LEM2-based Rule Extraction from Non-monotonic Decision Tables with Ordinal Attributes Based on Rough Sets

Masahiro Inuiguchi, Masayo Tsurumi and Daizo Fukuda
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, JAPAN
E-mail: {inuiguti,tsurumi}@sys.es.osaka-u.ac.jp, fukuda@inulab.sys.es.osaka-u.ac.jp

***Abstract*– In this paper, we discuss rule extraction from non-monotonic decision tables with ordinal attributes based on rough sets. In order to treat the non-monotonicity of a decision attribute with respect to ordinal condition attribute, we use generalized rough sets based on a family by intervals spanned by a set of objects in the decision table. We propose a LEM2-based rule extraction algorithm called INTLEM and show its advantages by numerical experiments.**

I Introduction

Rough sets were proposed by Pawlak [8]. The methods based on rough sets are useful to analyze information tables and applied to various fields, medical informatics, knowledge discovery, decision analysis and so on. Since rough sets were defined under equivalence relations referred to as indiscernibility relations, the attribute values in information tables were implicitly assumed to be nominal. Under this assumption, the rule induction system LERS (Learning from Examples based on Rough Sets) which produces certain and possible rules has been proposed. LEM2 (Learning from Examples Module, version 2) is a subsystem which computes local coverings of attribute-value pairs. Local coverings are constructed from minimal complexes. The minimal complex contains attribute-value pairs, selected on the basis of their relevancy to the concept. In the case of a tie, the next criterion is the maximum of conditional probability of the concept given the attribute-value pair.

In the classical rough set based rule extraction, the implicit weak assumption that the attribute values in information tables were nominal may sometimes lead to unacceptable results when some of attribute values are ordinal. Greco et al. [1] demonstrated this unacceptability by using a simple example. They proposed dominance based rough sets in order to treat ordinal attributes in information tables. The dominance based rough sets are powerful tool to analyze decision tables with ordinal attributes such as preference information tables (information tables about human preferences). Further, they introduced DOMLEM, which is based on LEM2 and which can deal with monotonic decision table [2].

Before the dominance based rough sets, numerical attributes had been treated in rough set literature (see [4, 13]). Numerical attributes are obviously ordinal attributes. Obtaining a proper partition of each attribute was a key issue to deal with numerical attributes in rough set analysis. Various approaches were proposed. Among them, the minimally necessary partitions for numerical attributes are obtained by RSBR [] through converting attribute values to binary values. Tanaka et al. [12] proposed another method to convert attribute values to binary values so that they obtained proper partitions of ordinal attributes.

In the analysis by dominance based rough sets, it is assumed that the decision attribute values are monotonic with respect to condition attribute values. This assumption is often acceptable when the given decision table shows the preference of a decision maker. However, in real world applications, there are a lot of attributes whose moderate values are preferable to extreme values. For example, we prefer moderate values to extremes for blood pressure, room temperature, sweetness of cakes, and so on. If we know the appropriate value for such an attribute, we may convert the attribute to another by a suitable non-monotonic function so as to possess the monotonicity with respect to the decision attribute. Unfortunately, it often occurs that we would not know the exact appropriate values or that we would like to know the appropriate values by the analysis. In such cases, the direct applications of dominance based rough sets would not be suitable.

Inuiguchi et al. [6] discussed the treatment of a decision attribute which is not monotonic with respect to condition attributes with ordinal property. To this end, rough sets suitable for the problem setting have been defined. It is assumed that each attribute is ordinal and that the objects in the same decision class are not very scattered in condition attribute space. Because of this assumption, the application of classical rough sets under equivalence relations will not be advantageous. Moreover, we may have non-monotonic attributes in the real world which we cannot treat properly by dominance based rough sets. From the problem setting, we would like to know what ranges of condition attribute values imply what ranges of decision attribute values. Namely, the extracted decision rules might have ranges of condition attribute values rather than single values in the conditional parts. Then they introduced rough sets defined under a family of intervals. In order to extract decision rules, they extended decision matrix methods[6]. Furthermore, since the problem is to obtain proper granules of condition attributes of decision rules, the approach is similar to BSBR. Then, they compared their proposed method with BSBR and shown the advantages of their proposed method. However, the decision matrix method requires a lot of computational effort. A rule extraction method with less computational effort is desired.

