Computational Intelligence, Volume 11, Number 2, 1995

DATA-BASED ACQUISITION AND INCREMENTAL MODIFICATION OF CLASSIFICATION RULES

NING SHAN, WOJCIECH ZIARKO

Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2

One of the most important problems in the application of knowledge discovery systems is the identification and subsequent updating of rules. Many applications require that the classification rules be derived from data representing exemplar occurrences of data patterns belonging to different classes. The problem of identifying such rules in data has been researched within the field of machine learning, and more recently in the context of rough set theory and knowledge discovery in databases. In this paper we present an incremental methodology for finding all maximally generalized rules and for adaptive modification of them when new data become available. The methodology is developed in the context of rough set theory and is based on the earlier idea of discernibility matrix introduced by Skowron.

Key words: rough sets, decision rules, knowledge discovery, machine learning, incremental learning, adaptive systems.

1. INTRODUCTION

Machine learning has been investigated by many researchers and applied to complex real-world problems (Kodratoff and Michalski 1990; Shavlik and Dietterich 1990; Gallant 1993). Currently, one can identify two fundamental approaches to machine learning: neural nets (Gallant 1993) and symbolic learning. Neural nets are concerned with modeling functional relationships existing in the training data by constructing a hierarchy of weighted linear functions. A symbolic learning system has the ability to automatically generate a rule-based knowledge base from a given set of training objects (Kodratoff and Michalski 1990; Shavlik and Dietterich 1990). The knowledge base, represented as a set of classification rules in an *if-then* or a decision tree form, can be used to build an expert system or to help the users better understand the relationships occurring in the domain of interest. The latter aspect is of particular importance in knowledge discovery or database mining applications of symbolic learning (Piatetsky-Shapiro and Frawley 1991).

A classification rule is an assertion that discriminates a concept, i.e., a set of objects characterized by a certain property from other concepts. In this article, we are interested in acquiring and maintaining all maximally generalized (with a minimized number of conditions) classification rules for a target concept. We present a methodology for identifying of such rules in data within the context of rough sets theory.

The theory of rough sets (Pawlak 1991) offers a new approach to machine learning, knowledge discovery in databases, and reasoning from data. This methodology is complementary to statistical methods of inference and provides the necessary formalism to conduct data analysis and knowledge discovery from imprecise and ambiguous data. The main idea behind rough sets is based on the observation that the universe from which we collect data is too refined for a clear identification and representation of knowledge hidden in data in the form of relationships among data items. Extracting such knowledge from the collected data is not a straightforward task. We need to have ways to analyze information at various abstract levels of knowledge representation, going from refined to coarse and vice versa, and we also need to extract useful information from incomplete or noisy data. All these problems have been investigated recently using the methodology of rough sets.

A number of algorithms and systems for rule extraction from data have been developed based on rough set theory (Slowinski 1992; Ziarko 1994). Most of them focus on generating the best minimal cover subset of decision rules that is consistent with a given set of training data. The algorithm incorporated in system LERS (Grzymala-Busse 1991, 1992; Grzymala-Busse and Grzymala-Busse 1993) can also compute all minimal-length rules, which is the main subject of this paper. However, these algorithms do not have incremental learning capability, by which we mean a property of the rule extraction algorithm which enables it to produce an updated set of decision rules (or a decision tree), without regenerating previous results (rules) when new data objects become available. An incremental learning system builds upon its previous results by gradually expanding and updating its knowledge base while more data become available. The incremental learning capability of a knowledge acquisition system offers hope of solving an important problem of maintaining knowledge bases in dynamic environments.

This paper reports an incremental algorithm for identifying all minimal-length decision rules in data and dynamically adjusting the knowledge base. The approach, based on the idea of the decision matrix and discernibility matrix defined in previous research works (Skowron and Rauszer 1992; Skowron and Suraj 1993; Ziarko and Shan 1993), provides a new way of incremental identification of decision rules. The method complements a previously published algorithm called *MINRUL* (Ziarko and Shan 1993) for finding all minimum-length or minimal rules in an attribute-value system. Since the process of computing all minimal rules is NP-hard, it is particularly essential to avoid recomputing the rules after adding new data records.

The primary application for the method seems to be data analysis in the context of a larger knowledge discovery system. Searching for new significant knowledge may involve checking *all* solutions rather than using only some of them, as produced by most of the other rule extraction methods.

This paper is organized as follows. Section 2 reviews some elementary notions and results of rough sets and information systems theory. Section 3 introduces the concept of a decision matrix and some related definitions. The principle of the incremental rule computation algorithm is presented and illustrated with simple examples in Section 4. In Section 5 we discuss our results and review directions for further research.

2. BASIC NOTIONS OF ROUGH SETS

In the process of data analysis using rough set theory (Pawlak 1991), the main computational effort is associated with the determination of attribute-value relationships in attribute-value systems. The rough set model is based on the intuitive observation that lower degree of precision in the representation of objects in such systems makes the data regularities more visible and easier to analyze and characterize in terms of rules. This section deals with some basic concepts as defined in rough set theory. These concepts play very important roles in rough set theory and are adapted by many application systems (Pawlak 1991; Slowinski 1992; Ziarko 1994).

