퍼지 및 지능시스템학회 논문지 2001, Vol. 11, No. 7, pp. 666-674

Rough Set-based Incremental Inductive Learning Algorithm: Theory and Applications

Won-Chul Bang* **and Z. Zenn Bien****

*Human-friendly Welfare Robot System Engineering Research Center, KAIST
**Division of Electrical Engineering, Dept. of EECS, KAIST

Abstract

Classical methods to find a minimal set of rules based on the rough set theory are known to be ineffective in dealing with new instances added to the universe. This paper introduces an inductive learning algorithm for incrementally retrieving a minimal set of rules from a given decision table. Then, the algorithm is validated via simulations with two sets of data, in comparison with a classical non-incremental algorithm. The simulation results show that the proposed algorithm is effective in dealing with new instances, especially in practical use.

Key Words : Incremental inductive learning, Rough sets, Minimal set of decision rules, Reduct change criteria

1. Introduction

For the rough set theory to be more effective in a dynamic environment, there should be some rough set-based decision algorithm that is capable of incremental rough approximation. A non-incremental learning algorithm may be utilized for learning tasks with a single fixed set of training instances. For online learning tasks, however, one would prefer an incremental learning algorithm since it is more efficient to revise an existing minimal set of rules than generate a minimal set of decision rules each time a new instance is observed. The cost of generating a minimal set of rules from scratch can be too expensive if a non-incremental method is applied for an online learning task. In this paper, we tackle the problem of adaptively and effectively producing a minimal set of rules from an increased universe.

Pawlak [1] showed that the principles of inductive learning can be precisely formulated and in a unified way within the framework of the rough set theory. Recently, Bang and Bien [2] proposed an algorithm for incremental inductive learning which does not require recalculation for the overall examples when a new instance is added onto a consistent decision table in the framework of rough set.

In this paper, a modified categorization is provided with several theorems and its applications are shown to validate the incremental algorithm. In Section II, we briefly review a typical classical algorithm for finding a minimal set of rules from a decision table. Section III introduces the theory of rough set-based incremental inductive learning. In Section IV, its applications are shown with two data sets. The last section concludes with a summary and discussion of further works.

This work has been supported by Human-friendly Welfare Robot System Research Center, KAIST, Daejeon, Korea.

2. Inductive Learning

In [3] is presented a method of minimization of a decision table consisting of the 3 steps. We denote three notions K, L and M as outcomes of each step. K is the set of all RQ-basic decision rules, where R is a Q-reduct of P. L is defined by the set of all reducts of each RQ-decision rule, that is, $L = \{\text{reduct of } \phi_i/R \rightarrow \varphi_i\} \mid \phi_i \rightarrow \varphi_i$ is a decision rule in S, $i \in U\}$. Finally, M is a minimal set of decision rules which can be obtained by calculating a reduct of L. Thus, the classical non-incremental inductive learning algorithm generates K, L and M in order as steps go on.

The following example is given to illustrate the above procedure in more detail.

Table 1. A decision table for BCD-to-seven-segment decoder

U	A	B	C	D	a
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1

Example 1. The seven outputs of a BCD-to-seven-segment decoder, $\{a, b, c, d, e, f, g\}$ select the corresponding

segments in the display. Let us denote A, B, C and D be four bit inputs and assume that the first output a is considered only and that 0-8 in BCD are of interest. The truth table for the input-output relationship can be considered as a decision table and can be derived as follows:

Suppose that $P = \{A, B, C, D\}$ and $Q = \{a\}$. We can easily find a unique reduct, $\mathbf{RED}(P, Q) = \{\{B, C, D\}\}$ of the algorithm (P, Q) by Step 1. Let $R = \{B, C, D\}$. Eliminating the duplicated row, object 8, from Table 1, we obtain Table 2.

Table 2. RQ-basic decision rules with $R = \{B, C, D\}$

U	B	C	D	a
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Computing all reducts of each RQ-basic decision rule in Table 2 by Step 2, we get Table 3.

Table 3. All reducts of each RQ-basic decision rule

U	B	C	D	A
0	0	×	0	1
1	0	0	1	0
2	×	1	×	1
2′	0	×	0	1
3	×	1	×	1
4	1	0	0	0
5	1	×	1	1
6	×	1	×	1
7	×	1	×	1
7′	1	×	1	1

Finally, according to Step 3, we find the following one minimal decision algorithm:

$$\begin{array}{ll} B_0D_0 \rightarrow a_1 & \cdots\text{From Rule 0} \\ B_0C_0D_1 \rightarrow a_0 & \cdots\text{From Rule 1} \\ C_1 \rightarrow a_1 & \cdots\text{From Rule 2, 3, 6, 7} \\ B_1C_0D_0 \rightarrow a_0 & \cdots\text{From Rule 4} \\ B_1D_1 \rightarrow a_1 & \cdots\text{From Rule 5} \end{array} \quad (1)$$

3. Rough Set-based Incremental Inductive Learning

The rough set approach to inductive learning reviewed in Section II is useful in the sense that it can provide a systematic way to find a minimal set of decision rules from a large data set. For non-incremental learning tasks, this algorithm is a good choice for building a classification rule. For incremental learning tasks, however, it would be far from a preferable method to accept instances incrementally without building a new minimal set of decision rules. A primary motivation for using incremental systems is that knowledge should be rapidly updated with each new observation.

