Incremental Learning of Decision Rules Based on Rough Set Theory

Tong Lingyun
School of Management
Hebei University of Technology
Tianjin, 300130
China

An Liping
College of International Business
Nankai University
Tianjin, 300071
China

***Abstract* – With the changes of databases, the former rule sets obtained from the data set require updating. We want that the algorithms of rule generation are incremental learning methodology for modification of the existing decision rules and their numerical measures when new objects are appended to the database, instead of running the whole learning process again. In this paper, based on rough set theory, the concept of ∂-indiscernibility relation is put forward in order to transform an inconsistent decision table to one that is consistent, called ∂-decision table, as an initial preprocessing step. Then the ∂-decision matrix is constructed. On the basis of this, by means of decision function, an algorithm for incremental learning of rules is presented. The algorithm can also incrementally modify some numerical measures of a rule.**

1. INTRODUCTION

Symbolic learning methods have the ability to automatically generate a set of classification rules in an if ..then ..or a decision tree form from a given set of training objects, which can be used to help the users better understand the relationships occurring in the domain of interest.

Rough sets theory, introduced by Pawlak [1], offers an approach to acquiring all maximally generalized classification rules with a minimized number of conditions from decision tables. But, most of the existing algorithms of rule induction cannot generate incrementally when new objects are derived. Then, the rule induction algorithms have to be run again when such new objects are given, which leads to the computational complexity to be expensive. Shan and Ziarko have discussed induction of deterministic rule in an incremental way [2]. Whereas, non-deterministic rules are ubiquitous. And, numerical measures of a rule such as coverage, accuracy, etc can not be updated simultaneously in their method.

In this paper, a new approach to incremental induction of deterministic as well as non-deterministic rules and their numerical measures is put forward based on the ∂-indiscernibility relation and ∂-decision matrix.

2. BASIC CONCEPTS

In this section, we first introduce some of the elementary concepts ([3], [4]) used in this paper, then the concept of ∂-decision matrix is presented to construct decision functions.

By an information table S, we mean $S = \{U, A, V\}$, where U is a finite set of objects, $U=\{x_1, x_2, \ldots, x_n\}$, A is a finite set of attributes, i.e., a: $U \rightarrow V_a$ for a A, where V_a is the domain of a, and $V = \bigcup_{a\in A} V_a$.

A decision table is any information table S of the form $S=(U, C\cup\{d\}, V)$, where $d \notin C$ is a distinguished attribute called decision attribute. The elements of C are called condition attributes.

Let $S=(U, A, V)$ be an information table, every non-empty subset of attributes $B \subseteq A$ is associated an indiscernibility relation on U, denoted by $IND(B)$:

$$IND(B) = \{(x, y) \in U^2 : \forall a \in B, a(x) = a(y)\}.$$

If $(x, y) \in IND(B)$, it is said that the objects x and y are B-indiscernible. Clearly, the indiscernibility relation is an equivalence relation. The family of all the equivalence classes of the relation $IND(A)$ is denoted by $U/IND(A)= \{X_1, X_2, \ldots, X_n\}$, where X_i $(i=1,2, \ldots, n)$ is called i-th condition equivalence class.

Let $S=(U, C\cup\{d\}, V)$ be a decision table. The decision attribute d induces a partition of the universe of objects U. Without loss of generality, we may assume that V_d is the set of integers $\{1, 2, \ldots, r(d)\}$, where $r(d)$ is said to the rank of d. The induced partition is therefore the collection of equivalence classes $\{Y_1, Y_2, \ldots, Y_{r(d)}\}$, called decision classes, where two objects are said to belong to the same decision class if they have the same value for the decision attribute, that is $Y_i = \{x \in U : d(x) = i\}$, $i= 1, 2, \ldots, r(d)$.

If $S=(U, C\cup\{d\}, V)$ be a decision table and $B \subseteq C$, then we define a function $\partial_B : U \rightarrow P\{1,2,\cdots,r(d)\}$, called the B-generalized decision in S, by

$$\partial_B(x) = \{i : \exists x' \in U \ s.t.\ x' IND(B) x \text{ and } d(x') = i\},$$

where $P\{1, 2, \ldots, r(d)\}$ is the power set of V_d.

