On rough set based approaches to induction of decision rules

Jerzy Stefanowski

Institute of Computing Science,
Poznań University of Technology,
3A Piotrowo Street,
60-965 Poznań, Poland,
E-mail: Jerzy.Stefanowski@cs.put.poznan.pl

Abstract: This paper discusses problems connected with using the rough set theory in induction of decision rules. If the input data table contains inconsistencies the use of the rough set theory leads to generating certain and approximate decision rules on the basis of approximations of decision classes. Three different approaches for inducing: minimum set, exhaustive set and satisfactory set of rules are distinguished. In the first case, the set contains the smallest number of decision rules sufficient to describe all learning examples. The exhaustive set contains all decision rules that can be generated from the given set of examples. The third set contains only such decision rules that satisfy requirements defined by the user/analyst. The paper presents different algorithms that induce all three distinguished categories of the sets of decision rules. The usefulness of these algorithms and the sets of decision rules produced by them for either creating classification systems or performing discovery tasks is discussed. This discussion is extended by a computational comparative study.

Keywords: Rough sets, decision rule induction, classification systems, knowledge discovery.

1 Introduction

The rough set theory has been introduced by Z.Pawlak as a mathematical approach to deal with vagueness and uncertainty in data analysis [50, 51]. The starting point of the rough set theory is an observation that objects may be indiscernible (similar or indistinguishable) due to limited available information. In general, one can distinguish classes of objects rather than individuals. As a consequence of the above indiscernibility it may not be possible to specify a set of objects in terms of available information (i.e. in terms of elementary sets of indiscernible objects). Therefore, the concept of the rough set is introduced which is a set characterized by a pair of precise concepts - called the lower and the upper approximations.

According to [52], the most important problems which can be solved using the rough set concept are the following: finding description of sets of objects in terms of attribute values, checking dependencies (full or partial) between attributes, reducing attributes, analysing significance of attributes, generating decision rules.

The following paper deals with the last of above problems, i.e. with using rough sets to induce decision rules from data sets represented in the form of decision tables.

It must be noticed, however, that the problem of inducing decision rules has been also extensively investigated in other fields. In particular it refers to *Machine Learning* where several efficient algorithms have been already proposed (see, e.g., reviews in [62, 88, 42]). Rough sets can be used on different stages of the process of rule induction and data processing. This can be done in different ways. As a result there is no unique "rough set approach" to induction of decision rules.

There is, however, one common aspect of using rough set theory which makes these systems different from typical machine learning systems (see, e.g. discussion in [20, 24]). It consists in a special way of handling inconsistencies in the input data. In the rough set approaches inconsistencies are not corrected or aggregated. Instead the lower and upper approximations of all decision concepts are computed. Thus, the task of rule induction from inconsistent data is reduced to rule induction from consistent data, since both lower and upper approximations of the concept make it feasible. This is, in fact, the main step of the process where the elements of the rough set theory are used in rule induction. As a consequence of using the approximations, induced decision rules are categorized into *certain* (exact) and *approximate* (possible) ones depending on the used lower and upper approximations, respectively.

Nowadays, the most representative approaches and software systems (see, e.g., [24, 69, 91, 39, 87]) which are used in the majority of last year applications are:

- System LERS (Learning from Examples based on Rough Sets) introduced by Grzymala (see, e.g., [8, 20]) which itself has four different options of rule induction; the most popular of them seems to be LEM2 algorithm. There are available extensions of this algorithm including processing of continuous attributes [9].
- Approaches based on a discernibility matrix and boolean reasoning techniques. These concepts have been introduced by Skowron [66, 63] and extended by several additional strategies connected with, e.g.: the approximation of reducts, looking for dynamic reducts, boundary region thinning, data filtration and tolerance relation [3, 4, 64, 65, 67]. Their implementations have been developed by Skowron and his collaborators and form a computational kernal of the system *Rosetta* [48].
- RoughDAS and RoughFamily software systems developed by Slowinski, Stefanowski and collaborators [45, 72] which offer several rule induction options, e.g. an approach inducing the set of decision rules satisfying the given user's requirements [83]. There are also available other generalizations of rough set model allowing to handle fuzzy input data [74] or generalization using similarity relation between objects instead of indiscernibility [76].

– Systems Dataquest and DataLogic, distributed commercially. The characteristic feature of DataLogic system [84] is the ability of identifying deterministic or probabilistic patterns in data, in form of rules, using the probabilistic extension of the original rough set model called *variable precision rough sets* model [90].
– System KDD-R, oriented towards data mining applications from large databases and capable of finding strongly supported rules, described in [92].

There are also known other learning algorithms inspired by the rough set theory (see, e.g., other proposals given in [69, 91, 39, 40, 87]). For instance, an interesting example of such an integration of statistical techniques at the stage of rule generation is the *probabilistic rough classifier* developed by Piasta [38].

As there are several rough algorithms for rule induction, it is interesting to make an attempt to create their classification. Let us notice that all algorithms aim at inducing the *rule descriptions of decision classes* in the input set of objects. These descriptions consist of the set of the decision rules. Condition parts of these rules are non-redundant. There are, however, different possible types of the induced rule description (see the discussions in [79, 83]). So, we propose to divide rough set based algorithms into three categories depending on the type of finally induced decision rules:

(1) algorithms inducing minimum set of rules,
(2) algorithms inducing exhaustive set of rules,
(3) algorithms inducing satisfactory set of rules.

The first category of algorithms is focused on describing input objects using the minimum number of necessary rules while the second group try to generate all possible decision rules in the simplest form.

The third category of algorithms gives as a result the set of decision rules which satisfy given a priori user's requirements. For example, the user can prefer to get strong decision rules, i.e. rules supported by a relatively large number of input objects.

We can also distinguish different perspectives of rule induction. In general, the induced decision rules can be mostly used for [44]:

(1) creating classification systems,
(2) performing data exploration and knowledge discovery process.

The first aim consists in finding automatically, from the set of learning examples, a collection of decision rules which will be used to *classify* new examples (this problem has been, in fact, extensively studied in the *Machine Learning* field).

The main purpose of the discovery perspective is to extract, from data sets, previously unknown information patterns and regularities (or sometimes exceptions) represented in the form of decision rules which are *interesting* and *useful* to different kinds of users [16]. Such discovered rules can facilitate understanding data sets and, in particular, should help in understanding and explaining dependencies between values of attributes and definitions of decision classes.

It must be noticed, however, that these both perspectives of rule induction are perceived as quite different. One of the basic distinctions consists in different evaluation criteria. In classification-oriented induction, rules are parts of a classification system; hence the evaluation refers to a complete set of rules. The evaluation criterion is usually unique and defined as classification (predictive) accuracy [88]. In discovery-oriented induction, each rule is evaluated individually and independently as a possible representant of an interesting pattern. The evaluation criteria are multiple and considering them together is not an easy task. Moreover, the definition of criteria depends on the application problem and the user's requirements (e.g., see [15, 41, 29]).

The three considered categories of sets of decision rules may have different usefulness for classification and discovery perspectives. The minimum set of rules seems to be more oriented to classification tasks while satisfactory and exhaustive set of rules are mainly oriented to rule discovery problems.

The algorithms for inducing all three kinds of sets of decision rules have been implemented by author and his collaborators in *RoughFamily* software system [45]. The *RoughFamily* system is a set of programs for rough set based data and knowledge analysis. The most known of these programs is *RoughDAS* [72]. The minimum set of rules is obtained by means of the author's reimplementation of LEM2 algorithm [20]. Exhaustive and satisfactory sets of rules are generated by *Explore* algorithm originally introduced by Stefanowski and Vanderpooten [83]. The main objectives of the following paper are:

- to present different rough set based approaches for inducing minimum, exhaustive and satisfactory sets of rules which are used in *RoughFamily* software system,
- to discuss the specificity of these approaches and their usefulness for classification and discovery purposes.

The discussion of the considered algorithms will be extended by a small computational comparative study performed on testing data sets. The sets of decision rules will be evaluated taking into account mainly classification accuracy and measures referring to getting strong and simple decision rules.

The paper is organized as follows. In section 2, basic concepts of the rough set theory, rule induction and criteria for evaluating decision rules are introduced. The algorithms for inducing the minimum, exhaustive and satisfactory sets of rules are presented in section 3. Then, computational experiments are described in section 4. Final remarks are given in the last section.

2 Rough sets and decision rule generation preliminaries

In this section we recall some basic notations related to information systems, rough sets and decision rule induction. More details can be found, e.g. in [51, 69, 53, 71, 93].

2.1 Rough set theory

Rough set theory deals with uncertainty and vagueness in the analysis of information systems. An *information system* (also called classification table, condition-decision table, etc.) is a formal representation of the analysed data set. It is defined as a pair $S = (U, A)$ where U is a finite set of *objects* and A is a finite set of *attributes*. With every attribute $a \in A$, set of its values V_a is associated. Each attribute a determines an information function $f_a\colon U \to V_a$ such that for any $a \in A$, and $x \in U$ $f_a(x) \in V_a$.

In practice, we are mostly interested in analysing a special case of information system called *decision table*. It is any information system $(U, A \cup \{d\})$, where $d \notin A$ is a distinguished *decision attribute*. The elements of A are called *condition* attributes. We can assume that the cardinality of the set V_d of values of the decision attribute d is equal to k. The decision attribute determines the partition of the set of all objets U into k disjoint classes $X_1, X_2, \ldots, X_k$. Sets X_i are called *decision classes*.

