Rough Set 이론을 이용한 연역학습 알고리즘

Inductive Learning Algorithm using Rough Set Theory

°방 원 철 변 증 남

한국과학기술원 전기 및 전자공학과
대전시 유성구 구성동 373-1

°Won-Chul Bang, Zeungnam Bien

Dept. of Electrical Engineering, KAIST
373-1 Kusong-Dong, Yusong-Gu, Taejon 305-701, KOREA
Tel: 042-869-3419, Fax: 042-869-8750

Abstracts In this paper we will discuss a type of inductive learning called learning from examples, whose task is to induce general descriptions of concepts from specific instances of these concepts. In many real life situations however new instances can be added to the set of instances. It is first proposed within the framework of rough set theory, for such cases, an algorithm to find minimal set of rules for decision tables without recalculation for overall set of instances. The method of learning presented here is based on a rough set concept proposed by Pawlak[2]. It is shown an algorithm to find minimal set of rules using reduct change theorems giving criteria for minimum recalculation and an illustrative example.

Keywords Inductive Learning, Rough Set Theory, Dynamic Learning, Semantics of Decision Logic

1 Introduction

The subject of machine learning has received considerable attention in recent decades. Inductive learning (learning from examples) is perhaps the oldest and best-understood problem in artificial intelligence[4]. Many existing expert systems were built by manually encoding the knowledge of human experts. Encoding processes as such can be very time consuming as they require close collaboration between computer professionals and experts of the subjects domain. To design expert system in this way is rather inefficient particularly because the same tedious task has to be performed for each specialized application. A better alternative of designing an expert system would be to construct and inductive algorithm that can, from a carefully chosen sample of expert decisions, infer and refine decision rules automatically independent of the subject of interest. The papers [5][6][8][9] describe some of the more recent research efforts made in this area.

Quinlan[1] suggested an inductive algorithm based on the statistical theory of information originally proposed by Shannon. The entropy function is used as a measure of uncertainty in the classification of objects characterized by attributes and attribute values. On the other hand, Pawlak[2] showed that the principles of inductive learning can be formulated precisely and in a unified way within the framework of rough set theory.

The set of instances, which is used for training set of learning, is usually constant and unchanged during the learning process. In many real life situations however this is not the case and new instances can be added to the set of instances. Our main objective in this paper is to find an algorithm for inductive learning without any recalculation for overall

instances when a new instance is added within the framework of rough set theory assuming that the minimized decision rules for the original decision table is already given.

In section 2, the preliminaries for rough set theory are reviewed, especially for the previous algorithm for minimizing of decision tables. In section 3, inductive learning concept in view of rough set theory is given. Based on this structure, an algorithm for learning from examples when a new instance is added to the examples in section 4.

2 Mathematical Preliminaries

In this section, in order to deal with decision tables mathematically, mathematical backgrounds on rough set theory and related definition are reviewed. In addition, the existing minimization method of decision table is followed up to catch an idea to lead the proposed algorithm.

2.1 Formal Definitions and Semantics of Decision Logic[7]

Decision tables can be defined in terms of *KR*(Knowledge Representation)-systems as follows. Let $K = (U, A)$ be a *KR*-system and 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 $T=(U,A,C,D)$, or in short *CD*-decision table.

Equivalence classes of the relations $IND(C)$ and $IND(D)$ will be called **condition** and **decision classes**, respectively.

With every $x \in U$ we associate a function $d_x: A \rightarrow V$, such that $d_x(a) = a(x)$, for every $a \in C \cup D$; the function d_x will be called a **decision rule** in T, and x will be referred as a **label** of the decision rule d_x.

Expressions of the form (a, v), or in short a_v, called **atomic formula**, are formulas of the *DL*(Decision Logic)-language for any $a \in A$ and $v \in V_a$. If ϕ and φ are formulas of the *DL*-language, then so are $\sim\phi$, $(\phi \vee \varphi)$, $(\phi \wedge \varphi)$, $(\phi \rightarrow \varphi)$, and $(\phi \equiv \varphi)$.

