

ORIGINAL ARTICLES

AN APPLICATION OF
ROUGH NON-DETERMINISTIC INFORMATION ANALYSIS
TO CLASS EVALUATION DATA BY STUDENTS

Hiroshi SAKAI*, Kazuhiro KOBAYASHI* and Michinori NAKATA**

* *Kyushu Institute of Technology, Tobata, Kitakyushu 804, Japan*

** *Josai International University, Gumyo, Togane, Chiba, 283, Japan*

Abstract: Non-deterministic Information Systems (NISs) have been recognized to be the most important framework for handling information incompleteness in tables, and several theoretical work has been examined. We follow this robust framework, and we have been developing algorithms and tool programs, which can handle the rough sets based concepts in NISs, on computers. We are simply calling this work Rough Non-deterministic Information Analysis (RNIA). This paper briefly surveys RNIA, and applies RNIA to class evaluation data, which consists of evaluation on 16 questions by 60 students. In this data, Question 16 “(Q16) Rate this class in either 1,2,3,4 or 5 grade” is the decision attribute, and data dependency and rules in the form of “Condition=>[Q16,5]” are considered. The most characteristic point is that [Q16,5] strongly depended upon question “(Q3) Was the teacher polite in every class?”. Question (Q3) is related to the emotional impression of a teacher, and the evaluation of a half-year class by every student seems to express every student’s emotional impression of a teacher.

Keywords: *Rough sets, Non-deterministic information, Data analysis, class evaluation data, Knowledge discovery tool*

1. INTRODUCTION

Rough set theory is seen as a mathematical foundation of soft computing. This theory usually handles tables with deterministic information. Many applications of this theory to rule generation, machine learning and knowledge discovery have been presented [1-4]. We follow rule generation in Deterministic Information Systems (DISs), and propose rule generation in NISs. NISs were proposed by Pawlak, Orłowska and Lipski in order to handle information incompleteness in DISs, like null values, unknown values, missing values. From the beginning of the research on incomplete information, NISs have been recognized to be the most important framework for handling information incompleteness [5-8]. Therefore, rule generation in NISs will also be an important framework for rule generation in incomplete information. However, very few work deals with rule generation from incomplete information on computers. Lipski showed a question-answering system besides the axiomatization of logic [6]. Grzymala-Busse developed a system named LERS, which depends upon LEM1 and LEM2 algorithms [7]. Kryszkiewicz proposed a framework of rules in incomplete information systems [8]. As far as authors know, these are the most important work for handling incomplete information, especially missing values, on computers.

In this paper, we briefly survey a framework of Rough Non-deterministic Information Analysis (RNIA), and we apply RNIA to analyzing class evaluation data, which consists of evaluation on 16 questions by 60 students.

2. AN OVERVIEW OF RNIA

This section summarizes a framework of RNIA, especially rule generation in NISs.

2.1 Basic Definitions

A Deterministic Information System (DIS) is a quadruplet $(OB, AT, \{VAL_A | A \in AT\}, f)$ [1,3]. Here, OB , AT and VAL_A are finite sets, and we sequentially call every element an object, an attribute and an attribute value, and f is a mapping $f: OB \times AT \rightarrow \bigcup_{A \in AT} VAL_A$. Let us consider two sets $CON \subseteq AT$ which we call condition attributes and $DEC \subseteq AT$ which we call decision attributes. An object $x \in OB$ is consistent, if $f(x,A) = f(y,A)$ for every $y \in OB$ and every $A \in CON$ implies $f(x,A) = f(y,A)$ for every $A \in DEC$. We call a pair $[A,e]$ for an attribute $A \in AT$ and an attribute value $e \in VAL_A$ a descriptor.

A Non-deterministic Information System (NIS) is also a quadruplet $(OB, AT, \{VAL_A | A \in AT\}, g)$, where g is a mapping $g: OB \times AT \rightarrow P(\bigcup_{A \in AT} VAL_A)$ (a power set of $\bigcup_{A \in AT} VAL_A$) [5]. Every set $g(x,A)$ is interpreted there is an actual value in this set but this value is not known.

For a $NIS = (OB, AT, \{VAL_A | A \in AT\}, g)$ and a set of attributes $ATR \subseteq AT$, we name a $DIS = (OB, ATR, \{VAL_A | A \in ATR\}, h)$, which satisfies $h(x,A) \in g(x,A)$ for every x and A , a derived DIS from NIS. We suppose that there exists a derived DIS^{real} with real information. In order to handle information based on DIS^{real} , two modalities the certainty and the possibility are usually defined.

(Certainty) If a formula F holds in every derived DIS

from a NIS, F also holds in DIS^{real} . In this case, we say F certainly holds (in DIS^{real}).

(Possibility) If a formula F holds in some derived DISs from a NIS, there exists such a possibility that F holds in DIS^{real} . In this case, we say F possibly holds (in DIS^{real}).

In order to handle these two modalities, we extended concepts in DISs to NISs, for example, concepts like the definability of a set, the consistency of an object, dependencies among attributes, rules and criteria including support, accuracy and coverage, reduction of attributes and so on. It is possible to handle most of these concepts on computer [9-12].

2.2 Equivalence Relations and Data Dependency

In rough set theory, the manipulation of equivalence relations is very important. Because, most of rough sets based concepts can be calculated by using equivalence relations. As an example, let us consider data dependency between attributes. The degree $dep(CON,DEC)$ of dependency from CON to DEC is a ratio such that (the number of consistent objects from CON to DEC)/(the number of all objects). This concept is extended to minimum $dep(CON,DEC)$ and maximum $dep(CON,DEC)$. In Appendix 1, we show the real execution based on equivalence relations in Table 1. There are 768 derived DISs for all attributes, and we see there is a derived DIS^{real} in 768 derived DISs.