The rough set theory is based on an observation that objects may be *indiscernible* (indistinguishable) due to limited available information. This leads to defining the indiscernibility relation. Formally, the *indiscernibility relation* is associated with every subset of attributes $B \subseteq A$ and is defined as $I(B) = \{(x, y) \in U \times U : f_a(x) = f_b(y), \forall a \in B\}$.

The indiscernibility relation defined in this way is an equivalence relation. The family of all equivalence classes of relation $I(B)$ is denoted by $U/I(B)$. These classes are called *elementary sets* or *atoms*. An elementary equivalence class (i.e. single block of the partition $U/I(B)$) containing element x is denoted by $I_B(x)$.

The main problem of the rough set theory is to express properties of any subset of objects $X \subseteq U$ in terms of available information, i.e. in terms of elementary sets. Such a precise definition is not always possible because some elementary sets may be *inconsistent*, i.e. contain examples described by the same values of attributes where some of these examples belong to the set while others not. Therefore, the concept of the *rough set* is introduced which is a set characterized by its *lower* and *upper* approximations.

Formally, if $B \subseteq A$ is a subset of attributes and $X \subseteq U$ is a subset of objects then the sets: $\{x \in U : I_B(x) \subseteq X\}$, $\{x \in U : I_B(x) \cap X \neq \emptyset\}$ are called B-lower and B-upper approximations of X, denoted by $\underline{B}X$ and $\bar{B}X$, respectively. The set $BN_B(X) = \underline{B}X - \bar{B}X$ is called the *B-boundary* of X.

The set $\underline{B}X$ is a maximal set including objects that can be certainly classified as elements of X while $\bar{B}X$ is a minimal set of objects which can be possibly classified to X, having the knowledge represented by attributes from B. The set $BN_B(X)$ reflects information ambiguity in describing the set X, i.e. it contains inconsistent objects.

The set X is *B-definable* iff $\underline{B}X = \bar{B}X$ otherwise it can be defined *roughly*. If $X_1, X_2, \ldots, X_k$ are decision classes in S then the set $\underline{B}X_1 \cup \underline{B}X_2 \cup \ldots \cup \underline{B}X_k$ is called the B-positive region of classification induced by d and is denoted by $Pos_B(d)$.

Approximations are used to define other basic concepts of the rough set theory, e.g.

- dependency between attributes,
- quality of approximation of the objects' classification (also called degree of dependency),
- reducts and a core of attributes.

2.2 Decision rules

In general, decision rules can be perceived as data patterns which represent relationships between values of attributes in the decision tables [93]. Before giving some formal definitions, let us discuss some specificity of integrating elements of the rough set theory and rule induction.

As it has been discussed in the previous section, the rough set theory is particularly well suited to deal with inconsistencies in the input decision table. If the input decision table is inconsistent then lower and upper approximations of the decision classes are computed. The decision rules are generated from these approximations. As a consequence of this way of treating inconsistencies two basic kinds of rules are distinguished (i.e. following the terminology used in [20, 56, 93]):

(1) *certain* (also called exact, deterministic or discriminant),
(2) *approximate* (also called possible, non-deterministic).

For each decision class, *certain decision rules* are generated from objects belonging to its lower approximation. *Possible/approximate decision rules* are generated either from the upper approximation or from the boundaries of this decision class with other classes. Let us notice that certain rules indicate unique decision to be made while approximate rules lead to a few possible decisions.

Considering the second type of rules one can notice that there are some differences among proposed approaches. In the LERS system created by Grzymala [20] possible rules are always induced from upper approximations. Their representations indicate one possible decision class (in the sense of its upper approximation). In the following study, we consider approximate rules which are induced from boundaries between the given decision class and other classes. Such a way of generating the second type of rules also occurs in Skowron's approach based on discernibility matrix [56, 63, 64].

Let us notice that the boundary region of the given decision class may generally consist of a few disjoint subsets of objects. These subsets are joint parts of the upper approximation of the given class with upper approximations of other decision classes. In such a case approximate rules are not induced from the complete boundary set but independently from all these subsets. For instance, let us assume that three decision classes $X_1, X_2, \ldots, X_3$ are roughly defined in the decision table. The boundary of the class X_1 consists of three disjoint subsets, i.e. $BN_B(X_1) = (\bar{B}X_1 \cap \bar{B}X_2 - \bar{B}X_3) \cup (\bar{B}X_1 \cap \bar{B}X_3 - \bar{B}X_2) \cup (\bar{B}X_1 \cap \bar{B}X_2 \cap \bar{B}X_3)$.

The approximate decision rules will be induced independently from each of these three subsets.

These aspects of generating certain and approximate decision rules could be expressed using the notion of a *generalized decision* in the decision table (cf. [56, 65]). For the simplicity of notation, let us further assume that the set V_d of values of the decision attribute d is equal to $\{1,\ldots,k\}$

If $(U, A \cup \{d\})$ is a decision table, then we define a function $\delta_A : U \rightarrow P(\{1,\ldots,k\})$, called the generalized decision, in the following way $\delta_A(x) = \{i : \exists x' \in U$ that $x' I(A) x$ and $f_d(x) = i\}$. A decision table is called consistent (deterministic) if $\mid \delta_A(x) \mid = 1$ for any $x \in U$, otherwise it is inconsistent.

Objects from any decision table could be partitioned into disjoint m subsets Y_j of objects (where $1 \leq j \leq k \leq m$). Each subset contains objects described by the same value of the generalized decision $\delta_A(x)$. If for all objects x in subset Y_j we have $\mid \delta_A(x) \mid = 1$ then Y_j is the lower approximation of the decision class indicated by index $\delta_A(x)$. If for all objects x in subset Y_j we have $\mid \delta_A(x) \mid > 1$ then this subset refers to boundary (or subboundary) between decision classes and it is used to induce approximate decision rules.

The decision rules are *induced iteratively* for each of subsets Y_j $(i = 1,\ldots,m)$. Thus, each subset is considered *independently* so as to be described by induced decision rules. The rule description must refer to *positive examples* of this decision concept (i.e. objects from Y_j) and cannot be satisfied by its *negative examples* (i.e. objects from $U \setminus Y_j$). Let us define it more formally.

In the following, K will represent the *decision concept* Y_j to be described.

An expression (a, v) where $a \in A$ and $v \in V_a$ is an *elementary condition* c of the decision rule which can be checked for any $x \in X$. Some authors call it also a *selector*. In the majority of the rough set based systems learning from attribute-value representations, these elementary conditions are expressed in the form $(a = v)$. An elementary condition c can be interpreted as a mapping $c : U \rightarrow \{\text{true, false}\}$.

A *conjunction* C of q elementary conditions is denoted by $C = c_1 \wedge c_2 \wedge \cdots \wedge c_q$. Its size will be denoted by $Size(C)$.

The *cover* of a conjunction C, denoted by $[C]$, is the subset of examples which satisfy the conditions represented by C. Formally, we have: $[C] = \{x \in U : C(x) = true\}$.

Considering the concept K to be described, the *positive cover* $[C]^+_K = [C] \cap K$ denotes the set of *positive examples* covered by C and the *negative cover* $[C]^-_K = [C] \cap (U \setminus K)$ denotes the set of *negative examples* covered by C.

A *rule* r (which partially describes K) is an assertion of the form

$$\textit{if } R \textit{ then } K$$

where R is a conjunction $c_1 \wedge c_2 \wedge \cdots \wedge c_q$, satisfying $[R]^+_K \neq \emptyset$.
A rule r is thus characterized by its condition part R and the concept K described by r. The set of attribute-value pairs occuring in the left hand side of the rule is referred to as the condition part and the right hand side is the decision part. Often in software systems the decision parts of rules are represented in other forms, i.e.:

- *if* R *then* $(d = i)$ for certain rules (where $i \in V_d$),
- *if* R *then* $(d = i) \vee (d = j) \vee \ldots \vee (d = l)$ for approximate rules (where $i, j, \ldots, l$ are values of generalized decision δ_A represented by objects satisfying the condition part R).

A rule r is *discriminant* (i.e. distinguishes positive examples belonging to K from negative ones) if its condition part $R = c_1 \wedge c_2 \wedge \cdots \wedge c_q$ is:

- *consistent*: $[R]_K^- = \emptyset$,
- *minimal*: removing any condition c_j from R would result in a conjunction which is no longer consistent.

In some data sets, in particular if they contain a lot of inconsistent examples, discovery systems could find only few certain rules which are strong enough, i.e. are supported by many examples. In such cases, it may make sense to look for *partly discriminant* rules. These are rules which apart from positive examples could cover a limited number of negative ones. For instance, such an approach is used in *variable precision rough set model* [90].

