An update algorithm of decision rules in expert systems based on rough sets theory

Yehong Han[1*], Lin Du[1]

[1] *Department of Computer, Qilu Normal University, Jinan, CHINA*

Received 1 January 2014, www.tsi.lv

Abstract

The decision table in rough sets theory is a kind of prescription, which specifies what actions should be undertaken when some of conditions are satisfied. Therefore, this tool can be used as knowledge representation system in expert systems. Decision rules, which are obtained by simplification of decision tables, can be used as rationale of decision reasoning. In order to compute new decision rules on the decision table in which a new instance is added, new instances are classified three cases according to the relation between the new instance and the original set of decision rules in the paper, and the category is proved that it is a partition of new instances. According to the category, an update algorithm of decision rules based on rough sets theory in expert systems is presented, and the complexity of the algorithm is obtained.

Keywords: rough sets, expert systems, incremental learning, decision table, algorithm.

1 Introduction

Expert systems [1] are computer intelligence systems, which can execute special tasks as experts. In other words, expert systems are that expertise is transferred from a human to a computer, and computers can be used as a human consultant. In order to make the computer gives advices and performances like an expert, some techniques must be employed. Now expert systems can provide very powerful and flexible methods for obtaining solutions to a lot of different problems that often cannot be dealt with by other, more traditional methods [2, 3]. Although many scholars have studied various methodologies for knowledge processing in expert systems, and the technology of expert systems has made great progress, there are still many problems, for example, most of research on expert systems limit to static data, and neglect update algorithm of knowledge bases in expert systems [4].

Rough sets theory [5] is a mathematical tool to data analysis. It was presented by Zdzislaw Pawlak in 1982. It can be used to deal with fuzzy and uncertainty information. Now, rough sets theory have been widely used in a variety of domains, such as decision support system, machine learning, expert systems, pattern recognition and others [6]. According to rough sets theory, a decision table is a kind of prescription, which specifies what actions should be undertaken when some of conditions are satisfied. Therefore, this tool especially suitable for expert systems and it can be used as knowledge representation system [7] of expert systems. Decision rules [8], which are obtained by simplification of decision tables, can be used as rationale of decision reasoning.

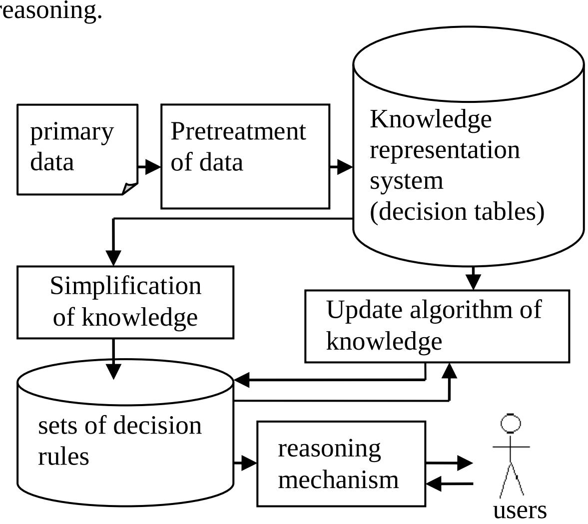

FIGURE 1 The model of expert systems based on rough

In the paper, a basic model of expert systems based on rough set theory is given as shown in figure1 by improving the traditional model. In order to improve the defect that the research of data in knowledge representation database of expert systems limit to static data, an update algorithm of knowledge based on the model of expert systems is presented. The new instance can be added to knowledge database of expert systems, and decision rules, which can be used as rationale of decision reasoning in expert systems will be updated by using the algorithm, and the algorithm is proved that it

* Corresponding author Tel: 13793161609
E-mail: sdzzhyh@163.com

can be used to consistent and inconsistent decision tables, and the complexity of the algorithm is given.

2 Rough Sets Theory

2.1 NOTATIONS AND DEFINITIONS OF ROUGH SETS

Knowledge representation system [9] is a pair $S=<U,A>$, where U is a nonempty and finite set called the universe, and A is a nonempty, finite set of primitive attributes. Every primitive attribute $a \in A$ is a total function $a: U \rightarrow Va$, where Va is the set of value of a, called the domain of a. Let $C,D \subset A$ be two subsets of attributes, called condition and decision attributes respectively. KR - system with distinguished condition and decision attributes will be called a decision table, and will be denoted as $S=<U,A>$.