We propose a new incremental inductive learning algorithm. The basic idea of the algorithm may be compared to the following example. Let us consider the case that we want to put a number of different sized objects in a box. Suppose it is considered that all the objects are stacked up neatly from the bottom of the box such that the size of the box is minimal when it is realized that one object is mistakenly neglected to be put. Then the problem is how we can put the object in the box in such a way that we rearrange only the smallest number of objects by taking out from the top and putting them back while the size of the completed box is still minimal.

Hinted by the nature of the above problem and its solution process, we have developed a method of how to update a minimal set of decision rules when a new instance is added as described in this section.

3.1 Refinement of Categorization for New Instances

Pawlak [1][3] considered three possibilities when adding a new instance to the universe. According to his categorization, a new instance is said to confirm the knowledge if it already appears in the universe, while it is said to contradict the knowledge if there exists an object that has the same predecessor with that of the instance but has a different successor, and is said to completely new if the instance does not belong to any decision class. But we find that there is a case which does not belong to any of the above category. For example, what happens if a new instance $x{:}A_1B_0C_0D_1 \rightarrow a_1$ is added to the universe presented in Example 1? We find that this instance does not belong to any of the 3 categories by Pawlak ([1][3]), and thus, need to refine the categorization for new instances.

We give formal definitions of four categories that cover all possibilities with respect to a minimal decision algorithm which, according to the designer's interest, is selected out of possibly more than one minimal decision algorithms.

Let M be the selected minimal decision algorithm. Suppose ϕ_i is the predecessor and φ_i is the successor of the rule associated with an object $i \in U$, respectively.

Definition 1. **Completely new instance** : A new instance x is said to be completely new to M if $\varphi_x \not\equiv \varphi$ for all decision rules $\phi \rightarrow \varphi \in M$.

Definition 2. **Confirmation** : A new instance x is said to confirm M if there exists a decision rule $\phi \rightarrow \varphi \in M$ such that $\phi_x \rightarrow \phi$ and $\varphi_x \equiv \varphi$.

Definition 3. **Partial contradiction**: A new instance x

is said to partially contradict M if 1) there exists a decision rule $\phi \rightarrow \varphi$ and $\varphi_x \neq \varphi$, and 2) $\phi_x \neq \phi_y$ for every basic decision rule $y \in U$.

***Definition 4*. Complete contradiction** : A new instance x is said to completely contradict M if there exists a basic decision rule $y \in U$ such that $\phi_x \equiv \phi_y$ and $\varphi_x \neq \varphi_y$.

Note that the definitions are made with respect to a *minimal decision algorithm*. Thus, the problem to determine which category a new instance belongs to depends on that of which minimal decision algorithm is selected.

The terminology partially contradict comes from the fact that there can be a new instance which does not contradict the acquired knowledge but contradicts a minimal decision algorithm. In Example 1, $A_1B_0C_0D_1 \rightarrow a_1$ does not contradict Table 1 even though it contradicts the minimal decision algorithm (1).

A completely new instance can also be considered as either a partially contradictory case or a completely contradictory case.

The following example illustrates the above categorization.

***Example 2*.** Let us consider again the KR-system shown in Table 1 with the same P and Q. Suppose we add a new instance $A_0B_1C_0D_1 \rightarrow a_1$. The new instance confirms the minimal decision algorithm (1). In fact, x is equal to the object 5 in U.

Now let us add $A_1B_1C_1D_1 \rightarrow a_1$. This instance also confirms the minimal decision algorithm but we cannot find an object in U which is the same as x. Note that for both cases, the minimal decision algorithm after adding x is the same as the original one.

If we add an instance $A_0B_1C_0D_1 \rightarrow a_2$, it corresponds to the completely new case and to completely contradictory case at the same time since we do not have decision class 2 in M and there are $A_0B_1C_0D_1 \rightarrow a_1$ in U.

Let us add an instance $A_0B_1C_1D_1 \rightarrow a_0$. The new instance completely contradicts the minimal decision algorithm (1).

If we add $A_1B_1C_1D_1 \rightarrow a_2$, the added instance is both completely new and partially contradictory since we do not have decision class 2 in M and there is B_1D_1 a_1 in M which has a predecessor implying $A_1B_1C_1D_1$ and a different successor from that of x.

Finally, we investigate the partially contradictory case in detail. If we add

$$x : A_1B_0C_0D_1 \rightarrow a_1, \tag{2}$$

this instance partially contradicts the minimal decision algorithm (1) since it does not contradict Table 1 but contradicts the minimal decision algorithm (1). In this case, we should calculate a new Q-reduct of P, $R' = \{A, B, C, D\}$ which is different from $R = \{B, C, D\}$ in Example 1. All reducts of each $R'Q$-basic decision rule are shown in Table 4.

