ELSEVIER

Information Sciences 112 (1998) 39–49

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

Rough set approach to incomplete information systems

Marzena Kryszkiewicz [1]

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

Received 1 February 1997; received in revised form 1 October 1997; accepted 30 December 1997

Communicated by Zdzislaw Pawlak

Abstract

In the paper we present Rough Set approach to reasoning in incomplete information systems. We propose reduction of knowledge that eliminates only that information, which is not essential from the point of view of classification or decision making. In our approach we make only one assumption about unknown values: the real value of a missing attribute is one from the attribute domain. However, we do not assume which one. We show how to find decision rules directly from such an incomplete decision table, which are as little non-deterministic as possible and have minimal number of conditions.

Keywords: Rough sets; Incomplete information systems; Decision rules

1. Introduction

Rough Set theory [1] has been conceived as a tool to conceptualize, organize and analyze various types of data, in particular, to deal with inexact, uncertain or vague knowledge in applications related to Artificial Intelligence.

In this paper we present Rough Set approach to incomplete information systems, i.e. to systems in which attribute values for objects may be unknown (*missing*, *null*). Our main concern is devoted to finding rules from such systems.

[1] E-mail: mkr@ ii.pw.edu.pl.

0020-0255/98/$19.00
PII: S0020-0255(98)10019-1

Different ways were described in which null values may be handled [2–4]. E.g. the methodology from [2] consists in transforming an incomplete system to a complete system, where each object with incomplete descriptor from the source system is represented by a set of quasi-objects in the target system. Another approach presented in [2] consists in removing objects with unknown values from the original system. Our approach is substantially different from those mentioned above since it does not require the changes in the original system and still is capable of reducing dispensable knowledge efficiently. We propose reduction of knowledge that eliminates only that information, which is not essential from the point of view of classification or decision making. We show how to find decision rules for an incomplete decision table, which are as little non-deterministic as possible and have minimal number of conditions. This type of knowledge reduction restricted to the case of complete information systems was discussed thoroughly in [5–9].

2. Incomplete information systems

Information system (IS) is a pair $\mathcal{S} = (\mathcal{O}, \mathrm{AT})$, where $\mathcal{O}$ is a non-empty finite set of *objects* and AT is a non-empty finite set of *attributes*, such that $a : \mathcal{O} \to V_a$ for any $a \in \mathrm{AT}$, where V_a is called the *value set* of a.

Each *subset* of attributes $A \subseteq \mathrm{AT}$ determines a binary *indiscernibility relation* IND(A), as follows:

$$\mathrm{IND}(A) = \{(x,y) \in \mathcal{O} \times \mathcal{O} \mid \forall a \in A, a(x) = a(y)\}.$$

The relation $\mathrm{IND}(A), A \subseteq \mathrm{AT}$, constitutes a partition of $\mathcal{O}$, which we will denote by $\mathcal{O}/\mathrm{IND}(A)$.

It may happen that some of attribute values for an object are missing. To indicate such a situation a distinguished value, so-called *null value*, is usually asssigned to those attributes.

If V_a contains null value for at least one attribute $a \in \mathrm{AT}$ then $\mathcal{S}$ is called an *incomplete information system*, otherwise it is *complete*. Further on, we will denote null value by $*$.

Let $\mathrm{SIM}(A), A \subseteq \mathrm{AT}$, denote binary *similarity relation* between objects that are possibly indiscernible in terms of values of attributes A (i.e. we cannot say with certainty that these objects are different). In general, $\mathrm{SIM}(A)$ could be any relation between objects that we want to treat as indiscernible.

Let us define similarity relation more precisely:

$$\mathrm{SIM}(A) = \{(x,y) \in \mathcal{O} \times \mathcal{O} \mid \forall a \in A, a(x) = a(y) \text{ or } a(x) = * \text{ or } a(y) = *\}.$$

Property 2.1. $\mathrm{SIM}(A)$ *is a tolerance relation;*

$$\mathrm{SIM}(A) = \bigcap_{a \in A} \mathrm{SIM}(\{a\}).$$

Several other properties and notions like *dispensability of attributes*, *indispensability of attributes*, *core*, *functional dependencies between attributes* may be introduced in the very similar way as in complete information systems (see [1]).

