

# Decision rule extraction for fault diagnosis by discernibility matrix

Mingzhu Lou<sup>1</sup>

<sup>1</sup>NanChang Institute of Technology, Nanchang, Jiangxi 330099, China

<sup>1</sup>email: loumingzhud@163.com

**Keywords:** Decision Table; Fault Diagnosis; Rule Extraction; Discernibility Matrix

**Abstract.** This article focuses on algorithm to directly and fast finding out the decision rule from decision tables by using rough set theory. The concept of relative rule is put forward and the method for finding absolute reduction decision rule through combining rules of relative decisional requirement sets is given in this paper. Using this concept, the relative discernibility matrix between decision classes can be created to decide the relative decision requirements which are conjoined to find the decision requirements of all the decision classes. The relative decision requirements are extracted by creating relative discernibility matrix, so that scales of computing can be cut down substantially and saving storage space. Theoretical analysis shows that the proved method is feasible. Last, the proposed method is applied in fault diagnosis of reciprocating machines.

## Introduction

Rough set theory, introduced by Pawlak [1], has been successfully applied in intelligence data analysis tasks, such as in machine learning, pattern recognition, knowledge discovery, and expert systems [2–7]. One fundamental aspect of rough set theory is attribute reduction in information systems, which is selecting or reserving those attributes that provide the same information for classification purposes as the entire set of available attributes. It is proved that finding the minimal reduction is a NP-hard problem. Traditional method for finding the minimal reduction is to reduce rules based upon attribute reduction, however, finding a random minimal attribute reduction cluster is another NP problem. Furthermore, rule generating on different minimal attribute reduction cluster means a different minimal rule cluster. In this paper, the concept of relative decision requirements is put forward and all the minimal rule clusters of decision table can be found on the basis of the combination relative rules between decision classes.

## Correlative conception

In rough set theory, a decision table is a four-tuple  $DT = (U, C \cup D, V, f)$ , where  $U = \{u_1, u_2, \dots, u_n\}$  is a finite and nonempty universe of objects,  $u_i$  is an object;  $A = C \cup D$  a finite set of attributes, where  $C$  and  $D$  are called conditional attributes set and decision attributes set respectively;  $V$  a set of values  $= \bigcup_{q \in A} V_q$ , where  $V_q$  is a value set of the attribute  $q$ ; and  $f: U \times A \rightarrow V$ , an information function which specifies attribute value of  $u_i \in U$ . On such basis, we have follow definitions.

**Definition 1:** Given two objects  $u_i, u_j \in U$ ,  $u_i, u_j$  concerning the condition  $a \in C$  can be set apart, iff  $a(u_i) \neq a(u_j)$ , we denotes  $u_i \propto_a u_j$ .

**Definition 2:** Given two objects  $u_i, u_j \in U$ ,  $f_{DNF(a)}(u_i, u_j)$  denotes the DNF of  $u_i$  and  $u_j$ 's distinguishable property.

$$f_{DNF(a)}(u_i, u_j) = \begin{cases} \vee a & , \text{iff } u_i \propto_a u_j \\ \phi & , \text{else} \end{cases} \quad (1)$$

**Definition 3:** Given two objects  $u_i, u_j \in U$ , if  $u_i$  and  $u_j$  are distinguishable, the function distinguishes  $u_i$  from  $u_j$ , denotes as  $f_{ui}(u_i, u_j)$  as follows:

$$u_i \propto u_j = \begin{cases} \wedge f_{ui}(u_i, u_{y_i}) & , (j=1, 2, \dots, n) \wedge f_{ui}(u_i, u_{y_i}) \neq \phi \\ \phi & , \text{else} \end{cases} \quad (2)$$

Where  $a_{a(ui)}$  denotes the abbreviation of  $(a, a(u_i))$ .