Formulas are meant to be used as descriptions of objects of the universe. Atomic formula (a, v) is interpreted as a description of all objects having value v for attribute a. Compound formulas are interpreted in the usual way. In order to express this problem more exactly, we define Tarski's style semantics of the decision logic language.

An object $x \in U$ **satisfies** a formula ϕ in $S = (U, A)$, denoted $x \models_S \phi$ or in short $x \models \phi$, if S is understood.

If ϕ is a formula the set $|\phi|_s$ defined as follows

$$|\phi|_s = \{x \in U \mid x \models_s \phi\}$$

will be called the **meaning** of the formula ϕ in S.

Formula of the form

$$(a_1, v_1) \wedge (a_2, v_2) \wedge \ldots \wedge (a_n, v_n),$$

where $v_i \in V_{ai}$, $P = \{a_1, a_2, \ldots, a_n\}$, and $P \subseteq A$, will be called a ***P*-basic formula** or in short *P*-formula. *A*-basic formula will be called **basic formulas**.

Any implication $\phi \rightarrow \varphi$ will be called a **decision rule** in the *KR*-language; ϕ and φ are referred to as the predecessor and the successor of $\phi \rightarrow \varphi$ respectively. If a decision rule $\phi \rightarrow \varphi$ is true in S, it is said that the decision rule is consistent in S. If $\phi \rightarrow \varphi$ is a decision rule and ϕ and φ are *P*-basic and *P*-basic formulas respectively, then decision rule $\phi \rightarrow \varphi$ will be called a ***PQ*-basic decision rule**, (in short *PQ*-rule), or **basic rule** when *PQ* is known.

Any finite set of basic decision rules will be called a **basic decision algorithm**. If all decision rules in a basic decision algorithm are *PQ*-decision rules, then the algorithm is said ***PQ*-decision algorithm**, or in short *PQ*-algorithm, and will be denoted by (P, Q). The *PQ*-algorithm is **consistent** in S, iff its decision rules are consistent in S.

2.2 Minimization of Decision Tables

The approach to table minimization presented in [7] consists of the following steps:

Step 1) Reduction of the Algorithm: Computation of reduct of condition attributes which is equivalent to elimination of some column from the decision table.

Step 2) Reduction of Decision Rules: Elimination of superfluous values of attributes.

Step 3) Minimization of the Decision Algorithm: Elimination of superfluous decision rules.

Now each step is introduced in detail one by one.

Let (P, Q) be a consistent algorithm and suppose that $a \in P$ and $R \subseteq P$.

Step 1) Reduction of the Algorithm:

An attribute $a \in P$ is called **dispensable** in the algorithm (P, Q) if the algorithm $(P-\{a\}, Q)$ is consistent; otherwise a is **indispensable** in (P, Q). The algorithm (P, Q) is called **independent** if all $a \in P$ are indispensable in (P, Q). The subset of attributes $R \subseteq P$ is called a **reduct** of the algorithm (P,Q) if (R,Q) is independent and consistent.

A basic decision algorithm is said to be **reduced** if every rule in the algorithm is reduced.

Step 2) Reduction of Decision Rules

Let $\phi \rightarrow \varphi$ be a PQ-rule.

An attribute $a \in P$ is called **dispensable** in the rule $\phi \rightarrow \varphi$ if $\models s\ \phi \rightarrow \varphi$ implies $\models s\ \phi/(P-\{a\}) \rightarrow \varphi$; otherwise a is **indispensable** in $\phi \rightarrow \varphi$. The PQ-rule $\phi \rightarrow \varphi$ is called **independent** if all $a \in P$ are indispensable in $\phi \rightarrow \varphi$. The subset of attributes $R \subseteq P$ is called a **reduct** of the PQ-rule $\phi \rightarrow \varphi$ if $\phi/R \rightarrow \varphi$ is independent and $\models s\ \phi \rightarrow \varphi$ implies $\models s\ \phi/R \rightarrow \varphi$.