Let P and Q are subsets of A. By P - Positive region of Q denoted $POS_F(Q)$, the set $POS_F(Q)$ can be computed by equation (1).

$$POS_F(Q)=\bigcup_{X \in U/IND(Q)} P\_(X) \tag{1}$$

where

$IND(Q)=\{(x,y) \in U^2 | f(x,b)=f(y,b), x \in U, y \in U, b \in Q\}$.

The family $R \subset C$ will be called a D - reduction of C, If and only if R is the D -independent subfamily of C and $POS_R(D)=POS_C(D)$.

In fact, the decision table can be viewed as a model for a set of propositions about reality, called here decision logic, which will be used to drive conclusions from data available in the Knowledge representation system. In decision logic language, there are definitions as follow:

The set of formulas in decision logic language is at least set satisfying the following conditions:

(1) Expression of the form (a,v) or in short a_v, called elementary formulas, are formulas of the DL - language for any $a \in A$ and $v \in V_a$.

(2) If θ, ψ are formulas of the DL - language, then so are $\neg\theta$, $\theta \vee \psi$, $\theta \wedge \psi$, $\theta \rightarrow \psi$ and $\theta \equiv \psi$.

An object $x \in U$ satisfies a formula θ in $S=<U,A>$, denoted $|x|=_s \theta$. Let $P=\{a_1,a_2,...,a_n\}$, and $P \subseteq A$, formula of the form $(a_1,v_1) \wedge (a_2,v_2) \wedge ... \wedge (a_n,v_n)$ will be called a P - basic formula. If θ is a P -basic formula and $R \subseteq P$, then by θ / R we mean the R -basic formula obtained from the formula θ by removing from θ all elementary formulas (a,v_a) such that $a \in P-R$.

In decision logic language, any implication $\theta \rightarrow \psi$ will be called a decision rule. θ and ψ are referred to as the predecessor and the successor of $\theta \rightarrow \psi$ respectively. If a decision rule $\theta \rightarrow \psi$ is true in S, we will say that the decision rule is consistent in S. If $\theta \rightarrow \psi$ is a decision rule, where θ and ψ are P -basic and Q - basic formulas respectively, then the decision rule will be called a PQ - basic decision rule, (in short PQ - rule). Any finite set of decision rules will be called a decision algorithm. The decision algorithm is consistent in S, if and only if all its decision rules are consistent in S. If all decision rules in a decision algorithm are PQ - basic decision rules, the algorithm is said to be PQ - decision algorithm, or in short PQ - algorithm and will be denoted as (P,Q).

2.2 SIMPLIFICATION OF DECISION TABLES

In [10], Zdzislaw Pawlak proposed that, in order to simplify a decision table, three steps as follows should be taken:

1) Reduce the set of attributes, i.e. remove all superfluous columns from decision table.

2) Simplify the decision rules, i.e. eliminate the unnecessary conditions in each rule of the algorithm separately.

3) Remove all duplicate decision rules from the algorithm.

Example 1

Supposing $S=<U,A>$, where $U=\{1,2,3,4,5,6,7,8\}$ and $A=\{a,b,c,d,e\}$. $C=\{a,b,c,d\}$ and $D=\{e\}$ are condition and decision attributes respectively. The decision table is shown in table 1.

TABLE 1 The decision table of example 1

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	0	0	1	0
6	1	1	0	2	2
7	2	2	0	2	2
8	2	2	2	2	2

It is easy to compute that the only e-dispensable condition attribute is c. Hence there is a D-reduction of the family C, and $R=\{a,b,d\}$ [11, 12].

The next step is to simplify the decision rules. In [10], only consistent decision rules will be reduced. In fact, inconsistent decision rule can be reduced too. In order to simplify both consistent and inconsistent decision rules, we give the following definitions.