**Definition 4:** Given  $u_i \in U$  and  $Y \subseteq U$ ,  $Y = \{u_{y1}, u_{y2}, \dots, u_{yj}, \dots, u_{yn}\}$ ,  $u_i \propto Y$  denotes the decision requirement of  $u_i$  which is distinguishable from the set  $Y$ .

$$u_i \propto u_j = \begin{cases} \wedge f_{ui}(u_i, u_{y_i}) & , (j=1, 2, \dots, n) \wedge f_{ui}(u_i, u_{y_i}) \neq \phi \\ \phi & , \text{else} \end{cases} \quad (3)$$

According to Definition 4,  $u_i \propto Y$  is not always a disjunctive normal form, however, one can always transform it into an equal disjunctive normal form, that is  $\vee s$ , in which  $s$  denotes an independence condition that can distinguish  $u_i$  from  $Y$ .  $s$  is a conjunctive normal form like  $\wedge a_{a(ui)}$ .  $u_i \propto Y$  is regard as the formal expression of DNF in the latter part of this paper.

**Definition 5 :** Given  $X, Y \subseteq U$  and  $X = \{u_{x1}, u_{x2}, \dots, u_{xi}, \dots, u_{xm}\}$ ,  $X \prec Y$  denotes the set  $\{u_{x1} \propto Y, u_{x2} \propto Y, \dots, u_{xm} \propto Y\}$  which consists of the rules of distinguishing every object in the set  $X$  from set  $Y$ . We call  $X \prec Y$  as the decision requirement set of  $X$  which is relative to set  $Y$ .

Obviously, in a decision table  $T$  and  $X \subseteq U$ , the absolute decision requirement of set  $X$  is  $X \prec \{U-X\}$ . The set which consists of the independence requirement from  $u_i \propto Y$ , in  $X \prec Y$ , is called the independence requirement set of  $X \prec Y$ .

Given a decision table  $T=(U, A, V, f)$ , Skowron introduces the discernibility matrix [8] of information system and the modified discernibility matrix of decision table was introduced by Hu is  $M=[C_{ij}]$ , in which  $(i=1, \dots, |U|, j=1, \dots, |U|)$ ,  $C_{ij} = \{a \in C: a(u_i) \neq a(u_j) \wedge D(u_i) \neq D(u_j), u_i, u_j \in U\}$  [9]. In most papers, attribute reduction is done by discernibility matrix. In order to get relative decision requirements of two sets, the definition of discernibility matrix are as follows.

**Definition 6:** Given  $X, Y \subseteq U$ , where  $X = \{u_{x1}, u_{x2}, \dots, u_{xi}, \dots, u_{xm}\}$  and  $Y = \{u_{y1}, u_{y2}, \dots, u_{yj}, \dots, u_{yn}\}$ , the relative Discernibility Matrix between  $X$  and  $Y$  is  $M_{(X,Y)}[i,j] = f_{DNF(a)}(u_{xi}, u_{yj})$ , where  $1 \leq i \leq m; 1 \leq j \leq n$ .

According to Definition 2 one knows  $f_{DNF(a)}(u_{xi}, u_{yj})$  denotes a DNF with discernibility attributes between  $u_{xi}$  and  $u_{yj}$ . Let  $M_{(X,Y)'}^*$  be the transposed matrix of  $M_{(X,Y)}$ , since  $M_{(X,Y)'}^* = M_{(Y,X)}$ , it is easy to obtain the decision requirement set  $X \prec Y$  and  $Y \prec X$  only by  $M_{(X,Y)}$ .

## Rule Generating by Discernibility Matrix

A decision rule,  $u_{xi} \propto Y$ , can be obtained once  $i$ th row created in  $M_{(X,Y)}$ , and another decision rule,  $Y_j \propto X$ , can be get once  $j$ th column created. The last decision rule of  $X \prec Y$ ,  $u_{xm} \propto Y$ , and the last decision rule of  $Y \prec X$ ,  $u_{yn} \propto X$  can be worked out once the last element of the  $M_{(X,Y)}(m,n)$  is figured out.

