ROUGH SETS IN KDD

Andrzej Skowron
Institute of Mathematics
Warsaw University
Banacha 2, 02–095, Warsaw,
POLAND
email: `skowron@mimuw.edu.pl`

Abstract: *In recent years we witness a rapid growth of interest in rough set theory and its applications, worldwide. The theory has been followed by the development of several software systems that implement rough set operations, in particular for solving knowledge discovery and data mining tasks. Rough sets are applied in domains, such as, for instance, medicine, finance, telecommunication, vibration analysis, conflict resolution, intelligent agents, pattern recognition, control theory, signal analysis, process industry, marketing, etc.*

We introduce basic notions and discuss methodologies for analyzing data and surveys some applications. In particular we present applications of rough set methods for feature selection, feature extraction, discovery of patterns and their applications for decomposition of large data tables as well as the relationship of rough sets with association rules. Boolean reasoning is crucial for all the discussed methods.

We also present an overview of some extensions of the classical rough set approach. Among them is rough mereology developed as a tool for synthesis of objects satisfying a given specification in a satisfactory degree. Applications of rough mereology in such areas like granular computing, spatial reasoning and data mining in distributed environment are outlined.

1 Basic rough set approach

We start by presenting the basic notions of classical rough set approach [41] introduced to deal with imprecise or vague concepts.

Information systems

A data set can be represented by a table where each row represents, for instance, an object, a case, or an event. Every column represents an attribute, or an observation, or a property that can be measured for each object; it can also be supplied by a human expert or user. This table is called an *information system.* More formally, it is a pair $\mathcal{A} = (U, A)$ where U is a non-empty finite set of *objects* called the *universe* and A is a non-empty finite set of *attributes* such that $a : U \to V_a$ for every $a \in A$. The set V_a is called the *value set* of a. By $Inf_B(x) = \{(a, a(x)) : a \in B\}$ we denote the *information signature of x with respect to B*, where $B \subseteq A$ and $x \in U$.

Decision systems

In many cases the target of the classification, that is, the family of concepts to be approximated is represented by an additional attribute called decision. Information systems of this kind are called *decision systems.* A decision system is any system of the form $\mathcal{A} = (U, A, d)$, where $d \notin A$ is the *decision attribute* and A is a set of *conditional attributes* or simply *conditions.*

Let $\mathcal{A} = (U, A, d)$ be given and let $V_d = \{v_1, \ldots, v_{r(d)}\}$. Decision d determines a partition $\{X_1, \ldots, X_{r(d)}\}$ of the universe U, where $X_k = \{x \in U : d(x) = v_k\}$ for $1 \le k \le r(d)$. The set X_i is called the *i-th decision class of $\mathcal{A}$.* By $X_d(u)$ we denote the decision class $\{x \in U : d(x) = d(u)\}$, for any $u \in U$.

One can generalize the above definition to a case of decision systems of the form $\mathcal{A} = (U, A, D)$ where the sets $D = \{d_1, ...d_k\}$ of decision attributes and A are assumed to be disjoint. Formally this system can be treated as the decision system $\mathcal{A} = (U, A, d_D)$ where $d_D(x) = (d_1(x), ..., d_k(x))$ for $x \in U$.

The decision tables can be identified with training samples known in Machine Learning and used to induce concept approximations in the process known as supervised learning [28].

Rough set approach allows to precisely define the notion of concept approximation. It is based [41] on the indiscernibility relation between objects defining a partition (or covering) of the universe U of objects. The indiscernibility of objects follows from the fact that they are perceived by means of values of available attributes. Hence some objects having the same (or similar) values of attributes are indiscernible.

Indiscernibility relation

Let $\mathcal{A} = (U, A)$ be an information system, then with any $B \subseteq A$ there is associated an equivalence relation $IND_{\mathcal{A}}(B)$:

$$IND_{\mathcal{A}}(B) = \{(x, x') \in U^2 \, : \, \forall a \in B \; a(x) = a(x')\}$$

$IND_{\mathcal{A}}(B)$ (or, $IND(B)$, for short) is called the B-*indiscernibility relation*, its classes are denoted by $[x]_B$. By X/B we denote the partition of U defined by the indiscernibility relation $IND(B)$.