Let $S_A(x)$ denote the object set $\{y \in \mathcal{O} \mid (x,y) \in \mathrm{SIM}(A)\}$. $S_A(x)$ is the maximal set of objects which are possibly indiscernible by A with x.

Let $D_A(x)$ denote the object set $\{y \in \mathcal{O} \mid (x,y) \notin \mathrm{SIM}(A)\}$. $D_A(x)$ is the maximal set of objects which are definitely discernible by A with x.

Of course, $S_A(x) \cap D_A(x) = \emptyset$ and $S_A(x) \cup D_A(x) = \mathcal{O}$ for any $x \in \mathcal{O}$.

Let $\mathcal{O}/\mathrm{SIM}(A)$ denote classification, which is the family set $\{S_A(x) \mid x \in \mathcal{O}\}$. Any element from $\mathcal{O}/\mathrm{SIM}(A)$ will be called a *tolerance/class*. Tolerance classes in $\mathcal{O}/\mathrm{SIM}(A)$ do not constitute a partition of $\mathcal{O}$ in general. They may be subsets/supersets of each other or may overlap. Of course, $\cup\mathcal{O}/\mathrm{SIM}(A) = \mathcal{O}$.

Example 2.1. Given descriptions of several cars as in Table 1 let us try to classify them according to the chosen subsets of attributes.

From Table 1 we have: $\mathcal{O} = \{1,2,3,4,5,6\}$, $\mathrm{AT} = \{P,M,S,X\}$ where P,M,S,X stand for Price, Mileage, Size, MaX-Speed.

Let us note that $\mathcal{O}/\mathrm{SIM}(\mathrm{AT}) = \{S_{\mathrm{AT}}(1), S_{\mathrm{AT}}(2), S_{\mathrm{AT}}(3), S_{\mathrm{AT}}(4), S_{\mathrm{AT}}(5), S_{\mathrm{AT}}(6)\}$, where $S_{\mathrm{AT}}(1) = \{1\}, S_{\mathrm{AT}}(2) = \{2,6\}, S_{\mathrm{AT}}(3) = \{3\}, S_{\mathrm{AT}}(4) = \{4,5\}, S_{\mathrm{AT}}(5) = \{4,5,6\}, S_{\mathrm{AT}}(6) = \{2,5,6\}$.

It can be also observed easily that $\mathcal{O}/\mathrm{SIM}(\{P,S,X\}) = \mathcal{O}/\mathrm{SIM}(\mathrm{AT})$, whereas $\mathcal{O}/\mathrm{SIM}(\{S,X\}) \neq \mathcal{O}/\mathrm{SIM}(\mathrm{AT})(\mathcal{O}/\mathrm{SIM}(\{S,X\}) = \{S_A(1), S_A(3), S_A(4), S_A(6)\}$, where $A = \{S,X\}$ and $S_A(1) = S_A(2) = \{1,2,6\}, S_A(3) = \{3\}, S_A(4) = S_A(5) = \{4,5,6\}, S_A(6) = \{1,2,4,5,6\})$.

In Example 2.1 car classification by AT is the same as that by $\{P,S,X\}$ and is different from car classification by $\{S,X\}$. Usually, we are interested in minimal subsets of AT, so-called *reducts*, that classify in the same way as AT.

Formally, a set $A \subseteq \mathrm{AT}$ is a *reduct* of IS iff

$$\mathrm{SIM}(A) = \mathrm{SIM}(\mathrm{AT}) \quad \text{and} \quad \forall B \subset A,\ \mathrm{SIM}(B) \neq \mathrm{SIM}(\mathrm{AT}).$$

For the information system from Example 2.1 we can find out that $\{P,S,X\}$ is its reduct. On the other hand, one can easily notice that it suffices to know

Table 1

Car	Price	Mileage	Size	Max-Speed
1	High	High	Full	Low
2	Low	*	Full	Low
3	*	*	Compact	High
4	High	*	Full	High
5	*	*	Full	High
6	Low	High	Full	*

- values (high,low) of Price and Max-Speed of an object to classify it to $S_{\mathrm{AT}}(1)$,
- values (low,low) of Price and Max-Speed of an object to classify it to $S_{\mathrm{AT}}(2)$,
- value (compact) of Size of an object to classify it to $S_{\mathrm{AT}}(3)$,
- values (high,full,high) of Price, Size and Max-Speed of an object to classify it to $S_{\mathrm{AT}}(4)$,
- values (full,high) of Size and Max-Speed of an object to classify it to $S_{\mathrm{AT}}(5)$,
- values (low,full) of Price and Size of an object to classify it to $S_{\mathrm{AT}}(6)$.

