{O(|C\|U'_{pos}\|IND(C)|), O(|U|)\}$.

V. SIMULATION EXAMPLE

To evaluate the performance of the proposed method, we give a decision system $S=(U,C,D,V,f)$ shown in System 1, where *a, b, c,* and *d* are conditional features, and *D* is the decision feature. It is verified that S is an inconsistent decision system. For the 15 objects in the decision system 1, according to Algorithm 1, here ($p_1 = a$, $p_2 = b$, $p_3 = c$, $p_4 = d$).

TABLE I. SYSTEM1: DECISION SYSTEM 1

	a	b	c	d	D
X1	1	1	1	1	0
X2	2	2	2	1	1
X3	1	1	1	1	0
X4	2	3	2	3	0
X5	2	2	2	1	1
X6	3	1	2	1	0
X7	1	2	3	2	2
X8	2	3	1	2	3
X9	3	1	2	1	1
X10	1	2	3	2	2
X11	3	1	2	1	1
X12	2	3	1	2	3
X13	4	3	4	2	1
X14	1	2	3	2	3
X15	4	3	4	2	2

According to Step 1 of Algorithm 1, we have $M_1 = 4$, $m_1 = 1$; $M_2 = 3$, $m_2 = 1$; $M_3 = 4$, $m_3 = 1$; $M_4 = 3$, $m_4 = 1$.

According to Step 2, we have

$$X_1 \to X_2 \to X_3 \to X_4 \to X_5 \to X_6 \to X_7 \to$$
$$\to X_8 \to X_9 \to X_{10} \to X_{11} \to X_{12}$$
$$X_{13} \to X_{14} \to X_{15}$$

For Step 3: The results of the first allocation are:

$$\mathit{front}[0] \to X_1 \to X_3 \to X_7 \to X_{10} \to X_{14} \to \mathrm{end}[0]$$
$$\mathrm{front}[1] \to X_2 \to X_4 \to X_5 \to X_8 \to X_{12} \to \mathrm{end}[1]$$
$$\mathrm{front}[2] \to X_6 \to X_9 \to X_{11} \to \mathrm{end}[2]$$
$$\mathrm{front}[3] \to X_{13} \to X_{15} \to \mathrm{end}[3]$$

The results of the first collection are:

$$X_1 \to X_3 \to X_7 \to X_{10} \to X_{14} \to X_2 \to X_4 \to$$
$$\to X_5 \to X_8 \to X_{12} \to X_6 \to X_9 \to X_{11}$$
$$\to X_{13} \to X_{15}$$

The results of the second allocation are:

$$\mathrm{front}[0] \to X_1 \to X_3 \to X_6 \to X_9 \to X_{11} \to \mathrm{end}[0]$$
$$\mathrm{front}[1] \to X_7 \to X_{10} \to X_{14} \to X_2 \to X_5 \to \mathrm{end}[1]$$
$$\mathrm{front}[2] \to X_4 \to X_8 \to X_{12} \to X_{13} \to X_{15} \to \mathrm{end}[2]$$

The results of the second collection are:

$$X_1 \to X_3 \to X_6 \to X_9 \to X_{11} \to X_7 \to X_{10} \to$$
$$\to X_{14} \to X_2 \to X_5 \to X_4 \to X_8 \to X_{12}$$
$$\to X_{13} \to X_{15}$$

The results of the third allocation are:

$$\mathrm{front}[0] \to X_1 \to X_3 \to X_8 \to X_{12} \to \mathrm{end}[0]$$
$$\mathrm{front}[1] \to X_6 \to X_9 \to X_{11} \to X_2 \to$$
$$\to X_5 \to X_4 \to \mathrm{end}[1]$$
$$\mathrm{front}[2] \to X_7 \to X_{10} \to X_{14} \to \mathrm{end}[2]$$
$$\mathrm{front}[3] \to X_{13} \to X_{15} \to \mathrm{end}[3]$$

The results of the third collection are:

$$X_1 \to X_3 \to X_8 \to X_{12} \to X_6 \to X_9 \to X_{11} \to$$
$$\to X_2 \to X_5 \to X_4 \to X_7 \to X_{10} \to X_{14}$$
$$\to X_{13} \to X_{15}$$

The results of the fourth allocation are:

$$\mathrm{fron}[0] \to X_1 \to X_3 \to X_6 \to X_9 \to$$
$$\to X_{11} \to X_2 \to X_5 \to \mathrm{end}[0]$$
$$\mathrm{front}[1] \to X_8 \to X_{12} \to X_7 \to X_{10}$$
$$\to X_{14} \to X_{13} \to X_{15} \to \mathrm{end}[1]$$
$$\mathrm{front}[2] \to X_4 \to \mathrm{end}[2]$$

The results of the fourth collection are:

$$X_1 \to X_3 \to X_6 \to X_9 \to X_{11} \to$$
$$\to X_2 \to X_5 \to X_8 \to X_{12} \to X_7 \to X_{10} \to X_{14}$$
$$\to X_{13} \to X_{15} \to X_4$$

According to Step 3, we know that $U/\{a, b, c, d\}$ is $\{\{X_1, X_3\}$, $\{X_6, X_9, X_{11}\}$, $\{X_2, X_5\}$, $\{X_8, X_{12}\}$, $\{X_7, X_{10}, X_{14}\}$, $\{X_{13}, X_{15}\}$, $\{X_4\}\}$.