Table 4. All reducts of each R'Q- basic decision rule

U	A	B	C	D	A
0	×	0	×	0	1
1	0	0	0	1	0
2	×	×	1	×	1
2′	×	0	×	0	1
3	×	×	1	×	1
4	×	1	0	0	0
5	×	1	×	1	1
6	×	×	1	×	1
7	×	×	1	×	1
7′	×	1	×	1	1
8	×	0	×	0	1
8′	1	×	×	×	1
9	1	×	×	×	1

One of the minimal decision algorithms is:

$$\begin{array}{ll} B_0D_0 \rightarrow a_1 & \cdots \text{From Rule 0, 8} \\ A_0B_0C_0D_1 \rightarrow a_0 & \cdots \text{From Rule 1} \\ C_1 \rightarrow a_1 & \cdots \text{From Rule 2, 3, 6, 7} \\ B_1C_0D_0 \rightarrow a_0 & \cdots \text{From Rule 4} \\ B_1D_1 \rightarrow a_1 & \cdots \text{From Rule 5} \\ A_1 \rightarrow a_1 & \cdots \text{From Rule 9} \end{array} \tag{3}$$

Again, we add

$$y : A_1B_0C_1D_0 \rightarrow a_0 \tag{4}$$

to the decision table to which (2) is already added. We now have 11 objects in the decision table. This instance also partially contradicts the minimal decision algorithm (3), but, the new Q-reduct of P is still $R'' = \{A, B, C, D\}$, which is equal to R'. One of the minimal decision algorithms is:

$$\begin{array}{rl} B_0C_0D_0 \rightarrow a_1 & \cdots \text{From Rule 0, 8} \\ A_0B_0C_0D_1 \rightarrow a_0 & \cdots \text{From Rule 1} \\ A_0C_1 \rightarrow a_1 & \cdots \text{From Rule 2, 3, 6, 7} \\ B_1C_0D_0 \rightarrow a_0 & \cdots \text{From Rule 4} \\ B_1D_1 \rightarrow a_1 & \cdots \text{From Rule 5} \\ A_1C_0 \rightarrow a_1 & \cdots \text{From Rule 9} \\ A_1C_1 \rightarrow a_0 & \cdots \text{From Rule } x \end{array} \tag{5}$$

If it is always true that the minimal decision algorithm does not change when the added instance confirms the minimal decision algorithm as shown in Example 2, we can use it as the first check item for incremental inductive learning. We shall present a result in a form of theorem which provides that this is true.

For easier explanation, we define $U' = U \cup \{x\}$ when x is added to U. We also define additional notions K', L', M' as well as K, L, M . These are for describing after adding the new instance x. That is, K' is the set of all $R'Q$-basic decision rules, where R' is defined by a new Q-reduct of P after the new instance is added, L' is defined by the set of all reducts of each $R'Q$-decision rule, and M' denotes a reduct of L', *i.e.*, a minimal set of decision rules of U'.

We now give the following theorem.

Theorem 1. Let M be a minimal decision algorithm of U and **M** be the set of all minimal decision algorithms of U'. Then the new instance x confirms M if and only if $M \in \boldsymbol{M}$.

Proof. Since $M \in \boldsymbol{M}$, we find that M is also a minimal decision algorithms of U'. Thus x confirms M.

To show the converse implication, let a decision rule be the same as the rule associated with x which is found in U. Then it is obvious that $M \in \boldsymbol{M}$. Now suppose that x confirms M but there is no $y \in U$ such that $y \models_S \phi_x /R$, where R, is a Q-reduct of P. Then a new rule associated with x does not conflict with any rules in (R, Q). Hence, the algorithm (R, Q) is consistent in U'. By definition, R is independent in U, that is, if any attribute in R is removed then (R, Q) becomes inconsistent in U. This is always true regardless of whether or not a new rule associated with x is added to U. Hence, R is independent in U'. Therefore, (R, Q) is consistent and independent in U' and thus R is still a reduct of P in (P, Q) in U'. By noting that L is the set of all reducts of each RQ-basic decision rule, we find that L remains unchanged because the rules that are indispensable in L remain so even after addition of x, and thus, $M \in \boldsymbol{M}$.

For the KR-system shown in Table 1, suppose we add a new instance $x: A_0B_1C_2D_1 \rightarrow a_1$. The new instance x confirms M in (1) but there is no $y(\in U)$ such that $y \models_S \phi_x/R$. We can easily find that, after adding x to U, **RED**$(P, Q) = \{\{B, C, D\}\}$ and $K' = K \cup \{B_1C_2D_1 \rightarrow a_1\}$ and $L' = L \cup \{C_2D_1 \rightarrow a_1, B_1D_1 \rightarrow a_1, B_1C_2 \rightarrow a_1\}$. We see that M is not altered since L already has the rule $B_1D_1 \rightarrow a_1$.

Among the proposed four categories for new instances, confirming instances can be added without changing any minimal decision algorithm by Theorem 1. If a new instance partially contradicts the minimal decision algorithm, it is reasonable to consider the level of how much the minimal decision algorithm should be changed. This issue will be considered in detail in the next section. Regarding the completely contradictory instances, it is more complicated to analyze the cases than a partially contradictory case and thus we need more investigation based on the study of partially contradictory instances.

3.2 Criteria for Reduct Change

The classical method to find a minimal decision algorithm consists of finding a Q-reduct of P, finding all reducts of each restricted decision rule to the Q-reduct of P, and finally finding a reduct of the set of decision rules from the previous step. Once it is known that the new instance x partially contradicts M, we must check if the Q-reduct of P is still available in order to decide to skip Step 1. To this end, we need a criterion on which to base the decision. Theorem 2 gives us such a criterion.