This observation encourages us to define a notion of a *reduct for an object* that should allow to classify objects with less number of required attributes then the number of attributes in a reduct of IS.

A set $A \subseteq \mathrm{AT}$ is a *reduct* of IS for $x, x \in \mathcal{O}$, iff

$$S_A(x) = S_{\mathrm{AT}}(x) \quad \text{and} \quad \forall B \subset A, \;\; S_B(x) \neq S_{\mathrm{AT}}(x).$$

3. Set approximations

Let $X \subseteq \mathcal{O}$ and $A \subseteq \mathrm{AT}$. $\underline{A}X$ is *lower approximation* of X, iff

$$\underline{A}X = \{x \in \mathcal{O} \mid S_A(x) \subseteq X\} = \{x \in X \mid S_A(x) \subseteq X\}.$$

$\overline{A}X$ is *upper approximation* of X, iff

$$\overline{A}X = \{x \in \mathcal{O} \mid S_A(x) \cap X \neq \emptyset\} = \cup\{S_A(x) \mid x \in X\}.$$

Like in complete IS, $\underline{A}X$ is a set of objects that belong to X with certainty, while $\overline{A}X$ is a set of objects that possibly belong to X.

Property 3.1.

$$\forall A \subseteq \mathrm{AT}, \forall X \subseteq \mathcal{O}, \quad (\underline{A}X \subseteq X \subseteq \overline{A}X);$$

$$\forall A, B \subseteq \mathrm{AT}, \forall X \subseteq \mathcal{O}, \quad (A \subset B \Rightarrow \underline{A}X \subseteq \underline{B}X);$$

$$\forall A, B \subseteq \mathrm{AT}, \forall X \subseteq \mathcal{O}, \quad (A \subset B \Rightarrow \overline{A}X \supseteq \overline{B}X).$$

4. Decision tables, decision rules, knowledge reduction

(*Incomplete*) *decision table* (DT) is an (incomplete) information system DT $= (\mathcal{O}, \mathrm{AT} \cup \{d\})$, where $d, d \notin \mathrm{AT}$ and $* \notin V_d$, is a distinguished attribute called *decision*, and the elements of AT are called *conditions*.

Let us define function $\partial_A : \mathcal{O} \to \mathcal{P}(V_d), A \subseteq \mathrm{AT}$, as follows:

$$\partial_A(x) = \{i \mid i = d(y) \text{ and } y \in S_A(x)\}.$$

∂_A will be called *generalized decision* in DT.

If $\mathrm{card}(\partial_{\mathrm{AT}}(x)) = 1$ for any $x \in \mathcal{O}$ then DT is *consistent* (*deterministic*, *definite*), otherwise it is *inconsistent* (*non-deterministic*, *nondefinite*).

Property 4.1. *The relation* $\mathrm{IND}(\partial_A), A \subseteq \mathrm{AT}$, *constitutes a partition of* $\mathcal{O}$.

Property 4.2. *The equation*

$$X \in \mathcal{O}/\mathrm{IND}(\partial_A) \ \Rightarrow\ \underline{A}X = X = \overline{A}X$$

does not hold for an incomplete DT (*though it holds for complete* DT).

Any decision table may be regarded as a set of (*generalized*) *decision rules* of the form:

$$\wedge\,(c, v) \rightarrow \vee(d, w), \quad \text{where } c \in \mathrm{AT},\ v \in V_c,\ w \in V_d.$$

In the sequel, we will consider decision rules only in the above form. $\wedge(c, v)$ $(\vee(d, w))$ will be called *condition* (*decision*) *part* of the rule.

Let X be a set of objects of property $\wedge(c, v)$ $(c \in \mathrm{AT}, v \in V_c)$ and let Y be a set of objects of property $\vee(d, w)$ $(w \in V_d)$.

A decision rule $\wedge(c, v) \rightarrow \vee(d, w)$ is true in DT iff $\overline{C}X \subseteq Y$, where C is the set of all attributes which occur in condition part of the rule.