2.3 Background of Rule Generation in NISs

For a set $ATR=\{A_1, \dots, A_n\} \subseteq AT$ and every $x \in OB$, let $PT(x,ATR)$ denote the Cartesian product $g(x,A_1) \times \dots \times g(x,A_n)$. We name every element a possible tuple of x. For every $ptup=(e_1, \dots, e_n) \in PT(x,ATR)$, let $[ATR,ptup]$ denote a formula $\bigwedge_{1 \leq i \leq n} [A_i, e_i]$, and let $PI(x,CON,DEC)$ denote a set $\{[CON,ptup] \Rightarrow [DEC,ptup'] \mid ptup \in PT(x,CON), ptup' \in PT(x,DEC)\}$. We name an element of $PI(x,CON,DEC)$ a possible implication (from CON to DEC) of x.

Now, we define six classes of possible implications, certain rules and possible rules. For any $pimp \in PI(x,CON,DEC)$, let $DD(pimp,x,CON,DEC)$ denote a set $\{\psi \mid \psi \text{ is such a derived DIS for } CON \cup DEC \text{ that an implication from } x \text{ in } \psi \text{ is equal to } pimp\}$.

Table 1 : A non-deterministic information system NIS_1 . Here, ‘{’ and ‘}’ are omitted for every singleton set.

OB	Color	Weight	Size	Shape	Price
1	green	{m,h}	1	triangle	expensive
2	blue	{l,m}	{m,l}	{round,s}	mean
3	{r,b}	light	{s,m}	square	cheap
4	red	mean	1	{t,s}	{c,m,e}
5	green	heavy	{s,m}	square	expensive

If $PI(x,CON,DEC)$ is a singleton set $\{pimp\}$, we say $pimp$ (from x) is definite. Otherwise we say $pimp$ (from x) is indefinite. If a set $\{\psi \in DD(pimp,x,CON,DEC) \mid x \text{ is consistent in } \psi\}$ is equal to $DD(pimp,x,CON,DEC)$, we say $pimp$ is globally consistent (GC). If this set is an empty set, we say $pimp$ is globally inconsistent (GI). Otherwise, we say $pimp$ is marginal (MA). By combining two cases, ‘D(efinite) or I(ndefinite)’ and ‘GC, MA or GI’, we define six classes DGC, DMA, DGI, IGC, IMA, IGI in Table 2, for possible implications.

A possible implication $pimp$ belonging to DGC class is consistent in all derived DISs, and this $pimp$ is not influenced by the information incompleteness, therefore we name $pimp$ (a candidate of) a certain rule. A possible implication $pimp'$ (from object x) belonging to either IGC, DMA or IMA class is consistent in some $\psi \in DD(pimp',x,CON,DEC)$. Therefore, we name $pimp'$ (a candidate of) a possible rule.

Table 2 : Six classes of possible implications in NISs.

	GC	MA	GI
Definite	DGC	DMA	DGI
Indefinite	IGC	IMA	IGI

Now, we give necessary and sufficient conditions for characterizing GC and MA classes. For every $ptup \in PT(x,ATR)$, we define two sets,

$$\begin{aligned} \inf(x,ATR,ptup) &= \{y \in OB \mid PT(y,ATR) = \{ptup\}\} \cup \{x\}, \\ \sup(x,ATR,ptup) &= \{y \in OB \mid ptup \in PT(y,ATR)\}. \end{aligned}$$

In the definition of $\inf(x,ATR,ptup)$, we implicitly fix the possible implication of x to $pimp$. Intuitively, $\inf(x,ATR,ptup)$ implies a set of objects whose tuples are $ptup$ and definite. A set $\sup(x,ATR,ptup)$ implies a set of objects whose tuples may be $ptup$. In every DIS, $[x]_{ATR}$ (an equivalence class for ATR with object x) is equal to both $\inf(x,ATR,ptup)$ and $\sup(x,ATR,ptup)$, however in NISs, $\{x\} \subseteq \inf(x,ATR,pimp) \subseteq [x]_{ATR} \subseteq \sup(x,ATR,pimp)$ holds. In Table 1, $\inf(1,\{W\},(m))=\{1,4\}$ and $\sup(1,\{W\},(m))=\{1,2,4\}$ hold, therefore $[1]_{\{W\}}$ is either $\{1,4\}$ or $\{1,2,4\}$. This is due to the information incompleteness in NIS_1 .

Theorem 1 [9]. For a NIS, let us consider a possible implication $pimp:[CON,ptup] \Rightarrow [DEC,ptup'] \in PI(x,CON,DEC)$. Then, the following holds.

- (1) Possible implication $pimp$ belongs to GC class if and only if $\sup(x,CON,ptup) \subseteq \inf(x,DEC,ptup')$.
- (2) Possible implication $pimp$ belongs to MA class if and only if $\inf(x,CON,ptup) \subseteq \sup(x,DEC,ptup')$.

In order to specify object x, which $pimp$ is extracted from, we employ $pimp^x, pimp^{x_1}, \dots, pimp^{x_n}$ instead of $pimp$, from now on.