A decision rule $\phi/R \rightarrow \varphi$ is said to be **reduced** if R is a reduct of the PQ-rule $\phi \rightarrow \varphi$.

Step 3) Minimization of the Decision Algorithm

The set of all rules in A having the same successor φ is denoted A_φ, and P_φ is the set of all predecessors of decision rules belonging to A_φ. A decision rule $\phi \rightarrow \varphi$ in A is called **dispensable** in A if $\models s \vee P_\varphi \equiv \{P_\varphi - \{\phi\}\}$, where $\vee P_\varphi$ denotes disjunction of all formulas in P_φ; otherwise $\phi \rightarrow \varphi$ is **indispensable** in A. The set of rules A_φ is called **independent** if all decision rules in A_φ are indispensable in A_φ. The subset A' of decision rules of A_φ is called a **reduct** of A_φ if all decision rules in A' are independent and $\models s \vee P_\varphi \equiv P_\varphi'$. A set of decision rules A_φ is said to be **reduced** if reduct of A_φ is A_φ itself.

A basic algorithm A is said to be **minimal** if every decision rule in A is reduced and for every decision rule $\phi \rightarrow \varphi$ in A, A_φ is reduced.

3 Inductive Learning

3.1 Learning from Examples

Assume that there are two agents: a "knower" and a "learner". Suppose that the knower has knowledge about certain universe of discourse U, that is, he is able to classify elements of the universe U, and classes of the knower's classification from concepts to be learned by the learner. Moreover, it is assumed that the knower has complete knowledge about the universe U and the universe U is closed that is nothing else besides U exists. This assumption is called the closed world assumption(CWA)[7].

Task of a learner is to learn knower's knowledge. Now the problem whether always the learner's knowledge can match the knower's knowledge or whether the knower's knowledge (attributes) depends on learner's knowledge (attributes). As a consequence the degree of dependency between the set of knower's and learner's attribute, can be viewed as a numerical measure of how exactly the knower's knowledge can be learned. To describe this concept, Pawlak[3] defined the quality of learning as follows:

$$k = \gamma_B(C) = \frac{cardPOS_B(C)}{cardU}$$

which is the same quantity with the degree of dependency. This number expresses what percentage of knower's knowledge can be learned by the learner.

3.2 Dynamic Learning

The learning under closed world assumption is viewed as a static learning[3]. The classification rules learned from training examples can be assumed as the background knowledge of the learner. The question arises whether the background knowledge can be used to classify correctly new object not occurring in training examples which occurs under open world assumption(OWA)[7].

Classification of new objects based on background knowledge previously acquired from training examples is considered dynamic learning[3]. Pawlak categorized the possibilities when a new instance is added as follows:

1) the new instance confirms actual knowledge

2) the new instance contradicts actual knowledge
3) the new instance is completely new case.

If the training set is in a certain sense complete, i.e., the decision table is consistent if provides the highest quality of learning and the learners knowledge cannot be improved by means of new instances. If however the training set is inconsistent, every new confirming instance increases learner's knowledge and any new borderline instance decreases his knowledge.

4 Proposed Algorithm

4.1 Refinement of Categorization for New Instances

Observing carefully the steps to minimize decision tables in the previous section gives an idea to make an update law of the minimized decision rules when a new instance is added.

While Pawlak[3][7] considered three possibilities when adding a new instance to the universe as introduced in the last section, we propose four possibilities. Before going ahead, it is required to define several concepts.