In a decision table S it may happen that two objects that are indiscernible with respect to attributes C may belong to different decision classes. The decision table S is said to be inconsistent with respect to C if this is the case, and consistent otherwise. Note that the decision table S is consistent if and only if $\partial_C(x)$ are singletons for all $x \in U$.

Let $S=(U, C\cup\{d\}, V)$ be a decision table, $B \subseteq C$, the ∂_B-indiscernibility relation on U is defined as:

$$IND(\partial_B) = \{(x, y) \in U^2 : \partial_B(x) = \partial_B(y)\},$$

denoted by $IND(\partial_B)$.

∂_B-indiscernibility relation determines a partition $\{Z_1, Z_2, \ldots, Z_m\}$, where Z_j (j=1, 2, ..., m) is called j-th ∂_B-decision class. Furthermore, by replacing d with ∂_C, the decision table obtained from S is necessarily consistent, called ∂-decision table. This property is often exploited in practice, where in some procedures a decision table is assumed to be consistent. An inconsistent decision table can thus always be transformed to one that is consistent as an initial preprocessing step.

If $U/IND(C)=\{X_1, X_2, \ldots, X_n\}$, and $x_p, x_q \in X_i$, i=1, 2, ..., n, then $\partial_C(x_p)=\partial_C(x_q)$. So we can define a function ∂_C : $U/IND(C) \rightarrow P\{1, 2, \ldots, r(d)\}$, called the generalized decision of equivalence class in S, by $\partial_c(X_i)=\{v : \exists x \in X_i \ s.t. \ d(x)=v\}$.

Let $S=(U, C \cup \{d\}, V)$ be a decision table, $B \subseteq C$, $U/IND(B)=\{X_1, X_2, \ldots, X_n\}$, a: $U/IND(B) \rightarrow P(V_a)$, for $a \in B$, the ∂-decision matrix of S is defined as

$$M(S)=(m_{ij})_{n \times n},$$

where $P(V_a)$ is the power set of V_a, and $m_{ij}=\{a \in C : a(X_i) \neq a(X_j) \text{ and } \partial_B(X_i) \neq \partial_B(X_j)\}$.

Suppose $M(S)=(m_{ij})_{n \times n}$ is the ∂-decision matrix of S, the decision function of X_i (i=1, 2 ..., n) is defined as

$$B_i = \bigwedge_j \bigvee_{a \in m_{ij}} a,$$

where j=1, 2, ..., n, $\wedge$ and $\vee$ are respectively generalized conjunction and disjunction operators. Turning B_i into disjunctive normal form and using the absorption law of Boolean algebra to simplify it, the conjuncts, i.e., the prime implicants of the simplified decision function correspond to the minimal decision rules.

3. NUMERICAL MEASURES OF A RULE

In this section, the computational method of the numerical measures of a rule is put forward based on the parameters describing the relation between condition equivalence classes and decision classes, that is $X_i \rightarrow Y_j$.

Let $S=(U, C \cup \{d\}, V)$ be a decision table, $U/IND(C)=\{X_1, X_2, \ldots, X_n\}$, $U/IND(\{d\})=\{Y_1, Y_2, \ldots, Y_m\}$. The relation $X_i \rightarrow Y_j$ (i=1, 2, ..., n; j=1, 2, ..., m) between $U/IND(C)$ and $U/IND(\{d\})$ is determined by S. In this paper, four parameters are used to describe $X_i \rightarrow Y_j$, i.e., (1) $|X_i|$; (2) $|Y_j|$; (3) $SI(X_i, Y_j)=(|X_i \cap Y_j|)/|X_i|$, denoted by SI_{ij} [5]; (4) $CI(X_i, Y_j)=(|X_i \cap Y_j|)/|Y_j|$, denoted by CI_{ij} [5].

Minimal decision rules of each X_i can be generated using ∂-decision matrix and decision functions. The parameters of $X_i \rightarrow Y_j$ are naturally passed to each rule and the numerical measures of a rule r is then obtained:

(1) $support(r)=|X_i \cap Y_j|$;

(2) $accuracy(r)=SI_{ij}$;

(3) $coverage(r)=CI_{ij}$.