Let $\theta \to \psi$ is a decision rule in S, $\overline{|\theta \to \psi|}_s$ are referred to as the set of all objects which corresponding decision rules' predecessor are the same as θ, and its successor are not the same as ψ. Obviously, if the decision rule is consistent, $\overline{|\theta \to \psi|}_s = \phi$. Otherwise $\overline{|\theta \to \psi|}_s \neq \phi$.

Let $\theta \to \psi$ is a RD-rule, and $a \in R$. We will say that attribute 'a' is dispensable in the rule $\theta \to \psi$ if and only if $\overline{|\theta \to \psi|}_s = \overline{|\theta / R - \{a\} \to \psi|}_s$. Otherwise, attribute '$a$' is indispensable.

If all attribute $a \in R$ are indispensable in $\theta \to \psi$, then $\theta \to \psi$ will be called independent.

The subset of attributes $R' \subseteq R$ will be called a reduction of RD - rule $\theta \to \psi$, when $\theta / R' \to \psi$ is independent and $\overline{|\theta \to \psi|}_s = \overline{|\theta / R' \to \psi|}_s$.

If R' is a reduction of the RD-rule $\theta \to \psi$, then $\theta / R' \to \psi$ is said to be reduced.

According to the above definitions, the reductions of each decision rule in the algorithm [13] as shown in Table 2 will be obtained.

TABLE 2. The reduction of table 1

U	a	b	d	e
1	*	0	1	1
2	1	*	0	1
3	0	*	*	0
4	*	1	1	0
5	*	0	1	0
6	*	*	2	2
7	2	*	*	2
7	*	2	*	2
7	*	*	2	2
8	2	*	*	2
8	*	2	*	2
8	*	*	2	2

Remove all superfluous decision rules from the Table 2, the following minimum set of decision rules will be got [14].

$b_0 d_1 \to e_1$ from rule 1

$a_1 d_0 \to e_1$ from rule 2

$a_0 \to e_0$ from rule 3

$b_1 d_1 \to e_0$ from rule 4

$b_0 d_1 \to e_0$ from rule 5

$d_2 \to e_2$ from rules 6, 7 and 8

The set of decision rules, which specifies what actions should be undertaken when some of conditions are satisfied is particularly useful in decision making of expert systems. It can be used as rationale of decision reasoning [15, 16, 17, 18].

Most of research on expert systems based on rough sets limit to static data. In other words, the set of instances U in knowledge representation system is constant and unchanged, which will be called Closed World Assumption. But in many real life situations however this is not the case, and new instances can be added to the set U. This situation will be called the Open World Assumption [19]. In order to compute the minimum set of rules of decision table when a new instance is added, all the data in the decision table should be recalculated in the classical method [20, 21, 22]. Obviously, this method is not effective. In the paper, an update algorithm of decision rules will be introduced in the following section.

3 An Update Algorithm of Decision Rules Based on Rough Sets Theory

3.1 THE CATEGORY OF NEW INSTANCES

In order to introduce the algorithm, new instances, which will be added to decision tables in expert systems should be classified according to the relationship between the new instance and the old decision table.

Let $S = (U, A)$ is a decision table, M is a minimum set of decision rules before added the new instance. The new instance is x, and its CD-basic rule is $\theta_x \to \psi_x$. On the premise, there are the following definitions.

If there is a rule or several rules $\theta \to \psi$ in the set of decision rules M satisfying $\theta_x \to \theta$, and every rule satisfying $\theta_x \to \theta$ implies $\psi_x \equiv \psi$, then this situation will be called x matches M.

If there is a rule or several rules $\theta \to \psi$ satisfying $\theta_x \to \theta$ in M, and every rule satisfying $\theta_x \to \theta$ implies $\psi_x \neq \psi$, then this situation will be called x is totally in contradiction with M.

A new instance is partially in contradiction with M when there are several rules $\theta \to \psi$ satisfying $\theta_x \to \theta$ in M, and not only exist rules satisfying $\theta_x \to \theta$ satisfies $\psi_x \equiv \psi$, but also exist rules satisfying $\theta_x \to \theta$ satisfies $\psi_x \neq \psi$ in M.

Both the totally and partially contradiction will be called x is in contradiction with M.