Definition 1. Completely contradict: A new instance x is said to be completely contradict if x contradicts the given minimized decision rules and there exists a $y \in U$ such that

$$\phi_x \equiv \phi_y,$$

where x is $\phi_x \rightarrow \varphi_x$ and y is $\phi_y \rightarrow \varphi_y$. ∎

Definition 2. Partially contradict: A new instance x is said to be partially contradict if x contradicts the given minimized decision rules and it doesn't completely contradict the given decision rules. ∎

Now we can suggest categorize the cases when a new instance to the universe as follows:

1) the new instance confirms actual knowledge
2) the new instance completely contradicts actual knowledge
3) the new instance partially contradicts actual knowledge
4) the new instance is completely new case,

4.2 Criteria to Change Reducts

In this paper, we only consider the case when a partially contradict instance is added to the universe.

The minimization method of decision tables find a Q-reduct of P, and then find reducts for each rule, that is, eliminate superfluous values of attributes, and finally find a reduct for the generated rules in the previous step, i.e., eliminate superfluous decision rules.

Now assume that the minimized decision rules for a decision table and all kinds of reduct corresponded are given before a new instance is added. Likewise the minimization steps, now when a new instance is added, the proposed algorithm first checks if the Q-reduct of P should be changed(step 1). If so, find rules for which the reduct for each rule should be updated and recalculate them(step 2). Finally, eliminate superfluous decision rules for newly created decision rules(step 3).

In order to check whether the Q-reduct of P should be changed or not, we have a criterion to decide it. The theorem 1 gives us such a criterion.

Theorem 1. Suppose $S = (U, A)$ is a KR-system and a new instance x is added to U with a basic decision rule $\phi_x \rightarrow \varphi_x$ which partially contradicts the acquired knowledge. Then the reduct of the PQ-basic algorithm, $RED(P, Q)$ changes if and only if there exists a $y \in U$ such that $y \models_S \phi_x / RED(P, Q)$. ∎

In order to find the rule set which the reduct for each rule should be updated, we use the following.

Remark 1. Suppose $S = (U, A)$ is a KR-system and a new instance x is added to U with a basic decision rule $\phi_x \rightarrow \varphi_x$ which partially contradicts the acquired knowledge. Then, we intuitively insist that, for each $i \in U'(= U \cup \{x\})$, the reduct of decision rule $\phi_i \rightarrow \varphi_i$, $RED(\phi_x \rightarrow \varphi_x)$ does not changes if and only if $\phi_i/RED(\phi_i \rightarrow \varphi_i) = \phi_x/RED(\phi_i \rightarrow \varphi_i)$ implies $\phi_i \rightarrow \varphi_i$.

From the above consideration we can make a set of label of which rules in the original minimized set of rules is contradicted by x. Let us denote such a label set be c_x. The overall flowchart of proposed algorithm is given in the figure 1.

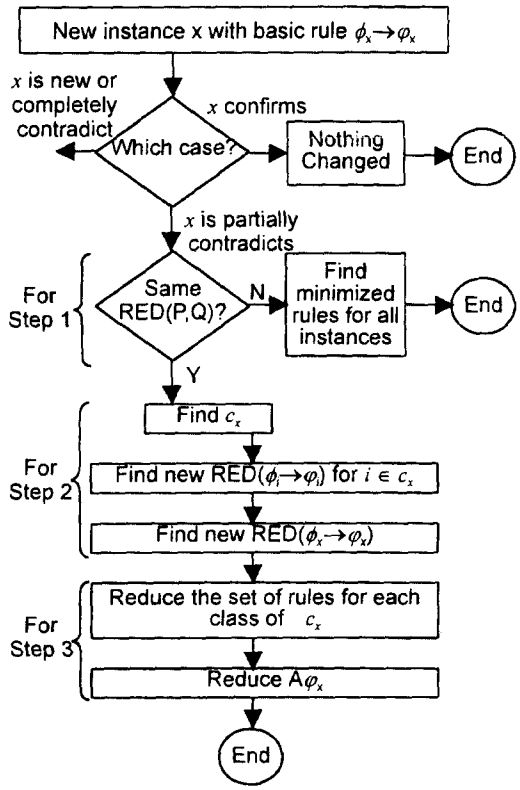

Fig. 1. Proposed inductive learning algorithm when a new instance is added

4.3 An Illustrative Example

Here is a *KR*-system $S = (U, A)$ where $U = \{1, 2, 3, 4, 5, 6, 7\}$, $A = P \cup Q$, $P = \{a, b, c, d\}$, and $Q = \{e\}$.