A decision rule $r: \wedge(c, v) \rightarrow \vee(d, w)$ $(c \in \mathrm{AT}, v \in V_c, w \in V_d)$ is *optimal* in DT iff it is true and no other rule constructed from a proper subset of conjuncts and disjuncts occurring in r is true.

Example 4.1. Let us consider decision table DT, constructed from information system presented in Table 1 and extended by decision attribute $d = $ *Acceleration* as shown in Table 2. Determine the family of decision classes $\mathcal{O}/\mathrm{IND}(d)$ and the family of generalized decision classes $\mathcal{O}/\mathrm{IND}(\partial_{\mathrm{AT}})$. For each decision class compute its lower and upper approximations and write down true decision rules.

Solution. From Table 2 we have: $\mathcal{O}/\mathrm{IND}(d) = \{X_{\mathrm{good}}, X_{\mathrm{poor}}, X_{\mathrm{excel.}}\}$, where $X_{\mathrm{good}} = \{1, 2, 4, 6\}, X_{\mathrm{poor}} = \{3\}, X_{\mathrm{excel.}} = \{5\}$.

Table 2

Car	Price	Mileage	Size	Max-Speed	d
1	High	High	Full	Low	Good
2	Low	*	Full	Low	Good
3	*	*	Compact	High	Poor
4	High	*	Full	High	Good
5	*	*	Full	High	Excel.
6	Low	High	Full	*	Good

So,

$$\underline{\mathrm{AT}}X_{\mathrm{good}} = \{1,2\}; \qquad \overline{\mathrm{AT}}X_{\mathrm{good}} = \{1,2,4,5,6\};$$

$$\underline{\mathrm{AT}}X_{\mathrm{poor}} = \{3\}; \qquad \overline{\mathrm{AT}}X_{\mathrm{poor}} = \{3\};$$

$$\underline{\mathrm{AT}}X_{\mathrm{excel.}} = \emptyset; \qquad \overline{\mathrm{AT}}X_{\mathrm{excel.}} = \{4,5,6\}.$$

In Table 3 we place the values of generalized decisions.

$\mathcal{O}/\mathrm{IND}(\partial_{\mathrm{AT}}) = \{X_{\{\mathrm{good}\}}, X_{\{\mathrm{poor}\}}, X_{\{\mathrm{good,excel.}\}}\}$, where $X_{\{\mathrm{good}\}} = \{1,2\}$, $X_{\{\mathrm{poor}\}} = \{3\}, X_{\{\mathrm{good,excel.}\}} = \{4,5,6\}$.

Hence,

$$\begin{aligned}&\underline{\mathrm{AT}}X_{\{\mathrm{good}\}} = \{1\}; \qquad \overline{\mathrm{AT}}X_{\{\mathrm{good}\}} = \{1,2,6\};\\ &\underline{\mathrm{AT}}X_{\{\mathrm{poor}\}} = \{3\}; \qquad \overline{\mathrm{AT}}X_{\{\mathrm{poor}\}} = \{3\};\\ &\underline{\mathrm{AT}}X_{\{\mathrm{good,excel.}\}} = \{4,5\}; \qquad \overline{\mathrm{AT}}X_{\{\mathrm{good,excel.}\}} = \{2,4,5,6\}.\end{aligned}$$

We list true decision rules for DT:

$$\begin{aligned}&r_1 : (P,\mathrm{high}) \wedge (M,\mathrm{high}) \wedge (S,\mathrm{full}) \wedge (X,\mathrm{low}) \to (d,\mathrm{good});\\ &r_2 : (P,\mathrm{low}) \wedge (M,*) \wedge (S,\mathrm{full}) \wedge (X,\mathrm{low}) \to (d,\mathrm{good});\\ &r_3 : (P,*) \wedge (M,*) \wedge (S,\mathrm{compact}) \wedge (X,\mathrm{high}) \to (d,\mathrm{poor})\\ &r_4 : (P,\mathrm{high}) \wedge (M,*) \wedge (S,\mathrm{full}) \wedge (X,\mathrm{high}) \to (d,\mathrm{good}) \vee (d,\mathrm{excel.});\\ &r_5 : (P,*) \wedge (M,*) \wedge (S,\mathrm{full}) \wedge (X,\mathrm{high}) \to (d,\mathrm{good}) \vee (d,\mathrm{excel.});\\ &r_6 : (P,\mathrm{low}) \wedge (M,\mathrm{high}) \wedge (S,\mathrm{full}) \wedge (X,*) \to (d,\mathrm{good}) \vee (d,\mathrm{excel.}).\end{aligned}$$