Proposition 2 [9]. For a NIS, let $ATR \subseteq AT$ be

$\{A_1, \dots, A_n\}$, and let $\text{ptup} \in \text{PT}(x, \text{ATR})$ be (e_1, \dots, e_n) . Then, the following holds.

$$(1) \inf(x, \text{ATR}, \text{ptup}) = \bigcap_i \inf(x, \{A_i\}, (e_i)).$$

$$(2) \sup(x, \text{ATR}, \text{ptup}) = \bigcap_i \sup(x, \{A_i\}, (e_i)).$$

Example 1. Let us consider NIS_1 in Table 1. There are 768 derived DISs for all attributes. For $\text{CON}=\{C, W\}$ and $\text{DEC}=\{P\}$, there are $24(=2^3 \times 3)$ derived DISs. Here, $\text{PT}(1, \{C, W\})=\{(g, m), (g, h)\}$, $\text{PT}(1, \{P\})=\{(e)\}$ and $\text{PI}(1, \{C, W\}, \{P\})$ consists of two possible implications $\text{pimp}^1_1: [C, g] \wedge [W, m] \Rightarrow [P, e]$ and $[C, g] \wedge [W, h] \Rightarrow [P, e]$. According to Proposition 2, $\sup(1, \{C, W\}, (g, m))=\sup(1, \{C\}, (g)) \cap \sup(1, \{W\}, (m))=\{1, 5\} \cap \{1, 2, 4\}=\{1\}$ holds. Since $\inf(1, \{P\}, (e))=\{1, 5\}$, $\sup(1, \{C, W\}, (g, m)) \subseteq \inf(1, \{P\}, (e))$ holds, we know pimp^1_1 belongs to IGC class according to (1) in Theorem 1. In pimp^1_1 , let us remove the descriptor $[W, m]$, then we obtain $\text{pimp}^1_2: [C, g] \Rightarrow [P, e]$. This pimp^1_2 also satisfies (1) in Theorem 1, and pimp^1_2 is definite, namely pimp^1_2 is a certain rule.

Let us consider $\text{pimp}^5_1: [C, g] \wedge [W, h] \Rightarrow [P, e]$ in NIS_1 . In this case, pimp^5_1 is definite and (1) in Theorem 1 holds. Thus, we know pimp^5_1 belongs to DGC class. However, we can reduce the descriptor $[W, h]$ from the condition part. Namely, $\text{pimp}^5_2: [C, g] \Rightarrow [P, e]$ is also belongs to DGC class. The condition part of pimp^5_2 is much simpler than that of pimp^5_1 , and we call such pimp^5_2 minimal rule. In order to obtain minimal possible implications, it is necessary to examine every combination of attributes. The number of all combinations is $2^{|\text{ATR}|}$ (ATR is a set of attributes, which can be conditions in possible implications). Therefore, it is necessary to employ another method for handling tables with large size of ATR, and we introduce discernibility functions [13] into NISs.

Example 2. Let us consider to pick up minimal certain rules, namely, which satisfies (1) in Theorem 1, definite and minimal. In object 5, $\sup(5, \{C, W, Sh\}, (g, h, s))=\{5\} \subseteq \inf(5, \{P\}, (e))$ holds, therefore it is possible to obtain minimal certain rules from object 5. Let $\text{DISC}(x, y)$ denote a set $\{[A, e] \mid y \text{ is not any element in } \sup(x, \{A\}, (e))\}$. Here, the discernibility function $\text{DF}(5)$ is $\text{DISC}(5, 2) \wedge \text{DISC}(5, 3) \wedge \text{DISC}(5, 4) = ([C, g] \vee [W, h]) \wedge ([C, g] \vee [W, h]) \wedge ([C, g] \vee [W, h]) = [C, g] \vee [W, h]$. Therefore, the minimal solutions of $\text{DF}(5)$ are $[C, g]$ and $[W, h]$, and we obtain minimal certain rules $\text{pimp}^5_2 (= \text{pimp}^1_2): [C, g] \Rightarrow [P, e]$ and $[W, h] \Rightarrow [P, e]$ from object 5. We show the real execution in Appendix 2.

2.4 Criteria Support, Accuracy and Coverage

For an implication imp in a DIS, $\text{criteria support}(\text{imp})$, $\text{accuracy}(\text{imp})$ and $\text{coverage}(\text{imp})$ may be applied to defining rules. In NISs, they are extended to minimum and maximum cases for every possible implication pimp , like minimum support $\text{minsupp}(\text{pimp})$ and maximum support $\text{maxsupp}(\text{pimp})$. These criteria depend upon all derived

DISs, however it is possible to calculate them by using $\inf()$ and $\sup()$ information.

Theorem 3 [9]. For a NIS, let us consider $\text{pimp}: [\text{CON}, \text{ptup}] \Rightarrow [\text{DEC}, \text{ptup}'] \in \text{PI}(x, \text{CON}, \text{DEC})$. Let INACC denote a set $[\sup(x, \text{CON}, \text{ptup}) - \inf(x, \text{CON}, \text{ptup})] \cap \sup(x, \text{DEC}, \text{ptup}')$, and let OUTACC denote a set $[\sup(x, \text{CON}, \text{ptup}) - \inf(x, \text{CON}, \text{ptup})] - \inf(x, \text{DEC}, \text{ptup}')$. Then, the following holds.

$$(1) \text{minsupp}(\text{pimp}) =$$

$$\text{linf}(x, \text{CON}, \text{ptup}) \cap \text{linf}(x, \text{DEC}, \text{ptup}') / |\text{OBl}|.$$

$$(2) \text{maxsupp}(\text{pimp}) =$$

$$\text{lsup}(x, \text{CON}, \text{ptup}) \cap \text{lsup}(x, \text{DEC}, \text{ptup}') / |\text{OBl}|.$$

$$(3) \text{minacc}(\text{pimp}) =$$

$$(\text{linf}(x, \text{CON}, \text{ptup}) \cap \text{linf}(x, \text{DEC}, \text{ptup}')) / (\text{linf}(x, \text{CON}, \text{ptup}) + |\text{OUTACC}|).$$

$$(4) \text{maxacc}(\text{pimp}) =$$

$$(\text{linf}(x, \text{CON}, \text{ptup}) \cap \text{lsup}(x, \text{DEC}, \text{ptup}')) / (\text{linf}(x, \text{CON}, \text{ptup}) + |\text{INACC}|).$$

As for $\text{mincov}()$ and $\text{maxcov}()$, it is possible to conclude the same formulas as (3) and (4) in Theorem 3.