**Algorithm 1:** Create  $M_{(X,Y)}$  for work out  $X \prec Y$  and  $Y \prec X$

Input :  $X = \{u_{x1}, u_{x2}, \dots, u_{xi}, \dots, u_{xm}\}$ ,  $Y = \{u_{y1}, u_{y2}, \dots, u_{yj}, \dots, u_{yn}\}$ , ( $X$  and  $Y$  are stored in Matrix  $X[m]$  and  $Y[n]$  respectively)

Output:  $X \prec Y$  and  $Y \prec X$

Step 1: Create an empty matrix,  $M_{(X,Y)}$ , with  $m$  rows and  $n$  columns. Evaluate variables,  $i=1, j=1$ .

Step 2: calculating  $M_{(X,Y)}[i,j] = f_{DNF(a)}(u_{xi}, u_{yj})$

Step 3: if  $i=m$

3.1 Conjoin  $j$ th column elements to get a CNF  $L_j$

3.2 Transform  $L_j$  into a DNF  $L_j'$

3.3 a rule  $u_{yj} \propto X$  can be get directly from  $L_j'$

Step 4: if  $j < n$ , then  $j=j+1$ , go to step 2

Step 5: if  $j=n$

5.1 Conjoin  $i$ th row elements to get a CNF  $L_i$

5.2 Transform  $L_i$  into a DNF  $L_i'$

5.3 a rule  $u_{xi} \propto Y$  can be get directly from  $L_i'$

Step 6: if  $i < m$ , then  $i=i+1, j=1$ , go to step 2

The time complexity of Algorithm 1 is  $O(|X||Y||C|)$ .  $A$  denotes an attribute set and  $|X|$  denotes the cardinality of  $X$ .

It may be abundant if use  $X \prec Y$  to denote the decision requirement set of  $X$ . For example, there are two objects  $u_i$  and  $u_j$  in set  $X$ , if  $(u_i \propto Y) \vee (u_j \propto Y) = u_j \propto Y$  then  $u_i \propto Y$  is redundant.

**Definition 7** For  $X \subseteq U$  and  $Y \subseteq U$ , denote  $d_X^Y = \bigvee_{u_i \in X} (u_i \propto Y)$ .

According to Definition 7,  $d_X^Y$  is a DNF that eliminating redundancy requirements in  $X \prec Y$ . It consists of all reduction requirements that can distinguish  $X$  from  $Y$ . It also equals to disjunction of all independence requirements in  $X \prec Y$ .

**Definition 8:** For  $X \subseteq U$  and  $Y \subseteq U$ , the independence requirement set of  $X \prec Y$  is  $S$ . If  $S$  has the subset  $S'$ , giving the set which consists of all decision requirement to the reflection of objects of  $S'$  is equal to  $X$ , the set we get from all independence requirements about any real subset of  $S'$  to the reflection of objects belongs to  $X$ , therefore  $S'$  is the minimal set of  $S$ .

By Algorithm 1, it is easy to work out  $X \prec Y$ . Algorithm 2 introduce process of get all the minimal requirement sets that  $X$  relative to  $Y$  according to  $X \prec Y$ .

**Algorithm 2:** Computing all minimal requirement sets.

**Input:**  $X \prec Y = \{u_{x1} \propto Y, u_{x2} \propto Y, \dots, u_{xm} \propto Y\}$ , where  $X = \{u_{x1}, u_{x2}, \dots, u_{xi}, \dots, u_{xm}\}$ , every  $u_{xi} \propto Y (1 \leq i \leq m)$  is a DNF about independent requirement.

**Output:** All minimal requirement set that  $X$  relative to  $Y$

Step 1: let  $W = (u_{x1} \propto Y) \wedge (u_{x2} \propto Y) \wedge \dots \wedge (u_{xm} \propto Y)$

Step 2: Transform  $W$  into a DNF,  $W = S_1 \vee S_2 \vee \dots \vee S_i \vee \dots \vee S_j$ , where  $S_i (1 \leq i \leq j)$  is a DNF about independent requirement of  $X \prec Y$ . The value of  $j$  equals to the number of minimal requirement sets. During the transformation, an independent requirement should be the minimal operating unit, it can not be split. For example, the conjoin of two independent requirements  $a_0 \wedge b_1$  and  $d_1$ , should be  $(a_0 \wedge b_1) \wedge d_1$ , rather than  $a_0 \wedge b_1 \wedge d_1$ .