Theorem 2. Suppose $S = (U, A)$ is a KR-system and let R be a Q-reduct of P. Suppose the new instance x is added to U with a PQ-basic decision rule $\phi_x \rightarrow \varphi_x$ which partially contradicts M. Then R is also one of the Q-reducts of P in $S' = (U', A)$ if and only if there does not exist a $y \in U$ such that $y \models_S \phi_x/R$.

Proof. Suppose first there does not exist any $y \in U$ such that $y \models_S \phi_x/R$. Then a new rule associated with x does not conflict with any rules in (R, Q). Hence, the algorithm (R, Q) is consistent in U'. By definition, R is independent in U, that is, if any attribute in R is removed then (R, Q) becomes inconsistent in U. This is naturally true whether a new object x is added to U or not. Hence, R is independent in U'. Therefore, (R, Q) is consistent and independent in U' and thus R is still a reduct of P in (P, Q) in U'.

To show the necessity, let a rule associated with z be in the minimized rule set which conflicts with the new rule associated with x. Then,

$$\models_S \phi_x \rightarrow \phi_z \text{ and } \varphi_x \neq \varphi_z \tag{6}$$

where $x \models_S \phi_x \rightarrow_x$ and $z \models_S \phi_z \rightarrow \varphi_z$. Since $\phi_z/R \equiv \phi_z$,

$$\models_S \phi_x/R \;\; \phi_z/R. \tag{7}$$

By assumption, there exists a $y \in U$ such that

$$y \models_S \phi_x/R, \text{ i. e., } \phi_y/R \equiv \phi_x/R. \tag{8}$$

(7) and (8) yield

$$\models_S \phi_y/R \rightarrow \phi_z/R. \tag{9}$$

And, since y and z are consistent in (R, Q) if $\models_S \phi_y/R \rightarrow \phi_z/R$ then $\varphi_y \equiv \varphi_z$. This and (9) yield

$$\varphi_y \equiv \varphi_z. \tag{10}$$

From (6) and (10),

$$\varphi_y \equiv \varphi_x. \tag{11}$$

As a result from (8) and (11), $\phi_x/R \rightarrow \phi_y/R$ and $\varphi_y \neq \varphi_x$. Since this implies that the new rule associated with x conflicts with the rule associated with y in (R, Q) in U, R cannot be a reduct of (P, Q) in U'.

Now we can decide whether R is still a Q-reduct of P by comparing the new instance and each object in U. If it is found that R needs to be recalculated, we must reconstruct all the reducts of RQ-basic decision rules, L'. Then a question arises whether we must calculate L' again. In Example 2, since the set of all reducts of each $R''Q$-basic decision rule has the three same rows with Table 4, we do not have to calculate the reducts of RQ-basic decision rules for {1, 4, 5} in Table 4. This implies we can skip Step 2 for some RQ-basic decision rules and can use the reduct of rules in L. The following theorem indicates which reducts of RQ-basic decision rules in L can be used again.

Theorem 3. Suppose S = (U, A) is a KR-system and let R be a Q-reduct of P. Suppose a new instance x is added to U with a PQ-basic decision rule $\phi_x \to \varphi_x$ which partially contradicts M and x does not make R changed. Then, for each rule $\phi_{ij} \to \varphi_{ij} \in M$, all the reducts of the rule $\phi_i \to \varphi_{ij} \in K$ are also in L' if and only if $\phi_x \to \varphi_x$ confirms the set of one rule $\{\phi_{ij} \to \varphi_{ij}\}$. Here $\phi_{ij} \to \varphi_{ij}$ is the j-th reduct of the i-th RQ-basic decision rule.

Proof. Consider a decision $\phi_i \to \varphi_i \in$ K. Since $\phi_x \to \phi_{ij}$ implies $\varphi_x \equiv \varphi_{ij}$, that is, $\phi_x \to \varphi_x$ does not contradict $\phi_{ij} \to \varphi_{ij}$, $\phi_i \to \varphi_i$ can remain in L'.
To show the necessity, for a decision rule $\phi_i \to \varphi_i$ in K', it is obvious that x cannot contradict any reduct of $\phi_i \to \varphi_i$. Thus, $\phi_i \to \phi_{ij}$ implies $\varphi_x \equiv \varphi_{ij}$.

Based on the above result, we shall consider a subset C_x of U defined as:

$$C_x = \{i \in U \mid \forall\, \phi_{ij} \to \varphi_{ij} \in M,\ x \text{ contradicts } \phi_{ij} \to \varphi_{ij} \in M\}$$

Note that, for a basic rule $\phi \to \varphi$, reduct of ϕ is generally not unique. Once we find C_x and calculate all the reduct of each RQ-basic decision rule in C_x, we can construct L' instead of using Step 2.

Next, to find a minimal set of rules M', we may try to find a reduct of L'. If we take into account, however, the fact that M' is made up of reducts of each decision class, it is enough to calculate only the reducts of the decision classes into which the RQ-basic decision rules in C_x, and the new instance are classified.

The overall flowchart of the algorithm is given in Fig. 1.

3.3 An Illustrative Example

Consider the KR-system shown in Table 1 with P = $\{A, B, C, D\}$ and $Q = \{a\}$ again. Note that, in Example 2, we have obtained a minimal set of rules (3) when a new instance $A_1B_0C_0D_1 \to a_1$ is added to the decision table given in Table 1. We rewrite the decision table in Table 5 after adding the new instance as well as its minimal set of rules in (12).