In this paper, it is assumed that some attributes are ordinal and that the decision classes are not very scattered in ordinal condition attribute space. We shall discuss a method for extracting a set of decision rules covering all objects with minimal conditions based on this rough set. This method requires less computational effort than the method by an extended decision matrix. It is an extension of LEM2 and then it does not extract all decision rules but decision rules composing a minimal cover of a given definable set. We call the method, INTLEM in this paper. After we propose INTLEM, we examine efficiency of our proposed method through a numerical experiment.

II Generalized Rough Sets under a Given Family

Various generalizations of rough sets have been proposed [5, 9, 10]. Two interpretations of rough sets have been proposed by Inuiguchi [5]: one is an interpretation of rough sets as classification of objects into positive, negative and boundary regions and the other is an interpretation of rough sets as approximation of sets by means of elementary sets of a given family. In applications, such interpretations are important to obtain proper results. The former interpretation extracts decision rules which infer positive members of a given set and the conditional parts of decision rules are single valued specifications of condition attribute values. The latter interpretation extracts decision rules which infer members of the lower approximation of a given set and the conditional parts of decision rules are regional specifications of condition attribute values. Since we would like to extract decision rules with condition attribute ranges in conditional parts, we introduce the latter interpretation.

In the latter interpretation, we assume that a family $\mathcal{F} = \{F_1, F_2, \dots, F_p\}$ on a universe U is given. The lower and upper approximations of a set $X \subseteq U$ is defined by

$$\mathcal{F}_*(X) = \bigcup\{F_i \mid F_i \subseteq X,\ i = 1, 2, \dots, p\}, \tag{1}$$

$$\mathcal{F}^*(X) = U - \bigcup\{F_i \mid F_i \subseteq U - X,\ i = 1, 2, \dots, p\}. \tag{2}$$

A pair $(\mathcal{F}_*(X), \mathcal{F}^*(X))$ is called a rough set of X under a family $\mathcal{F}$. The pair is a classical rough set of X when $\mathcal{F}$ is a partition, i.e., $F_i \cap F_j = \emptyset$ for $i \neq j$ and $\bigcup\{F_i \mid i = 1, 2, \dots, p\} = U$. The fundamental properties of this kind of rough sets are discussed in [5].

Corresponding to the definition of the lower approximation of X, we may extract the following type of decision rules (see [5]):

$$\text{if } x \in F_i \text{ then } x \in X, \tag{3}$$

where F_i satisfies $F_i \subseteq X$. Such rules are extracted from decision tables. A set X is defined by using decision attribute values while a set F_i is defined by using condition attribute values.

III Decision Tables, Decision Classes and the Family

In this paper, we consider a decision table represented by a 4-tuple $\mathcal{D} = (U, C \cup \{d\}, V, \rho)$, each components of which are defined in the following. $U = \{x_1, x_2, \dots, x_n\}$ represents a finite set of objects, $C = \{c_1, c_2, \dots, c_m\}$ is a finite set of condition attributes and d is a decision attribute. Let $V = \bigcup_{a \in C \cup \{d\}} V_a$ and V_a be a domain of the attribute a. If there is a total order $\leq_a$ on V_a, then $a \in C$ is called ordinal. For convenience, we use simply $\leq$ instead of $\leq_a$ if there is no confusion. The set of all ordinal condition attributes is denoted by C^O and let $C^N = C - C^O$. The order $\leq_d$ with respect to the decision attribute d may represent the preference of a decision maker. Further, let $\rho : U \times C \cup \{d\} \to V$ be a function such that $\rho(x, a) \in V_a$ for every $x \in U$ and $a \in C \cup \{d\}$, which is called an information function. If d is not ordinal, a target class will be represented by $X = \{x \in U \mid \rho(x, d) = v\}$ for some $v \in V_d$ and otherwise it is represented by $X = \{x \in U \mid v_1 \leq \rho(x, d) \leq v_2\}$ for some $v_1, v_2 \in V_d$.