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

Table 1. A decision table

The minimized decision rules and a Q-reduct of P can be found from the steps in the previous section as follows:

$$a_1 b_0 \rightarrow e_1$$
$$a_0 \rightarrow e_0$$
$$b_1 d_1 \rightarrow e_0$$
$$d_2 \rightarrow e_2$$

and

$$RED(P, Q) = \{a, b, d\}.$$

Suppose a new instance x

U	a	b	c	D	e
x	0	1	2	1	1

is added which partially contradicts the previously acquired knowledge.

By the theorem 1, we can know $RED(P, Q)$ doesn't change since no $y \in U$ and $y \models_s \phi_X / RED(P, Q)$.

Moreover, the remark 1 yields the label set and corresponding reduct for each rule:

$$c_X = \{3, 4\}$$
$$RED(\phi_3 \rightarrow \varphi_3) = \{a, b\}, \{a, d\}$$
$$RED(\phi_4 \rightarrow \varphi_4) = \{a, b, d\}$$
$$RED(\phi_X \rightarrow \varphi_X) = \{a, b\}, \{a, d\}$$

For e_0,

3: $a_0 b_0 \rightarrow e_0$ or $a_0 d_0 \rightarrow e_0$

4: $a_1 b_1 c_0 \rightarrow e_0$,

we can choose

$$a_0 b_0 \rightarrow e_0, a_1 b_1 c_0 \rightarrow e_0$$

On the other hand, for $A\varphi_x$, i.e., e_1,

1, 2: $a_1 b_0 \rightarrow e_1$

x: $a_0 b_1 \rightarrow e_1$ or $a_1 d_0 \rightarrow e_1$,

we can choose

$$a_1 b_0 \rightarrow e_1, a_0 b_1 \rightarrow e_1$$

Therefore the total minimized decision rules are

$$a_0 b_0 \rightarrow e_0$$
$$a_1 b_1 c_0 \rightarrow e_0$$
$$a_1 b_0 \rightarrow e_1$$
$$a_0 b_1 \rightarrow e_1$$

which is identical to the result of recalculation for overall set of instances.

5 Concluding Remarks

In this paper, it is shown a type of inductive learning, whose task is to induce general descriptions of concepts from specific instances of these concepts.

It is proposed, when a new instance is added to the universe of discourse, an algorithm to find minimal

set of rules for decision tables without recalculation for overall set of instances.

The main contribution of this paper is to provide an algorithm of inductive learning for additive instances to efficiently increase the total learned knowledge within the framework of rough set theory assuming that the minimized decision rules for the original decision table is already given.

At last, an illustrative example shows that both the proposed algorithm and the calculation considering all instances at once give us the same result.

This paper still leaves several further works to make the algorithm perfect for any kinds of new instances.

The proposed algorithm does not consider the cases when the new instance is completely new case to the previously acquired knowledge and when it completely contradicts the knowledge. An algorithm to cope with these two cases would be acquired from the mathematical analysis with semantics of decision logic language.

A complexity analysis should be also done for the proposed algorithm to compare the performance of calculation time.