Therefore, by using the algorithm 1, we have $U'_{pos} = \{X_1, X_2, X_8, X_4\}$, $U'_{neg} = \{X_6, X_7, X_{13}\}$.

According to Step 2 and Step 3 of Algorithm2, the binary discernibility matrix is calculated. Shown in System 2, according to Step 4, we know that m(1,6)={0,0,1,0}, so we can obtain that core attributes is $\{a\}$. According to Step 5, attribute c is added to the result of attribute reduction, due to the binary discernibility matrix $M=\varnothing$, the algorithm terminates. Therefore, the result of the attribute reduction is $\mathrm{Re}d=\{a,c\}$.

TABLE II. SYSTEM 1: THE BINARY DISCERNIBILITY MATRIX

m(i,j)	a	b	c	d
m(1,2)	1	1	1	0
m(1,8)	1	1	0	1
m(1,4)	0	0	0	0
m(1,6)	0	0	1	0
m(1,7)	0	1	1	1
m(1,13)	1	0	1	0
m(2,8)	0	1	1	1
m(2,4)	0	1	1	1
m(2,6)	1	1	0	0
m(2,7)	1	0	1	1
m(2,13)	1	1	1	0
m(8,4)	1	0	1	1
m(8,6)	1	1	1	1
m(8,7)	0	1	1	0
m(8,13)	1	0	1	0
m(4,6)	0	1	0	1
m(4,7)	1	1	1	0
m(4,13)	0	0	1	1

VI. CONCLUSIONS

In this paper, attribute reduction is one of the issues in rough set theory. In order to improve the efficiency of the attribute reduction algorithm, the definition of simplified decision system is introduced at first, and then the binary discernibility matrix based on the simplified decision system is constructed. This novel binary discernibility matrix can reduce the search space of the attribute reduction algorithm, which can greatly improve the performance. To make use of the bitmap technology conveniently, a fast attribute reduction algorithm based on the binary discernibility matrix is designed. In this new algorithm, we can quickly obtain the result of the attribute reduction. In addition, an example is given to demonstrate the feasibility and effectiveness of the proposed algorithm.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of Zhejiang Province (No.LY13F020024, No.LZ14F010002), the Foundation of Science and Technology Department of Zhejiang Province (No.2014C31079), and the National Natural Science Foundation of China (61501331)

REFERENCE

[1] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data System Theory, Knowledge Engineering and Problem Solving. Dordrecht: Kluwer, 1991.
[2] Pawlak Z., J. Grzymala-Busse, Rough sets, Communications of the ACM, 1995, 8(11): 89-95.
[3] Z. Pawlak and A. Skowron, Rudiments of rough sets, Information Sciences, 2007, 177(1): 3-27.
[4] B. Guo, R. I. Damper, S. R. Gunn, J. D. B. Nelson. A fast reparability-based feature selection method for high-dimensional remotely sensed image classification, Pattern Recognition, 2008, 41(5): 1670-1679.
[5] W.H. Xu, Y. Li, X.W. Liao. Approaches to attribute reductions based on rough set and matrix computation in inconsistent ordered information systems, Knowledge- Based Systems, 2012, 27(3): 78-91.
[6] J. Blaszczynski, R. Slowinski, Incremental induction of decision rules from dominance-based rough approximations, Electronic Notes in Theoretical Computer Science, 2003, 82(4): 40-51.
[7] G.Y. Wang, Rough selection in algebra view and information view, International Journal of Intelligent System, 2003, 18(3): 679-688.
[8] Qian J, D. Q. Miao, Z. H. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, International Journal of Approximate Reasoning, 2011, 52(2): 212-230.
[9] H. Zhao, K.Y. Qin, Mixed feature selection in incomplete decision table. Knowledge-Based Systems, 2014. 57(6): 181-190
[10] D.G. Chen, Q.H. Hu, Y.P. Yang, Parameterized attribute reduction with Gaussian kernel based fuzzy rough sets, Information Sciences, 2011, 181(23): 5169-5179.
[11] Q. H. Hu, X. J. Che, L. Zhang, Rank entropy-based decision trees for monotonic classification. IEEE Transaction on Knowledge and Data Engineering, 2012, 24(11): 2052-2064.
[12] Y. Qian, J. Liang, W. Pedrycz, C. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, Pattern Recognition. 44 (2011) 1658-1670.
[13] M. Kryszkiewicz, K. Cichon, Torwards scalable algorithms for discovery rough set reducts[J]. Transcations on rough sets, LNCS 3100, 2000, pp:120-143.
[14] S.Y. Li, T.R. Li. Incremental update of approximations in dominance-based rough sets approach under the variation of attribute values, Information Sciences, 2015, 294(7): 348-361
[15] Neil Mac Parthalain, Qiang Shen, Exploring the boundary region of tolerance rough sets for feature selection, Pattern Recognition, 2009, 42(5): 655-667.
[16] Xu Z Y, Lin Z P. A quick attribution reduction algorithm with complexity of max{O(|U||C|), O(|C|2|U/C|)} , Journal of Computer, 2006, 29(3): 391-399.