In order to evaluate discovered rules several measures could be used (see, e.g., [1, 49, 42, 43, 44, 92, 93]). If one is interested in discovery perspective, measures refer usually to evaluation of a single rule independently of other rules. In many of discovery-oriented systems users are interested in getting decision rules which are mainly strong (i.e. refer to a large number of covered objects) and simple (i.e. whose condition parts consist of a rather limited number of elementary conditions). These could be expressed by measures considered below. For a given rule r with condition part R, the following measures are considered:

a) the *strength* of r, denoted by $Strength(r)$:

$$Strength(r) = | [R]_K^+ |$$

b) the *length* of the rule r, denoted by $Length(r)$:

$$Length(r) = Size(R)$$

In a case of classification systems, the quality of complete set of rules is evaluated using the *classification accuracy rate* (or conversely the misclassification error rate) (see, e.g. [88])) It is defined by the ratio:

$$n_c/n$$

(expressed in percentage) where n is the number of classified testing examples, n_c is the number of these examples which have been correctly classified.

2.3 Categories of rule description

The rough set based algorithms create in an iterative way a set of rules being the description of approximations of decision classes. Let us denote by $\mathcal{R}$ the set of decision rules which are induced in each iteration for the considered decision concept K. It is assumed that all rules $r \in \mathcal{R}$ are discriminant.

We distinguish three possible categories of the set of rules $\mathcal{R}$:

- minimum set of decision rules,
- exhaustive set of decision rules,
- satisfactory set of decision rules.

The set of rules $\mathcal{R}$ is *minimum* if it describes the concept K in the following way:

(1) condition part R of each rule $r \in \mathcal{R}$ is minimal
(2) $\bigcup_{r \in \mathcal{R}} [R] = K$
(3) does not exist any rule $r' \in \mathcal{R}$ such that $\mathcal{R}$ - r' satisfies conditions (1) and (2)

In other words the minimum set of rules contains the smallest number of discriminant rules sufficient to cover the set of objects K. This idea follows the notion of the *minimal discriminant description* introduced by Michalski in [42] and similar one considered by Grzymala in the context of rough set as the *single local covering* [20].

The set of decision rules is called *exhaustive* if it contains **all** discriminant decision rules that can be induced on the basis of positive examples of K. Each rule is produced in the simplest form.

The definition of the above two categories of induced rules (i.e. minimum and exhaustive) is inspired by the classification introduced by Grzymala in [20] where he described two main options of rule induction: *machine learning* approach and *knowledge acquisition* approach (*all rules* option).

Let us comment briefly motivations for inducing the exhaustive rule description. One can easily notice that the minimum set approach produces a limited part of rules which could be interesting for the user (e.g. in the sense of strength or length measures). Only some attribute-value pairs are involved in obtained rules and some other possibly important rule patterns may still remain hidden in the data. If the user wants to learn them, he should use the exhaustive (or knowledge acquisition) approach where all possible rules are induced from the given input data.

However, time complexity for the second choice is exponential and using this kind of approach may be not practical for larger input data files. Moreover, the data analyst could be 'overloaded' by getting too many rules to be considered. In fact, only a small part of them is usually interesting for him.

These limitations of the above two approaches for discovery purposes have encouraged us to consider the third kind of induced rules (for more motivations see [83, 79]), i.e. *satisfactory set of rules.*

The set of decision rules is called *satisfactory* if it contains *only* such rules describing K that satisfy requirements defined by the user.

These requirements express the user's expectations to discovering '*interesting*' rules, regarding e.g. the minimum strength or the maximal length of rules to be induced as well as an emphasis on some specific elementary conditions. Thus, this approach is focused on an interactive discovery controlled by user's preferences.

The introduced categories can be used to classify the known rough set based rule induction algorithms as:

- *Approaches inducing minimum set of decision rules.* The best examples of such approaches are LEM2 algorithm [20], probabilistic rough classifiers [38]. Some other older proposals are CPLA learning algorithm [27], and procedures based on discriminant coefficient [89] can also be included into this category.
- *Approaches inducting exhaustive set of decision rules.* This perspective seems to be very often considered in the rough set literature, e.g. all rules or knowledge acquisition option of the LERS system [20, 18], techniques based on relative rule cores [51], approaches based on discernibility matrix and boolean reasoning [63, 64], approach based on decision matrix (either in non- incremental or incremental version) used in KDD-R system [61, 92]. For two last approaches there are also known their versions which filter final rules or modify the rule discovery process as to get more general rules instead of inducing all of them [92]. The exhaustive rule set can be also obtained by using a version of the *Explore* procedure [83] where no stopping conditions are used.
- *Approaches inducing satisfactory set of decision rules.* Such decision rules can be directly obtained as a result of using the *Explore* procedure [44]. For some data sets (which are rather of limited size) it is also possible to obtain such rules into a two stage approach: first to generate all rules, then to look for relevant rules only [83, 92].

3 Algorithms for induction of decision rules

In this section we present algorithms, implemented in *RoughFamily* software, which allow to obtain all three kinds of rule description, i.e. minimum, exhaustive and satisfactory. All these algorithms follow the general scheme of rough set based rule induction presented in previous sections. Certain and approximate decision rules are induced from lower approximations and boundaries of decision classes. As in all cases the process is repeated iteratively for the given decision concept (i.e. the subset of objects being either lower approximation or boundary), we present the notation of algorithms in the form referring to one iteration. In all notations K represents the set of object to be described.

3.1 Algorithm for extracting minimum set of rules

In the *RoughFamily* system, the minimum set of decision rules is obtained by reimplementation of LEM2 algorithm. This algorithm has been proposed by

Grzymala in [20]. Originally, this algorithm produces certain and possible decision rules. The complete presentation of this algorithm and its comparison to other systems can be found, e.g. in [20, 8, 22].

The elementary conditions of induced decision rules are represented as attribute-value pairs: *(attribute=value)*. Here, we shortly present the general scheme of the algorithm in pseudo-code. We will be using the notation introduced in section 2.2. According to it, c is an elementary condtion, and C is a conjunction of such conditions being a candidate for condition part of the decision rule. Additionally, $\mathcal{C}(G)$ denotes the set of conditions c currently considered to be added to the conjunction C. Rule r is characterized by its condition part R.

```
Procedure LEM2 ( K set of objects; var R set_of_rules )
begin
    G := K;
    R := ∅;
    while G ≠ ∅ do
    begin
        C := ∅;
        C(G) := {c : [c] ∩ G ≠ ∅};
        while (C = ∅) or (not([C] ⊆ K)) do
        begin
            select a pair c ∈ C(G) such that | [c] ∩ G | is maximum;
            if ties occur then select a pair c ∈ C(G) with
            the smallest cardinality | [c] |;
            if further ties occur then select the first pair from the list.
            C := C ∪ [c];
            G := [c] ∩ G;
            C(G) := {c : [c] ∩ G ≠ ∅};
            C(G) := C(G) − C;
        end;
        for each elementary condition c ∈ C do
            if [C − c] ⊆ K then C := C − {c};
        Create rule r basing on the conjunction C and add it to the set of rules R;
        G := K − ⋃_{r∈R} [R] ;
    end;
    for each r ∈ R do
        if ⋃_{s∈R−r} [S] = K then R := R − r
    end;
```

Let us consider the simple illustrative example of the decision table (see Table 1). It contains 17 objects described by 5 condition attributes $(a_1, \ldots, a_5)$. Objects are partitioned into two decision classes according to the value of the decision attribute d.

For objects presented in Table 1 three decision concepts can be found: lower approximation of decision class $(d = 1)$ represented by the set of objects {1, 4,

Table 1. The illustrative decision table

No.	a_1	a_2	a_3	a_4	a_5	d
1	2	2	1	2	3	1
2	3	2	2	1	3	2
3	1	1	3	2	2	2
4	1	1	1	1	2	1
5	3	1	2	2	1	2
6	1	2	3	3	1	2
7	1	1	1	1	2	1
8	1	1	2	1	2	1
9	2	1	1	2	3	1
10	3	1	2	2	1	2
11	1	1	3	2	2	1
12	1	1	2	2	2	2
13	1	2	1	1	3	1
14	3	1	3	1	1	2
15	1	2	3	1	2	1
16	1	2	2	2	2	1
17	3	1	3	1	1	2

7, 8, 9, 13, 15, 16}, lower approximation of decision class ($d = 2$) represented by the set {2, 5, 6, 10, 12, 14, 17}, and boundary of decision classes ($d = 1$) and ($d = 2$) represented by objects {3, 11}.

The set of the following decision rules can be induced from these examples using the LEM2 algorithm (each rule is additionally described by the set of covered objects):

rule 1. if $(a_1 = 1) \wedge (a_4 = 1)$ then $(d = 1)$ $\{4, 7, 8, 13, 15\}$
rule 2. if $(a_1 = 2)$ then $(d = 1)$ $\{1, 9\}$
rule 3. if $(a_2 = 2) \wedge (a_4 = 2)$ then $(d = 1)$ $\{1, 16\}$
rule 4. if $(a_1 = 3)$ then $(d = 2)$ $\{2, 5, 10, 14, 17\}$
rule 5. if $(a_4 = 3)$ then $(d = 2)$ $\{6\}$
rule 6. if $(a_2 = 1) \wedge (a_3 = 2) \wedge (a_4 = 2)$ then $(d = 2)$ $\{5, 10, 12\}$
rule 7. if $(a_3 = 3) \wedge (a_4 = 2)$ then $(d = 1)$or$(d = 2)$ $\{3, 11\}$

As one can notice from the listing, the algorithm LEM2 follows the general heuristic strategy which is typical for many well known machine learning techniques (e.g. *AQ* [42], *CN2* [10], *PRISM* [6]). This strategy consists in creating a first rule by choosing sequentially the 'best' elementary conditions according to some heuristic criteria. Then, learning examples that match this rule are removed from consideration. The process is repeated iteratively while some learning examples remain uncovered.