Step 3: Extracting all independent requirements from  $S_i$  to form a new set  $S_i'$ , which is one minimal requirement set that  $X$  relative  $Y$ .  $S_1', S_2', \dots, S_j'$  are all minimal requirement sets that  $X$  relative to  $Y$ .

Next, a fast decision rule extraction algorithm is as follows.

**Algorithm 3:** Decision rule extraction

**Step 1:** Partition decision classes  $U/D = \{D_1, D_2, D_3, \dots, D_n\}$ .

**Step 2:** For every couple of decision class  $(D_i, D_j)$ , where  $1 \leq i < j \leq n$ , creating relative discernibility matrix  $M_{(D_i, D_j)}$  to get  $D_i \prec D_j$  and  $D_j \prec D_i$

**Step 3:** Conjoin the relative decision requirement sets extracted in step 2 according to Theorem 1 to finding the absolute decision requirement sets of all decision classes,  $D_1 \prec \{U - D_1\}$ ,  $D_2 \prec \{U - D_2\}$ ,  $D_3 \prec \{U - D_3\}$ , ...,  $D_n \prec \{U - D_n\}$

**Step 4:** For every absolute decision requirement set that get in step 3, eliminate redundant requirements according definition 8 to get all reduced decision requirements of decision classes, that is  $d_{D_1}^{U-D_1}, d_{D_2}^{U-D_2}, d_{D_3}^{U-D_3}, \dots, d_{D_n}^{U-D_n}$

**Step 5:** Find all minimal decision requirement set for every decision class based on Algorithm 2 and step 3.

In what follows, we estimate the time complexity of Algorithm 3.

For the step 1: partitioning decision class according every decision attribute step by step is a faster method. For example, for decision attribute set  $D = \{D_a, D_b, D_c, \dots\}$ , first partitioning  $U$  into  $D_{a1}, D_{a2}, \dots, D_{ai}$  according to attribute  $D_a$  should be done  $|U|$  times throughout the whole set  $U$ . Then partitioning  $D_{a1}, D_{a2}, \dots, D_{ai}$  according to attribute  $D_b$  should be done  $|D_{a1}| + |D_{a2}| + \dots + |D_{ai}| = |U|$  times. Each further partition should be done  $|U|$  times. So the time complexity to partition the decision classes is  $O(|U||D|)$ , where  $|D|$  is the cardinality of decision attribute set.

For step 2: For decision class cluster  $\{D_1, D_2, D_3, \dots, D_n\}$ , all discernibility matrix  $M_{(D_i, D_j)}$  should

be worked out, where  $1 \leq i < j \leq n$ . Total times of comparison is  $|C|(|U|^2 - \sum_{i=1}^n |D_i|^2)/2$ , which is only the half of total comparison times of finding decision rules for each decision class directly.

For step 3: Conjoin all decision rule relating to  $D_i$ , which involves  $|U/D|-1$  relative requirement set. Since sequences of rules in every relative decision set relating to  $D_i$  are same, rules conjunction can be done in sequence. Conjunction all decision rules relating to  $D_i$  need  $|D_i|(|U/D|-1)$  time. Time complexity of step 3 can be worked out by summing up time taken by all decision classes, that is  $O(|U|(|U/D|-1))$ .

For step 4: Disjoin every rule in  $D_i \prec \{U-D_i\}$  to get  $d_{D_i}^{U-D_i}$ , which will takes  $|D_i|$  time. In order to get all  $d_{D_i}^{U-D_i}$  need throughout  $|U|$  times.

For step 5: Conjoin rules in  $D_i \prec \{U-D_i\}$ , which will takes  $|D_i|$  time. A minimal requirement set can be found according to each disjunction term.