If some of the rules, which have the same form (i.e., same antecedents and consequents), are induced from different equivalence classes within one ∂-decision class, these rules should be regarded as a single rule of which numerical measures require computing based on the parameters of the equivalence classes inducing the rule. Suppose a rule r is induced from $X_i \rightarrow Y_j$ and $X_{i'} \rightarrow Y_j$ of which parameters are ($|X_i|$, $|Y_j|$, SI_{ij}, CI_{ij}) and ($|X_{i'}|, |Y_j|, SI_{i'j}, CI_{i'j}$), respectively. Then the numerical measures of the rule r are computed as follows:

$$support(r)=|(X_i \cup X_{i'}) \cap Y_j|$$

$$accuracy(r)=(|X_i \cap Y_j|+|X_{i'} \cap Y_j|)/(|X_i|+|X_{i'}|)$$
$$=(SI_{ij} \times |X_i|+SI_{i'j} \times |X_{i'}|)/(|X_i|+|X_{i'}|)$$

$$coverage(r)=(|X_i \cap Y_j|+|X_{i'} \cap Y_j|)/|Y_j|=CI_{ij}+CI_{i'j}$$

The computing process can be extended to more than two equivalence classes inducing one rule.

4. RULE INDUCTION

The algorithm of rule induction from a decision table presented by the authors [6] is in the following way:

STEP 1. Examine if the decision table is consistent. If the table is inconsistent, then the ∂-indisnibility relation is used to transform it into a consistent one, i.e., a ∂-decision table S';

STEP 2. Arrange all of the objects in the order of ∂-decision classes. Compute the parameters of $X_i \rightarrow Y_j$;

STEP 3. Construct ∂-decision matrix of S', i.e., $M(S')=(m_{ij})_{n \times n}$. We can only consider the upper diagonal part of the $M(S')$ for the matrix is symmetrical;

STEP 4. Construct the decision function B_i of each equivalence class X_i (i=1, 2, ..., n):

$B_i=(\bigwedge_{k<i} \bigvee_{a \in m_{ki}} a) \wedge (\bigwedge_{l>i} \bigvee_{a \in m_{il}} a)$, where k=1, 2, ..., i-1, l = i+1, ..., n;

STEP 5. Using the absorption law, calculate the disjunctive normal form of B_i where each conjunct corresponds to a minimal decision rule;

STEP 6. Repeat the STEP 4 and 5, then the decision rules of all the equivalence classes are obtained. Calculate the numerical measures of the rules and the eventual rule set is obtained.

Example 1 Table 1 depicts a decision table whose equivalence classes are X_i, i = 1, ...,11.

The decision table is inconsistent. Thus, the following ∂-decision classes are constructed: $Z_1=X_1 \cup X_2 \cup X_3$, $Z_2=X_4 \cup X_5$, $Z_3=X_6 \cup X_7$, $Z_4=X_8 \cup X_9$, $Z_5=X_{10}$, $Z_6=X_{11}$. Using ∂-decision matrix, the rules and parameters of $X_i \rightarrow Y_j$ are computed as shown in Table 2.

The rule $(d,0)\rightarrow(e,2)\vee(e,1)$ is induced from X_8 as well as X_9 in Table 2. Therefore, the numerical measures of the rule must be calculated further. With regard to $(d,0)\rightarrow(e,2)$,

$$support((d,0)\rightarrow(e,2))$$
$$=|(X_8 \cup X_9)\cap Y_2|$$
$$=41$$

$$accuracy((d,0)\rightarrow(e,2))$$
$$=(SI_{82}\times|X_8|+SI_{92}\times|X_9|)/(|X_8|+|X_9|)$$
$$=(0.888889\times 45+0.5\times 2)/(45+2)$$
$$=0.87234$$

$$coverage((d,0)\rightarrow(e,2))$$
$$=CI_{82}+CI_{92}$$
$$=0.412371+0.010309$$
$$=0.42268$$

Similarly, the numerical measures of $(d,0)\rightarrow(e,1)$, $(b,0)\wedge(c,1)\rightarrow(e,0)$ and $(a,0)\wedge(d,2)\rightarrow(e,1)$ are computed. The measures of the other rules are obtained directly from the parameters of $X_i\rightarrow Y_j$.