Table 5. New decision rule with 10 objects

U	A	B	C	D	A
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	1	1
7	0	1	1	1	1
8	1	0	0	0	1
10	1	0	0	1	1

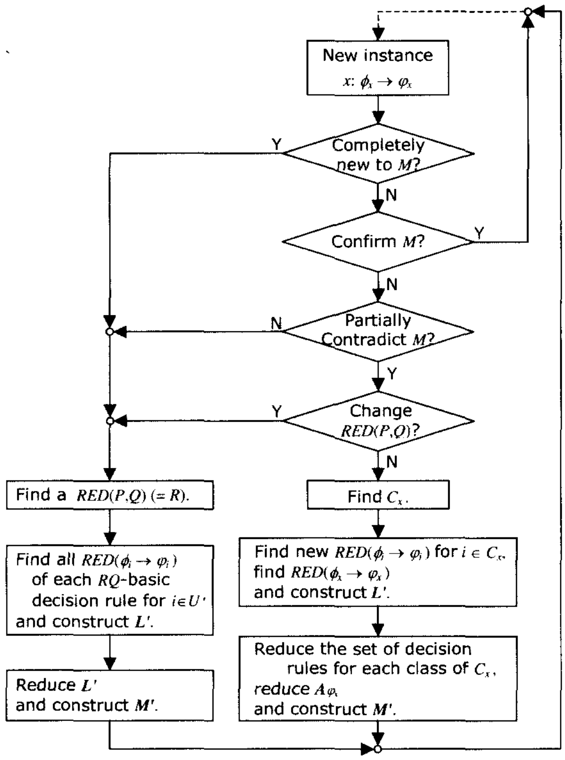


Fig. 1. Rough set-based incremental inductive learning algorithm

$$\begin{array}{ll} B_0D_0 \to a_1 & \cdots \text{From Rule } 0, 8 \\ A_0B_0C_0D_1 \to a_0 & \cdots \text{From Rule } 1 \\ C_1 \to a_1 & \cdots \text{From Rule } 2, 3, 6, 7 \\ B_1C_0D_0 \to a_0 & \cdots \text{From Rule } 4 \\ B_1D_1 \to a_1 & \cdots \text{From Rule } 5 \\ A_1 \to a_1 & \cdots \text{From Rule } 9 \end{array} \quad (12)$$

Suppose that we choose (12) as our minimal decision algorithm M and we add another instance x, (4), in Example 2.

The algorithm examines first whether x is completely new to M by comparing its successor with successors of the rules in M. Since this is not the case, the algorithm examines next if x confirms M. By Theorem 1, if it confirms M then a new minimal decision algorithm becomes M itself. By comparing x with the decision rules in M, x is proved not to confirm M, and thus the algorithm examines whether it partially contradicts M. Since, the instance $x: A_1B_0C_1D_0 \to a_0$ contradicts the first and the sixth rule in (12), x is either completely contradictory or partially contradict. Since $\phi_x (= A_1B_0C_1D_0)$ does not appear in the predecessors of any objects in U, one can find that x is a partially contradictory instance.

Next, by Theorem 2, R is still a Q-reduct of P in S = (U, A) since there is no $y \in U$ such that $y \models_S \phi_x/R$. Now, the algorithm determines the set of labels as noted in Theorem 3. Since the first and sixth rules in (12), which x contradicts, come from rule 0, 8 and rule 9 in Table 5, we find that Cx = {0, 8, 9}.

Thus, we obtain **RED**($\phi_0 \to \varphi_0$) = {{B, C, D}, {A, B, D}}, **RED**($\phi_8 \to \varphi_8$) = {{B, C, D}, {A, C}}, **RED**($\phi_9 \to \varphi_9$) = {{A, C}, {A, D}}, and **RED**($\phi_x \to \varphi_x$) = {{A, C}}, which are also presented in a tabular form in Table 6.

Table 6. All reducts of each RQ-basic rule in C_x

U	A	B	C	D	a
0	×	0	0	0	1
0′	0	0	×	0	1
8	×	0	0	0	1
8′	1	×	0	×	1
9	1	×	0	×	1
9′	1	×	×	1	1
x	1	×	1	×	0

Combining Table 6 with the set of all $R''Q$-basic rules in Example 2, we get L'. Finally, we should reduce L' to extract a minimal set of rules. As mentioned above, however, we just have to reduce the classes which reducts of the new instance x and each RQ-basic rule in C_x = {0, 8, 9} are classified into. Since the reducts of x are classified into a_0 and those of each RQ-basic rule in C_x are classified into a_1, we calculate the reducts for the classes a_0 and a_1. We do not have to calculate the other reducts of the other classes if any (but no other classes in this example).