Our algorithms mentioned above are effective to decision table only with discrete attribute. For those decision tables which contain continuous attributes, many discretization algorithms can be used as an effective preprocessing step.

## Fault Diagnosis Application

In this section, we apply our proposed method to fault diagnosis of a diesel engine fuel system [10]. Table 1 is a decision table formed from historical data of fault diagnosis of engine fuels system.

Table1. Fault diagnosis table

$U$	$k$	$a$	$b$	$c$	$d$	$g$
1	15	0	2	1	2	1
2	16	1	0	0	0	1
3	15	2	1	1	2	2
4	10	2	1	1	0	2
5	15	1	2	1	2	3
6	4	0	0	0	0	3

Table2. Matrix  $M_{(\{g_1\}, \{g_2\})}$

$\{g_1\}, \{g_2\}$	$u_3$	$u_4$	
$u_1$	$a \vee b$	$a \vee b \vee d$	$a_0 \vee b_2$
$u_2$	$a \vee b \vee c \vee d$	$a \vee b \vee c$	$a_1 \vee b_0 \vee c_0$
	$a_2 \vee b_1$	$a_2 \vee b_1 \vee (d_0 \wedge c_1)$	

Where  $U=\{1,2,\dots,6\}$  denotes 6 fault states of engine fuels system, and  $k$  is the number of samples corresponding each fault state.  $C=\{a,b,c,d\}$  is a condition attribute set in which each one express a kind wave shape feature of the pressure on pipe of diesel engine. The meaning of the values of omen condition attribute  $a$ ,  $b$ , and  $d$  are: 0 denotes down, 1 denotes invariant, and 2 denotes increase; the meaning of the values of omen condition attribute  $c$  are: 0 denotes none, 1 denotes have. There are three kinds of diagnose results in decision condition  $g$  that respectively denotes 1-abnormal in oil provide, 2-oil provided forward or lag, and 3-abnormal in the pressure of spray.

Firstly, the partition of decision classes can be calculated as :  $U/\{g\}=\{\{g_1\},\{g_2\},\{g_3\}\}=\{\{u_1,u_2\},\{u_3,u_4\},\{u_5,u_6\}\}$

Secondly, one can obtains discernibility matrixs  $M_{(\{g_1\},\{g_2\})}$ ,  $M_{(\{g_3\},\{g_4\})}$ ,  $M_{(\{g_5\},\{g_6\})}$  as show in Tables 2-4.

Table3. Matrix  $M_{(\{g_1\}, \{g_2\})}$

$\{g_1\}$	$\{g_2\}$	$u_5$	$u_6$	
$u_1$	$a$	$b \vee c \vee d$	$(a_0 \wedge b_2) \vee (a_0 \wedge c_1) \vee (a_0 \wedge d_2)$	
$u_2$	$b \vee c \vee d$	$A$	$(a_1 \wedge b_0) \vee (a_1 \wedge c_0) \vee (a_1 \wedge d_0)$	
	$(a_1 \wedge b_2) \vee (a_1 \wedge c_1) \vee (a_1 \wedge d_2)$	$(a_0 \wedge b_0) \vee (a_0 \wedge c_0) \vee (a_0 \wedge d_0)$		

Table 4. Matrix  $M_{(\{g_3\}, \{g_4\})}$

$\{g_3\}$	$\{g_4\}$	$u_5$	$u_6$	
$u_3$	$a \vee b$	$a \vee b \vee c \vee d$	$a_2 \vee b_1$	
$u_4$	$a \vee b \vee d$	$a \vee b \vee c$	$a_2 \vee b_1 \vee (d_0 \wedge c_1)$	
	$a_1 \vee b_2$	$a_0 \vee b_0 \vee c_0$		

In tables 2-4, last rows and columns of these respectively express the corresponding object's relative decision rule.

Lastly, we can obtain all minimal requirement sets of all decision classes as follows.