Example 3. Now, let us consider criterion values of $\text{pimp}^4_1: [W, m] \Rightarrow [P, e]$. $\text{minsupp}(\text{pimp}^4_1) = (\text{linf}(4, \{W\}, (m)) \cap \text{linf}(4, \{P\}, (e))) / 5 = (\{4\} \cap \{1, 4, 5\}) / 5 = 0.2$, and $\text{maxsupp}(\text{pimp}^4_1) = (\text{lsup}(4, \{W\}, (m)) \cap \text{lsup}(4, \{P\}, (e))) / 5 = (\{1, 2, 4\} \cap \{1, 4, 5\}) / 5 = 0.4$ holds. Now, let us consider $\text{accuracy}()$. $\text{OUTACC} = [\{1, 2, 4\} - \{4\}] - \{1, 4, 5\} = \{2\}$ holds, so it is possible to extract a possible implication, whose condition part is $[W, m]$ and the decision part is different from $[P, e]$, from object 2. Such implication is counted in the denominator, and is not counted in the numerator. In this way, $\text{minacc}(\text{pimp}^4_1) = (\{4\}) / (\{4\} + \{2\}) = 0.5$. Since $\text{INACC} = [\{1, 2, 4\} - \{4\}] \cap \{1, 4, 5\} = \{1\}$ holds, it is possible to extract pimp^4_1 from object 1, and $\text{maxacc}(\text{pimp}^4_1) = (\{4\} + \{1\}) / (\{4\} + \{1\}) = 1.0$.

3. CLASS EVALUATION DATA BY STUDENTS

In order to know students' opinions for a half-year class, we asked 60 students to answer 16 questions at the last class. We see every question is an attribute, and the attribute values to every question are 1, 2, 3, 4, 5. The value 5 is the most positive, and the value 1 is the most negative. The value 3 is mean. Let us enumerate some questions.

- (Q1) Did the teacher have enough classes for students?
- (Q3) Was the teacher polite in every class?
- (Q4) Was the notation on the blackboard good?
- (Q5) Could you hear the teacher's explanation well?
- (Q6) Was the short survey at the beginning of every class good?
- (Q11) Suppose you had enough credits for your promotion. Then, did you have this class?
- (Q15) Was this class meaningful for you?

(Q16) Rate this class in either 1,2,3,4 or 5.

In 16 questions, we intentionally put some uncomfortable questions for students, which some students may not answer to, like Q11. However, for such uncomfortable questions, we agreed to use not only an attribute value but also a set of attribute values. Namely, students may answer {3,4,5} instead of 3. If the attribute value is either 1,2,3,4 or 5, we say this attribute value is deterministic to this attribute. If the attribute value is a set of values, we say this attribute value is non-deterministic to this attribute. In this way, we obtained a table, which consists of 60 objects and 16 attributes. In reality, 15 students answered non-deterministic values to some questions. We name this data class evaluation data by students.

4. AN APPLICATION OF RNIA TO CLASS EVALUATION DATA

As for teachers, the most important attribute is Q16, and the implications like $[CON,ptup] \Rightarrow [Q16,5]$ and $[CON,ptup] \Rightarrow [Q16,1]$ give us some remarks in the next half-year class. According to this consideration, we cope with the issue in the following.

Issue. Let attribute Q16 be the decision attribute, and examine which attributes influence attribute Q16, and obtain implications which reflect the character of data in the following two cases.

(Case 1) We handle 45 students' data Table₄₅, which consists of only deterministic attribute values. We may see Table₄₅ is a deterministic information system.

(Case 2) We handle 60 students' data Table₆₀, which consists of both deterministic and non-deterministic attribute values. We may see Table₆₀ is a non-deterministic information system.

5. ANALYSIS ON TABLE₄₅

This section applies tool programs for RNIA to Table₄₅ with deterministic information.

5.1 An Application of Regression Analysis to Table₄₅

It is possible to apply multivariate analysis to Table₄₅, and we easily obtained a regression line by using Microsoft Excel.

$$Q16 = -0.14 + 0.26Q3 + 0.14Q4 + 0.30Q6 + 0.13Q8 + 0.11Q9 + 0.19Q11 - 0.11Q12 - 0.19Q13 + 0.25Q15.$$

However, it seems difficult to know the character and the tendency of data from this regression line.

5.2 Data Dependency between Attributes in Table₄₅

In this subsection, let us consider data dependency between attributes. We have applied tool programs to Table₄₅, and examined 256 (=16×16) combinations of

attributes for 16 attributes. In most of combinations, the degrees of the dependency were 0. This seems reasonable. Because, every student selects either 1,2,3,4 or 5, and 45 students are divided into 5 groups according to the selection. Every group may consist of more than 7 or 8 students. For keeping the consistency, such 7 or 8 students must select the same value in the decision attribute. This seems rare, therefore the degree of the dependency will be 0 in every combination.

In this situation, we revised data dependency to a ratio such that $dep'(CON,DEC) = (|\{x | \text{accuracy ratio of object } x \text{ from CON to DEC is more than } val_1\}|) / (\text{the number of all objects})$ for a threshold value val_1 ($0 < val_1 \leq 1.0$). If $val_1 = 1.0$, this equals to the original definition. By means of controlling val_1 , we may find data dependencies. In reality, we employed $val_1 = 0.7$, and obtained Table 3 in Appendix 3. In Table 3, $dep(Q3, Q16) = 0.04$ for $val_1 = 1.0$, however $dep'(Q3, Q16) = 0.36$ for $val_1 = 0.7$. In this way, we may find weak data dependencies.

5.3 Minimal Rules in Table₄₅ with Consistency

Now, let us consider minimal consistent implications $[CON, tup] \Rightarrow [Q16, 5]$. We can apply tool programs for RNIA to both tables with deterministic information and tables with non-deterministic information. In programs, Theorem 1 and Proposition 2 are employed, and a discernibility function is generated. A minimal solution, which is a set of descriptors, of this function becomes the condition part of a minimal rule with consistency. In order to assure the minimality of a solution, every subset of this solution is examined. The most characteristic implications are the following. We remark that we employ a term 'rule' for simplicity, however a term 'candidate of a rule' is more correct in every case.