If there is not any rule $\theta \to \psi$ satisfying $\theta_x \to \theta$ in a minimum set of decision rules, then we will say the new instance x is completely new in M.

According to above definitions, new instances will be classified three cases according to the relationship between the new instance and the old decision table.

1) x matches M;
2) x is in contradiction with M;
3) x is completely new in M.

This classification is completely, and covered all situations that new instances may be, in other words, it is a partition of new instances.

Suppose a new instance's CD -basic rule is $\theta_x \rightarrow \psi_x$ .The set of all objects which corresponding decision rules $\theta \rightarrow \psi$ satisfy $\theta_x \rightarrow \theta$ and $\psi_x \neq \psi$ will be called the contradicting domain of the new instance x, and will be denoted Cx, and Cx can be obtained by equation (2).

$$Cx = \{i | i \in U, \forall \theta_i \rightarrow \psi_i \in M \; satisfy \; \theta_x \rightarrow \theta_i \; and \; \psi_x \neq \psi_i\} \quad (2)$$

The set of all the objects which corresponding decision rules $\theta \rightarrow \psi$ satisfy $\theta_x \rightarrow \theta$ and $\psi_x \equiv \psi$ will be called the matching domain of the new instance x, and will be denoted Mx, and Mx can be obtained by equation (3).

$$Mx = \{i | i \in U, \forall \theta_i \rightarrow \psi_i \in M \; satisfy \; \theta_x \rightarrow \theta_i \; and \; \psi_x \neq \psi_i\} \quad (3)$$

The rule $\theta_i \rightarrow \psi_i$ is referred to as the reduction of the corresponding rule of object i. Let $S = (U, A)$ is a decision table, and there are not same rows in the decision table. R is a reduction of condition attributes. M is a minimum set of decision rules before adding a new instance. A new instance is *x*, and let its CD - basic rule be $\theta_x \rightarrow \psi_x$. The new decision table will be denoted $\theta_x \rightarrow \psi_x$ after adding the new instance, where $U' = U \cup \{x\}$. By the above definitions, the following proposition hold.

Proposition 1

If for every object $y \in Cx$ does not imply $y |=_S \theta_x / R$ in S, a reduction of condition attributes of the new decision table $S' = (U', A)$ is R.

Proof: Because there is no an object $y \in Cx$ such that $y |=_S \theta_x / R$ in S, the new instance is not in contradiction with all the RD - basic rules including in S. When the new instance x is added to S, $POS_R(D)' = POS_R(D) \cup \{x\}$ and $POS_C(D)' = POS_C(D) \cup \{x\}$, where $POS_R(D)'$ and $POS_C(D)'$ are R - Positive region and C - Positive region of D respectively. Because $POS_C(D) = POS_R(D)$, $POS_C(D)' = POS_R(D)'$ is true, and R is the D - independent subfamily of C in universe U', a reduction of condition attributes of the new decision table $S' = (U', A)$ is R.

Proposition 2

If there is an object $y \in Mx$ implies $y |=_S \theta_x$ in S, then the decision table $S' = S$, and its reduction of attributes and minimum set of rules will not change.

Proof: there is a object $y \in Mx$ satisfying $y |=_S \theta_x$ in S, then the new instance have existed in the decision table. Hence, the decision table does not change when we add the new instance to it. Obviously, the reduction of condition attributes and the minimum set of rules of decision table S' will not change.

Proposition 3

If there is only one object $y \in Cx$ such that $y |=_S \theta_x$, and there is not any other object $i \in Cx$ and $i \neq y$ satisfying θ_x / R, then there is a reduction R' of condition attributes satisfying $R' \subseteq R$ of $S' = (U', A)$.

Proof: If there is only one object $y \in Cx$ satisfying $y |=_S \theta_x$ in S, then the new instance x will be in contradiction with the object satisfying θ_x. There is not any other object $i \in Cx$ satisfying θ_x / R, then $POS_R(D)' = POS_R(D) - \{y | y |=_S \theta_x\}$ and $POS_C(D)' = POS_C(D) - \{y | y |=_S \theta_x\}$. Hence, we will conclude that $POS_C(D)' = POS_R(D)'$. Because the number of elements of the positive region is reduced when the new instance is added to S, R may be not the D-independent subfamily of C in universe U'. There is a reduction R' of condition attributes satisfying $R' \subseteq R$ of decision table $S' = (U', A)$.