References

[1] Quinlan, J.R., "Learning Efficient Classification Procedures and Their Application to Chess End Games", *Machine Learning: An Artificial Intelligence Approach*, Michalski, R. S., et al.(eds.), Morgan Kaufmann Publishers, Inc., 1983

[2] Forsyth, R., *Machine Learning: Applications in Expert Systems and Information Retrieved*, John Wiley & Sons, 1986

[3] Pawlak, Z., "On Learning – A Rough Set Appoach", *Proc. of the 5th Int. Symp. on Computation Theory* or *Lecture Notes in Computer Science*, Goos, G., et al.(eds.), Vol.208, pp.197-227, 1984

[4] Ras, Z. W., et al., "Rough-Sets Based Learning System", *Proc. of the 5th Int. Symp. on Computation Theory* or *Lecture Notes in Computer Science*, Goos, G., et al.(eds.), Vol.208, pp.265-275, 1984

[5] Arciszewski, T., et al., "A Methodology of Design Knowledge Acquisition for Use in Learning Expert Systems", *Int. J. of Man-Machine Studies*, Vol.27, pp.23-32, 1987

[6] Yasdi, R., et al., "An Expert System for Conceptual Schema Design: A Machine Learning Approach", *Int. J. of Man-Machine Studies*, Vol.29, pp.351-376, 1988

[7] Pawlak, Z., *Rough Sets: Theoretical Aspects and Reasoning about Data*, Kluwer Academic Publishers, 1991

[8] Grzymala-Busse, J. W., et al., "On the Unknown Attribute Values in Learning from Examples", *Proc. of the 6th Int. Symp. on Methodologies for Intelligent Systems* or *Lecture Notes in Artificial Intelligence*, Ras, Z. W., et al.(eds.), Vol.542, pp.368-377, 1991

[9] Slowinski, R., "Rough Set Learning of Preferential Attitude in Multi-Criteria Decision Making", *Proc. of the 7th Int. Symp. on Methodologies for Intelligent Systems* or *Lecture Notes in Artificial Intelligence*, Komorowski, J., et al.(eds.), Vol.689, pp.642-651, 1993

[10] Pawlak, Z., "Data Analysis with Rough Set Theory", *Proc. of KFIS Fall Conference '96*, Vol.6, No. 2, pp.3-19, 1996

Appendix

Proof of theorem 1.

Suppose first there does not exist any $y \in U$ such that $y \models s\ \phi_x/R$ where $R = RED(P, Q)$. Then a new rule x does not conflict to any previous rules in (R, Q). Hence (R, Q) is consistent in the universe of discourse after adding the new instance, denoted by U'.

By definition of $RED(P, Q)$, $RED(P, Q)$ is indispensable in (R, Q) 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 rule x is added to U or not.

Hence, $RED(P, Q)$ is indispensable in (R, Q) in U' and thus $RED(P, Q)$ is still a reduct in U'.

For reverse implication, let a rule z in the minimized rule set which conflicts to the new rule x. Then,

$$|=s\ \phi_x \rightarrow \varphi_x \text{ and } \phi_x \neq \varphi_x \qquad \text{(A.1)}$$

where $x\ |=s\ \phi_x \rightarrow \varphi_x$ and $z\ |=s\ \phi_z \rightarrow \varphi_z$.

Since $\phi_z/R \equiv \phi_z$,

$$|=s\ \phi_x/R \rightarrow \phi_z/R \qquad \text{(A.2)}$$

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

$$y\ |=s\ \phi_x/R, \text{ i.e., } \phi_y/R \equiv \phi_x/R \qquad \text{(A.3)}$$

(A.2) and (A.3) yields

$$|=s\ \phi_y/R \rightarrow \phi_z/R \qquad \text{(A.4)}$$

And, since y and z are consistent in (R, Q) if $|=s$ $\phi_y/R \rightarrow \phi_z/R$ then $\phi_y \equiv \phi_z$. This and (A.4) yields

$$\phi_y \equiv \phi_z \qquad \text{(A.5)}$$

From (A.1) and (A.5),

$$\phi_y \neq \phi_x \qquad \text{(A.6)}$$

In result, from (A.3) and (A.6),

$$\phi_x/R \rightarrow \phi_y/R \text{ and } \phi_y \neq \phi_z.$$

Because the new rule x conflicts to y in (R, Q) in U, which implies that $R = RED(P, Q)$ cannot be a reduct of (P, Q) in U', the proof is complete. ∎