The reducts of the class a_1 are

$$\begin{aligned}
&0:\ B_0C_0D_0 \to a_1 \quad \text{or} \quad A_0B_0D_0 \to a_1\\
&8:\ B_0C_0D_0 \to a_1 \quad \text{or} \quad A_1C_0 \to a_1\\
&9:\ A_1C_0 \to a_1 \quad \text{or} \quad A_1D_1 \to a_1\\
&2:\ A_0C_1 \to a_1 \quad \text{or} \quad A_0B_0D_0 \to a_1\\
&3:\ C_1D_1 \to a_1 \quad \text{or} \quad A_0C_1 \to a_1\\
&5:\ B_1D_1 \to a_1\\
&6:\ A_0C_1 \to a_1 \quad \text{or} \quad B_1C_1 \to a_1\\
&7:\ B_1C_1 \to a_1,\ B_1D_1 \to a_1,\ C_1D_1 \to a_1,\ A_0C_1 \to a_1.
\end{aligned} \tag{13}$$

And those of $A_{\varphi x}$, *i.e.*, the class a_0 are

$$\begin{aligned}
&0:\ A_0B_0C_0D_1 \to a_0\\
&4:\ B_1C_0D_0 \to a_0\\
&x:\ A_1C_1 \to a_0
\end{aligned} \tag{14}$$

Then, a minimal decision algorithm is found to be

$$\left.\begin{aligned}
B_0C_0D_0 &\to a_1\\
A_0C_1 &\to a_1\\
B_1D_1 &\to a_1\\
A_1C_0 &\to a_1
\end{aligned}\right\}\ \text{Class } a_1 \text{ from (13)}$$

and

$$\left.\begin{aligned}
A_0B_0C_0D_1 &\to a_0\\
B_1C_0D_0 &\to a_0\\
A_1C_1 &\to a_0
\end{aligned}\right\}\ \text{Class } a_0 \text{ from (14).}$$

This is identical to(5), which is the result of recalculation for U'.

4. Applications to Two Sets of Data

In order to evaluate the performance of the proposed incremental inductive learning algorithm, we must compare the minimal set of rules from the classical non-incremental algorithm with that from the proposed incremental algorithm, and also compare the times to produce the minimal sets of rules from both algorithms. For a given decision table and an object to be added, the minimal set of rules extracted by the proposed algorithm should be the same as that from the decision table to which the object is added. Moreover, if N_{new} objects are to be added one by one, the minimal set of rules for the final decision table can be obtained by applying the proposed algorithm N_{new} times. Again, the final minimal set of rules should be the same as that from the decision table to which the all N_{new} objects are added. Here, we assume that N_0 objects are given at the beginning and then N_{new} objects are consecutively added to meet totally N (= N_0 + N_{new})objects. The proposed algorithm is applied to a couple of data sets. N_0 objects are to be initially considered. Then, N_{new} objects are consecutively added to produce new minimal set of rules by 1) the classical non-incremental algorithm and 2) the proposed algorithm.

The simulation has been done with Ultra Sparc 1 170 and Rough Set Library (RSL) is used to find reducts and minimal sets of rules. The simulation procedure is shown in Fig. 2.

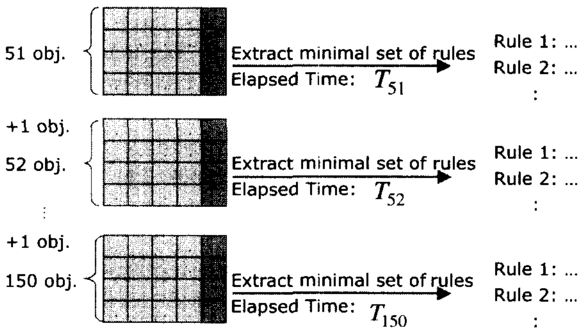


Fig. 2. Simulation procedure for incremental learning

4.1 Iris Data

Iris data is a classification data set first used by Fisher [4]. The data, of which a part is shown in Table 7, contains 150 cases of three iris species, 50 cases from each class. Each case is described by 4 attributes. The first 50 cases correspond to the first class, the next 50 cases to the second, and the rest to the third.

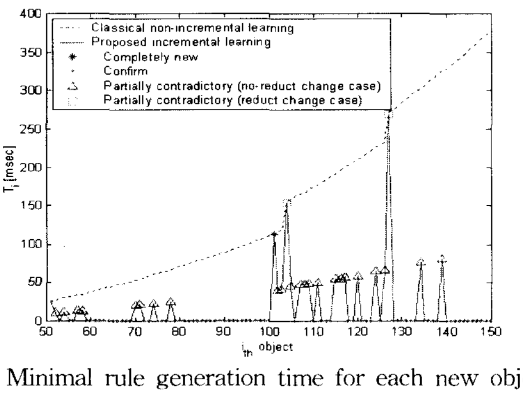
For this data, we let N_0 = 51 so that the initial decision table has at least two classes. Then, we find a minimal set of rules for the given initial decision table by classical non-incremental algorithm. Adding 99(=N_{new})

objects one by one, we find the corresponding minimal set of rules both by the classical algorithm and by the proposed algorithm. After N_{new} times of iteratively applying both algorithms to the incremental decision table, we get the final minimal set of rules for overall decision table in which totally 150($N = N_0 + N_{new}$) objects are contained.