TABLE I
A DECISION TABLE

Equivalence classes	Cardinal of X_i	Condition attributes				Decision attribute
		a	b	c	d	e
X_1	15	0	0	1	1	0
X_2	10	1	0	1	2	0
X_3	5	1	1	0	1	0
X_4	60	0	1	0	2	1
X_5	10	0	2	1	2	1
X_6	30	1	2	0	1	2
X_7	25	2	0	0	1	2
X_8	40	2	1	2	0	2
	5	2	1	2	0	1
X_9	1	1	2	1	0	2
	1	1	2	1	0	1
X_{10}	1	1	2	2	1	0
	1	1	2	2	1	1
	1	1	2	2	1	2
X_{11}	1	1	1	2	2	3

TABLE 2
∂-DECISION MATRIX AND RULES FOR TABLE 1

∂-decision classes	Equivalence classes	∂-Decision matrix								Rules		Parameters of $X_i\rightarrow Y_j$
		Z_2		Z_3		Z_4		Z_5	Z_6			
		X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	Antecedents of the rules	Consequents of the rules	(SI_{ij}, CI_{ij})
Z_1	X_1	bcd	bd	abc	ac	abcd	abd	abc	abcd	$(a,0)\wedge(b,0)$, $(a,0)\wedge(d,1)$, $(b,0)\wedge(c,1)$, $(c,1)\wedge(d,1)$	$(e,0)$	(1.0, 0.483871)
	X_2	abc	ab	bcd	acd	abcd	bd	bcd	bc	$(a,1)\wedge(b,0)$, $(a,1)\wedge(c,1)\wedge(d,2)$, $(b,0)\wedge(c,1)$, $(b,0)\wedge(d,2)$	$(e,0)$	(1.0, 0.322581)
	X_3	ad	abcd	b	ab	acd	bcd	bc	cd	$(a,1)\wedge(b,1)\wedge(c,0)$, $(b,1)\wedge(d,1)$	$(e,0)$	(1.0, 0.16129)
Z_2	X_4			abd	abd	acd	abcd	abcd	ac	$(a,0)\wedge(b,1)$, $(a,0)\wedge(c,0)$, $(a,0)\wedge(d,2)$, $(c,0)\wedge(d,2)$	$(e,1)$	(1.0, 0.779221)
	X_5			acd	abcd	abcd	ad	acd	abc	$(a,0)\wedge(b,2)$, $(a,0)\wedge(d,2)$, $(b,2)\wedge(d,2)$	$(e,1)$	(1.0, 0.12987)
Z_3	X_6					abcd	cd	c	bcd	$(b,2)\wedge(c,0)$	$(e,2)$	(1.0, 0.309278)
	X_7					bcd	abcd	abc	abcd	$(a,2)\wedge(b,0)$, $(a,2)\wedge(c,0)$, $(a,2)\wedge(d,1)$, $(b,0)\wedge(c,0)$	$(e,2)$	(1.0, 0.257732)
Z_4	X_8							abd	ad	$(a,2)\wedge(b,1)$, $(d,0)$, $(a,2)\wedge(c,2)$	$(e,2)\vee(e,1)$	(0.888889,0.412371), (0.111111,0.064935)
	X_9							cd	bcd	$(d,0)$, $(a,1)\wedge(b,2)\wedge(c,1)$	$(e,2)\vee(e,1)$	(0.5,0.010309), (0.5,0.012987)
Z_5	X_{10}								bd	$(b,2)\wedge(c,2)$, $(c,2)\wedge(d,1)$	$(e,1)\vee(e,2)\vee(e,0)$	(0.333333,0.012987), (0.333333,0.010309), (0.333333,0.032258)
Z_6	X_{11}									$(c,2)\wedge(d,2)$, $(a,1)\wedge(b,1)\wedge(c,2)$, $(a,1)\wedge(b,1)\wedge(d,2)$	$(e,3)$	(1.0, 1.0)

5. RULE MODIFICATION

With the changes of the objective world, new objects belonging to the present classes and objects with new class will be appended to a data set. Therefore, the former rules require updating [7]. When new objects are added to a decision table, it is an efficient point of view to accept and modify the existing rules and their numerical measures incrementally, instead of regenerating them.