This characteristic feature can be noticed while analysing the exemplary decision rules induced from Table 1. Only two of them (no. 1 and no. 3) are strong and general while others seem to be weak. The set of all rules, however, covers all learning examples.

The LEM2 algorithm seems to be mainly useful for building classification systems (see discussion in [20, 21, 83]). The induced set of rules is used to classify new objects, i.e. objects unseen in the learning phase. It is performed by matching a new object description to condition parts of decision rules. This may lead, however, to some difficulties because in general three possible cases may happen:

(a) the new object matches exactly one rule,
(b) the new object matches more than one rule,
(c) the new object does not match any of the rules.

In case **(a)**, if the matched rule is an exact one then the classification suggestion is clear. In case of matching to approximate rule the suggestion may be ambiguous. Similar difficulties occur in determining suggestions for case **(b)**. Case **(c)** must be also handled.

So, considering classification perspective it is necessary to extend rule induction by techniques allowing to solve the above difficulties.

Some solutions have been already introduced in machine learning rule classification systems. For instance, in a rule option of the system $C4.5$ [58] rules are ordered. The matching is done starting from the first rule. The earliest matched rule from the list is used to classify the new object. The remaining rules are not tried for matching. The last rule is a default rule. It does not contain any condition attribute-value pairs and is used when none of the previous rules works. Similar idea was used in the first version of the $CN2$ system [10]. Other solutions are proposed for unordered set of rules (like in classical Michalski's AQ system [43]). During classification, the whole rule set is scanned and decision is done by using additional information about learning examples covered by matched rules. In case of no matching, the partly matched rules are considered and the most probable decision is chosen. Some of the above solution can be adopted to handle difficulties in cases **(b)** and **(c)**.

There are also other specific approaches proposed by Grzymala [21] or by Slowinski and Stefanowski [70, 73]. The first one takes into account additional coefficients characterizing rules: the strength of matched or partly matched rules, the number of non-matched conditions, the rule specificity (i.e. length of condition parts). All these coefficients are combined and the strongest decision wins. Another version of choosing the strongest decision is also used in [78] where the rules nearest to the classified object are looked for. The concept of *nearest rules* requires using a *distance measure* – a proposal of such metric (generalized also for nominal attributes) is proposed in [78]. The more sophisticated solution employing the idea of looking for close rules is introduced in [70] and developed in [73]. It is based on the notion of, so called, *valued closeness relation* where two kinds of arguments are considered while calculating the closeness: arguments for being similar between compared objects/rules and arguments for being different. These arguments are aggregated in a non-compensatory way using specific thresholds expressing possible similarity, small difference, strong difference and indifference between an object and a rule. Results of using *valued closeness relation* are discussed in [80].

It must be stressed that LEM2 algorithm is one of the most often used rough set based rule induction algorithm in real-life applications. For instance, original Grzymala's LERS system has proven its applicability in such fields as, e.g.: developing expert systems in NASA Johnson Space Center (Automation and Robotics Division) to be used in medical decision-making on board the Space Station Freedom, supporting tasks considered by the U. S. Environmental Protection Agency, in medicine to assess preterm labor risk for pregnant women and to compare the effects of warming devices for postoperative patients linguistic analysis of semantics for some of English words (see, e.g. [25, 23, 26]).

In addition, the RoughDAS system (being a part of *RoughFamily* software system) – offering author's reimplementation of LEM2 version – has been successfully used in several applications concerning, e.g.: medical diagnosis of different diseases and the patients' treatments (see, e.g., [12, 68, 54, 81]), analysis of chemical structures of pharmaceutical compounds (see, e.g., [35]), processing of histological images (see, e.g. [28]), technical diagnostics of industrial machinery (see, e.g., [47]), analysis of maintenance procedures in transportation system (see, e.g. [75]), financial data analysis and analysis of multi-attribute decision problems (see, e.g. [77]), software engineering [60], or geology [86].

3.2 Algorithms for extracting exhaustive set of rules

This part of *RoughFamily* system generates all rules that can be induced from the input decision table. All rules have non-redundant condition parts where elementary conditions are formed as *(attribute=value)*. In the *RoughFamily* system, such a rule description is obtained by using the *Explore* procedure while no *stopping conditions* are defined. This procedure is described in section 3.3. Let us notice that such a set of rules may be also obtained by other approaches: all rules option of the LERS system [20], techniques based on relative rule cores [51], approaches based on discernibility matrix and boolean reasoning [63, 64], approach based on decision matrix [61].

All decision rules which can be generated from objects represented in Table 1 are listed below.

rule 1. if $(a_1 = 1)$ then $(d = 1)$ $\{1, 9\}$
rule 2. if $(a_3 = 1)$ then $(d = 1)$ $\{1, 4, 7, 9, 13\}$
rule 3. if $(a_1 = 3)$ then $(d = 2)$ $\{2, 5, 10, 14, 17\}$
rule 4. if $(a_4 = 3)$ then $(d = 2)$ $\{6\}$
rule 5. if $(a_5 = 1)$ then $(d = 2)$ $\{5, 6, 10, 14, 17\}$
rule 6. if $(a_1 = 1) \wedge (a_4 = 1)$ then $(d = 1)$ $\{4, 7, 8, 13, 15\}$
rule 7. if $(a_1 = 1) \wedge (a_5 = 3)$ then $(d = 1)$ $\{13\}$
rule 8. if $(a_2 = 1) \wedge (a_5 = 3)$ then $(d = 1)$ $\{9\}$
rule 9. if $(a_2 = 2) \wedge (a_4 = 2)$ then $(d = 1)$ $\{1, 16\}$
rule 10. if $(a_2 = 2) \wedge (a_5 = 2)$ then $(d = 1)$ $\{15, 16\}$
rule 11. if $(a_4 = 1) \wedge (a_5 = 2)$ then $(d = 1)$ $\{4, 7, 8, 15\}$

rule 12. if $(a_4 = 2) \wedge (a_5 = 3)$ then $(d = 1)$ $\{1, 9\}$
rule 13. if $(a_3 = 2) \wedge (a_5 = 3)$ then $(d = 2)$ $\{2\}$
rule 14. if $(a_3 = 3) \wedge (a_4 = 2)$ then $(d = 1)$or$(d = 2)$ $\{3, 11\}$
rule 15. if $(a_1 = 1) \wedge (a_2 = 2) \wedge (a_3 = 2)$ then $(d = 1)$ $\{16\}$
rule 16. if $(a_2 = 1) \wedge (a_3 = 2) \wedge (a_4 = 1)$ then $(d = 1)$ $\{8\}$
rule 17. if $(a_2 = 2) \wedge (a_3 = 3) \wedge (a_4 = 1)$ then $(d = 1)$ $\{15\}$
rule 18. if $(a_2 = 1) \wedge (a_3 = 2) \wedge (a_4 = 2)$ then $(d = 2)$ $\{5, 10, 12\}$
rule 19. if $(a_2 = 1) \wedge (a_3 = 3) \wedge (a_4 = 1)$ then $(d = 2)$ $\{14, 17\}$
rule 20. if $(a_2 = 2) \wedge (a_3 = 2) \wedge (a_4 = 1)$ then $(d = 2)$ $\{2\}$
rule 21. if $(a_1 = 1) \wedge (a_2 = 1) \wedge (a_3 = 3)$ then $(d = 1)$or$(d = 2)$ $\{3, 11\}$
rule 22. if $(a_2 = 1) \wedge (a_3 = 3) \wedge (a_5 = 2)$ then $(d = 1)$or$(d = 2)$ $\{3, 11\}$

This kind of rule description provides the user the richest information about patterns existing in the analysed data table. On the other hand it is the most demanding from the viewpoint of time and memory complexity. So, it could be used for some data sets only.

One can notice that for the considered example the number of induced rules is larger than induced by LEM2 algorithm. Besides strong rules one can find rules that are weak and very specific. There are also rules referring to the same or overlapping sets of objects and rules having similar condition parts. These features of the discovered rules may restrict their readability for larger data sets.

On the other, the literature review show several examples where such approach was useful both for classification and discovery applications [3, 18, 23]. In study [4] it is reported that the approach based on discernibility matrix and boolean reasoning outperformed the well known C4.5 machine learning system [58] in classification tasks.

The use of this set of rules in classifying new objects often leads to multiple matching of objects to rules. It is usually handled in a similar way as described in the previous subsection, i.e. all matched rules are analysed and the strongest (majority) decision class wins. There are several versions of this heuristics depending on the authors (see, e.g. voting strategy [4], counting rule strength [21, 18] or [73, 78]). The RoughFamily system uses the same technique of solving these situations for all implemented algorithms.

3.3 Algorithm for extracting satisfactory set of rules

The main purpose of this algorithm is to discover the set of all decision rules which satisfy user–defined *requirements*. Discovering of all such rule may be impossible by using approaches for extracting the minimum set of rules due to their greedy heuristic scheme. Following this scheme, some important rules may still remain hidden in the data, in particular when different patterns are shared by a large proportion of common examples. This is due to the elimination of learning examples once they are covered by an induced rule. Conversely, the minimum set of rules may also include very specific rules, consisting of many elementary conditions, which refer only to one or very few learning examples. This is due to the last iterations of the heuristic strategy which impose to cover the remaining examples.