(Rule 1) $[Q3,5] \wedge [Q5,5] \wedge [Q6,5] \Rightarrow [Q16,5]$, [0.11,1.0]

(Rule 2) $[Q3,5] \wedge [Q4,5] \wedge [Q6,5] \Rightarrow [Q16,5]$, [0.11,1.0]

The list [0.11,1.0] implies that the support value is 0.11 and the accuracy value is 1.0. Because, we handled consistent implications, the accuracy must be 1.0. These two implications are minimal, namely the condition part is simplest for concluding $[Q16,5]$.

5.4 Rules in Table₄₅ without Consistency and Apriori Algorithm

In the previous section, we obtained two characteristic rules, however the support value 0.11 seems to be low for the accuracy value 1.0. In this subsection, we consider implications whose decision attribute is Q16, and these implications may not be consistent. We characterize rules by means of support and accuracy values. Namely, we handle the problem in the following.

Problem: Pick up every implication imp such that accu-

racy is maximum under the condition $\text{support}(\text{imp}) \geq \text{val}_2$ for a threshold value val_2 ($0 < \text{val}_2 \leq 1.0$).

For handling such implications, the simple method is to examine support and accuracy values in every implication sequentially. However in Table₄₅, the number of condition attributes is 15, therefore there are $2^{15}-1$ implications for an object. In reality, we executed this simple method on a computer. However, this execution was so time-consuming that we ceased to do it.

In this situation, we employed Apriori algorithm [14]. The total number of descriptors is much smaller than $2^{15}-1$. Since the attribute value is either 1,2,3,4 or 5 for every attribute, there are $80(=16 \times 5)$ distinct descriptors, like $[Q1,1], \dots, [Q2,1], \dots, [Q16,5]$. We first fix a value val_2 , and pick up candidates of descriptors. Since $\text{support}(\text{imp}) = (\text{the number of imp}) / (\text{the number of total objects})$, if support value is more than 0.5, the number of $\text{imp} (=|[x]_{\text{CON}} \cap [x]_{\text{DEC}}| \text{ for every object } x)$ must be more than the half number of total objects. Then, we generate conjunctions of candidates of descriptors. Since $\text{accuracy}(\text{imp}) = (\text{the number of imp}) / (\text{the number of condition part in imp})$, accuracy value increases for every generated conjunction. However, support value decreases for every generated conjunction. In this way, we have realized tool programs in C. The following is the characteristic implications for $\text{val}_2=0.2$.

(Rule 3) $[Q3,4] \Rightarrow [Q16,4]$, [0.33,0.71]

(Rule 4) $[Q4,4] \Rightarrow [Q16,4]$, [0.24,0.64]

(Rule 5) $[Q5,5] \Rightarrow [Q16,4]$, [0.22,0.47]

(Rule 6) $[Q15,4] \Rightarrow [Q16,4]$, [0.28,0.72]

(Rule 7) $[Q3,4] \wedge [Q15,4] \Rightarrow [Q16,4]$, [0.20,0.81]

5.5 Rules with Intervals in Table₄₅

In the previous subsection, we generated conjunctions of descriptors. As we have shown, support value decreases for every generated conjunction. In reality, the maximum of support values is 0.33 in Table₄₅.

In order to obtain implications with higher support values, we handle intervals for attribute values. Namely, we handle descriptors like $[Q1,3\#5] (= [Q1,3] \cup [Q1,4] \cup [Q1,5])$, which we name interval descriptor. We first fix the threshold value val_2 , then we pick up candidates of descriptors. For higher threshold value, fewer descriptors may become candidates. In this case, we employ interval descriptors. For example, if $[Q1,1]$ can not be a candidate, we sequentially examine $[Q1,1\#2]$, $[Q1,1\#3]$, $[Q1,1\#4]$ and $[Q1,1\#5]$. In this way, we obtain candidate of descriptors. For these descriptors, we generate conjunctions of descriptors, and apply Apriori based algorithm. We have also realized such tool programs in C. The following is the characteristic implications for $\text{val}_2=0.5$.

(Rule 8) $[Q1,5] \Rightarrow [Q16,4\#5]$, [0.51,0.74]

(Rule 9) $[Q3,4\#5] \Rightarrow [Q16,4\#5]$, [0.55,0.86]

(Rule 10) $[Q5,4\#5] \Rightarrow [Q16,4\#5]$, [0.55,0.73]

(Rule 11) $[Q6,4\#5] \Rightarrow [Q16,4\#5]$, [0.60,0.71]

(Rule 12) $[Q3,4\#5] \wedge [Q6,4\#5] \Rightarrow [Q16,4\#5]$, [0.53,0.88]

(Rule 13) $[Q5,4\#5] \wedge [Q6,4\#5] \Rightarrow [Q16,4\#5]$, [0.53,0.77]

6. ANALYSIS ON TABLE₆₀

In this section, we cope with data analysis on Table₆₀ with non-deterministic information.

6.1 Minimal Certain Rules in Table₆₀

Now, let us consider minimal certain rules, which belong to DGC class and the condition part is minimal, like $\text{pimp}_2^5: [C,g] \Rightarrow [P,e]$ in Table 1. The algorithm for rule generation is the same as the algorithm in section 5.3. The following is the characteristic minimal certain rules obtained.