To define a rough set, it is necessary to define a family of sets $\mathcal{F}$. Given a set of condition attributes $A \subseteq C$, we define a family of sets $\mathcal{F}_A$ as follows;

$$\begin{aligned} \mathcal{F}_A &= \{\langle Y \rangle_A \mid \emptyset \neq Y \subseteq U\} \\ \langle Y \rangle_A &= \{x \in U \mid \min_{y \in Y} \rho(y, a) \leqq \rho(x, a) \leqq \max_{y \in Y} \rho(y, a), \\ &\quad \forall a \in A \cap C^O,\ \rho(x, a) = \rho(y, a), \forall y \in Y, \\ &\quad \forall a \in A \cap C^N\}. \end{aligned}$$

By (1), (2) and this family, we can define a rough set in a decision table with ordinal attributes for any $X \subseteq U$.

For any $a \in C^N$ and $v \in V_a$, let $[(a,v)]$ denote a set of objects matching (a,v), namely, $[(a,v)] = \{x \in U \mid \rho(x,a) = v\}$. Similarly, for any $a \in C^O$ and $v_1, v_2 \in V_a$ such that $v_1 \le v_2$, let $[(a,[v_1,v_2])]$ denote a set of objects matching $(a,[v_1,v_2])$, namely, $[(a,[v_1,v_2])] = \{x \in U \mid v_1 \le \rho(x,a) \le v_2\}$. Let T denote a complex being a candidate for a conditional part of the rule. Then $[T]$ denotes the set of objects matching T, namely, $[T] = \{x \in U \mid \rho(x,a^1) = v, \forall (a^1,v) \in T \text{ s.t. } a^1 \in C^N \text{ and } v_1 \le \rho(x,a^2) \le v_2, \forall (a^2,[v_1,v_2]) \in T \text{ s.t. } a^2 \in C^O\}$.

Let $B \subseteq U$ be a nonempty lower or upper approximation of a target class. Given a complex T, it is said that $B \subseteq U$ does not depend on T if and only if $[T] \neq \emptyset$ and $[T] \subseteq B$. A set T is called a minimal complex of B if and only if B depends on T and B does not depend on T' for any $T' \subsetneq T$. Let $\mathcal{T}$ be a nonempty collection of attribute-value pairs. Then $\mathcal{T}$ is said to be a local covering of B if and only if the following conditions are satisfied:

1. each member T of $\mathcal{T}$ is a minimal complex of B,
2. $\bigcup_{T \in \mathcal{T}} [T] = B$,
3. $\mathcal{T}$ is minimal, i.e., $\forall \mathcal{T}' \subseteq \mathcal{T}, \cup_{T \in \mathcal{T}'}[T] \neq B$.

IV LEM2

LEM2 [3] is an algorithm to produce a minimal discriminant description for the classical rough set setting, which implicitly assumes all attributes are nominal, i.e., $C^O \neq \emptyset$. LEM2 is an algorithm to produce a single local covering and is described as follows;

Procedure LEM2
(**input:** a set B;
output: a single local covering $\mathcal{T}$ of set B;)
begin
$G := B$;
$\mathcal{T} := \emptyset$;
while $G \neq \emptyset$ **do begin**
$T := \emptyset$;
$T(G) := \{(a,v) \mid [(a,v)] \cap G \neq \emptyset, a \in C, v \in V_a\}$;
while $(T \neq \emptyset)$ **or** $([T] \not\subseteq B)$ **do begin**
select a pair $(a,v) \in T(G)$;
$T := T \cup \{(a,v)\}$;
$G := G \cap [(a,v)]$;
$T(G) := T(G) - T$;
end{while $([T] \not\subseteq B)$};
for each $(a,v) \in T$ **do begin**
if $[T - \{(a,v)\}] \subseteq B$ **then** $T := T - \{(a,v)\}$;
end{for};
$\mathcal{T} := \mathcal{T} \cup \{T\}$;
$G := B - \cup_{T \in \mathcal{T}}[T]$;
end {while$(G \neq \emptyset)$};
for each $T \in \mathcal{T}$ **do begin**
if $\bigcup_{S \in \mathcal{T} - \{T\}}[S] = B$ **then** $\mathcal{T} := \mathcal{T} - \{T\}$;
end{procedure};

In the step of pair selection, select a pair with the highest attribute priority when such a priority is given; if a tie occurs or such a priority is not given, select a pair such that $|[(a,v)] \cap G|$ is maximum; if another tie occurs, select a pair with the smallest cardinality of $[(a,v)]$; if a further tie occurs, select the first pair.