It follows from the definition of generalized decision and the definitions of true and optimal decision rules that the decision part of an optimal rule for $x, x \in \mathcal{O}$, is equal to $(d,w_1) \vee (d,w_2) \vee \cdots \vee (d,w_n)$, where $\{w_1,w_2,\ldots,w_n\} = \partial_{\mathrm{AT}}(x)$. Thus the problem of finding optimal rules is restricted to the problem of reduction of condition attributes.

Reduction of knowledge that preserves generalized decisions for all objects in DT is lossless from decision making standpoint. Thus, we will want to define a *reduct* A of DT as minimal subset of AT, such that $\partial_A(x)$ of the reduced DT $= (\mathcal{O}, \mathrm{AT} \cup \{d\})$ is equal to $\partial_{\mathrm{AT}}(x)$ for any $x \in \mathcal{O}$.

Formally, a set $A \subseteq \mathrm{AT}$ is a *reduct* of DT (*relative reduct*) iff

$$\partial_A = \partial_{\mathrm{AT}} \quad \text{and} \quad \forall B \subset A,\ \partial_B \neq \partial_{\mathrm{AT}}.$$

Table 3

Car	∂_{AT}
1	{Good}
2	{Good}
3	{Poor}
4	{Good, Excel.}
5	{Good, Excel.}
6	{Good, Excel.}

In order to determine decision rules with minimal number of conditions we may employ the notion of a *reduct* for an object in DT.

A set $A \subseteq \mathrm{AT}$ is a *reduct* of DT for x (*relative reduct* for x), $x \in \mathcal{O}$, in DT, iff

$$\partial_A(x) = \partial_{\mathrm{AT}}(x); \qquad \forall B \subset A, \quad \partial_B(x) \neq \partial_{\mathrm{AT}}(x).$$

Property 4.3. *Let A be a relative reduct. The equations:*

$$X \in \mathcal{O}/\mathrm{IND}(\partial_{\mathrm{AT}}) \;\Rightarrow\; \underline{A}X = \underline{\mathrm{AT}}X;$$

$$X \in \mathcal{O}/\mathrm{IND}(\partial_{\mathrm{AT}}) \;\Rightarrow\; \overline{A}X = \overline{\mathrm{AT}}X.$$

do not hold for incomplete DT.

Example 4.2. Let us illustrate Property 4.3 for a relative reduct of DT described in Table 2.

Solution. We can easily check that $A = \{\text{Size, Max-Speed}\}$ is a reduct for DT from Table 2. Below we present lower and upper aproximations of classes from the family $\mathcal{O}/\mathrm{IND}(\partial_{\mathrm{AT}})$ with regard to attribute set A:

$$\underline{A}X_{\{\text{good}\}} = \emptyset; \qquad \overline{A}X_{\{\text{good}\}} = \{1, 2, 6\};$$
$$\underline{A}X_{\{\text{poor}\}} = \{3\}; \qquad \overline{A}X_{\{\text{poor}\}} = \{3\};$$
$$\underline{A}X_{\{\text{good,excel}\}.} = \{4, 5\}; \qquad \overline{A}X_{\{\text{good,excel}\}.} = \{1, 2, 4, 5, 6\}.$$

where $X_{\{\text{good}\}}, X_{\{\text{poor}\}}$ and $X_{\{\text{good,excel.}\}}$ have the same meaning as in Example 4.1.

Comparing the above set approximations with the set approximations computed in Example 4.1, we can state for instance the following:

$$\underline{A}X_{\{\text{good}\}} \subset \underline{\mathrm{AT}}X_{\{\text{good}\}};$$
$$\overline{A}X_{\{\text{good,excel.}\}} \supset \overline{AT}X_{\{\text{good,excel}\}.}$$

It can be also easily shown that $\{$*Max-Speed*$\}$ is a reduct for objects 1 and 2 from Table 2 and Size is a reduct for objects 3–6. These reducts allow us to obtain the following three optimal decision rules:

$r'_1 : (X, \text{low}) \to (d, \text{good});$
$r'_2 : (S, \text{compact}) \to (d, \text{poor})$
$r'_3 : (S, \text{full}) \to (d, \text{good}) \vee (d, \text{excel.});$
instead of six initial ones.