(Rule 14) $[Q2,5] \wedge [Q3,5] \Rightarrow [Q16,5]$,

[4608/4608:DGC], [14,33,34,37,46,49],

[(0.1,0.1),(1.0,1.0),(0.42,0.54)]

(Rule 15) $[Q2,5] \wedge [Q4,5] \Rightarrow [Q16,5]$,

[6144/6144:DGC], [14,33,34,37,46,49],

[(0.1,0.1),(1.0,1.0),(0.42,0.54)]

(Rule 16) $[Q3,5] \wedge [Q10,1] \Rightarrow [Q16,5]$,

[331776/331776:DGC], [14,33,40,50],

[(0.06,0.06),(1.0,1.0),(0.28,0.36)]

The result in Rule 14 implies that there exist 4608 derived DISs for a set $\{Q2, Q3, Q16\}$ of attributes, and this is consistent in every derived DISs, furthermore this is minimal. Namely, Rule 14 is a minimal certain rule. The list [14,33,34,37,46,49] implies that Rule 14 occurs in these 6 objects. The list [(0.1,0.1),(1.0,1.0),(0.42, 0.54)] implies (minsupp, maxsupp), (minacc, maxacc), (minimum coverage, maximum coverage) of this implication, respectively.

6.2 Possible Rules in Table₆₀

Possible rules are implications, which belong to either IGC, DMA or IMA class, like $\text{pimp}_1^1: [C,g] \wedge [W,m] \Rightarrow [P,e]$ in Table 1. For handling possible rules in IGC class, it is necessary to consider $|\prod_{A \in \text{AT-DECg}(x,A)} \text{kind of discernibility functions}|$ kinds of discernibility functions. Furthermore in either DMA or IMA class, there may be huge number of $[CON, \text{ptup}]$ satisfying $\text{inf}(x, CON, \text{ptup}) \subseteq \text{sup}(x, DEC, \text{ptup})$, which is (2) in Theorem 1. Because of these reasons, we are employing an order of attributes. We employed an order $Q2, Q3, Q4, Q5, Q6, Q11, Q12$ for attributes. The following is the characteristic rules obtained.

(Rule 17) $[Q3,5] \wedge [Q4,5] \wedge [Q6,5] \Rightarrow [Q16,5]$,

[12288/12288:DMA], [14,34,37,40,43,46,49,50],

[(0.11,0.13),(0.87,1.0),(0.5,0.66)]

(Rule 18) $[Q4,5] \wedge [Q10,3] \Rightarrow [Q16,5]$,

[73728/442368:IGC], [34,37,43,46],
 [(0.05,0.06),(1.0,1.0),(0.21,0.33)]
 (Rule 19) [Q4,5]^[Q11,2]⇒[P,5],
 [294912/1179648:IGC], [34,37,43,49],
 [(0.03,0.06),(1.0,1.0),(0.14,0.33)]
 (Rule 20) [Q3,5]^[Q10,3]⇒[Q16,5],
 [18432/331776:IMA], [34,37,43,46,53],
 [(0.05,0.08),(0.75,1.0),(0.21,0.38)]

7. CONCLUDING REMARKS

This paper surveyed a framework of RNIA, and applied tool programs to analyzing a class evaluation data by students. Realized tool programs can easily handle tables like Table₄₅ and Table₆₀. In this paper, we mainly showed how we obtain rules from tables, and we omitted the evaluation and the interpretation of every obtained rule. Intuitively, question Q3 (Was the teacher polite in every class?) seems the most important factor for the decision part [Q16,4] or [Q16,5]. In the next step, it is necessary to cope with the combinations of multivariate analysis and rough sets based analysis as well as the interpretation of obtained rules.

ACKNOWLEDGMENT

The authors would be grateful to Prof. H.Shiizuka and Prof. K.Inoue.

This work is partly supported by the Grant-in-Aid for Scientific Research (C) (No.16500176, No.18500214), Japan Society for the Promotion of Science.

REFERENCES

1. Z.Pawlak; Rough Sets; Kluwer Academic Publisher (1991).
2. Z.Pawlak; Some Issues on Rough Sets, Transactions on Rough Sets; Int'l. Rough Set Society, 1, 1-58 (2004).
3. J.Komorowski, Z.Pawlak, L.Polkowski and A.Skowron; Rough Sets: a tutorial. Rough Fuzzy Hybridization; Springer, 3-98 (1999).
4. Rough Set Software; Bulletin of Int'l. Rough Set Society; 2, 15-46 (1998).
5. E.Orlowska and Z.Pawlak; Representation of Nondeterministic Information, Theoretical Computer Science; 29, 27-39 (1984).
6. W.Lipski; On Semantic Issues Connected with Incomplete Information Data Base, ACM Trans. DBS; 4, 269-296 (1979).
7. J.Grzymala-Busse and P.Werbrouck; On the Best Search Method in the LEM1 and LEM2 Algorithms.

In; E.Orlowska (Ed.) Incomplete Information: Rough Set Analysis; Physica-Verlag, 75-91 (1998).

8. M.Kryszkiewicz; Rules in Incomplete Information Systems, Information Sciences; 113, 271-292 (1999).
9. H.Sakai and A.Okuma; Basic Algorithms and Tools for Rough Non-deterministic Information Analysis, Transactions on Rough Sets; Int'l. Rough Set Society, 1, 209-231 (2004).
10. H.Sakai; Effective Procedures for Handling Possible Equivalence Relations in Non-deterministic Information Systems, Fundamenta Informaticae; 48, 343-362 (2001).
11. H.Sakai; Effective Procedures for Data Dependencies in Information Systems. Rough Set Theory and Granular Computing, Studies in Fuzziness and Soft Computing; Springer-Verlag, 125, 167-176 (2003).
12. H.Sakai and M.Nakata; Discernibility Functions and Minimal Rules in Non-deterministic Information Systems. Lecture Notes in AI; Springer-Verlag, 3641, 254-264 (2005).
13. A.Skowron and C.Rauszer; The Discernibility Matrices and Functions in Information Systems. In Intelligent Decision Support-Handbook of Advances and Applications of the Rough Set Theory; Kluwer Academic Publishers, 331-362 (1992).
14. R.Agrawal and R.Srikant; Fast Algorithms for Mining Association Rules. Proc. 20th Very Large Data Base; 487-499 (1994).