Time complexity of LEM2 is $O(k^3m^2)$ where $k = |B|$ and $m = |C|$, which means polynomial time.

V THE METHOD PROPOSED BY TANAKA ET AL.

Tanaka et al. [12] proposed a method to obtain minimally requisite divisions of ordinal condition attributes. The main idea behind this method is to transform an ordinal condition attribute to a certain number of binary (categorical) condition attributes. Applying this transformation, we obtain a 2-valued decision table. Then the minimally requisite divisions of ordinal condition attributes can be obtained by the classical rough set analysis of the 2-valued decision table.

The procedure of transformation for the case of $C^N = \emptyset$ is described as follows; Let the set of attribute values for $a \in C^O$ be $V_a = \{v_a^1, v_a^2, \ldots, v_a^m\}$. For a relation $>$ on V_a, assume that $v_a^1 > v_a^2 > \cdots > v_a^m$. For each attribute value v_a^i, consider an attribute r_a^i, which means that we derive new $|V_a|$ attributes from each attribute $a \in C^O$. Given $a \in C^O$, for any $x \in U$ and for any $r_a^i, i = 1, \ldots, m$, the attribute value $\overline{\rho}(x, r_a^i)$ is defined by

$$\overline{\rho}(x, r_a^i) = \begin{cases} 1, & \rho(x,a) \ge v_a^i, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For any ordinal attribute, we can obtain a 2-valued decision table by using (4).

Applying classical rough set analysis to the 2-valued decision table, we find reducts and decision rules with 2-valued attributes r_a^i's. By the inverse application of (4) to the obtained reducts, we may find requisite divisions. Similarly, decision rules with 2-valued attributes can be interpreted as decision rules with original condition attributes by the inverse application of (4). For example, by the inverse application of (4), a decision rule with 2-values attributes $(i > j)$,

$$\text{if } \overline{\rho}(x, r_a^i) = 1 \text{ and } \overline{\rho}(x, r_a^j) = 0 \text{ then } \overline{\rho}(x, r_d^u) = 1 \tag{5}$$

can be interpreted as

$$\text{if } v_a^i \le \rho(x,a) < v_a^j \text{ then } \rho(x,d) \ge v_d^u. \tag{6}$$

Time complexity of this method is $O(k^5m^2)$ when we apply LEM2 to extract decision rules. Thus this method is more computationally complex than LEM2.

VI INTLEM

A. INTLEM

To analyze a non-monotonic decision table with ordinal attributes, Inuiguchi et al.[6] have introduced an extended decision matrices. However, time complexity for this method is not polynomial. In this paper, we shall propose an algorithm which derives a minimal set of decision rules covering all objects in a polynomial time. The algorithm is based on LEM2 and will be called INTLEM (Interval attribute value based on LEM).

As in LEM2, we can choose a lower approximation or an upper approximation of each target class for an input B in INTLEM. Given an input B, we obtain a minimal set of decision rules covering all objects in B as an output. More precisely, a minimal collection composed of conditional parts of decision rules which cover all objects in X.

In INTLEM, a condition is continuously added to the temporal condition set T until objects satisfying all conditions in T are included in B. The selection of added condition is done by a function *evaluate* which is discussed in the next section. After T is obtained, we erase unnecessary conditions from T so that T is minimal. Then, we add T to the temporal set $\mathcal{T}$ of condition parts of decision rules. Once T is obtained, we got a decision rule explaining objects in $[T]$. Then we erase those objects $[T]$ from a set G of unexplained objects and extract another decision rule explaining some of unexplained objects by continuing the same procedure. The procedure stops when G becomes empty.