Therefore, it is necessary to develop specific approaches for discovery-oriented induction. The procedure *Explore* implemented as a part of *RoughFamily* system is an example of such a specific approach. It is based on an algorithm originally introduced by Stefanowski and Vaderpooten in [83].

The exploration of the rule space is here controlled by parameters intervening in *stopping conditions* (reflecting the *user's requirements*). The stopping conditions guarantee desirable properties of the rules and significantly reduce the computational costs of the algorithm.

In the implementation discussed in the following paper the main attention is focused on the strength of rules. Even if several control parameters can be used simultaneously, and more effectively, in order to guide the discovery of interesting rules, the strength of rules is treated as a main control parameter. A first reason for this is that some control parameters (related, e.g., to the selection of some specific elementary conditions or specific subsets of learning examples) can be tuned only in relation with a specific application and user. Moreover, the strength parameter is probably both the simplest and most significant parameter at least for users with a limited level of expertise. Even for expert users, strong rules can be useful at least as a starting point. The interests in looking for strong rules is also shared by other researchers creating *Knowledge Discovery* systems (see, e.g. [2, 49, 17, 29, 59, 92]).

The main part of the algorithm *Explore* is based on a *breadth-first* strategy which generates rules of increasing size starting from the shortest ones. The strategy begins with the initial rule having an empty condition part. During the search process this empty conjunction is extended with elementary conditions from the list of allowed conditions. The extended conjunctions are evaluated as candidates for being condition parts of rules.

The main part of the algorithm, i.e. the phase of *breadth-first* search is presented in pseudo-code:

Procedure Explore(SC: stopping_conditions; var $\mathcal{R}$: set_of_rules)
begin
  $\mathcal{R} \leftarrow \emptyset$
  for each available elementary condition c **do**
  begin
    if $[c]^{+}_{K} = \emptyset$ or c satisfies SC **then** discard c;
    if $[c]^{+}_{K} \neq \emptyset$ and $[c]^{-}_{K} = \emptyset$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \{c\}$ and discard c
  end;
  form a queue with all the remaining elementary conditions $c_1, \ldots, c_n$;

  while the queue is not empty **do**
  begin
    remove the first conjunction C from the queue;
    let h be the highest index of the condition involved in C;
    generate all the conjunctions $C \wedge c_{h+1}, C \wedge c_{h+2}, \ldots, C \wedge c_n$;

```
        let 𝒞 be the set of these conjunctions;
        for each C' ∈ 𝒞 do
        begin
            if [C']⁺_K = ∅ or C' satisfies SC then 𝒞 ← 𝒞 \ {C'};
            if [C]⁺_K ≠ ∅ and [C]⁻_K = ∅ then
            begin
                if C' is minimal then ℛ ← ℛ ∪ {C'};
                𝒞 ← 𝒞 \ {C'}
            end;
        end;
        place all the conjunctions from 𝒞 at the end of the queue
    end
end
```

The exploration space of candidate rules is controlled by *stopping conditions* SC connected with user defined requirements, here referring to minimal strength of the rule. It could be defined in the following way:
Let C be the comjunction currently examined,

SC: $\frac{|[C]_K^+|}{|K|} < l$, where l is the smallest percentage of positive examples that a rule must cover

As pointed out before, other requirements related e.g. to maximal length of rules can be easily incorporated into stopping conditions SC. One can also incorporate other conditions introduced so as to satisfy additional specific user's requirements (incompatibility of some selectors, relaxing stopping conditions when specific selectors are present in the candidate conjunction, etc.).

Notice finally, that some learning examples may not be covered by decision rules. However, this may not be damaging; it is even instructive to check the examples which are difficult to cover. Such examples can be presented to the user or expert as possible untypical cases. If they appear to be typical, it is possible to focus the search and use 'weaker' stopping conditions.

Let us consider the discovery of decision rules from examples presented in Table 1 by using algorithm *Explore*.

Let us assume that the user's level of interest to the possible strength of a rule is expressed by assigning a value $l = 33\%$ in SC for each decision concepts *(d=1)* and *(d=2)*. *Explore* gives the following decision rules:

rule 1. if $(q_3 = 1)$ then $(d = 1)$ $\{1, 4, 7, 9, 13\}$
rule 2. if $(q_1 = 1) \wedge (q_4 = 1)$ then $(d = 1)$ $\{4, 7, 8, 13, 15\}$
rule 3. if $(q_4 = 1) \wedge (q_5 = 2)$ then $(d = 1)$ $\{4, 7, 8, 15\}$
rule 4. if $(q_1 = 3)$ then $(d = 2)$ $\{2, 5, 10, 14, 17\}$
rule 5. if $(q_5 = 1)$ then $(d = 2)$ $\{5, 6, 10, 14, 17\}$
rule 6. if $(a_2 = 1) \wedge (a_3 = 2) \wedge (a_4 = 2)$ then $(d = 2)$ $\{5, 10, 12\}$

This set of rules can be extended by inducing approximate rules if it is necessary to cover examples 3 and 11. Object 16 is not covered by any of discovered decision rules.

One can notice that the proposed algorithm creates a set of 'relatively' strong rules, as a direct consequence of condition SC. Moreover, these rules are generally shorter than those induced by LEM2 algorithm or rules being the part of the exhaustive set. Some of the rules induced by *Explore* (no. 1, 3, 5) were not discovered by LEM2 and are also strong rules.

4 Computational experiments

The presented algorithms induce different sets of rules. Some particularities of these algorithms and sets of rules induced by them have been already discussed in the previous section. This discussion is extended by small computational experiments. The aim of the performed experiments is to evaluate the usefulness of the sets of rules produced by the three algorithms for two tasks: classification of objects and discovery of decision rules. In the discovery tasks, we restrict our interests to getting a *limited number of relatively strong and short decision rules*. Some of the results are summarized from previous computational experiments performed by the author and presented more precisely in [73] – classification and [44] – discovery points of view.

To evaluate the sets of induced rules, the following measures are taken into account:

- number of rules,
- average rule strength (expressed in number of covered learning examples),
- average rule length (expressed in number of elementary conditions),
- classification accuracy.

The first three points refer mainly to discovery perspective while the last criterion is typical for classification systems.

Classification accuracy was calculated by performing one of the two standard reclassification techniques: either 10 *fold cross-validation* or *leaving-one-out* (see [88]). While performing reclassification tests, the possible ambiguity in matching the testing example to condition parts of decision rules was solved using the so-called VCR approach (described, e.g. in [73]).

The experiments were performed on several different real-life data sets. These data sets were coming from either known applications of rough set theory or they were well-known benchmark sets of examples from machine learning literature. The machine learning data sets were obtained from the *UCI repository of machine learning databases* at the University of California at Irvine [46]. The author is grateful to the creators of these databases and all other people who allowed him to use their data sets for these experiments.

Let us notice that the data sets used in experiments were assumed to be completely defined, i.e. they did not contain any missing values. Due to this fact some of the data were slightly modified by removing few examples or attributes. The data sets *iris, buses* originally contained continuous-valued attributes which were discretized by means of Fayyad and Irani's method [13].

First, we evaluated the usefulness of rough set based algorithms for classifying of objects. As the minimum sets of rules is particularly well suited for discriminating between decision classes, we examined first the LEM2 algorithm. The results of using this algorithm (together with the VCR technique) are presented in Table 4. In all data sets the use of VCR technique increased the classification accuracy (except *small soybean diseases* where the accuracy remained unchanged). More precise analysis of the influence of the VCR technique is presented in [73].

While considering classification perspective of the rough set based rule induction it is necessary to compare the obtained classification accuracy to results of using other well known systems from machine learning field. This is the important issue and in fact, there are still some critical papers (e.g. by Kononenko and Zorc [30, 31]) where it is even claimed that the systems of rule induction based on rough set theory are not sufficiently compared with other, well known systems of machine learning and give, in general, worse results. So, such a comparative study has been performed and results are also given in Table 4.

In this table, algorithm inducing the minimum set of rules is compared with implementation of other machine learning algorithms: *PRISM* algorithm [6], classical version of Quinlan's *ID3* algorithm [57], probabilistic tree classifier *PT* – modification of ID3 with pre-pruning technique (similar idea as in [7]), *ELYSEE* method [85], and *C*4.5 system in a rule option [58].

Table 2. Classification accuracy (in %) of minimum sets of rules obtained by LEM2 algorithm compared to other machine learning systems; — denotes that a classification test for PRISM could not be performed because of inconsistent examples; ? denotes that the "leaving-one-out" test could not be performed for a given implementation.

data set	compared systems					
	ID3	PT	PRISM	ELYSEE	C4.5rules	LEM2
large soybean	81.2	76.6	62.5	86.8	88.4	87.9
election	84.4	88.8	76.7	84.0	89.6	89.4
iris	90.7	90.7	90.0	94.7	91.3	95.3
hsv4	50.0	60.7	—	52.5	61.4	58.2
hsv2	68.0	71.3	—	68.0	78.1	77.1
concretes	86.6	92	—	89.4	87.1	88.9
breast cancer	62.5	68.1	—	63.5	68.1	67.1
imidasolium	35.3	35.3	35.8	59.7	52.1	53.3
lymphograpy	75	82.4	66.9	79.7	80.4	85.2
oncology	78.6	84	73.2	81.9	79.8	83.8
buses	94.7	97.4	?	?	98.7	98.7
small soybean	97.8	97.8	?	?	97.8	97.8

The above results show that the algorithm for getting minimum set of rules enhanced by VCR strategy gives similar classification accuracy as other learning systems.