The absolute decision requirement for decision class  $\{g_1\}$  is:  $\{g_1\} \prec \{\{g_2\} \cup \{g_3\}\} = \{u_1 \propto \{\{g_2\} \cup \{g_3\}\}, u_2 \propto \{\{g_2\} \cup \{g_3\}\}\}$ . Conjoin the counterpart's unit in the last rows of the table2 and table 3 we get  $\{g_1\} \prec \{\{g_2\} \cup \{g_3\}\} = \{(a_0 \wedge b_2) \vee (a_0 \wedge c_1) \vee (a_0 \wedge d_2), (a_1 \wedge b_0) \vee (a_1 \wedge c_0) \vee (a_1 \wedge d_0)\}$ , Disjoin every requirement in set  $\{g_1\} \prec \{\{g_2\} \cup \{g_3\}\}$  we obtained  $d_{\{g_1\}}^{U-\{g_1\}} = (a_0 \wedge b_2) \vee (a_0 \wedge c_1) \vee (a_0 \wedge d_2) \vee (a_1 \wedge b_0) \vee (a_1 \wedge c_0) \vee (a_1 \wedge d_0)$ , so find decision rules of decision class  $\{g_1\}$  is  $(a_0 \wedge b_2) \vee (a_0 \wedge c_1) \vee (a_0 \wedge d_2) \vee (a_1 \wedge b_0) \vee (a_1 \wedge c_0) \vee (a_1 \wedge d_0) \rightarrow g_1$ . Finding out 9 minimal requirement set of class  $\{g_1\}$  are  $\{a_0 \wedge b_2, a_1 \wedge b_0\}, \{a_0 \wedge b_2, a_1 \wedge c_0\}, \{a_0 \wedge b_2, a_1 \wedge d_0\}, \{a_0 \wedge c_1, a_1 \wedge b_0\}, \{a_0 \wedge c_1, a_1 \wedge c_0\}, \{a_0 \wedge c_1, a_1 \wedge d_0\}, \{a_0 \wedge d_2, a_1 \wedge b_0\}, \{a_0 \wedge d_2, a_1 \wedge c_0\}, \{a_0 \wedge d_2, a_1 \wedge d_0\}$ .

The absolute decision requirement for decision class  $\{g_2\}$  is:

$\{g_2\} \prec \{\{g_1\} \cup \{g_3\}\} = \{u_3 \propto \{\{g_1\} \cup \{g_3\}\}, u_4 \propto \{\{g_1\} \cup \{g_3\}\}\}$ . Conjoin the counterpart's unit in the columns row of the table2 and table 4 we get  $\{g_2\} \prec \{\{g_1\} \cup \{g_3\}\} = \{a_2 \vee b_1, a_2 \vee b_1 \vee (d_0 \wedge c_1)\}$ , Disjoin every requirement in set  $\{g_2\} \prec \{\{g_1\} \cup \{g_3\}\}$  we obtained  $d_{\{g_2\}}^{U-\{g_2\}} = a_2 \vee b_1 \vee (d_0 \wedge c_1)$ , so find decision rules of decision class  $\{g_2\}$  is  $a_2 \vee b_1 \vee (d_0 \wedge c_1) \rightarrow g_2$ . Finding out 9 minimal requirement set of class  $\{g_2\}$  are  $\{a_2\}, \{b_1\}$ .