A concrete procedure of INTLEM is given as follows;

Procedure INTLEM
(**input:** a set B;
output: a single local covering $\mathcal{T}$ of set B;)
begin
  $G := B$;
  $\mathcal{T} := \emptyset$;
  while $(G \neq \emptyset)$ **do begin**
    $T := \emptyset$;
    $S := G$;
    $Cond := \emptyset$;
    while $(T = \emptyset)$ **or** $([T] \not\subseteq B)$ **do begin**
      $best := \emptyset$
      for each attribute $a \in C^O$ **do begin**
        $Cond := Cond \cup \{(a, [\min(\rho(x_i, a), \rho(x_j, a)), \max(\rho(x_i, a), \rho(x_j, a))]) \mid x_i \in S, x_j \in B\}$
      end{for};
      for each attribute $a \in C^N$ **do begin**
        $Cond := Cond \cup \{(a, \rho(x, a)) \mid x \in S\}$;
      end{for};
      for each $e \in Cond$ **do begin**
        if $evaluate(\{e\} \cup T)$ is better than $evaluate(\{best\} \cup T)$ **then** $best := e$;
      end {for};
      $T := T \cup \{best\}$;
      $S := S \cap [best]$;
    end{while($[T] \not\subseteq B$)};
    for each elementary condition $e \in T$ **do begin**
      if $[T - \{e\}] \subseteq B$ **then** $T := T - \{e\}$;
    end {for};
  $\mathcal{T} := \mathcal{T} \cup \{T\}$
  $G := G - \cup_{\{T \in \mathcal{T}\}}[T]$;
  end{while{$G \neq \emptyset$}};
for each $T \in \mathcal{T}$ **do begin**
if $\bigcup_{S \in \mathcal{T} - \{T\}}[S] = B$ **then** $\mathcal{T} := \mathcal{T} - \{T\}$;
end {procedure};

Time complexity of this procedure depends on evaluation bases, i.e., function *evaluate* for the selection of condition. Indeed, if we use the same evaluation basis as LEM2 for $evaluate(T)$, then the time complexity will be $O(k^5m^2)$.

If we use the following evaluation basis for $evaluate(T)$, we can find a different time complexity: select a condition with the highest attribute priority when such a priority is given; if a tie occurs or such a priority is not given, select a condition such that $|[T] \cap G|/|[T]|$ is maximum; if a tie occurs, select a pair such that $|[T] \cap G|$ is maximum; if a further tie occurs, select the first pair. Then the time complexity of INTLEM is $O(k^4m^2)$. Here we note that this evaluation basis is used in DOMLEM [2], a rule extraction algorithm based on LEM2 from monotonic decision tables.

B. Bases to select conditions

In INTLEM, we use an evaluation basis, i.e., a function *evaluate* to select a condition to be added to T. For a given complex T, we consider the following evaluation criteria:

- $|B \cap [T]|$: the number of objects which are included in a set $B \subseteq U$ and which are covered by the candidate of conditional parts T
- $|G \cap [T]|$: the number of objects which should be covered in this or further steps and which are covered by the candidate of conditional parts T
- $|B \cap [T]|/|[T]|$: the ratio of $|B \cap [T]|$ to $|[T]|$
- $|G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|)$: the ratio of $|G \cap [T]|$ to the sum of the number of objects in

$U - B$ which are covered by T and the number of $|G \cap [T]|$)

- $|att(T)|$: the number of condition attributes included in T, i.e., $|att(T)| = |\{a \in C \mid (a, v) \in T\}|$.

We can use these criteria in an arbitrary order of priority as an evaluation basis.

VII A Numerical Experiment

In this section, we deal with data of classification of wine as a numerical experiment. The condition attributes are Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline. The decision attribute values are represented by 1, 2 and 3. The number of objects are 178. Target classes characterized by decision attribute values 1, 2 and 3 have 59, 71 and 48 objects, respectively.

We consider the following 6 priority orders of evaluation criteria as $evaluate(T)$:

- e_1 : $|G \cap [T]|, |G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|), -|att(T)|$
- e_2 : $|G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|), |G \cap [T]|, -|att(T)|$
- e_3 : $|G \cap [T]|, |B \cap [T]|, |G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|), -|att(T)|$
- e_4 : $|G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|), |B \cap [T]|/|[T]|, |G \cap [T]|, -|att(T)|$
- e_5 : $|G \cap [T]| \geq 1, |B \cap [T]|, |B \cap [T]|/|[T]|, -|att(T)|$
- e_6 : $|G \cap [T]| \geq 1, |B \cap [T]|/|[T]|, |B \cap [T]|, -|att(T)|$

In those evaluation bases, we transform the signs of each value as the larger number represents the better.