In this section, an algorithm for rule modification is presented using the concepts of ∂-indiscernibility relation and ∂-decision matrix.

Let X_i (i=1, 2, ..., n) be an equivalence class of decision table S, Y_j (j=1, 2, ..., m) be a decision class. The parameters of $X_i \to Y_j$ are $|X_i|$, $|Y_j|$, SI_{ij}^0 and CI_{ij}^0. B_i^0 is the decision function of X_i. Suppose $U/IND(\partial)=\{Z_1, Z_2, \ldots, Z_P\}$.

When a new object is appended to the decision table, both of the forms and numerical measures of the rules from $X_i \to Y_j$ may change. Thus, rule modification consists of rule's form updating (abbreviate to **FU**) and rule's measures updating. Before the numerical measures of the rules are computed, the parameters of $X_i \to Y_j$ must be updated (abbreviate to **PU**) first. Suppose that object x will be appended to the decision table. The algorithm is implemented by distinguishing the following cases:

CASE 1. $x \notin X_i$ ($\forall$ i = 1, 2, ..., n) and $x \notin Y_j$ ($\forall j$=1, 2, ..., m).

FU: A new equivalence class X_{n+1} and a new decision class Y_{m+1} are constituted with x. Create a new row and a new column of decision matrix and compute the decision function of X_{n+1}:

$$B_{n+1}^1 = \bigwedge_i \bigvee_{a \in m_{i,n+1}} a,$$

where i=1, 2, ..., n. The associated decision rules from X_{n+1} are then induced. The original set of rules are modified by

$$B_i^1 = B_i^0 \wedge (\bigvee_{a \in m_{i,n+1}} a),$$

where i=1, 2, ..., n.

PU: With regard to $X_{n+1} \to Y_{m+1}$, $SI_{n+1,m+1} = CI_{n+1,m+1} = 1$. Other parameters will not be affected by the modification.

CASE 2. $x \notin X_i$ ($\forall$ i = 1, 2, ..., n) and $x \in Y_j$ ($\exists j$=1, 2, ..., m).

FU: A new equivalence class X_{n+1} is constituted with x. Suppose $X_{n+1} \subset Z_f$, where $Z_f = \bigcup_{i_f} X_{i_f}$. Create a new row (column) of decision matrix and compute the decision function of X_{n+1}:

$$B_{n+1}^1 = \bigwedge_i \bigvee_{a \in m_{i,n+1}} a,$$

where i=1, 2, ..., n and $i \neq i_f$. The associated decision rules of X_{n+1} are then induced. The original set of rules are modified by

$$B_i^1 = B_i^0 \wedge (\bigvee_{a \in m_{i,n+1}} a),$$

where i=1, 2, ..., n and $i \neq i_f$. $B_{i_f}^1$ will not be affected by the modification.

PU: With regard to $X_{n+1} \to Y_j$, $SI_{n+1,j}^1 = 1$, $CI_{n+1,j}^1 = 1/(|Y_j|+1)$; With regard to $X_i \to Y_j$ ($i \neq n+1$), SI_{ij}^1 will not be affected, $CI_{ij}^1 = (|X_i \cap Y_j|)/(|Y_j|+1) = (CI_{ij}^0 \times |Y_j|)/(|Y_j|+1)$.

CASE 3. $x \in X_i$ and $x \in Y_k$, where $X_i \cap Y_k = \varnothing$ and $X_i \subseteq Z_g$, $Z_g = \bigcup_{i_g} X_{i_g}$. Suppose $Z_f = \bigcup_{i_f} X_{i_f}$. After object x is appended, suppose $X_i \subseteq Z_f$ ($f \neq g$) and $Z_f \subset U/IND(\partial)$.