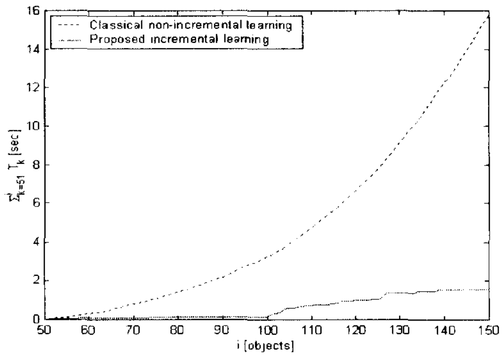
Table 7. A part of the iris data

U	sepal length	sepal width	petal length	petal width	type
⋮	⋮	⋮	⋮	⋮	⋮
47	4.6	3.2	1.4	0.2	setosa
48	5.3	3.7	1.5	0.2	setosa
49	5.0	3.3	1.4	0.2	setosa
50	7.0	3.2	4.7	1.4	versicolor
51	6.4	3.2	4.5	1.5	versicolor
52	6.9	3.1	4.9	1.5	versicolor
⋮	⋮	⋮	⋮	⋮	⋮
97	6.2	2.9	4.3	1.3	versicolor
98	5.1	2.5	3.0	1.1	versicolor
99	5.7	2.8	4.1	1.3	versicolor
100	6.3	3.3	6.0	2.5	virginica
101	5.8	2.7	5.1	1.9	virginica
102	7.1	3.0	5.9	2.1	virginica
⋮	⋮	⋮	⋮	⋮	⋮

The calculation time for finding each minimal set of rules is shown in Fig. 3 (a). The classical algorithm without considering incremental aspect must recalculate it for all of the objects in the current decision table. The minimal rule generation time for each new object is shown in a dotted line in Fig. 3 (a). The time when incremental algorithm is applied is shown in a solid line. There is a wide gap between both lines. Fig. 3 (a) also shows which objects are new, confirmative, partially contradictory, or completely new. For the iris data, 73 objects out of 99 added objects are confirmative, with which the minimal sets of rules are most easily obtained by Theorem 1, 23 objects are partially contradictory with



(a) Minimal rule generation time for each new object



(b) Cumulative time for minimal rule generation when each new object added

Fig. 3. Performance evaluation with iris data

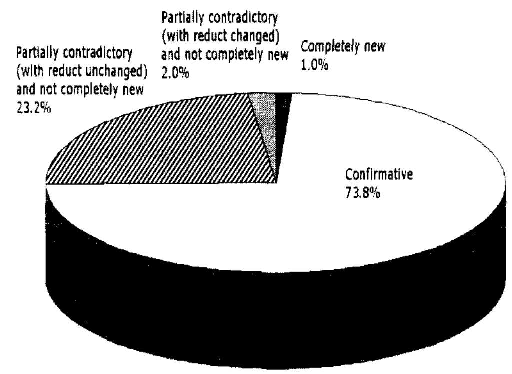


Fig. 4. Distribution of iris data

reduct unchanged, for which the calculation time of finding the rules is decreased by Theorem 2, and 2 objects are partially contradictory with reduct changed, for which it is slightly faster to obtain the rules by Theorem 3. Thus, we can think that the outperforming result of the proposed algorithm for the iris data comes from that the added objects are usually confirmative or partially contradictory. The distribution of iris data is shown in Fig. 4.

Actually, it is also important to compare the cumulative time to find the final minimal set of rules. Fig. 3 (b) shows the result. We may observe that the classical algorithm yields an exponentially increasing curve while the proposed algorithm gives a steadily increasing piece-wise linear curve as new objects are added. It is observed that the proposed algorithm is 10.5 times faster than the classical algorithm after 99 objects are added.

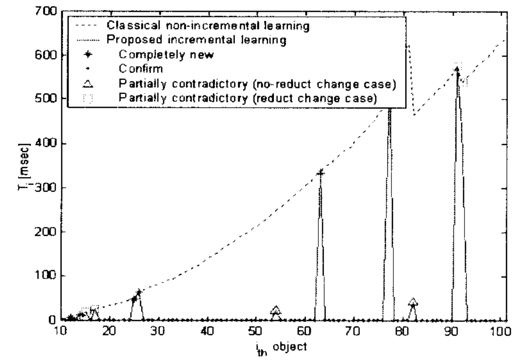
4.2 Zoo Data

Zoo data is a simple database containing 16 condition attributes and 1 decision attribute [5]. The data contains 101 objects from seven species of animals. The objects are randomly distributed so that the same species of animals, i.e. the same classes of objects, do not gather together in successive labels of objects. A part of the data is shown in Table 8.

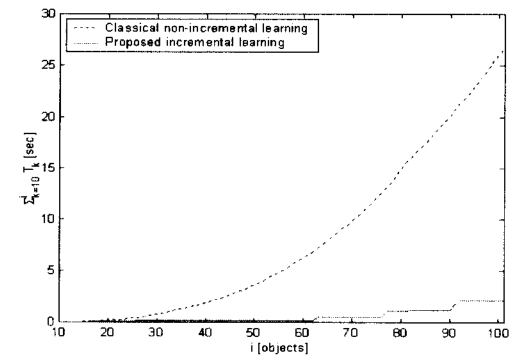
Table 8. A part of the zoo data

U	hair	feathers	eggs	milk	tails	domestic	catesize	type
0	1	0	0	1	0	0	1	1
1	1	0	0	1	1	0	1	1
2	0	0	1	0	1	0	0	4
3	1	0	0	1	0	0	1	1
4	1	0	0	1	1	0	1	1
5	1	0	0	1	1	0	1	1
6	1	0	0	1	1	1	1	1
7	0	0	1	0	1	1	0	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

For this data, we let $N_O = 10$, in which two species appear. Then, we find a minimal set of rules for the given initial decision table by classical non-incremental algorithm. Then, adding 91 (= N_{new}) objects one by one, we find the corresponding minimal set of rules both by the classical algorithm and by the proposed algorithm, respectively. Similarly to iris data, the calculation time for finding each minimal set of rules is shown in Fig. 5 (a). There exists also a wide gap between the curves of the classical algorithm and the proposed algorithm. And also, note that 79 objects out of 91 added objects are