We extract decision rules by 6 kinds of INTLEM with evaluation bases corresponding to $e_1, e_2, \ldots, e_6$, and also by LEM2 and the method proposed by Tanaka et al. for comparison. In order to compare extracted bodies of decision rules, we apply the 5-fold closs validation. The results of the experiment are shown in Tables 1, 2, 3 and 4. Table 1 shows the result of all target class while Tables 2, 3 and 4 shows the result of each target class. The numerical experiment is carried out under the following environments; CPU : Pentium III 600MHz, Memory: 256MB, OS: Windows ME, Compiler: Visual C++ 6.0.

In Tables 1, 2, 3 and 4, we observe the following facts:

- When evaluation bases e_1, e_3 and e_5 are used, $|G \cap [T]|$ or $|B \cap [T]|$ has a high priority. Since both $|G \cap [T]|$ and $|B \cap [T]|$ express the high priority on the generality rather than the accuracy, the number of extracted decision rules will be smaller than others, i.e., $|\mathcal{T}|$ will be smaller.
- On the other hand, when evaluation bases e_2, e_4 and e_6 are used, $|G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|)$ or $|B \cap [T]|/|[T]|$ has a high priority and both $|G \cap [T]|/(|(U - B) \cap [T]| + |G \cap [T]|)$ and $|B \cap [T]|/|[T]|$ express the high priority on the accuracy rather than the generality. Then the length of the condition part of each extracted decision rule will be smaller than others, i.e., $|T|$ will be smaller.
- From Table 1, we can see that most of INTLEM brings a good result. The rate of correctly classified objects are better than LEM2. Computational time is shorter than the method proposed by Tanaka et al. Expect for computational time, e_1, e_3 and e_5 are better than the other evaluation bases.

VIII Concluding Remarks

In this paper, we discussed an algorithm to extract decision rules from a non-monotonic decision table with ordinal attributes based on rough sets. By a numerical experiment, we compare our algorithm with other techniques. In the algorithm in this paper, we can change the evaluation bases and their priorities. However, if we fix those, we can have more efficient algorithms. The development of the more efficient algorithms is one of the future topic. A similar approach is developed in [7]. A comparison with this approach is also a future topic. Moreover, it might be interesting to extract decision rules with nested conditions to infer ordinal decision attribute values.

References

[1] S. Greco, B. Matarazzo, R. Słowiński. The Use of Rough Sets and Fuzzy Sets in MCDM, in: T. Gal, T.J. Stewart, T. Hanne, (Eds.) *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, Kluwer Academic Publishers, Boston, pp. 14-1–14-59, 1999.

[2] S. Greco, B. Matarazzo, R. Słowiński, J. Stefanowski. An Algorithm for Induction of Decision Rules Consistent with the Dominance Principle, in: W. Ziarko and Y. Yao (Eds.) *RSCTC 2000, LNAI 2005*, pp. 304–313, 2001.

[3] J.W. Grzymala-Busse. LERS - a system for learning from examples based on rough sets. In: R. Slowinski (ed.) *Intelligent Decision Support. Handbook of*

Table 1: Result of analysis to classify types of wine

	e_1	e_2	e_3	e_4	e_5	e_6	LEM2	Tanaka
The average number of rules	6.40	15.6	7.60	15.6	10.8	58.4	87.0	5.20
The average number of conditions in the rule	3.00	1.00	3.15	1.00	3.27	1.00	1.33	2.47
The average number of objects supporting rule	38.0	13.5	38.7	13.5	41.2	3.89	1.75	38.3
The average rate of correctly classified objects	83.10	76.46	83.7	76.46	82.49	61.71	16.29	85.32
The average rate of misclassified objects	1.67	3.35	1.11	3.35	1.67	9.56	15.17	3.94
The average rate of inconsistent objects	3.41	7.30	3.37	7.30	3.98	15.76	5.60	7.90
The average rate of not classified objects	11.83	12.89	11.8	12.89	11.86	12.97	62.94	2.84
The average computational time	116.9	6.40	194.3	6.20	185.8	41.6	1.40	390.8