FU: Delete m_{i,i_f} ($i < i_f$) or $m_{i_f,i}$ ($i_f < i$) and add m_{i,i_g} ($i < i_g$) or $m_{i_g,i}$ ($i_g < i$) in i-th row (column) of ∂-decision matrix. Compute the decision function B_i^1 of X_i, $B_{i_f}^1$ of X_{i_f} and $B_{i_g}^1$ of X_{i_g}. The associated rules of X_i, X_{i_f} and X_{i_g} are then induced. Other rules will not be affected by the modification.

PU: With regard to $X_i \to Y_k$, $SI_{ik}^1 = 1/(|X_i|+1)$, $CI_{ik}^1 = 1/(|Y_k|+1)$; With regard to $X_i \to Y_j$ ($j \neq k$), $SI_{ij}^1 = (|X_i \cap Y_j|)/(|X_i|+1) = (SI_{ij}^0 \times |X_i|)/(|X_i|+1)$, CI_{ij}^1 will not be affected; With regard to $X_h \to Y_k$ ($h \neq i$), SI_{hk}^1 will not be affected,

$CI_{hk}^1 = (|X_h \cap Y_k|)/(|Y_k|+1) = (CI_{hk}^0 \times |Y_k|)/(|Y_k|+1)$.

CASE 4. $x \in X_i$ and $x \in Y_k$, where $X_i \cap Y_k = \varnothing$ and $X_i \subseteq Z_g$, $Z_g = \bigcup_{i_g} X_{i_g}$. After object x is appended, suppose $X_i \subseteq Z_f$ and $Z_f \not\subset U/IND(\partial)$.

FU: Add m_{i,i_g} ($i < i_g$) or $m_{i_g,i}$ ($i_g < i$) in the i-th row (column) of ∂-decision matrix. Compute the decision function B_i^1 of X_i and $B_{i_g}^1$ of X_{i_g}. The associated rules of X_i and X_{i_g} are then induced. Other rules will not be affected by the modification.

PU: With regard to $X_i \to Y_k$, $SI_{ik}^1 = 1/(|X_i|+1)$, $CI_{ik}^1 = 1/(|Y_k|+1)$; With regard to $X_i \to Y_j$ ($j \neq k$), $SI_{ij}^1 = (|X_i \cap Y_j|)/(|X_i|+1) = (SI_{ij}^0 \times |X_i|)/(|X_i|+1)$, CI_{ij}^1 will not be affected; With regard to $X_h \to Y_k$ ($h \neq i$), SI_{hk}^1 will not be affected,

$CI_{hk}^1 = (|X_h \cap Y_k|)/(|Y_k|+1) = (CI_{hk}^0 \times |Y_k|)/(|Y_k|+1)$.

CASE 5. $x \in X_i$ and $x \notin Y_j$ ($\forall j$=1, 2, ..., m), where $X_i \subseteq Z_g$.

Suppose $Z_g = \bigcup_{i_g} X_{i_g}$.

FU: A new decision class Y_{m+1} is constituted with x. Add m_{i,i_g} $(i < i_g)$ or $m_{i_g,i}$ $(i_g < i)$ in the i-th row (column) of ∂-decision matrix. Compute the decision function B_i^1 of X_i and $B_{i_g}^1$ of X_{i_g}. The associated rules of X_i and X_{i_g} are then induced. Other rules will not be affected by the modification.

PU: With regard to $X_i \to Y_{m+1}$, $SI_{i,m+1}^1 = 1/(|X_i|+1)$, $CI_{i,m+1}^1 = 1$; With regard to $X_i \to Y_j$ $(j \neq m+1)$, $SI_{ij}^1 = (|X_i \cap Y_j|)/(|X_i|+1) = (SI_{ij}^0 \times |X_i|)/(|X_i|+1)$, CI_{ij}^1 will not be affected.

CASE 6. $x \in X_i$ and $x \in Y_j$, where $X_i \subseteq Y_j$.

FU: All the rules' form will not be affected by the modification.

PU: With regard to $X_i \to Y_j$, SI_{ij}^1 will not be affected and $SI_{ij}^1 = 1$, $CI_{ij}^1 = (|X_i \cap Y_j|+1)/(|Y_j|+1) = (|X_i|+1)/(|Y_j|+1)$; With regard to $X_h \to Y_j$ $(h \neq i)$, SI_{hj}^1 will not be affected, $CI_{hj}^1 = |X_h \cap Y_j|/(|Y_j|+1) = (CI_{hj}^0 \times |Y_j|)/(|Y_j|+1)$.