Proposition 4

If there are two objects $y \in Cx$ satisfying $y |=_S \theta_x$ at least, then a reduction of attributes of decision table $S' = (U', A)$ is R.

Proof: there are two objects $y \in Cx$ satisfying $y |=_S \theta_x$ at least, then $POS_R(D)' = POS_R(D)$ and $POS_C(D)' = POS_C(D)$. Hence, $POS_C(D)' = POS_R(D)'$. Because the positive region doesn't change after adding the new instance, R is the D-independent subfamily of C in universe U'. So a reduction of attributes of decision table $S' = (U', A)$ is R.

Proposition 5

Let $\theta_i \rightarrow \psi_i$ is a reduction of the corresponding CD - rule of the object i in S, and $i \notin$ $Cx \cup \{ j | \theta_j' \equiv \theta_i'$, $i \in Cx$, and $j \in Mx$, where θ_i' and θ_j' are corresponding C - basic formulas of objects i and j respectively$\}$. Let W be the set of all minimum set of rules that can be obtained by reducing S'. If a reduction of attributes doesn't change when a new instance is added to S, there is a minimum set of rules $M' \in W$ satisfying $\theta_i \rightarrow \psi_i \in M'$.

Proof: because $i \notin Cx \cup \{ j \mid \theta_j' \equiv \theta_i',\ i \in Cx$, and $j \in Mx$, where θ_i' and θ_j' are corresponding C - basic formulas of objects i and j respectively$\}$, obviously, the rule $\theta_x \to \psi_x$ is not in contradiction with the decision algorithm composed by rules satisfying $\theta_i \to \psi_i$. Hence there is a minimum set of rules $M' \in W$ satisfying $\theta_i \to \psi_i \in M'$.

According to above propositions, following corollaries will be established.

1) If the new instance x matches M, a minimum set of decision rules of decision table S' is still M.

2) If the new instance x is completely new in M, a reduction of condition attributes of S' is still R, and a minimum set of rules, which can be denoted M' can be computed by equation (4),

$$M' = M \cup \{\theta_x' \to \psi_x \mid \theta_x' \to \psi_x \text{ is a reduction of } \theta_x \to \psi_x\} \quad (4)$$

3) if a reduction of attributes doesn't change after adding a new instance, in order to compute the minimum set of rules, we need simplify corresponding RD -rules of objects in $Cx \cup \{ j \mid \theta_j' \equiv \theta_i',\ i \in Cx$, and $j \in Mx$, where θ_i' and θ_j' are corresponding C -basic formulas of objects i and j respectively$\} \cup \{x\}$.

According to above propositions and corollaries, the following method shown in figure 2 can be used to compute the minimum set of rules. And the method can be used in the update algorithm of decision rules in expert systems.

3.2 AN UPDATE ALGORITHM OF DECISION RULES

Input: $S = (U, A)$ (S is a decision table);

R (R is a reduction of attributes of S);

M (M is a minimum set of rules of decision table S based on the attribute reduction R);

$\theta_x \to \psi_x$ ($\theta_x \to \psi_x$ is the new instance's CD-basic Rule).

Output: M' (M' is a Minimum Set of Rules of Decision Table $S' = (U' = U \cup \{x\}, A)$).