2.1. Rough Sets

Let *OBJ* be a non-empty *finite set* called the *universe*, and let *IND* be an *equivalence relation* over the universe *OBJ*, called an *indiscernibility relation*. The universe is our domain of interest, such as patients in medical application, speech sounds in speech recognition application, etc. The indiscernibility relation represents a classification of the domain objects into classes of objects which are indistinguishable or identical in terms of available information about them. The primary component of the rough set formal model is the notion of *approximation space* defined as an ordered pair $Apr = (OBJ, IND)$. In this approach the indiscernibility relation can be viewed as a formal model of information about objects belonging to the universe *OBJ*. It partitions the universe *OBJ* into *equivalence classes* $[e_i]$, where $e_i \in OBJ$. Equivalence classes of the relation are also referred to as *elementary sets* or *indiscernibility classes* of the partitioning. Any finite union of elementary sets is called a *definable* set in the approximation space *Apr*.

Let X be a subset of *OBJ* to represent a *concept* or *target* of learning. The *lower approximation* or lower bound of X, denoted as $\underline{Apr}(X)$, is the union of all those elementary sets each of which can be classified as definitely belonging to the set X based on the classification information represented by the indiscernibility relation R, that is,

$$\underline{Apr}(X) = \{e_i \in OBJ : [e_i] \subseteq X\}.$$

The *upper approximation* or upper bound of set X, denoted as $\overline{Apr}(X)$, is the union of all those elementary sets each of which can be classified as possibly belonging to the set X based on the classification information represented by the indiscernibility relation R, that is,

$$\overline{Apr}(X) = \{e_i \in OBJ : [e_i] \cap X \neq \emptyset\}.$$

Any set specified approximately in terms of its lower and upper bounds is called a *rough set*. The *boundary region* of set X, denoted as $BND(X)$, is the union of all those elementary sets for each of which it cannot be determined with certainty whether the object from this area belongs to the set X or X complement, $-X$. The boundary area formally can be expressed as a set difference

$$BND(X) = \overline{Apr}(X) - \underline{Apr}(X).$$

In other words, in the boundary region, none of elementary sets can be classified with certainty as associated with the concept X or the concept complement $-X$.

If the information about objects is sufficient to classify all of elementary sets, that is, if $\overline{Apr}(X) = \underline{Apr}(X)$, then the boundary region of the set X disappears and the rough set becomes equivalent to the standard set. Only in this case the precise classification rules can be identified by a data analysis system. However, if the lower and upper bounds of a concept are not identical, then approximate classification rules can be produced. That is, for any target concept in the learning system, we can derive two kinds of classification rules by using the lower and upper approximations of the concept X. The rules obtained from the lower approximation of the concept are called deterministic rules. Whenever the description of an object matches a deterministic rule, this object is definitely in the target concept. The rules obtained from the upper approximation of the concept are called nondeterministic rules. Whenever the description of an object matches a nondeterministic rule, this object may or may not be included in the target concept.

2.2. Information Systems

Information systems are formal models of attribute-value systems. Their logical properties have been investigated extensively by many researchers (Pawlak 1991; Skowron and Rauszer 1992; Skowron and Suraj 1993; Orlowska and Orlowski 1992; Ziarko 1993, 1994; Ziarko and Shan 1993). In this article, we assume that the information available to a rule computation system can be represented in the framework of a formal information system model. The basic component of an information system S is a set of objects. It can be conveniently represented by a data table called an information table, the columns of which are labeled by attribute names. The rows of the information table correspond to objects of the universe. Each row represents the information about an object in the information system S in terms of attribute values. The attributes used to describe objects of the information system are typically some elementary features of the objects.

TABLE 1. Information Table.

OBJ	SIZE	HAIR	EYES	COMPLEXION	CLASSIFICATION
obj_1	short	dark	blue	pale	0
obj_2	tall	dark	brown	matt	1
obj_3	tall	red	blue	pale	0
obj_4	short	blond	blue	matt	1
obj_5	tall	blond	blue	pale	1
obj_6	tall	dark	blue	pale	0
obj_7	tall	blond	brown	matt	1
obj_8	short	dark	brown	matt	1

Formally, the information system can be represented as a quadruple $S = \langle OBJ, C, D, \{VAL_a\}_{a \in A} \rangle$, where OBJ is a non-empty set of objects, C is a non-empty set of condition attributes, and D is a non-empty set which contains decision attributes and $A = C \cup D$, the set of all attributes. VAL_a is a domain of an attribute a with at least two elements. The elements of VAL_a are called values of attribute a ($a \in A$). Each attribute $a \in A$ can be perceived as a function assigning a value $a(obj) \in VAL_a$ to each object $obj \in OBJ$. Every object which belongs to OBJ is therefore associated with a set of values corresponding to the condition attributes C and decision attributes D. Given a limited number and a limited resolution capability of attributes, some objects may become indistinguishable. This results in the partitioning of the universe into disjoint classes of identical objects as formally described below.