(a) Minimal rule generation time for each new object



(b) Cumulative time for minimal rule generation when each new object added

Fig. 5. Performance evaluation with zoo data

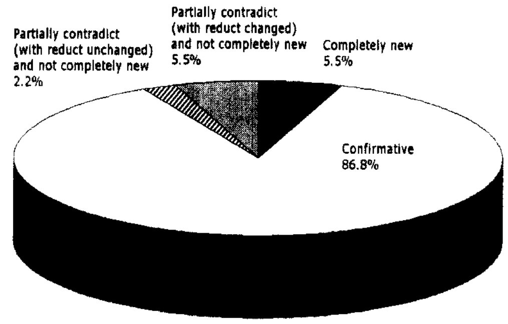


Fig. 6. Distribution of zoo data

confirmative, 2 objects are partially contradictory with reduct unchanged, and 5 objects are partially contradictory with reduct changed. The distribution of the iris data is shown in Fig 6.

Fig. 5 (b) shows that the classical algorithm yields an exponentially increasing curve while the proposed algorithm gives a steadily increasing curve as new objects are added in view of the cumulative time to find the final minimal set of rules. It is also observed that the proposed algorithm is 11.6 times faster than the classical algorithm after 91 objects are added.

5. Concluding Remarks

The rough set-based incremental inductive learning algorithm is incremental and computationally economical, thus sustaining a continual basis for reacting to new stimuli. This is an important desirable property of any system under real-world constraints. The algorithm has been evaluated with two sets of data where all the added objects are almost confirmative or partially contradictory with reduct unchanged. It is shown that its performance is much faster than the classical one as the time passes. In case of the data sets used in the simulation, the proposed algorithm is effective in dealing with new instances added to a given set.

The proposed algorithm needs to be further improved to handle the case when a completely contradictory instance is introduced to the minimal set of rules.

References

[1] Z. Pawlak, "On Learning A Rough Set Approach," *Proc. Int. Symp. on Computation Theory* and *Lecture Notes in Computer Science*, G. Goos, et al. (eds.), vol. 208, 1984, pp. 197-227.

[2] Won-Chul Bang and Zeungnam Bien, "New Incremental Inductive Learning Algorithm in the Framework of Rough Set Theory," *International Journal of Fuzzy Systems*, vol. 1, no. 1, pp. 25-36, Sep, 1999.

[3] Z. Pawlak, *Rough Sets: Theoretical Aspects and Reasoning*

about Data, Kluwer Academic Publishers, 1991.
[4] R. A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.
[5] R. Forsyth, *Machine Learning: Principles and Techniques*, Chapman and Hall Computing Series, 1989.

Won-Chul Bang
received the B.S. degree with Summa Cum Laude from Hanyang University, Seoul, Korea, in 1994, and the M.S. and Ph.D. degrees from KAIST, Taejon, Korea, in 1996 all in electrical engineering.
He is currently working at Human-friendly Welfare Robot System Engineering Research Center at KAIST. His research interests include intelligent control, measuring machine intelligence, rough set-based data analysis, machine learning and knowledge discovery, and soft computing techniques, especially their applications in the area of control system design.

Zeung-nam Bien
received his B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1969 and the M.S. and Ph.D. degrees in electrical engineering from the University of Iowa, Iowa City, Iowa, U.S.A., in 1972 and 1975, respectively. During 1976-1977 academic years, he taught as assistant professor at the Department of Electrical Engineering, University of Iowa. Then Dr. Bien joined Korea Advanced Institute of Science and Technology, Summer, 1977, and is now Professor of Control Engineering at the Department of Electrical Engineering and Computer Science, KAIST. He was a visiting faculty at the University of Iowa during his 1981-1982 sabbatical year and a visiting researcher at CASE Center of Syracuse University, New York, and visiting professor at Dep't of Control Engineering, Tokyo Institute of Technology during 1987-1988 academic year. Dr. Bien was the president of the Korea Fuzzy Logic and Intelligent Systems Society during 1990-1995 and also, the general chairs for IFSA World Congress 1993, and for FUZZ-IEEE99, respectively. He is currently co-Editor-in-Chief for International Journal of Fuzzy Systems (IJFS), Associate Editor for IEEE Transactions on Fuzzy Systems, and a regional editor for the International Journal of Intelligent Automation and Soft Computing. He has been serving as Vice President for IFSA since 1997, and now President for Institute of Electronics Engineers of Korea. At KAIST, Prof. Bien had served in particular as Dean of Academic Affairs, and then Dean of College of Engineering, among his several administrative positions. Since 1999, he has been working as the Director of Human-Friendly Welfare Robot System (HWRS) Research Center, which is sponsored by KOSEF within the national Center of Excellence Program.

His current research interests include intelligent automation and learning control methods, softcomputing techniques with emphasis on fuzzy logic systems, service robotics and rehabilitation engineering systems, and large-scale industrial application systems. Prof. Bien has published more than 250 international journal/proceedings papers, 200 domestic journal papers, 10 contributed papers for edited volumes and has authored/coauthored 5 technical books. Dr. Bien is a member of Eta Kappa Nu, KAST and NAEK..