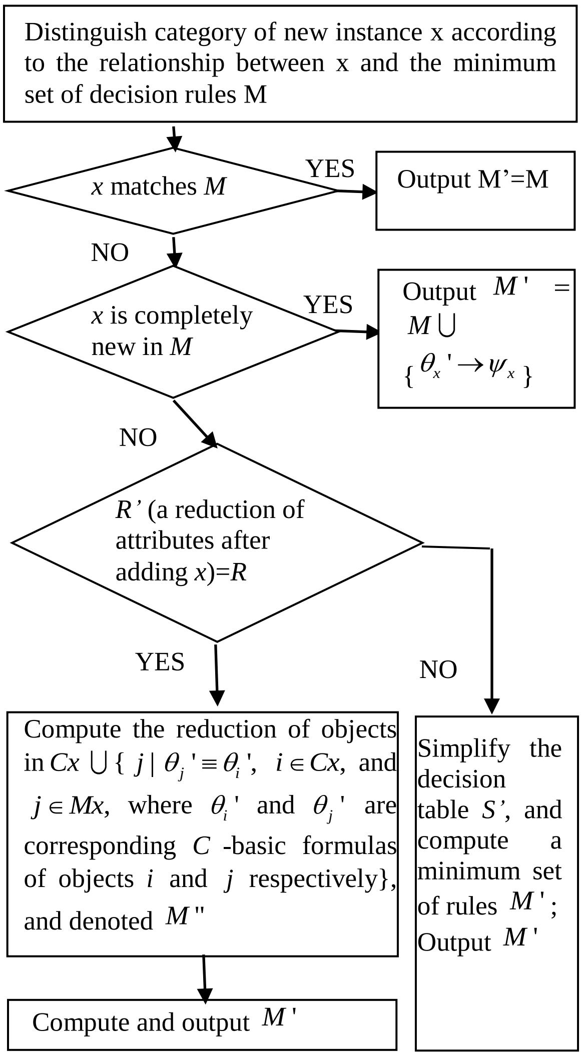

FIGURE 2 Algorithm to simplify decision tables

Algorithm:

Step1: $M1 = M2 = M3 = \phi$;

Step2: FOR every rule $\theta \to \psi$ in M

IF $\theta_x \to \theta$ is not true

THEN $M1 = M1 \cup \{\theta \to \psi\}$

ELSE IF $\psi_x \equiv \psi$

THEN $M2 = M2 \cup \{\theta \to \psi\}$

ELSE $M3 = M3 \cup \{\theta \to \psi\}$

Step3.1:

IF $M2 = M3 = \phi$

THEN {

Compute reductions of the rule $\theta_x \to \psi_x$,

and a result is denoted as $\theta_x' \to \psi_x$;

$M' = M \cup \{\theta_x' \to \psi_x\}$;

}

Step3.2:

ELSE IF $M3 = \phi$

THEN $M' = M$;

Step3.3:

ELSE {

Step3.3.1: Compute the contradicting domain of x:

$$Cx = \left\{ \begin{array}{l} i \mid i \text{ is an object which} \\ \text{corresponding rule is in} M3 \end{array} \right\};$$

Step3.3.2: Compute the matching domain of x:

$$Mx = \left\{ \begin{array}{l} i \mid i \text{ is an object which} \\ \text{corresponding rule is in} M2 \end{array} \right\};$$

Step3.3.3: IF $\exists y \in Mx$ satisfying $y \models_S \theta_x$

THEN $M' = M$

Step3.4:

ELSE {

Step3.4.1 Compute the set:

$C_{Cx} = \{ \theta_j \to \psi_j \mid \theta_j \equiv \theta_i, i \in Cx, j \in Mx \}$;

(Where $\theta_j \to \psi_j$ is the corresponding rule of object j)

IF do not exist $y \in Cx$

Satisfying $y \models_S \theta_x / R$

THEN $R' = \{R\}$

ELSE IF at least exist two objects

$y \in Cx$ Satisfying $y \models_S \theta_x$

THEN $R' = \{R\}$

ELSE IF only exist an object

$y \in Cx$ satisfying $y \models_S \theta_x$

THEN compute the set of all

The D-reductions of R in

U' and denoted R'

ELSE compute the set of all

the D-reductions of C in

U' and denoted R';

Step3.4.2 IF $R \in R'$

THEN {

Compute the reduction of Corresponding decision rules of objects in $Cx \bigcup \{ j \mid \theta_j' \equiv \theta_i'$, $i \in Cx$, and $j \in Mx$, where θ_i' and θ_j' are corresponding C -basic formulas of objects i and j respectively} $\bigcup \{x\}$, and denoted M'';

$M' = M'' \bigcup M1 \bigcup M2 - C_{C_x}$;

}

Step3.5

ELSE {

Simply all the decision rules in S' according to a reduction of attributes which have computed, and obtain a minimum set of rules M'

}

}

}

Step4: OUTPUT M'.