For any subset of attributes $B \subseteq A$ in an information system, let $IND(B)$ be an equivalence relation over the universe OBJ, called the indiscernibility relation and defined as follows:

$$IND(B) = \{(obj_x, obj_y) : \forall a \in B, a(obj_x) = a(obj_y)\}, \quad obj_x, obj_y \in OBJ\,.$$

If $(obj_x, obj_y) \in IND(B)$, then obj_x and obj_y are called indiscernible with respect to B. In other words, two objects are indiscernible with respect to condition attributes B whenever they have the same values of all conditions belonging to the set B. That is, an information system represents a classification of the domain of interest. In practical applications, the construction of an information system from empirical data involves defining higher order attributes, for example, by discretizing the original numeric values or by using some other concept generalization techniques (Han *et al.* 1992).

Table 1 shows an example of an information system. The universe of discourse OBJ consists of eight objects, $OBJ = \{ obj_1, obj_2, \ldots, obj_8 \}$. Each object is described by set of condition attributes $C = \{$ SIZE, HAIR, EYES, COMPLEXION $\}$, with attribute values $VAL_{size} = \{short, tall\}$, $VAL_{hair} = \{dark, red, blond\}$, $VAL_{eyes} = \{blue, brown\}$, and $VAL_{complexion} = \{pale, matt\}$. The set of values $VAL_{classification} = \{0, 1\}$ of the decision attribute D represents the set of concept descriptions which are to be learned based on the attribute values of C.

In our terminology, the concept is a subset of objects with a common value of the decision attribute. All indiscernibility classes belonging to the lower bound of the concept are said to be positive, whereas all classes of objects outside the concept lower bound are called negative.

2.3. Reduction of Attributes

It is possible that some attributes in an information system may be redundant or irrelevant and could be eliminated without losing any essential classification information. The process of obtaining a minimal nonredundant set of attributes is called *attribute reduction* (Pawlak 1991). At the end of this process, we obtain an "incomplete" decision table containing only those attributes absolutely necessary to make a decision.

To describe the basics of attribute reduction, let us denote by C^* the collection of equivalence classes of the relation $IND(C)$ and let D^* be a family of equivalence classes of the relation $IND(D)$. The $POS(C, D)$ is a union of lower approximations of all elementary sets of the partition D^* in the approximation space $Apr = (OBJ, IND(C))$. We say that the set C of condition attributes is independent with respect to the set D of decision attributes if, for every proper subset P of attributes C,

$$POS(C, D) \neq POS(P, D);$$

otherwise C is said to be dependent on D.

The set R of attributes is a reduct of C if R is an independent subset of C with respect to D such that

$$POS(R, D) = POS(C, D).$$

In the context of this paper, we are more interested in a special kind of reduct called *relative reduct*. The definition of the relative reduct is identical to the above definition except that the reduct is computed with respect to the lower approximation $POS(C, D = d)$ of the set of objects with the value d of the decision attribute D. That is, in this case $POS(C, D)$ is substituted by $POS(C, D = d)$ in the reduct definition.

In general, attribute reducts are the minimal subsets of condition attributes C, preserving a given relationship of interest with a set of decision attributes D. The collection of all reducts will be denoted here as $RED(D)$. None of the attributes of the minimal subset can be eliminated without affecting the relationship with the decision attribute D. Any of the reduct sets, $R \in RED(D)$, can be used instead of all condition attributes C of the information system S. The idea of attribute reduction can be used to find one or more minimal subsets in the set of condition attributes C. These minimal subsets can discern decision classes as well as the complete set of condition attributes.

2.4. Reduction of Decision Rules

The generation of decision rules (knowledge base) is an important aspect of the rough set methodology. Decision rules can be obtained directly from an information system. Some condition values may be unnecessary in a decision rule produced directly from such a table. Such values can then be dropped to create a simpler rule. The process in which the maximum number of condition attribute values are removed without loosing essential information is called *value reduction* and the resulting rule is called minimal.

To describe the process of value reduction, let $C^* = \{X_1, X_2, \ldots, X_n\}$ be the set of equivalence classes generated by the set of condition attributes C. We can apply the concept of reduct of attributes to distinguish the equivalence class X_i from the remaining equivalence classes $X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$, thus obtaining the set of attributes $R_i \subseteq C$ characteristic to each equivalence class X_i. This process leads to n reducts (possibly different) such that each reduct R_i is associated with the equivalence class X_i.

It is possible for one set of attributes to have more than one group of minimal decision rules covering the target concept. The rule-generating process in such case depends much on the sequence in which attributes are processed. Most machine learning algorithms attempt to find the minimal, with respect to inclusion relation, set of such rules. Intuitively, reducing knowledge representation means dropping all dispensable condition attributes and condition attribute values from the information system. In reality, however, it is very difficult to determine which covering set of minimal decision rules or which attribute-reduced table should be used. One possible way of dealing with this problem is by computing all minimal rules by using the idea of decision matrix described in the next section.