CASE 7. $x \in X_i$ and $x \in Y_j$, where $X_i \cap Y_j \neq \varnothing$ and $X_i \not\subset Y_j$.

FU: All the rules' form will not be affected by the modification.

PU: With regard to $X_i \to Y_j$, $SI_{ij}^1 = (|X_i \cap Y_j|+1)/(|X_i|+1) = (SI_{ij}^0 \times |X_i|+1)/(|X_i|+1)$, $CI_{ij}^1 = (|X_i \cap Y_j|+1)/(|Y_j|+1) = (CI_{ij}^0 \times |Y_j|+1)/(|Y_j|+1)$; With regard to $X_h \to Y_j$ $(h \neq i)$, SI_{hj}^1 will not be affected, $CI_{hj}^1 = |X_h \cap Y_j|/(|Y_j|+1) = (CI_{hj}^0 \times |Y_j|)/(|Y_j|+1)$; With regard to $X_i \to Y_k$ $(k \neq j)$, $SI_{ik}^1 = |X_i \cap Y_k|/(|X_i|+1) = (SI_{ik}^0 \times |X_i|)/(|X_i|+1)$, CI_{ik}^1 will not be affected.

After updating the rules and the parameters of $X_i \to Y_j$, the numerical measures of the modified rules must be computed. Then the rule set is come into being.

Example 2 Consider the decision table as shown in Table 1 and its ∂-decision matrix as shown in Table 2 again. Suppose a new object depicted in Table 3 will be appended to Table 1.

In Table 1, $Z_3 = X_6 \cup X_7$, $Z_4 = X_8 \cup X_9$. Due to adding of $x \in X_7$, $Z_3 = X_6$ and $Z_4 = X_7 \cup X_8 \cup X_9$. Therefore, add $m_{67} = ab$, delete $m_{78} = bcd$ and $m_{79} = abcd$ of the ∂-decision matrix. The corresponding rules induced from X_6, X_7, X_8 and X_9 may be change as shown in the shadowy part in Table 2. The results of the updated rules and the parameters are shown in Table 4.

TABLE 3

A NEW OBJECT

	a	*b*	*c*	*d*	*e*
x	2	0	0	1	1

TABLE 4

RESULTS OF RULE MODIFICATION FOR X_6, X_7, X_8 AND X_9

	Rules		(SI_{ij}, CI_{ij})
	Antecedents	Consequents	
X_6	$(b,2) \wedge (c,0)$	$(e,2)$	(1.0, 0.309278)
X_7	$(a,2)$, $(b,0) \wedge (c,0)$	$(e,2) \vee (e,1)$	(0.961538,0.257732), (0.038462,0.012821)
X_8	$(a,2)$, $(d,0)$	$(e,2) \vee (e,1)$	(0.888889, 0.412371), (0.111111, 0.064935)
X_9	$(a,1) \wedge (b,2) \wedge (c,1)$, $(d,0)$,	$(e,2) \vee (e,1)$	(0.5, 0.010309), (0.5, 0.012821)

Furthermore, the numerical measures of the rules are computed. The eventual rules and their measures are shown in Table 5.