Let m and n be referred to as the quantity of elements in the set of condition attributes and universe respectively. Obviously, if x matches M, or x is completely new in M, the time-complexity of the algorithm is $O(mn)$. If x is in contradiction with M and there is not an object $y \in Cx$ satisfying $y \models_S \theta_x / R$ or there are two objects $y \in Cx$ satisfying $y \models_S \theta_x$ at the same, the time-complexity of the algorithm is $O(mn^2)$. At those situations, the algorithm is better than the classical algorithm obviously. The time-complexity of the classical algorithm is $O(2^m n^2)$ [23, 24]. But in other situations, the efficiency of this algorithm is worse than the classical algorithm little, and its time-complexity is $O(2^m n^2)$ too.

3.3 EXAMPLES

In order to understand the algorithm, examples of decision rules update will be given as follows:

Example 2:

If a new instance $a_0 b_1 c_2 d_1 \to e_0$ will be added to the decision table as shown in table1. Because $a_0 b_1 c_2 d_1 \to a_0$, and $e_0 \equiv e_0$, the new instance matches the minimum set of decision rules obtained by example 1, According to above algorithm, the new minimum set of decision rules will be obtained as follows:

$b_0 d_1 \to e_1$ from rule 1
$a_1 d_0 \to e_1$ from rule 2
$a_0 \to e_0$ from rule 3, 9
$b_1 d_1 \to e_0$ from rule 4
$b_0 d_1 \to e_0$ from rule 5
$b_0 d_1 \to e_0$ from rule 6, 7, 8

Example 3:

If the new instance is $a_0 b_0 c_1 d_1 \to e_1$, the new minimum set of decision rules is as follows:

* $a_1 b_0 d_1 \to e_1$ from rule 1
$a_1 d_0 \to e_1$ from rule 2
* $a_0 d_0 \to e_0$ from rule 3
$b_1 d_1 \to e_0$ from rule 4
* $a_1 b_0 d_1 \to e_0$ from rule 5
$d_2 \to e_2$ from rule 6, 7, 8
* $a_0 d_1 \to e_1$ from rule 9

Because the new instance is in contradiction with M, and there is not an object $y \in Cx$ satisfying $y \models_S \theta_x / R$, the reduction of corresponding decision rules of objects in $Cx \cup \{ j \mid \theta_j' \equiv \theta_i', i \in Cx$, and $j \in Mx$, where θ_i' and θ_j' are corresponding C-basic formulas of objects i and j respectively$\} \cup \{x\}$ should to be computed only.

Example 4:

If the new instance is $a_1b_2c_1d_1 \to e_2$, the new instance is completely new in M. Because If the new instance is completely new in M, a reduction of condition attributes of S' is still R, and a minimum set of rules is M', and

$$M' = M \cup \left\{ \theta_x' \to \psi_x \middle| \theta_x' \to \psi_x \text{ is a reduction of } \theta_x \to \psi_x \right\},$$

the minimum set of decision rules is as follows:.

$b_0d_1 \to e_1$	from rule 1
$a_1d_0 \to e_1$	from rule 2
$a_0 \to e_0$	from rule 3
$b_1d_1 \to e_0$	from rule 4
$b_0d_1 \to e_0$	from rule 5
$d_2 \to e_2$	from rule 6, 7, 8
$* b_2 \to e_2$	from rule 9

4 Conclusions

In the paper, decision tables are used as knowledge representation in expert systems and the minimum set of rules are used as rationale of decision reasoning, based on this, a basic model of expert systems based on rough set theory is given. And the update algorithm of knowledge representation system in expert systems is discussed in detail. In order to compute new decision rules, new instances, which will be added to decision tables in expert systems are classified three cases according to the relation between the new instance and the minimal set of decision rule that have been computed. The category is a partition of all new instances. According to the category, an update algorithm of decision rules is presented. The algorithm can be used in a variety of other domains, such as machine learning, data analysis and so on. But when the new instance is in contradiction with M and the reduction of condition attributes is changed after adding the new instance, all the data must be recalculated. This situation must be researched forward.