3. USING A DECISION MATRIX FOR COMPUTING RULES AND REDUCTS

A decision matrix (Ziarko and Shan 1993), or the related concept of discernibility matrix (Skowron and Rauszer 1992; Skowron and Suraj 1993), can be used to compute all minimal decision rules and reducts of information system S. It provides a way to generate the simplest set of decision rules, while preserving all classification information. The approach to generation of rules, as presented here, is based upon the construction of a number of Boolean functions (Skowron and Suraj 1993; Ziarko and Shan 1993) from decision matrices.

Before we introduce the concept of a decision matrix, we assume that all classes of positive and negative objects in a classification corresponding to an indiscernibility relation are separately numbered with subscripts i (i.e., $i = 1, 2, \ldots, \gamma$) and j (i.e., $j = 1, 2, \ldots, \rho$) respectively. The *decision matrix* $M(S) = (M_{ij})$ of an information system S is defined as a $\gamma \times \rho$ matrix whose entry at the position (i, j) is a set of attribute-value pairs:

$$M_{ij} = \{(a, a(i)) : a(i) \neq a(j)\}, \qquad (i = 1, 2, \ldots, \gamma;\ j = 1, 2, \ldots, \rho),$$

where $a(i)$ is a value of attribute a, common to all objects in the class i. The set M_{ij} contains all attribute-value pairs (*attribute*, *value*) whose values are not identical between classes i and j. In other words, M_{ij} represents the complete information distinguishing class i from class j of objects. An example decision matrix is shown in Table 2.

By using the concept of the decision matrix, the set of all minimal-length decision rules $|B_i|$ for a given class i ($i = 1, 2, \ldots, \gamma$) can be computed (Ziarko 1993). To find the rules, a Boolean expression

$$B_i = \bigwedge_j \bigvee M_{ij},$$

where $\bigwedge$ and $\bigvee$ are respectively generalized conjunction and disjunction operators, is formed first. The Boolean expression, called a decision function B_i, is constructed out of a row i of the decision matrix, which is row $(M_{i1}, M_{i2}, \ldots, M_{i\rho})$, by formally treating each attribute-value pair occurring in the component M_{ij} as a Boolean variable and then forming a Boolean conjunction of disjunctions of components belonging to each set M_{ij} ($j = 1, 2, \ldots, \rho$). The decision rule sets $|B_i|$ are obtained by turning such an expression into disjunctive normal form and, for example, using the absorption law of Boolean algebra to simplify it. The conjuncts, or prime implicants of the simplified decision function correspond to the minimal decision rules (Skowron and Suraj 1993; Ziarko and Shan 1993). Similarly, by treating the complement of the decision class as a target concept, a set of decision rules can be computed for each object of the complement of the target class, using the same approach. Some of the decision functions, obtained from decision matrices given in Table 2. and derived from the information contained in Table 1, are presented in Example 1.

Once all the decision rule sets $|B_i|$ have been computed, a set of all minimal, that is, maximally generalized, decision rules *RUL* for the lower bound $POS(C, D = d)$ of the target concept corresponding to the value d of the decision attribute (or attributes) D can be obtained by taking the union of rule sets computed for all positive classes of the information system, that is,

$$RUL = \bigcup |B_i| \qquad (i = 1, 2, \ldots, \gamma).$$

To compute the reducts of an information system, we introduce the notion of the *phantom* decision function $\tilde{B}_i$ and a Boolean function called the *reduct* function F_{RED}. A phantom decision function $\tilde{B}_i$ is a Boolean expression defined by the conjunction of all Boolean expressions $\bigvee \tilde{M}_{ij}$ of row i in the given decision matrix, where $\bigvee \tilde{M}_{ij}$ represents the disjunction of the attribute names of the component M_{ij}, that is,

$$\tilde{B}_i = \bigwedge_j \bigvee \tilde{M}_{ij} \qquad (j = 1, 2, \ldots, \rho).$$

Informally speaking, a phantom decision function $\tilde{B}_i$ is similar to the decision function B_i with except that the elements of the Boolean expression do not include the attribute values. One can directly derive the phantom decision function $\tilde{B}_i$ from the corresponding decision function B_i by eliminating the values of attributes from the result.

The reduct function, F_{RED}, is a Boolean function constructed by logical conjunction of all phantom decision functions $\tilde{B}_i$, that is,

$$F_{RED} = \bigwedge_i \tilde{B}_i \qquad (i = 1, 2, \ldots, \gamma)$$

or

$$F_{RED} = \bigwedge_i \left(\bigwedge_j \bigvee \tilde{M}_{ij} \right) \qquad (i = 1, 2, \ldots, \gamma; \ \ j = 1, 2, \ldots, \rho)$$

The set of all reducts for the lower bound $POS(C, D = d)$ of the target class, denoted as *RED*, is obtained by performing the required multiplications and simplification by applying the absorption law of Boolean algebra to the Boolean expression F_{RED}. The conjuncts, or prime implicants of the simplified reduct function, are the relative reducts with respect to the lower approximation $POS(C, D = d)$ of the target concept.