The absolute decision requirement for decision class  $\{g_3\}$  is:  $\{g_3\} \prec \{\{g_1\} \cup \{g_2\}\} = \{u_5 \propto \{\{g_1\} \cup \{g_2\}\}, u_6 \propto \{\{g_1\} \cup \{g_2\}\}\}$ . Conjoin the counterpart's unit in the columns row of the table 3 and table 4 we get  $\{g_3\} \prec \{\{g_1\} \cup \{g_2\}\} = \{(a_1 \wedge b_2) \vee (a_1 \wedge c_1) \vee (a_1 \wedge d_2), (a_0 \wedge b_0) \vee (a_0 \wedge c_0) \vee (a_0 \wedge d_0)\}$ , Disjoin every requirement in set  $\{g_3\} \prec \{\{g_1\} \cup \{g_2\}\}$  we obtained  $d_{\{g_3\}}^{U-\{g_3\}} = (a_1 \wedge b_2) \vee (a_1 \wedge c_1) \vee (a_1 \wedge d_2) \vee (a_0 \wedge b_0) \vee (a_0 \wedge c_0) \vee (a_0 \wedge d_0)$ , so find decision rules of decision class  $\{g_3\}$  is  $(a_1 \wedge b_2) \vee (a_1 \wedge c_1) \vee (a_1 \wedge d_2) \vee (a_0 \wedge b_0) \vee (a_0 \wedge c_0) \vee (a_0 \wedge d_0) \rightarrow g_3$ . Finding out 9 minimal requirement set of class  $\{g_3\}$  are  $\{a_1 \wedge b_2, a_0 \wedge b_0\}, \{a_1 \wedge b_2, a_0 \wedge c_0\}, \{a_1 \wedge b_2, a_0 \wedge d_0\}, \{a_1 \wedge c_1, a_0 \wedge b_0\}, \{a_1 \wedge c_1, a_0 \wedge c_0\}, \{a_1 \wedge c_1, a_0 \wedge d_0\}, \{a_1 \wedge d_2, a_0 \wedge b_0\}, \{a_1 \wedge d_2, a_0 \wedge c_0\}, \{a_1 \wedge d_2, a_0 \wedge d_0\}$ .

## Conclusion

Decision rules can be extracted through knowledge reduction from a decision table by using rough set theory. A lot of different methods of reduction be using as preprocess of rule generating. Since different methods of attributes reduction may lead to different results, so the reduction rules may not be the optimum, which may induce too much attributes or rules. In this paper, we put the concept of relative decision requirement, in which the relative discernibility matrix between decision classes can be created to extract the relative decision requirements which are conjoined to find the absolute decision requirements of all decision classes. The relative decision requirements are extracted by creating smaller discernibility matrix, so that scales of computing can be cut down substantially and saving storage space. In future, we intend to develop a parallel calculation method to calculate relative decision requirement of decision categories. Such research efforts are expected to further accelerate the extracting speed of the decision rule from decision table.

## Acknowledgement

In this paper, the research was sponsored by the grants from Nature Science Foundation of China (Project No. 61363047).

## References

- [1] Z.Pawlak, Rough sets, International Journal of Computer and Information Science, 11 (5) (1982) 341-356.
- [2] A. Skowron, S.K Pal, Rough Sets, Pattern Recognition, and Data Mining, Pattern Recognition Letters 24 (6) (2003) 829-933.
- [3] Hsu-Hao Yang, Tzu-Chiang Liu, Yen-Ting Lin, Applying rough sets to prevent customer complaints for IC packaging foundry, Expert Systems with Applications 32 (2007) 151-156.
- [4] Chitcharoen. D, Palttaraintakorn. P, Novel matrix forms of rough set flow graphs with applications to data integration, Computers & Mathematics with Applications 60 (2010) 2880-2897.
- [5] Qing Liu, Rough Set and Rough Inference. Beijing Science Press, 2001.
- [6] Min Li, Changxing Shang, Shengzhong Feng, Jianping Fan, Quick attribute reduction in inconsistent decision tables, Information Sciences, 2014(254):155-180.
- [7] S. Tsumoto, Automated extraction of medical expert system rules from clinical databases based on rough set theory, Information Sciences 112 (1-4) (1998) 67-84.
- [8] Skowron A , Rauszer C. The discernibility functions matrices and functions in information systems [ C ]. Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht : Kluwer Academic Publisher, 1992: 331-362.
- [9] Wang Jue, Wang Ju. Reduction algorithms based on discerniblility matrix: The ordered attributed method [J ] . Journal of Computer Science and Tecnology, 2001,11 (6) : 489-504.
- [10]Huang Wen-tao, Zhao Xue-zeng. Extraction method of decision rules for fault diagnosis based on rough set theory: Proceeding of the Chinese Society for Electrical Engineering, 2003,23:151-155.