Acknowledgements

This work was supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2013FL010).

References

[1] Russell S J, Norvig P 1995 *Artificial intelligence: a modem approach*, Prentice hall

[2] Jackson P 1998 *Introduction to Expert Systems* Boston, MA, USA: Addison-wesley Longman publishing Co Inc

[3] Jonassen, D H, Wilson B G, Wang S and Grabinger R S 1993 *Journal of Computer Based Instruction* **20**(3) 86-94

[4] Siler W, Buckley J J 2005 *Fuzzy Expert systems and Fuzzy Reasoning*, John Wiley & Sons Inc.

[5] Pawlak Z 1991 *ROUGH SETS: Theoretical Aspects of Reasoning about Data* Kluwer Academic Publishers

[6] Wang Guo-Yin, Yao Yi-Yu, Yu Hong 2009 A Survey on Rough Set Theory and Applications *Chinese Journal of Computers* **7**, 1229-46

[7] Zdzislaw Pawlak 2002 *Information Sciences* **147**(1-4), 1-12

[8] Yunliang Jiang, Congfu Xu, Jin Gou, Zuxin Li 2004 Research on Rough Set Theory Extension and Rough Reasoning, *2004 IEEE International Conference on Systems, Man and Cybernetics*, 5888-93

[9] Hong Shi, Jin-Zong Fu 2005 A Global Discretization Method Based on Rough Sets, *Machine Learning and Cybernetics*, 3053-57

[10] Zdzislaw Pawlak, Andrzej Skowron 2007 *Information Sciences*, **177** 3-27

[11] HU Qing-Hua, YU Da-Ren, XIE Zong-Xia 2008 *Journal of Software* **19**(3), 640-649

[12] Xiaodong Liu, Witold Pedrycz 2009 *IEEE Transactions On Knowledge And Data Engineering* **21**(3) 443-462

[13] Thangavel K, Pethalakshmi A 2009 *Applied soft computing* **9**(1) 1-12

[14] Yee Leung, Wei-Zhi Wu, Wen-Xiu Zhang 2006 *European Journal of Operational Research* **168** 164-80

[15] Huang B, Li H, Wei D 2012 *Knowledge-Based Systems* **28** 115-123

[16] Fazel Zarandi M H, Rezaee B, Turksen I B, Neshat E 2009 *Expert Systems with Applications* **36**(1) 139-154

[17] Yuhua Qian, Jiye Liang, Chuangyin Dang 2010 *IEEE* **40**(2) 420-31

[18] Zdzislaw Pawlak 1997 *European Journal of Operational Research* **99**(1) 48-57

[19] Jian Yu, Shuigeng Zhou, Lipo Wang, Jingsheng Lei 2009 *Computers & Mathematics with Applications* **57**(6) 865-6

[20] Yunfei Yin, Guanghong Gong, Liang Han 2009 *Computers & Mathematics with Applications* **57**(1) 117-26

[21] Qinghua Hu, Daren Yu, Maozu Guo 2010 *Information Sciences* **180**(10) 2003-22

[22] Hong Yu, Dachun Yang, Hong Tang, Zhongfu Wu 2003 *Computer Engineering and Application* **33** 38-41

[23] Dongya Li, Baoqing Hu 2007 A Kind of Dynamic Rough Sets, *Fourth International Conference on Fuzzy Systems and Knowledge Discovery* (FSKD 2007)

[24] Herbert J P, Jing Tao Yao 2009 *Computers & Mathematics with Applications* **57**(6),908-18

Authors

Yehong Han, born in 1981

Current position, grades: a lecturer with Department of Computer at Qilu Normal University.
Scientific interest: M.S. degree in Management Science and Engineering from Shandong Normal University.
Scientific interest: data mining, data analysis, decision support systems and patter recognition.

Lin Du, born in 1978

Current position, grades: an associate professor with Department of Computer at Qilu Normal University.
University studies: M.S. degree in Computer Science and Technology from Shandong University.
Scientific interest: software reengineering, system comprehension, software measurement and image retrieval.