Example 1. Table 2 depicts two decision matrices obtained from the information system given in Table 1. In these decision matrices, S is an abbreviation for *SIZE*, H for *HAIR*, and so on. $VAL_S = \{0, 1\}$ represents $VAL_{SIZE} = \{short, tall\}$, $VAL_H = \{0, 1, 2\}$ represents $VAL_{HAIR} = \{dark, red, blond\}$, $VAL_E = \{1, 2\}$ represents $VAL_{EYES} = \{blue, brown\}$, and $VAL_C = \{0, 1\}$ represents $VAL_{COMPLEXION} = \{pale, matt\}$. Each cell (i, j) in a decision matrix is a collection of attribute-value pairs distinguishing row i of the target class from column j of its complement.

Based on these decision matrices we can obtain the following decision functions B_i^0 $(i = 1, 2, 3)$ from the class 0 decision matrix, and B_i^1 $(i = 1, 2, \ldots, 5)$ from the class 1 decision matrix.

Class 0 decision functions:

$$\begin{aligned}
B_1^0 = {} & ((S,0) \vee (E,1) \vee (C,0)) \wedge ((H,0) \vee (C,0)) \wedge ((S,0) \vee (H,0)) \wedge ((S,0) \vee (H,0) \vee (E,1) \vee (C,0)) \\
& \wedge((E,1) \vee (C,0)) = ((S,0) \wedge (C,0)) \vee ((H,0) \wedge (E,1)) \vee ((H,0) \wedge (C,0)) \\
B_2^0 = {} & ((H,1) \vee (E,1) \vee (C,0)) \wedge ((S,1) \vee (H,1) \vee (C,0)) \wedge ((H,1)) \wedge ((H,1) \vee (E,1) \vee (C,0)) \\
& \wedge((S,1) \vee (H,1) \vee (E,1) \vee (C,0)) = (H,1) \\
B_3^0 = {} & ((E,1) \vee (C,0)) \wedge ((S,1) \vee (H,0) \vee (C,0) \wedge ((H,0)) \wedge ((H,0) \vee (E,1) \vee (C,0)) \\
& \wedge((S,1) \vee (E,1) \vee (C,0)) = ((H,0) \wedge (E,1)) \vee ((H,0) \wedge (C,0))
\end{aligned}$$

TABLE 2. Decision Matrices for Table 1

Class	j	1	2	3	4	5
i	OBJ	obj_2	obj_4	obj_5	obj_7	obj_8
1	obj_1	(S,0)(E,1)(C,0)	(H,0)(C,0)	(S,0)(H,0)	(S,0)(H,0)(E,1)(C,0)	(E,1)(C,0)
2	obj_3	(H,1)(E,1)(C,0)	(S,1)(H,1)(C,0)	(H,1)	(H,1)(E,1)(C,0)	(S,1)(H,1)(E,1)(C,0)
3	obj_6	(E,1)(C,0)	(S,1)(H,0)(C,0)	(H,0)	(H,0)(E,1)(C,0)	(S,1)(E,1)(C,0)

(a) A decision matrix for class 0.

$CLASS$	j	1	2	3
i	OBJ	obj_1	obj_3	obj_6
1	obj_2	(S,1)(E,2)(C,1)	(H,0)(E,2)(C,1)	(E,2)(C,1)
2	obj_4	(H,2)(C,1)	(S,0)(H,2)(C,1)	(S,0)(H,2)(C,1)
3	obj_5	(S,1)(H,2)	(H,2)	(H,2)
4	obj_7	(S,1)(H,2)(E,2)(C,1)	(H,2)(E,2)(C,1)	(H,2)(E,2)(C,1)
5	obj_8	(E,2)(C,1)	(S,0)(H,0)(E,2)(C,1)	(S,0)(E,2)(C,1)

(b) A decision matrix for class 1.

Class 1 decision functions:

$$B_1^1 = ((S,1) \vee (E,2) \vee (C,1)) \wedge ((H,0) \vee (E,2) \vee (C,1)) \wedge ((E,2) \vee (C,1)) = (E,2) \vee (C,1)$$

$$B_2^1 = ((H,2) \vee (C,1)) \wedge ((S,0) \vee (H,2) \vee (C,1)) \wedge ((S,0) \vee (H,2) \vee (C,1)) = (H,2) \vee (C,1)$$

$$B_3^1 = ((S,1) \vee (H,2)) \wedge ((H,2)) \wedge ((H,2)) = (H,2)$$

$$\begin{aligned} B_4^1 = {} & ((S,1) \vee (H,2) \vee (E,2) \vee (C,1)) \wedge ((H,2) \vee (E,2) \vee (C,1)) \\ & \wedge ((H,2) \vee (E,2) \vee (C,1)) = (H,2) \vee (E,2) \vee (C,1) \end{aligned}$$

$$B_5^1 = ((E,2) \vee (C,1)) \wedge ((S,0) \vee (H,0) \vee (E,2) \vee (C,1)) \wedge ((S,0) \vee (E,2) \vee (C,1)) = (E,2) \vee (C,1)$$

Some example decision rules derived from B_1^0 are

$$(SIZE = short) \wedge (COMPLEXION = pale) \rightarrow (CLASS =' 0')$$
$$(HAIR = dark) \wedge (EYES = blue) \rightarrow (CLASS =' 0')$$
$$(HAIR = dark) \wedge (COMPLEXION = pale) \rightarrow (CLASS =' 0')$$