Next, we extend comparison for algorithms inducing exhaustive and satisfactory sets of decision rules. The classification performance of all three kinds of algorithms implemented in *RoughFamily* are given in Table 3.

Let us discuss briefly the way of getting these results. First of all, one can notice that the computations were performed only for some of data sets presented in Table 4. This is due to much harder computation requirements connected with getting exhaustive and satisfactory set of decision rules than in the case of getting the minimum set of rules.

Although the exhaustive set of rules is computed once for each data set, for some large data sets this approach could not be used due to time and memory restrictions. It occurred even for computations performed on quite 'strong' SGI Power Challenge computing server belonging to Poznan Supercomputing and Networking Center. For instance, for data set *election* we had to stop the rule induction and as a result we got a part of all necessary rules (in this case around 260000 rules).

For the algorithm inducing satisfactory set of rules, the main interest was put on getting strong decision rules. So, only one parameter of the algorithm was used in stopping conditions - *relative rule strength* threshold. This threshold was tuned for different data set in order to get a limited number of relatively strong decision rules having also a classification accuracy comparable to that obtained by other systems. The technique of looking for interesting values of SC threshold is described and precisely tested in the study [44]. Let us shortly comment here that it consists in systematical testing the set of different values of this threshold and examining its influence on the discovered decision rules. For each tested threshold value the chosen evaluation measures are calculated (number of rules, average rule strength, average rule length and classification accuracy). Then, the results are scanned to find a set or sets of rules having required properties. In our case these sets of rules should consist of limited number of rules having satisfactory classification accuracy (comparable to the number of classification rules induced by the algorithm LEM2 and their classification accuracy) and characterized by an average strength higher than in minimum and exhaustive sets of rules. It was possible to find at least one set of rules having satisfactory properties in the above described sense.

The process of looking for such satisfactory sets of rules can be even more time consuming than in the case of looking for exhaustive sets of rules. This is an additional reason that we decided to restrict the number of analysed data sets. So, the process of tuning the SC value was performed in the largest range for *iris, tic-tac-toe, voting* and *election* data set (this computational study is described in [44]). Here, it is extended by the smaller examination for three additional data sets *breast cancer, hsv4, buses.* The results presented in Table 3 for the satisfactory set of rules refer to the best chosen threshold value from these experiments.

As one can notice in Table 3 classification performance of all the compared algorithms is quite comparable. In some cases the minimum rule description leads to the highest accuracy but results obtained for exhaustive set are at the

Table 3. Classification performance of three compared algorithms (classification accuracy expressed in %)

Data set	Exhaustive set of rules	Satisfactory set of rules	Minimum set of rules
Iris	92.67	92.00	95.33
Tic-tac-toe	91.35	97.19	98.96
Voting	95.87	93.31	95.87
Election	—	85.59	89.41
Breast Cancer	70.93	68.54	67.07
Buses	98.70	98.70	98.70
Hsv-4	58.33	53.83	58.20

similar level for the most of analysed data sets.

Table 4. Characteristics of decision rules obtained by three compared algorithms (1 — Number of rules; 2 — Average rule length [# conditions]; 3 — Average rule strength [# examples]; SC — minimum relative strength [%] and refers to the best tested threshold value)

Data Set	Exhaustive set of rules			Satisfactory set of rules				Minimum set of rules		
	1	2	3	1	2	3	SC	1	2	3
Iris	80	2.10	6.03	22	1.86	17.27	10%	23	1.91	11.0
Tic-tac-toe	2858	4.63	4.26	16	3.00	60.25	10%	24	3.67	40.83
Voting	1502	4.72	10.61	50	3.10	104.70	30%	26	3.69	43.77
Election	>260000	—	—	48	2.84	48.89	16%	48	3.27	21.18
Breast Cancer	1163	3.58	2.32	59	3.03	11.46	10%	109	3.43	3.93
Buses	31	3.54	13.09	11	1.64	33.55	50%	3	1.33	33.0
HSV122	1589	1.76	1.75	89	1.74	6.07	8%	53	1.83	3.02

Then, the decision rules induced by all three compared algorithms (inducing minimum, satisfactory and exhaustive sets of rules) have been compared from viewpoint of the other evaluation criteria: number of rules, average rule strength and average rule length. These results are presented in Table 4. The satisfactory sets of rules are the same as evaluated in Table 3. Comparing the sets of rules obtained by all three approaches we can notice that:

- The exhaustive sets usually consist of a large number of relatively long decision rules. Most of these rules are very week. (see e.g. Brest Cancer, Tic-tac-toe, hsv4 in Table 4).

- The minimum set contains the smallest number of rules. However, the obtained sets of rules only partially contain the strongest ones. The average rule strength is better than for exhaustive set of rules but lower than in the case of satisfactory sets of rules (see, e.g. Table 4 — Breast Cancer, Iris, Election, etc.). The same observation refers to average rule length.
- The satisfactory set of rules gives as a result the rules that have the average strength about twice higher than for other sets of rules. They are also shorter. Moreover the number of rules is acceptable and much lower than for the case of exhaustive sets.

The satisfactory sets of rules are, however, dependent on the choice of stopping conditions. Considering possible application problems the threshold values should be defined rather in an interaction with the user/analyst and in accordance with her/his knowledge about the problem and meaning of being 'satisfactory'. Moreover, we observed that the satisfactory sets did not cover all learning examples in the majority of analysed data sets. This is a consequence of their bias to get rules consistent with given requirements to the rule strength.

5 Final remarks

The paper deals with problems of using the rough set theory in induction of decision rules from data represented in decision tables. If the input data are inconsistent then these inconsistencies are handled using the rough set approach. It results in inducing certain and approximate decision rules on the basis of lower approximations and boundaries of each decision class.

We distinguished three types of sets of decision rules describing approximations of decision classes: minimum set, exhaustive set, and satisfactory set of decision rules. First of these sets contains the smallest number of decision rules necessary to describe/cover all learning examples. The second one contains all decision rules that can be generated from the given set of examples, while the third set contains only such decision rules that satisfy requirements defined for the user. The first and second sets of rules describe all learning examples while the satisfactory set of rules may cover only a part of examples relevant to the user defined requirements.

We presented different algorithms, implemented in *RoughFamily* software, which induce all three considered sets of decision rules. The first of these algorithms is based on the idea of local covering introduced by Grzymala and builds the minimum set of rules following the heuristic strategy typical for machine learning systems. This algorithm is the less demanding from the computational point of view. Exhaustive and satisfactory sets of rules are induced by the *Explore* algorithm. This algorithm tries to discover only such decision rules which satisfy user's requirements expressed by proper stopping conditions. In this study, we focused on discovering only strong discriminant decision rules. The scheme of the algorithm is, however, very general and can be easy customized to take into account other requirements related to various criteria of rule evaluation, e.g. the length of discovered rules, the level of discrimination and also requirements on the syntax of condition parts of rules. If the *Explore* algorithm is used without any stopping conditions it produces the exhaustive set of decision rules.

These algorithms may have different usefulness for classification and discovery perspectives. The minimum set of rules produced by the first algorithm is particularly useful for creating classification systems. The greedy heuristic strategy used inside this algorithm restricts the usefulness of its output for discovering rules potentially 'interesting' for the user. On the other hand, exhaustive and satisfactory sets of rules seem to be more oriented to other tasks. The exhaustive set of rules can be used when it is necessary to know about the analysed problem as much as possible. However, inducing of this kind of rules is the most demanding both from the time complexity and memory point of view. Looking for satisfactory set of rules seems to be particularly well suited for user driven interactive knowledge discovery tasks.

The results of the computational experiments show that classification performance of all three rough set based approached to rule induction is quite comparable. In some cases the minimum set of rules leads to the highest accuracy but results obtained for exhaustive set are quite similar for the most of analysed data sets. Moreover, the rough set based algorithms give results similar to that of well known machine learning algorithms. Comparing the sets of rules obtained by each of three approaches from viewpoint of other criteria: number of rules, average rule strength and average rule length one can notice that the satisfactory set of rules can lead to discovering a limited number of the strongest and the most general decision rules. It depends, however, on choosing the proper threshold values for the stopping condition of the *Explore* algorithm.

This paper does not cover all aspects of integrating the rough set theory and rule induction. There are still many other topics which have not been mentioned in this paper but are very important and need more research. In our opinion, it refers, e.g. to the following problems:

- including continuous attributes to the learning process,
- handling incomplete input decision tables (i.e. containing objects with missing values of some attributes),
- integrating the rough set learning algorithms with the strategies of constructive induction,
- comparing the rough set approaches to other similar methods.

Preliminary results have been already obtained in the above topics. For instance, several techniques of discretizing continuous attributes prior to inducing decision rules were introduced by Grzymala-Busse and other authors (see, e.g., [9]). There are already known other different approaches to handle continuous attributes directly in the decision rule learning phase: probabilistic rough classifier [37, 38], LEM2 with interval extension [23], Skowron and Nguyen's approach [67]. Moreover, currently at least few authors are introducing other approaches to create rough approximations which are based on similarity relation instead of strict indiscernibility of objects (see, e.g. [76, 32]). In these proposals continuous attributes may be directly handled in the rough set operations and rule induction techniques without any prior discretizations.