Let us note that the above decision rules will remain true when all or some missing values in DT will be replaced by arbitrary values.

5. Discernibility function and computing reducts

Computing reducts of incomplete IS and incomplete DT we will exploit the idea of so-called *discernibility functions* [5–9]. Their main properties are that they are monotonic Boolean functions and their prime implicants determine reducts uniquely.

Let $\alpha_A(x,y)$ be a set of attributes $a \in A$ such that $(x,y) \notin SIM(\{a\})$. Hence, if $(x,y) \in \mathrm{SIM}(\{a\})$ then $\alpha_A(x,y) = \emptyset$. Let $\sum \alpha_A(x,y)$ be a Boolean expression which is equal to 1, if $\alpha_A(x,y) = \emptyset$. Otherwise, let $\sum \alpha_A(x,y)$ be a disjunction of variables corresponding to attributes contained in $\alpha_A(x,y)$.

Δ is a *discernibility function* for IS iff

$$\Delta = \prod_{(x,y)\in\mathcal{O}\times\mathcal{O}} \sum \alpha_{\mathrm{AT}}(x,y).$$

$\Delta(x)$ is a *discernibility function* for *object* x in IS iff

$$\Delta(x) = \prod_{y\in\mathcal{O}} \sum \alpha_{\mathrm{AT}}(x,y).$$

Δ^* is a *discernibility function* for DT iff

$$\Delta^* \prod_{(x,y)\in\mathcal{O}\times\{z\in\mathcal{O} \mid d(z)\notin\partial_{\mathrm{AT}}(x)\}} \alpha_{\mathrm{AT}}(x,y).$$

$\Delta^*(x)$ is a *discernibility function* for *object* x in DT iff

$$\Delta^*(x) = \prod_{y\in\{z\in\mathcal{O} \mid d(z)\notin\partial_{\mathrm{AT}}(x)\}} \sum \alpha_{\mathrm{AT}}(x,y).$$

Example 5.1. Determine all reducts for IS presented in Table 1 by computing prime implicants of discernibility functions Δ.

Solution. To construct a discernibility function we will use Table 4, in which values of $\alpha_{\mathrm{AT}}(x,y)$ for any pair (x,y) of objects from $\mathcal{O}$ are placed.

Hence we have,

$\Delta = P(S \vee X)X(P \vee X)S = PSX;$
$\Delta(1) = P(S \vee X)X = PX;$
$\Delta(2) = P(S \vee X)(P \vee X)X = PX;$
$\Delta(3) = (S \vee X)S = S;$
$\Delta(4) = X(P \vee X)SP = PSX;$

Table 4

$x \setminus y$	1	2	3	4	5	6
1		P	SX	X	X	P
2	P		SX	PX	X	
3	SX	SX		S	S	S
4	X	PX	S			P
5	X	X	S			
6	P		S	P		

$\Delta(5) = SX;$
$\Delta(6) = PS.$

Thus, {Price, Size, Max-Speed} is a reduct for IS, {Price, Max-Speed} is a relative reduct for objects 1 and 2 etc.

Example 5.2. Determine all reducts for DT presented in Table 2 by computing prime implicants of discernibility functions Δ^*.

Solution. To construct a discernibility function we build Table 5, in which values of $\alpha_{AT}(x,y)$ for any pair (x,y) of objects, such that $x \in \mathcal{O}$ and $y \in \{z \in \mathcal{O} \mid d(z) \notin \partial_{\mathrm{AT}}(x)\}$ are placed.

Hence we have,

$\Delta^* = (S \vee X)XS = SX;$
$\Delta^*(1) = (S \vee X)X = X;$
$\Delta^*(2) = (S \vee X)X = X;$
$\Delta^*(3) = (S \vee X)S = S;$
$\Delta^*(4) = S;$
$\Delta^*(5) = S;$
$\Delta^*(6) = S.$

Thus, {Size, Max-Speed} is a reduct for DT, {Max-Speed} is a relative reduct for objects 1 and 2 etc.