The $\bigcup |B_i^0|$ corresponds to all the minimal decision rules RUL for the class 0 of the information system shown in Table 1:

$$(SIZE = short) \wedge (COMPLEXION = pale) \rightarrow (CLASS =' 0')$$
$$(HAIR = dark) \wedge (EYES = blue) \rightarrow (CLASS =' 0')$$
$$(HAIR = dark) \wedge (COMPLEXION = pale) \rightarrow (CLASS =' 0')$$
$$(HAIR = red) \rightarrow (CLASS =' 0')$$

Similarly, $\bigcup |B_i^1|$ represents the set of all minimal rules for the class 1:

$$(EYES = brown) \rightarrow (CLASS =' 1')$$
$$(COMPLEXION = matt) \rightarrow (CLASS =' 1')$$
$$(HAIR = blond) \rightarrow (CLASS =' 1')$$

Next, let us compute the reduct functions and reduct sets $RED(0)$ and $RED(1)$ for the classes 0 and 1 respectively. The reduct function for class 0 is

$$\begin{aligned} F_{RED(0)} &= \bigwedge \tilde{B}_i^0 \qquad (i = 1, 2, 3) \\ &= ((S \wedge C) \vee (H \wedge E) \vee (H \wedge C,)) \wedge (H) \wedge ((H \wedge E) \vee (H \wedge C)) \\ &= (H \wedge E) \vee (H \wedge C), \end{aligned}$$

whereas for class 1 we have

$$\begin{aligned} F_{RED(1)} &= \bigwedge \tilde{B}_j^1 \qquad (j = 1, 2, 3, 4, 5) \\ &= ((E) \vee (C)) \wedge ((H) \vee (C)) \wedge (H) \wedge ((H) \vee (E) \vee (C)) \wedge ((E) \vee (C)) \\ &= (H \wedge E) \vee (H \wedge C). \end{aligned}$$

Consequently, from the simplified result we obtain the following sets of reducts for class 0 and class 1:

$$RED(0) = \{HE, HC\}; \qquad RED(1) = \{HE, HC\}$$

The primary objectives behind introducing the construct of a decision matrix are generating minimal decision rules and providing an efficient way to find all reducts from the given data set. In practical application, the decision matrix approach may result in a large number of decision rules, some of which may be weakly supported by the data. On the other hand, one may prefer to focus on the strongest patterns and use only some subjectively interesting rules or the ones that passed some statistical significance tests. One way of handling this problem is to give the user all minimal decision rules and reducts and let him select only reducts and rules that pass predefined selection criteria.

4. INCREMENTAL ADAPTATION OF RULES AND REDUCTS

An important objective of incremental update of information is maintaining the knowledge base in a dynamic environment. In machine learning or knowledge discovery problems in which new objects are added to an information system S, it is highly beneficial from an efficiency point of view to accept objects and modify existing rules and reducts in an incremental fashion, instead of regenerating them. The update mechanism presented here processes new objects by modifying current decision matrices and decision rules. The basic steps of the knowledge base update process called Increment are described here in detail. When presenting Increment, we assume that a new object has been added to the information system, and this addition has resulted in creation of a new class in a classification of objects with respect to the indiscernibility relation $IND(C)$. If the new object falls into a positive class and its value of the decision attribute is consistent with the target concept, or if it falls into any negative class, then the decision matrix, rules, and reducts will not be affected by this modification. If the new object falls into a positive region and its decision value is inconsistent with the target concept, then the decision matrix, rules, and reducts have to be modified. This problem, however, can be shown to be a special case of the process Increment described below. It essentially requires the elimination of one row of the decision matrix (corresponding to the positive class absorbing the new inconsistent object), along with rules obtained from this row, and the addition of one column to the matrix. On the other hand, the case of adding one column to the matrix and its implications on the set of rules are part of the process Increment. The process Increment specifies necessary modifications to the system of rules to maintain their consistency with the set of stored objects as new objects are added.

Process Increment

Step 1. Determine, based on value of the decision attribute of the new object, which target concept the new object belongs to. If the new object does not belong to any of the current target concepts, then define a new target concept and add it to the set of target concepts. Next, for each decision value d, consider the problem of modifying rules and reducts for the positive region $POS(C, D = d)$ of the set of objects with the decision value d. Let (M_{ij}) be the current decision matrix for this problem and let us denote the new class which resulted from the addition of new object as k.

Step 2. If the new class k is included in the positive region of the current concept, then

Step 2.1. Create a new row of decision matrix (M_{kj}):

$$(M_{kj}) = \{(a, a(k)) : a(k) \neq a(j), \quad j = 1, 2, \ldots, \rho\}$$

Step 2.2. Compute the decision function of class k

$$B_k = \bigwedge_j \bigvee M_{kj}$$

and the associated set of decision rules $|B_k|$.

Step 2.3. Update the current set of decision rules RUL to become

$$RUL' = RUL \cup |B_k|$$

Step 2.4. Update the reduct function F_{RED} of the current concept,

$$F'_{RED(V)} = F_{RED} \bigwedge \left(\bigwedge_j \bigvee \tilde{M}_{kj} \right)$$

and the associated set of reducts RED'.