There are also some approaches to handle the incomplete information systems developed within the context of rough sets. For instance in the paper [19] the approach of substituting missing values by a subset of possible values is introduced. Also, in [74] yet other approaches are presented, where a distribution of possible values or fuzzy membership measures are taken into account. Moreover, quite new and comprehensive approach has been recently introduced by Kryszkiewicz [36].

Integration of rough sets, rule induction and constructive induction techniques could be useful for cases of 'difficult' data sets (i.e. the constructive induction point of view [5] could lead to changing the definitions of original attributes into new ones that improve the finally discovered decision rules [82]).

Although there are some theoretical and computational comparative studies of rough set theory with other methodologies (see e.g. [33, 34, 24]) it is necessary to investigate further this subject.

In spite of these achievements, we think that research in the mentioned problems still require further attention.

Last but not least, it is still necessary to work on efficient software development for rough set based data analysis, particularly for large data sets.

Acknowledgment

The work on this paper was partly supported from KBN grant no. 8-T11C 013 13 and CRIT II research grant.

References

1. Agrawal R., Imelinski T., Swami A., Database mining: A performance perpective. *Journal IEEE: Special Issue on Learning and Discovery in Knowledge-Based Databases*, vol. 5, no. 6, 1993, 914–925.
2. Agrawal R., Mannila H., Sirkant R., Toivonen H., Verkamo A.I., Fast discovery of association rules. In Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge Mass., 1996, 307–328.
3. Bazan J., Skowron A., Synak P., Dynamic reducts as a tool for extracting laws from decision tables. In *Proc. of the Symp. on Methodologies for Intelligent Systems*, Charlotte, NC. October 16-19, 1994, Lecture Notes in Artificial Intelligence, no. 869, Springer-Verlag, Berlin 1994, 346-355.
4. Bazan J., Skowron A., Synak P., Discovery of decision rules from experimental data. In Lin T. Y., Wildberger A. (eds.), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*, Simulation Councils, Inc., San Diego CA 1995, 276-279.
5. Bloedorn E.,Michalski R.S., Data-driven constructive induction in AQ17-PRE: a method and experiments. In *Proc. of the Third Conference on Tools for AI*, San Jose. CA, 1991, 25-35.
6. Cendrowska J., PRISM, an algorithm for inducing modular rules. *Int. J. Man-Machine Studies*, 27, 1987, 349-370.
7. Cestnik B., Kononenko I., Bratko I., Assistant 86, a knowledge elicitation tool for sophisticated users. In Bratko I., Lavrac N. (eds.), *Progress in Machnine Learning*, Sigma Press, Wilmshow, 1987, 31-45.

8. Chan C.C., Grzymala-Busse J.W., On the two local inductive algorithms: PRISM and LEM2. *Foundations of Computing and Decision Sciences*, vol. 19 (4), 1994, 185-204.
9. Chmielewski M. R., Grzymala-Busse J. W., Global discretization of continuous attributes as preprocessing for machine learning. In T. Y. Lin and A. M. Wilderberger,(eds.), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*, Simulation Councils, Inc., 1995, 294–297.
10. Clark P., Niblett T., The CN2 induction algorithm. *Machine Learning*, 3, 1989, 261–283.
11. Deogun J., Raghavan V., Sarkar A., Sever H., Data mining: Trends in research and development. In Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 9–46.
12. Fibak, J., Pawlak, Z., Slowinski, K., Slowinski, R.: Rough sets based decision algorithm for treatment of duodenal ulcer by HSV. *Bulletin of the Polish Academy of Sciences. Biological Sciences* vol. 34, 10/12, 1986, 227-246.
13. Fayyad U.M., Irani K.B., Multi–interval discretization of continuous-valued attributes for classification learning. In *Proc. of 13th Int. Conf. on Machine Learning*, Morgan Kaufmann, 1993, 1022–1027.
14. Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge Mass., 1996.
15. Fayyad U.M., Piatetsky-Shapiro G, Smyth P., The KDD Process for extraacting useful knowledge from volumes of data. *Communications of the ACM*, vol. 39, no. 11, 1996, 27–35.
16. Frawley W.J., Piatetsky-Shapiro G, Christopher Matheus, Knowledge Discovery in Databases - An Overview. In G. Piatetsky-Shapiro and Christopher Matheus, (eds.), *Knowledge Discovery in Databases*, AAAI/MIT Press, 1991, 1–30.
17. Golan R. and Ziarko W., A Methodology for stock market analysis utilizing rough set theory. *Proceedings of IEEE/IAFE 1995 Conference on Computational Intelligence in Financial Engineering*, New York City, April 9–11, 1995, 32–40.
18. Grzymala-Busse D. M., Grzymala-Busse J. W., The usefulness of machine learning approach to knowledge acquisition. *Computational Intelligence* 11, 1995, 268–279.
19. Grzymala-Busse J. W., On the unknown attribute values in learning from examples. *Proc. of Int. Symp. on Methodologies for Intelligent Systems*, 1991, 368–377.
20. Grzymala-Busse J.W., LERS - a system for learning from examples based on rough sets. In R. Słowiński, (ed.) *Intelligent Decision Support*, Kluwer Academic Publishers, 1992, 3–18.
21. Grzymala-Busse J. W., Managing uncertainty in machine learning from examples. In Dabrowski M, Michalski R., Ras Z. (eds.) *Proceedings of 3rd Int. Symp. on Intelligent Systems*, Wigry, Poland, IPI PAN Press, Warszawa, 1994, 70–84.
22. Grzymala-Busse J. W., Lakshmanan A., LEM2 with Interval Extension: An Induction Algorithm for Numerical Attributes. In Tsumoto S. (ed.) *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996, 67-73.
23. Grzymala-Busse J. W., Sedelow S.Y., Sedelow W.A., Machnie learning and knowledge acquisition, rough sets and the English semantic code. In Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 91-108.
24. Grzymala-Busse J.W., Stefanowski J., Ziarko W., Rough Sets: Facts versus Misconceptions. *Informatica*, 20, 1996, 455-464.

25. Grzymala-Busse J. W., Woolery, L., Improving prediction of preterm birth using a new classification scheme and rule induction. *Proc. of the 18-th Annual Symposium on Computer Applications in Medical Care (SCAMC)*, Washington, DC, November 5–9, 1994, 730–734.
26. Gunn J., Grzymala-Busse J. W., Global temperature stability by rule induction: An interdisciplinary bridge. *Human Ecology* 22, 1994, 59–81.
27. Hadjmichael M., Wasilewska A., Rough sets-based study of voter preferences in 1988 USA presidental election. In R. Słowiński, editor, *Intelligent Decision Support*, Kluwer Academic Publishers, 137–152, 1992.
28. Jelonek J., Krawiec K., Slowinski R., Szymas J., Rough set reduction of features for picture-based reasoning. In Lin T. Y. and Wildberger A. (eds.) *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*, Simulation Councils, Inc., San Diego CA 1995, 89–92.
29. Klemettinen M., Mannila H.,Ronkainen P., Toivonen H., A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the third int. conference on information and knowledge management* (CIKM'94), Nov. 29-Dec 2, 1994, Geithersburg, Maryland, ACM Press 401–407.
30. Kononenko I., Zorc S., Critical analysis of rough sets approach to machine learning. *Informatica* 18, 1994, 305–313.
31. Konenenko I., On Facts versus Misconceptions about Rough Sets. *Informatica* 20, 1996, 465-468 .
32. Krawiec K., Slowinski R., Vanderpooten D., Construction of rough classifiers based on application of similarity relation In Tsumoto S. (ed.) *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996,, 23-30.
33. Krusinska E., Slowinski R., Stefanowski J., Discriminant versus rough sets approach to vague data analysis. *Applied Stochastic Models and Data Analysis*, 8, 1992, 43–56.
34. Krusinska E., Stefanowski J., Stromberg J.E., Comparibility and usefulness of newer and classical data analysis techniques. Application in medical domain classification. In Didey E. et al. (eds.) *New Approaches in Classification and Data Analysis*, Springer Verlag, Studies in Classification, Data Analysis and Knowledge Organization, 1993, 644-652.
35. Krysinski J., Rough set approach to the analysis of the structure activity relationship of quaternary imidazolium compounds. *Arzneim. Forsch*, 40, 1990, 795–799.
36. Kryszkiewicz M., Rough set approach to incomplete information system. ICS *Research Report* 55/95, Warsaw University of Technlogy.
37. Lenarcik A., Piasta Z., Discretization of condition space. In Slowinski R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of Rough Set Theory.* Kluwer Academic Publishers, 1992, 373–390.
38. Lenarcik A., Piasta Z., Probabilistic rough classifiers with mixture of discrete and continuous attributes. In Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 373-384.
39. Lin T. Y. and Wildberger A. (eds.), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*, Simulation Councils, Inc., San Diego CA 1995.
40. Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997.
41. Mannila H., Data mining: machine learning, statistics, and databases. *Eight International Conference on Scientific and Statistical Database Management*, Stock-

holm June 18-20, 1996, 1–8.