6. Other approaches to generation rules from incomplete information systems

Our next papers [10,11] examine relationship among different kinds of rules generated by different Rough Set methods directly or indirectly from an incomplete information system. In particular, the method described in this paper that allows to generate generalized rules is compared with the replacing examples' method and the removing examples' method [2]. Let us report shortly the results we obtained and proved in [10,11]. To this end let us remind the definition

Table 5

$x \setminus y$	1	2	3	4	5	6
1			SX		X	
2			SX		X	
3	SX	SX		S	S	S
4			S			
5			S			
6			S			

of a certain rule in a complete system. A rule is certain if it is deterministic and set of objects satisfying the conditional part of the rule is a subset of objects satisfying the decision part of the rule. Now let us introduce informally the definition of a certain rule in an incomplete system. After [12], we regard rule as certain in an incomplete IS if it is certain in every complete extension of the original IS (i.e. in every complete system consistent with the original incomplete IS). In [10,11] we prove that the example's replacing method allows to generate all rules from replaced system which are certain in original incomplete IS. To the contrary, the method of removing examples may cause generation of false certain rules, i.e. rules that are certain in modified destination system used for rule generation, but which are not certain in original IS. Finally, it is shown in [10,11] that the set of all deterministic rules generated as generalized according to the method presented in this paper is a subset of certain rules in original IS. One more interesting property of generalized deterministic rules is that the set of objects supporting them is the same in all extensions of the initial IS. Hence, they seem to constitute very important class of certain rules.

7. Conclusion

In the paper we have shown that Rough Set approach is suitable one to reasoning in incomplete information systems.

The proper definitions of reducts allow to define knowledge reduction that does not diminish the original system's abilities to classify objects or to make decisions. Unlike classical information systems, an incomplete IS allows to achieve much less number of decision rules, which is implied by the character of a tolerance similarity relation.

Both reduction of dispensable knowledge and finding of optimal decision rules are transformable to the problem of computing prime implicants of discernibility functions. We have shown that discernibility functions for incomplete information systems may be constructed in conjunctive normal form. This is a particular feature of incomplete information systems, since in general, the formula defining the discernibility function of a tolerance information system is much more complex [9].

We believe that the type of knowledge reduction offered here will turn out to be useful also in other tolerance information systems, in particular in systems with multivalued attributes.

Acknowledgements

This work was supported by grant: No. 8 T11C 038 08 from State Committee for Scientific Research.

References

[1] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, vol. 9, Kluwer Academic Publishers, Dordrecht, 1991.

[2] M.R. Chmielewski, J.W. Grzymala-Busse, N.W. Peterson, S. Than, The rule induction system LERS – A version for personal computers, Found. Comput. Decision Sci. 18 (3/4) (1993) 181–212.

[3] R. Slowinski, J. Stefanowski, Rough classification in incomplete information systems, Math. Comput. Modelling 12 (10/11) (1989) 1347–1357.

[4] R. Slowinski, J. Stefanowski, Handling various types of uncertainty in the rough set approach, in: W. Ziarko (Ed.), Rough Sets Fuzzy Sets and Knowledge Discovery (RSKD '93), Springer, Berlin, 1994.

[5] M. Kryszkiewicz, Knowledge Reduction in Information Systems, Ph.D. Thesis, Warsaw University of Technology, 1994.

[6] Z. Pawlak, A. Skowron, A rough set approach to decision rules generation, ICS Research Report 23/93, Warsaw University of Technology, 1993.

[7] A. Skowron, Management of uncertainty in AI: A rough set approach, ICS Research Report 46/93, Warsaw University of Technology, December 1993.

[8] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowinski (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory, Kluwer Academic Publisher, Dordrecht, 1992, pp. 331–362.

[9] A. Skowron, J. Stepaniuk, Generalized approximation spaces, in: Proceedings of the Third International Workshop on Rough Sets and Knowledge Discovery RSSC '94, San Jose, USA, 1994, pp. 156–163.

[10] M. Kryszkiewicz, Rules in incomplete information systems, in: Proceedings from the Third Joint Conference on Information Sciences, North Carolina, USA, 2–5 March 1997.

[11] M. Kryszkiewicz, Generation of rules from incomplete information systems, submitted to PKDD'97, Trondheim Norway, 25–27 June 1997.

[12] W.J. Lipski, On semantic issues connected with incomplete information databases, ACM Trans. Databases Systems 4 (1979) 262–296.