TABLE 3. A New Object.

i	OBJ	SIZE	HAIR	EYES	COMPLEXION	CLASSIFICATION
6	obj_9	tall	dark	brown	pale	1

Step 3. If the new class k is not included in the positive region of the current concept, then

Step 3.1. Update the current decision matrix (M_{ij}) by creating a new column of decision matrix:

$$(M_{ik}) = \{(a, a(i)) : a(i) \neq a(k), \ \ i = 1, 2, \ldots, \gamma\}$$

Step 3.2. Update the current decision functions B_i to become B_i' by constructing and simplifying the Boolean expression

$$B_i' = B_i \bigwedge \bigvee M_{ik}$$

Step 3.3. Update the rule set by taking the union of the associated rule sets, that is,

$$RUL' = \bigcup |B_i'|$$

Step 3.4. Update the current reduct function F_{RED} to become the new reduct function F_{RED}' by constructing and simplifying the following Boolean expression

$$F_{RED}' = F_{RED} \bigwedge \left(\bigvee \tilde{M}_{ik}\right)$$

and then update the set of reducts RED' by using prime implicants of the above expression. ■

The algorithm offers an approach to incremental updating of all minimal decision rules and reducts. Although the algorithm essentially applies to extraction of rules in deterministic cases, it can be extended to nondeterministic cases by constructing rules with decision probabilities using the extended model of rough sets, the Variable Precision Rough Sets model (VPRS) (Ziarko 1993).

Example 2. In Example 1, we obtained all minimal decision rules RUL for the information system shown in Table 1. Now let us consider adding a new object as shown in Table 3. Suppose that the new object obj_9 belongs to class 1. To update the knowledge base, we create the following new row in the decision matrix for class 1:

$$M^1_{6,1} = (S, 1)(E, 2) \quad M^1_{6,2} = (H, 0)(E, 2) \quad M^1_{6,3} = (E, 2)$$

and compute the decision function B^1_6 for the new object:

$$B^1_6 = ((S, 1) \vee (E, 2)) \wedge ((H, 0) \vee (E, 2)) \wedge ((E, 2)) = (E, 2)$$

By adding the rule corresponding to the term (E, 2) to the set of rules obtained in Example 1, we obtain the following updated set of rules for class 1:

$$(EYES = brown) \rightarrow (CLASS =' 1')$$
$$(COMPLEXION = matt) \rightarrow (CLASS =' 1')$$
$$(HAIR = blond) \rightarrow (CLASS =' 1')$$

Now we compute the new set $RED'(1)$ of reducts for class 1 by updating the reduct function and the reduct set $RED(1)$ obtained in Example 1 by conjuncting it with the new elements of $\bigvee \tilde{M}^1_{6,j}$ (i.e., $j = 1, 2, 3$) or with the phantom decision function $\tilde{B}^1_6 = (E)$, so that

$$\begin{aligned} F'_{RED(1)} &= F_{RED(1)} \bigwedge \left(\bigwedge_j \bigvee \tilde{M}_{6,j} \right) \qquad (j = 1, 2, 3) \\ &= F_{RED(1)} \bigwedge \tilde{B}^1_6 = ((H \wedge E) \vee (H \wedge C)) \wedge (E) = H \wedge E \end{aligned}$$

Thus, the updated set of reducts of class 1, $RED'(1)$ is

$$RED'(1) = \{HE\}$$

To update the rules for class 0, we create a new column in the decision matrix for class 0,

$$M^0_{1,4} = (S, 0)(E, 1) \qquad M^0_{2,4} = (H, 1)(E, 1) \qquad M^0_{3,4} = (E, 1)$$

and then we update the decision functions B^0_i ($i = 1, 2, 3$) as derived from Table 2(a).

$$\begin{aligned} B^{0'}_1 &= B^0_1 \wedge ((S, 0) \vee (E, 1)) = ((S, 0) \wedge (C, 0)) \vee ((H, 0) \wedge (E, 1)) \\ B^{0'}_2 &= B^0_2 \wedge ((H, 1) \vee (E, 1)) = (H, 1) \\ B^{0'}_3 &= B^0_3 \wedge ((E, 1)) = ((H, 0) \wedge (E, 1)) \end{aligned}$$

From the above computation, we obtain the new set of minimal decision rules for class 0.

$$\begin{aligned} &(SIZE = short) \wedge (COMPLEXION = pale) \rightarrow (CLASS =' 0') \\ &(HAIR = dark) \wedge (EYES = blue) \rightarrow (CLASS =' 0') \\ &(HAIR = red) \rightarrow (CLASS =' 0') \end{aligned}$$

Similarly, to update the set of reducts for the class 0 by conjuncting the reduct function $F_{RED(0)}$ obtained in Example 1 with the elements of $\bigvee \tilde{M}^0_{i,4}$ (i.e., $i = 1, 2, 3$). Thus,

$$\begin{aligned} F'_{RED'(0)} &= F_{RED(0)} \bigwedge_i \bigvee \tilde{M}_{i,4} \qquad (i = 1, 2, 3) \\ &= ((H \wedge E) \vee (H \wedge C)) \wedge (S \vee E) \wedge (H \vee E) \wedge (E) = (H \wedge E). \end{aligned}$$