42. Michalski R.S., A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (eds.), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufman, 1983, 83–134.
43. Michalski R. S., Mozetic I., Hong J., and Lavrac N., The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. of the 5th Nat. Conf. on AI*, 1986, 1041–1045.
44. Mienko R., Stefanowski J., Toumi K., Vanderpooten D., Discovery-Oriented Induction of Decision Rules. *Cahier du Lamsade* no. 141, Paris, Universite de Paris Dauphine, septembre 1996.
45. Mienko R., Slowinski R., Stefanowski J., Susmaga R. RoughFamily - software implementation of rough set based data analysis and rule discovery techniques. In Tsumoto S. (ed.) *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996, 437-440.
46. Murphy P.M., Aha D.W., UCI Repository of machine learning databases. University of California at Irvine, Dept. of Computer Science.
47. Nowicki R., Slowinski R. and Stefanowski J., Rough sets analysis of diagnostic capacity of vibroacoustic symptoms. *Computers and Mathematics with Applications*, vol. 24 no. 7, 1992, 109-123.
48. Ohrn A., Komorowski J., Skowron A., Synak P., Rosetta - Part I: System Overview, Technical report, Dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 1997.
49. Piatetsky-Shapiro G., Discovery, Analysis and Presentation of Strong rules. In G. Piatetsky-Shapiro and Christopher Matheus, (eds.), *Knowledge Discovery in Database* AAAI/MIT Press, 1991, 229–247.
50. Pawlak Z., Rough sets. *Int. J. Computer and Information Sci.*, 11, 1982, 341–356.
51. Pawlak Z., *Rough sets. Theoretical aspects of reasoning about data.* Kluwer Academic Publishers, Dordrecht, 1991.
52. Pawlak Z., Rough Sets. In Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 3-8.
53. Pawlak Z., Grzymala-Busse J., Slowinski R., Ziarko, W., Rough sets. *Communications of the ACM*, vol. 38, no. 11, 1995, 89-95.
54. Pawlak Z., Slowinski K., Slowinski R., Rough classification of patients after highly selected vagotomy for duodenal ulcer. *International J. Man-Machine Studies*, 24, 1986, 413–433.
55. Pawlak Z., Slowinski R., Rough set approach to multi- attribute decision analysis. Invited Review in *European Journal of Operational Research*, 72, 1994, 443-459.
56. Pawlak Z., Skowron A., A rough set approach for decision rules generation. *ICS Research Report*, 23/93 Warsaw University of Technology, 1993.
57. Quinlan J.R., Discovering rules from large collections of examples - a case study. In Michie D., (ed.), *Experts Systems in the Microelectronics Age.* Edinburgh University Press, 1979, 33-46.
58. Quinlan J. R., *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo CA, 1993.
59. Riddle P., Segal R., Etzioni O., Representation Design and Brute-force Induction in a Boening Manufacturing Domain. *Applied Artificial Intelligence*, vol. 8, 1994, 125-147.
60. Ruhe, G., Rough set based data analysis in goal oriented software measurement. *Proceedings of the Third Symposium on Software Metrics*, Berlin, March 1996, IEEE Computer Society Press, 10-19.

61. Shan N., Ziarko W., Data-based acquisition and incremental modification of classification rules. *Computational Intelligence: An International Journal*, vol. 11, no. 2, 1995, 357–370.
62. Shavlik J.W., Dietterich T.G., (eds.) *Readings in Machine Learning*, Morgan Kaufman, 1990.
63. Skowron A., Boolean reasoning for decision rules generation. In Komorowski J., Ras Z. (des.) *Methodologies for Intelligent Systems*, LNAI 689, Springer-Verlag, Berlin, 1993, 295–305.
64. Skowron A., Extracting Laws from Decision Tables. *Computational Intelligence: An International Journal* 11(2), 1995, 371-388.
65. Skowron A., Polkowski L., Synthesis of decision systems from data tables. In Lin T.Y., Cecrone N.(ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 289-299.
66. Skowron A., Rauszer C., The discernibility matrices and functions in information systems. In Slowinski R. (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of Rough Set Theory.* Kluwer Academic Publishers, Dordrecht/ Boston/ London, 1992, 331–362 .
67. Skowron A., Son Nguyen H., Quantization of real value attributes, *Proceedings of the Second Joint Conference on Information Sciences*, Wrightsville Beach, North Carolina, September 28 - October 1, 1995, 34-37.
68. Slowinski K., Slowinski R., Stefanowski J., Rough sets approach to analysis of data from peritoneal lavage in acute pancreatitis. *Medical Informatics*, vol. 13, 1988, 143-159.
69. Slowinski R. (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publishers, 1992.
70. Slowinski R., Rough set learning of preferential attitude in multi-criteria decision making. In Komorowski J., Ras Z., (eds.) *Methodologies for Intelligent Systems*, LNAI 689, Springer-Verlag, 1993, 642–651.
71. Slowinski R., Rough set approach to decision analysis. *AI Expert*, March 1995, 19-25.
72. Slowinski, R., Stefanowski, J., 'RoughDAS' and 'RoughClass' software implementations of the rough set approach. In Slowinski R. (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of Rough Set Theory.* Kluwer Academic Publishers, Dordrecht/ Boston/ London, 1992, 445-456.
73. Slowinski R., Stefanowski J., Rough classification with valued closeness relation. In Diday E. (ed.) *New Approaches in Classification and Data Analysis*, Studies in Classification, Data Analysis and Knowledge Organisation, Springer Verlag, 1994, 482-489.
74. Slowinski R., Stefanowski J., Handling various types of uncertainty in the rough set approach. In Ziarko W. (ed.) *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, 366–376.
75. Slowinski, R., Stefanowski, J., Susmaga, R., Rough set analysis of attribute dependencies in technical diagnostics. In Tsumoto S. (ed.) *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996, 384-291.
76. Slowinski R., Vanderpooten D., Similarity relation as a basis for rough approximations, ICS *Research Report* 53/95, Institute of Computer Science, Warsaw University of Technology, (1995). Also accepted to Wang P. (ed.) *Advances in Information Sciences*, vol. IV.

77. Slowinski R., Zopounidis C., Applications of the rough set approach to evaluation of bankruptcy risk, *Intelligent Systems in Accounting, Finance and Management*, vol. 4, 1995, 27–41.
78. Stefanowski J., Classification support based on the rough sets. *Foundations of Computing and Decision Sciences*, vol. 18, no. 3-4, 1993, 371-380.
79. Stefanowski J., *Interactive decision making using knowledge acquired by means of the rough set approach.* Ph.D. Thesis, Poznan University of Technology 1994 (In Polish).
80. Stefanowski J., Using valued closeness relation in classification support of new objects. In Lin T. Y., Wildberger (eds.) *Soft computing: rough sets, fuzzy logic, neural networks uncertainty management, knowledge discovery*, Simulation Councils Inc., San Diego CA, 1995, 324–327.
81. Stefanowski J., Slowinski K. Rough sets s a tool for studying attribute dependencies in the urinary stones treatment data set. In T.Y. Lin, N. Cecrone (ed.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, 1997, 177-198.
82. Stefanowski J., Slowinski K., Rough set theory and rule induction techniques for discovery of attributes dependencies in medical information systems. In Komorowski J., Zytkow J. (eds.) *Principles of Knowledge Discovery.* Proceedings of the First European Symposium: PKDD'97, Trondhiem, Norway, Springer LNAI no. 1263, Springer Verlag, 1997, 36–46.
83. Stefanowski J., Vanderpooten D., A general two stage approach to rule induction from examples. In W. Ziarko, editor, *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, 317-325.
84. Szladow A., Datalogic/R for databases mining and decision support. *Proc. of 2nd Int. workshop on rough sets and knowledge discovery, RSKD-93, Banff, Canada*, 1993, 511–513.
85. Teghem J., Benjelloun M., Some experiments to compare rough sets theory and ordinal statistical methods. In Slowinski R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of Rough Set Theory.* Kluwer Academic Publishers, 1992, 267-287.
86. Teghem J., Charlet J.M., Use of 'rough sets' method to draw premonitory factors for earthquakes by emphasing gas geochemistry: The case of a low seismic activity context, in Belgium'. In Slowinski R. (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of Rough Set Theory.* Kluwer Academic Publishers, 165-180.
87. Tsumoto S. (ed.), *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996, Tokyo University Press.
88. Weiss S. M., Kulikowski C.A., *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*, Morgan Kaufmann, 1991.
89. Wong S.K.M., Ziarko W., Ye Li W., Comparison of rough set and statistical methods in inductive learning. *Int. Journal of Man-machine Studies*, 24, 1986, 53–72.
90. Ziarko W., Variable precision rough sets model. *Journal of Computer and Systems Sciences*, vol. 46. no. 1, 1993, 39–59.
91. Ziarko W.(ed.), *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994.
92. Ziarko, W. and Shan, N., KDD-R: A comprehensive system for knowledge discovery in databases using rough sets. In Lin T. Y, Wildberg A.M., (eds.), *Soft Com-*

puting. rough sets, fuzzy logic, neural networks uncertainty management, knowledge discovery, San Jose, CA., USA, Simulation Council Inc., San Diego, 1994, 93–96.

93. Ziarko W., Review of basics of rough sets in the context of data mining. In Tsumoto S. (ed.) *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo Nov. 6-8 1996, 447-457.

This article was processed using the LaTeX macro package with LMAMULT style