TABLE 5

MODIFIED RULE SET

Rules	*support*	*accuracy*	*coverage*
$a(0) \wedge b(0) \to e(0)$	15	1.0	0.483871
$a(1) \wedge b(0) \to e(0)$	10	1.0	0.322581
$a(0) \wedge b(1) \to e(1)$	60	1.0	0.769231
$a(0) \wedge b(2) \to e(1)$	10	1.0	0.128205
$b(0) \wedge c(1) \to e(0)$	25	1.0	0.806452
$b(2) \wedge c(0) \to e(2)$	30	1.0	0.309278
$b(0) \wedge c(0) \to e(2) \vee e(1)$	25, 1	0.961538, 0.038462	0.257732, 0.012821
$b(2) \wedge c(2) \to e(1) \vee e(2) \vee e(0)$	1, 1, 1	0.333333, 0.333333, 0.333333	0.012821, 0.010309, 0.032258
$a(0) \wedge d(1) \to e(0)$	15	1.0	0.483871
$a(0) \wedge d(2) \to e(1)$	70	1.0	0.897436
$c(1) \wedge d(1) \to e(0)$	15	1.0	0.483871
$c(0) \wedge d(2) \to e(1)$	60	1.0	0.769231
$c(2) \wedge d(2) \to e(3)$	1	1.0	1.0
$c(2) \wedge d(1) \to e(1) \vee e(2) \vee e(0)$	1, 1, 1	0.333333, 0.333333, 0.333333	0.012821, 0.010309, 0.032258
$a(1) \wedge c(1) \wedge d(2) \to e(0)$	10	1.0	0.322581
$b(0) \wedge d(2) \to e(0)$	10	1.0	0.322581
$b(1) \wedge d(1) \to e(0)$	5	1.0	0.16129
$b(2) \wedge d(2) \to e(1)$	10	1.0	0.128205
$a(1) \wedge b(1) \wedge c(0) \to e(0)$	5	1.0	0.16129
$a(1) \wedge b(2) \wedge c(1) \to e(2) \vee e(1)$	1, 1	0.5, 0.5	0.010309, 0.012821
$a(1) \wedge b(1) \wedge c(2) \to e(3)$	1	1.0	1.0
$a(0) \wedge c(0) \to e(1)$	60	1.0	0.769231
$a(2) \to e(2) \vee e(1)$	65, 6	0.915493, 0.084507	0.670103, 0.076923
$d(0) \to e(2) \vee e(1)$	41, 6	0.87234, 0.12766	0.42268, 0.076923
$a(1) \wedge b(1) \wedge d(2) \to e(3)$	1	1.0	1.0

The update mechanism presented in this paper processes the new objects by modifying ∂-decision matrix and induced rules. Therefore, the ∂-decision matrix must be saved after the original rules are modified every time. Sometimes the induced rules require filtering to obtain the concise rule set.

Similarly, when the objects are modified or deleted, the rules are updated in the same way. In addition, the reducts of a decision table are also updated based on the ∂-decision matrix and decision functions, which is not given details in this paper.

6. CONCLUSIONS

An incremental algorithm of rule generation based on the ∂-decision matrix from decision table is presented. The characteristics of the method are:

(1) The computation of decision rules is reduced to the problems of simplification and adaptation of the associated Boolean expressions. Since the Boolean expressions simplification problem can be decomposed into a number of disjoint sub-problems by independently simplifying parts of the whole formula, the presented method is suitable for implementation with parallel multiprocessor systems.

(2) With this approach, the numerical measures of rules, i.e., support, coverage and accuracy, can be updated in the same time without searching all rules and their measures.

REFERENCES

[1] Z. Pawlak, "Rough set", *International J. Computer and Information Science*, vol. 11, 1982, pp. 341-356.

[2] N. Shan and W. Ziarko, "Data-based acquisition and incremental modification of classification rules", *Computational Intelligence*, vol. 11, no. 2, 1995, pp. 357-370.

[3] Z. Pawlak, *Rough Sets – Theoretical Aspects of Reasoning about Data*, Dordrecht: Kluwer Academic Publishers, 1991.

[4] A. Skowron and C.Rauszer, "The discernibility matrices and functions in information table", *in Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory*, R. Slowinski, eds., pp. 331-362, Dordrecht: Kluwer Academic Publishers, 1991.

[5] S. Tsumoto and H.Tanaka, "Primerose: probabilistic rule induction method based on rough sets and resampling methods", *Computational Intelligence*, vol. 11, no. 2, 1995, pp. 389-405.

[6] An Liping, Wu Yuhua and Tong Lingyun, "An approach for rule generation based on rough set theory", *Journal of Mangement Sciences in China*, vol. 4, no. 5, 2001, pp. 74-78.

[7] Zhang Xuemin and An Liping, "Obtaining and updating rules based on rough set theory in dynamic databases", *Journal of Tianjin University Sciences and Technology*, vol. 34, no. 4, 2001, pp.529-532.