That is, the set of reducts $RED'(0)$ for class 0 is

$$RED'(0) = \{HE\}.$$

5. SUMMARY AND CONCLUSIONS

An incremental process for construction and adaptation of all minimal-length rules derived from an information table has been developed. The process can be used to produce and update rules and reducts, i.e., alternative minimal representations of the classification information. The approach is based on a decision matrix method stemming from the earlier concept of discernibility matrix. The core of the rule discovery algorithm offers an improvement over traditional approaches in that it dynamically constructs and modifies all rules in the rule identification process. The key feature of the presented method is that the main computation of decision rules and reducts is reduced to the problem of simplification and adaption of a group of associated Boolean expressions. That is, there is no need for a search mechanism to find and adapt all rules with this method. To our knowledge, no other rule generation algorithm has these capabilities.

Today, one of the main obstacles in the application of the learning and knowledge discovery systems to the real-world problems is computational time. In case of this method, the size of the original database has only linear effect on the processing time. What really matters in that respect is the size of the information table, i.e., the final table representing the classes of objects obtained after grouping original data records into identity classes according to values of higher order, generalized attributes. This makes this method applicable to practically any size database as long as the corresponding information table is not exceedingly large. The latter aspect, however, can be controlled, i.e., the size of the information table can be bounded by proper selection of higher order attributes. Nevertheless, if the values of information system attributes are relatively fine or the number of attributes is large, the computational requirements of a knowledge discovery system using this method may exceed the capabilities of typical sequential processor architecture. It is our belief that in cases of this kind, only through the use of parallel processing will major progress and achievements be possible. Since the Boolean expression simplification problem can be decomposed into a number of disjoint subproblems by independently simplifying parts of the whole formula, the presented approach is suitable for implementation with parallel multiprocessor systems.

ACKNOWLEDGMENTS

The research reported in this paper was supported in part by an operating grant from the Natural Sciences and Engineering Research Council of Canada. The authors acknowledge the referees' valuable contributions to this paper.

REFERENCES

GALLANT, S. I. 1993. Neural network learning and expert systems. MIT Press, Cambridge, MA.

GRZYMALA-BUSSE D. M., and J. W. GRZYMALA-BUSSE. 1993. Comparision of machine learning and knowledge acquisition methods of rule induction based on rough sets. *Edited by* W. Ziarko. *In* Rough sets, fuzzy sets and knowledge discovery. Springer-Verlag, New York, pp. 282–289.

GRZYMALA-BUSSE, J. W. 1991. Managing uncertainty in expert systems. Kluwer, Boston.

GRZYMALA-BUSSE, J. W. 1992. LERS—a system for learning from examples based on rough sets. *In* Intelligent decision support: handbook of applications and advances of rough sets theory. *Edited by* R. Slowinski. Kluwer, Boston, pp. 3–18.

HAN, J., Y. CAI, and N. CERCONE. 1992. Knowledge discovery in databases: an attribute-oriented approach. *In* Proceeding of the 18th VLDB Conference, Vancouver, B.C., Canada, pp. 335–350.

KODRATOFF, Y., and R. S. MICHALSKI. Editors. 1990. Machine learning: an artificial intelligence approach. Vol. 3. Morgan Kaufmann, San Mateo, CA.

ORLOWSKA, E., and M. ORLOWSKI. 1992. Maintenance of knowledge in dynamic information systems. *In* Intelligent decision support: handbook of applications and advances of rough sets theory. *Edited by* R. Slowinski. Kluwer, Boston, pp. 315–330.

PAWLAK, Z. 1991. Rough sets: theoretical aspects of reasoning about data. Kluwer, Boston.

PIATETSKY-SHAPRO, G., and W. J. FRAWLEY. Editors. 1991. Knowledge discovery in databases. AAAI Press/MIT Press, Cambridge, MA.

SHAVLIK, J. W., and T. G. DIETTERICH. Editors. 1990. Readings in machine learning. Morgan Kaufmann, San Mateo, CA.

SKOWRON, A., and C. RAUSZER. 1992. The discernibility matrices and functions in information systems. *In* Intelligent decision support: handbook of applications and advances of rough sets theory. *Edited by* R. Slowinski. Kluwer, Boston, pp. 331–362.

SKOWRON, A., and Z. SURAJ. 1993. A rough set approach to real-time state identification for decision making. ICS Report 18/93. Warsaw University of Technology.

SLOWINSKI, R. Editor. 1992. Intelligent decision support: handbook of applications and advances of rough sets theory. Kluwer, Boston.

ZIARKO, W. 1993. Variable precision rough set model. Journal of Computer and System Sciences, **46**(1):39–59.

ZIARKO, W. Editor. 1994. Rough sets, fuzzy sets and knowledge discovery. Springer-Verlag, New York.

ZIARKO, W., and N. SHAN. 1993. A rough set-based method for computing all minimal deterministic rules in attribute-value systems. Technical Report CS-93-02. Department of Computer Science, University of Regina, Canada.