Table 2: Result of analysis to classify types of wine w.r.t. target class 1

	e_1	e_2	e_3	e_4	e_5	e_6	LEM2	Tanaka
The average number of rules	1	6.6	1	6.6	1	11.2	27.8	1
The average number of conditions in the rule	3.2	1	3.2	1	3.2	1	1.27	2.2
The average number of objects supporting rule	47.2	8.46	47.2	8.46	47.2	5.09	1.83	47.2
The average computational time	0.636	2.296	0.616	2.178	0.624	7.702	0.404	14.62

Table 3: Result of analysis to classify types of wine w.r.t. target class 2

	e_1	e_2	e_3	e_4	e_5	e_6	LEM2	Tanaka
The average number of rules	4.4	5.4	5.6	5.4	8.8	25.8	35.2	3.2
The average number of conditions in the rule	3.20	1	3.64	1	4.00	1	1.38	3.2
The average number of objects supporting rule	28.4	16.9	30.5	16.9	38.0	3.85	1.72	29.4
The average computational time	115.9	3.074	193.4	3.03	184.8	24.47	0.690	366.8

Table 4: Result of analysis to classify types of wine w.r.t. target class 3

	e_1	e_2	e_3	e_4	e_5	e_6	LEM2	Tanaka
The average number of rules	1	3.6	1	3.6	1	21.4	24	1
The average number of conditions in the rule	2.6	1	2.6	1	2.6	1	1.35	2
The average number of objects supporting rule	38.4	15.2	38.4	15.2	38.4	2.73	1.69	38.4
The average computational time	0.354	1.03	0.322	1.032	0.318	9.404	0.33	9.348

Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht, pp. 3-18, 1992.

[4] J.W. Grzymala-Busse, J. Stefanowski. Three approaches to numerical attribute discretization for rule induction, *International Journal of Intelligent Systems*, vol. 16 no. 1, pp. 29-38, 2001.

[5] M. Inuiguchi. Generalizations of Rough Sets and Rule Extraction, in: J.F. Peters, A. Skowron, J.W. Grzymala-Busse, B. Kostek, R.W. Świniarski, M.S. Szczuka (Eds.) *Transactions on Rough Sets I*, LNCS 3100, Springer, pp. 96-119, 2004.

[6] M. Inuiguchi, T. Yoneda, T. Tanino. A Rough Set Approach to Non-monotonic Preference Information Tables, *Bulletin of International Rough Set Society*, vol. 7, no. 1/2, pp. 29–33, 2003.

[7] M. Kudo, S. Yanagi, M. Shimbo. Construction of Class Regions by a Randomized Algorithm: A Randomized Subclass Method, *Pattern Recognition*, vol. 29, no. 4, pp. 581–588, 1996.

[8] Z. Pawlak. *Rough Sets*, Kluwer Academic Publishers, Dordrecht, 1991.

[9] R. Słowiński, D. Vanderpooten. A Generalized Definition of Rough Approximations Based on Similarity, *IEEE Transactions on Data and Knowledge Engineering*, vol. 12, no. 2, pp. 331–336, 2000.

[10] Y.Y. Yao, T.Y. Lin. Generalization of Rough Sets Using Modal Logics, *Intelligent Automation and Soft Computing*, vol. 2, no. 2, pp. 103–120, 1996.

[11] N. Shan, W. Ziarko. Data-Based Acquisition and Incremental Modification of Classification Rules, *Computational Intelligence*, vol. 11, pp. 357-370, 1995.

[12] H. Tanaka, K. Koyama, Y. Maeda. A Method for reducing information systems with binary data by rough sets, *Fifth IEEE Int. Conference on Fuzzy Systems*, pp. 755-760, 1996.

[13] N. Zhong. Rough sets in knowledge discovery and data mining, *Journal of Japan Society for Fuzzy Theory and Systems*, vol. 13, no. 6, pp.581-591, 2001.