Appendix 1.

Let us show a real execution of manipulating possible equivalence relations and data dependency in Table 1. The ordinal number is employed for displaying every attribute in the following execution, namely Q1 and Q3 are displayed as 1 and 3. The underlined parts show the user's input. These programs are realized on a workstation with 450MHz UltraSparc in prolog and C.

```
% more data.pl [Operation 1]
object(5,5).
data(1,[g,[m,h],l,t,e]).
data(2,[b,[l,m],[m,l],[r,s],m]).
data(3,[[r,b],l,[s,m],s,c]).
data(4,[r,m,l,[t,s],[c,m,e]]).
data(5,[g,h,[s,m],s,e]).
?-transall. [Operation 2]
File Name for Read Open: data.pl.
EXEC_TIME=0.093(sec)
yes
?-eqall. [Operation 3]
Attribute 1
[1] [[1,5],[2,3],[4]] 1
[2] [[1,5],[2],[3,4]] 1
```

```

POSSIBLE CASES 2
Attribute 2
[1] [[1,5],[2,3],[4]] 1
[2] [[1,4],[2,3],[5]] 1
[3] [[1,5],[2,4],[3]] 1
[4] [[1,2,4],[3],[5]] 1
POSSIBLE CASES 4
Attribute 3
[1] [[1,2,4],[3,5]] 2
[2] [[1,2,4],[3],[5]] 2
[3] [[1,4],[2,3,5]] 1
: : : : :
Attribute 5
[1] [[1,4,5],[2],[3]] 1
[2] [[1,5],[2,4],[3]] 1
[3] [[1,5],[2],[3,4]] 1
POSSIBLE CASES 3
EXEC_TIME=0.148(sec)
yes
    In Operation 1, the original data is displayed. Realized
    programs can handle any data in the form of this syntax.
    We first translate original data into internal expressions in
    Operation 2, and we execute program 'eqall' in Operation
    3. We obtain all possible equivalence relations for each
    attribute. We can generate possible equivalence relations
    for any set of attributes by using program 'merge'.
% ./merge [Operation 4]
Merging 1.pe ...
Merging 2.pe ...
Merging 3.pe ...
EXEC_TIME=0.010(sec)
% ./depend [Operation 5]
File Name for Condition: 123.pe
File Name for Decision: 5.pe
----- Dependency Check -----
CRITERION 1(Num of Consistent DISs/Num of All
DISs)
Number of Derived DISs: 192
Number of Derived Consistent DISs: 180
Degree of Consistent DISs: 0.938
CRITERION 2(Min and Max Degrees)
Minimum Degree of Dependency: 0.600
Maximum Degree of Dependency: 1.000
EXEC_TIME=0.040(sec)

```

We define the details for merging in 'merge.dat', and execute program 'merge' in Operation 4. Here, possible equivalence relations for attributes {C,W,Si} are generated. In reality, there are two distinct possible equivalence relations, {{1},{2},{3},{4},{5}} for 60 derived DISs and {{1},{2,3},{4},{5}} for 4 derived DISs. For a set of attributes {C,W,Si,P}, there exist 192 derived DISs, so we may sequentially examine the degree of dependency in

192 derived DISs. However in Operation 5, the 6 combinations of two possible equivalence relations for {C,W,Si} and three possible equivalence relations for {P} are examined, and the results are obtained.

Appendix 2.

Here, we generate minimal certain rules in the form of [CON,ptup]⇒[P,e] in Table 1. Before executing Operation 6, we prepare attribute definition file, which stores decision attributes, decision attributes values and candidates of condition attributes. Here, [P,e] and condition attributes C,W,Si,Sh are specified in 'attrib.pl'. Operation 6 generates internal expressions.

```

?-transattrib. [Operation 6]
File Name for Read Open: data.pl.
Decision Definition File: attrib.pl.
File Name for Write Open: data.rs.
EXEC_TIME=0.025(sec)
yes
?-init. [Operation 7]
Rs File: data.rs.
DECLIST: <inf=[1,5]>
Certain Rules come from [1,5]
EXEC_TIME=0.001(sec)
yes
?-minimal. [Operation 8]
<<Minimal Certain Rules from object 1>>
Descriptor [1,g] is a core for object 4
[1,g]⇒[5,e],[6/6(=2/2,3/3),DGC,Core],
[1,5],[ (0.4,0.4), (1.0,1.0), (0.66,1.0) ]
<<Minimal Certain Rules from object 5>>
[1,g]⇒[5,e],[6/6(=2/2,3/3),DGC,Common],
[1,5],[ (0.4,0.4), (1.0,1.0), (0.66,1.0) ]
[2,h]⇒[5,e],[12/12(=4/4,3/3),DGC,Common],
[5],[ (0.2,0.4), (1.0,1.0), (0.33,1.0) ]
EXEC_TIME=0.016(sec)
yes

```

In Operation 7, objects, which can generate certain rules, are examined. In this execution, we mark objects 1 and 5. The details of this examination are like in Example 2. For objects 1 and 5, program 'minimal' in Operation 8 picks up minimal condition part [CON,ptup] by using each discernibility function DF(1) and DF(5), respectively. In Operation 8, core and common descriptors exist in both discernibility functions, therefore we obtained minimal certain rules without interactive process.

Appendix 3.

Let us show the degrees dep'(CON,DEC) (Row: a condition attribute CON, Column: a decision attribute DEC, and the threshold value is 0.7 in Table_{4,5}) of data dependencies in Table 3.

Appendix 4.

Let us show a real execution to obtain minimal certain rules from 14th student's data in Table₆₀.

```
?-solall(14).
Input a number to Start Exhaustive Search: 10.
Exhaustive Search for less than 1024 Cases !!
<<Minimal Certain Rules from object 14>>
Core Descriptors: []
DF without Core Descriptors: [[1,[1,5],[2,5],
[3,5],[4,5],[5,5],[6,5],[7,4],[8,1],[9,1],[10,
1],[11,5],[12,1],[13,5],[15,5]],[2,[2,5],[3,5],
[4,5],[5,5],[6,5],[7,4],[8,1],[10,1],[11,5],
[12,1],[14,4],[15,5]],..., [59,[1,5],[3,5],[4,5],
[5,5],[6,5],[7,4],[8,1],[9,1],[10,1],[11,5],
[12,1],[13,5],[14,4],[15,5]]]
Currently Selected Descriptors: []
[Loop: 1]
Descriptors in DF: [[1,5],[2,5],[3,5],[4,5],[5,
5],[6,5],[7,4],[8,1],[9,1],[10,1],[11,5],[12,
1],[13,5],[14,4],[15,5]]
Select a Descriptor: [2,5].
[Loop: 2]
Common Descriptors: [[3,5],[4,5]]
Exhaustive Search for [[2,5],[3,5]], 4 Cases !!
[2,5]&[3,5]=>[16,5],
[4608/4608(=96/96,48/48),DGC],
[14,33,34,37,46,49],
[(0.1,0.1),(1.0,1.0),(0.42,0.54)]
Exhaustive Search for [[2,5],[4,5]], 4 Cases !!
[2,5]&[4,5]=>[16,5],
[6144/6144(=128/128,48/48),DGC],
[14,33,34,37,46,49],
[(0.1,0.1),(1.0,1.0),(0.428,0.545)]
EXEC_TIME(for Exhaustive Search)=0.000(sec)
yes
```

Table 3 (No.1) : Degrees of data dependencies

	1	2	3	4	5	6	7	8
1	1.00	0.51	0.58	0.27	0.38	0.40	0.69	0.24
2	0.00	1.00	0.00	0.00	0.00	0.04	0.02	0.00
3	0.07	0.00	1.00	0.00	0.00	0.04	0.27	0.00
4	0.07	0.00	0.13	1.00	0.00	0.04	0.20	0.00
5	0.07	0.00	0.13	0.20	1.00	0.04	0.22	0.00
6	0.18	0.00	0.13	0.16	0.00	1.00	0.13	0.00
7	0.00	0.00	0.22	0.09	0.22	0.13	1.00	0.22
8	0.00	0.00	0.04	0.00	0.00	0.04	0.02	1.00
9	0.00	0.00	0.04	0.00	0.00	0.04	0.02	0.00
10	0.07	0.00	0.04	0.27	0.04	0.13	0.02	0.00
11	0.07	0.00	0.04	0.00	0.00	0.04	0.02	0.00
12	0.00	0.04	0.00	0.00	0.00	0.04	0.02	0.00
13	0.04	0.00	0.00	0.00	0.00	0.04	0.02	0.00
14	0.11	0.04	0.00	0.09	0.00	0.04	0.02	0.00
15	0.00	0.00	0.04	0.00	0.00	0.04	0.20	0.00
16	0.07	0.04	0.38	0.09	0.16	0.13	0.09	0.00

Table 3 (No.2) : Degrees of data dependencies

	9	10	11	12	13	14	15	16
1	0.18	0.27	0.27	0.29	0.38	0.18	0.42	0.53
2	0.00	0.00	0.02	0.07	0.00	0.02	0.00	0.02
3	0.00	0.07	0.31	0.00	0.00	0.16	0.00	0.36
4	0.00	0.07	0.02	0.00	0.00	0.16	0.00	0.16
5	0.16	0.13	0.02	0.00	0.00	0.11	0.11	0.18
6	0.16	0.09	0.02	0.07	0.16	0.11	0.11	0.18
7	0.29	0.00	0.02	0.20	0.11	0.18	0.27	0.02
8	0.00	0.00	0.02	0.00	0.00	0.09	0.00	0.02
9	1.00	0.00	0.13	0.18	0.00	0.11	0.00	0.02
10	0.27	1.00	0.02	0.00	0.00	0.02	0.09	0.02
11	0.00	0.00	1.00	0.00	0.00	0.09	0.00	0.02
12	0.00	0.00	0.02	1.00	0.00	0.02	0.00	0.02
13	0.00	0.00	0.02	0.09	1.00	0.02	0.00	0.02
14	0.22	0.00	0.02	0.07	0.00	1.00	0.00	0.02
15	0.00	0.00	0.02	0.00	0.00	0.09	1.00	0.02
16	0.00	0.07	0.02	0.00	0.00	0.09	0.47	1.00



Hiroshi SAKAI

Hiroshi Sakai received the D.S. degree in applied mathematics and computer science from Kyushu University in 1988.

He is a professor in the Department of Mathematics and Computer Aided Science, Faculty of Engineering at Kyushu Institute of Technology.

His current research interests are rough sets, mathematical logic, logic programming, etc. He is a member of Japan Society for Fuzzy Theory and Intelligent Informatics, Japanese Society for Artificial Intelligence and The Mathematical Society of Japan.



Kazuhiro Koba

Kazuhiro Koba received the B.E. degree in computer engineering from Kyushu Institute of Technology in 2006, and he is currently studying at the graduate school of engineering.

His current research interests are rough sets and data mining, etc.



Michinori NAKATA

Michinori Nakata received the B.Sci in physics from Hokkaido University, Japan, in 1977, the Dr.Sci in astronomy from Tohoku University, Japan, in 1986.

He is a professor in the Department of Management Science, Faculty of Management & Information Science at Josai International University.

His current research interests are imperfect databases, rough sets, logic programming, soft computing, and intelligent informatics. He is a member of Japan Society for Fuzzy Theory and Intelligent Informatics, Association for Computing Machinery and The Institute of Electrical and Electronics Engineers.