Now we will discuss what sets of objects can be expressed (defined) by formulas constructed by means of attributes and their values. The simplest formulas, called *descriptors*, are of the form $a = v$ where $a \in A$ and $v \in V_a$. One can consider *generalized descriptors* of the form $a \in S$ where $S \subseteq V_a$. The descriptors can be combined into more complex formulas using propositional connectives. The meaning $\|\varphi\|_{\mathcal{A}}$ in $\mathcal{A}$ of formula φ is defined inductively by

1. if φ is of the form $a = v$ then $\|\varphi\|_{\mathcal{A}} = \{x \in U \, : \, a(x) = v\}$;
2. $\|\varphi \wedge \varphi'\|_{\mathcal{A}} = \|\varphi\|_{\mathcal{A}} \cap \|\varphi'\|_{\mathcal{A}}$; $\|\varphi \vee \varphi'\|_{\mathcal{A}} = \|\varphi\|_{\mathcal{A}} \cup \|\varphi'\|_{\mathcal{A}}$; $\|\neg\varphi\|_{\mathcal{A}} = U - \|\varphi\|_{\mathcal{A}}$.

The above definition can be easily extended to generalized descriptors.

Any set of objects $X \subseteq U$ definable in $\mathcal{A}$ by some formula φ (i.e., X=$\|\varphi\|_{\mathcal{A}}$) is referred to as a *crisp* (exact) set – otherwise the set is *rough* (*inexact, vague*). Vague concepts may be only approximated by crisp concepts; these approximations are defined now [41].

Lower and upper approximation of sets, boundary regions

Let $\mathcal{A} = (U, A)$ be an information system and let $B \subseteq A$ and $X \subseteq U$. We can approximate X using only the information contained in B by constructing the B-*lower* and B-*upper approximations of* X, denoted $\underline{B}X$ and $\overline{B}X$ respectively, where $\underline{B}X = \{x : [x]_B \subseteq X\}$ and $\overline{B}X = \{x : [x]_B \cap X \neq \emptyset\}$.

The lower approximation corresponds to certain rules while the upper approximation to possible rules (rules with confidence greater than 0) for X. The B-lower approximation of X is the set of all objects which can be with certainty classified to X using attributes from B. The set $U - \overline{B}X$ is called the B-*outside region of* X and consists of those objects which can be with certainty classified as not belonging to X using attributes from B. The set $BN_B(X) = \overline{B}X - \underline{B}X$ is called the B-*boundary region of* X thus consisting of those objects that on the basis of the attributes from B cannot be unambiguously classified into X. A set is said to be *rough* (respectively *crisp*) if the boundary region is non-empty (respectively empty). Consequently each rough set has boundary-line cases, i.e., objects which cannot be with certainty classified neither as members of the set nor of its complement. Obviously crisp sets have no boundary-line elements at all. That means that boundary-line cases cannot be properly classified by employing the available knowledge. The size of the boundary region can be used as a measure of the quality of set approximation (in U).

It can be easily seen that the lower and upper approximations of a set are, respectively, the interior and the closure of this set in the topology generated by the indiscernibility relation.

One can consider weaker indiscernibility relations defined by tolerance relations defining coverings of the universe of objects by tolerance (similarity) classes. An extension of rough set approach based on tolerance relations has been used for pattern extraction and concept approximation (see, e.g., [60], [64], [35], [32]).

Quality measures of concept approximation and measures of inclusion and closeness of concepts

We now present some examples of measures of quality approximation as well as of inclusion and closeness (approximate equivalence). These notions are instrumental in evaluating the strength of rules and closeness of concepts as well as being applicable in determining plausible reasoning schemes [46], [51]. Important role is also played by entropy measures (see e.g., [11]).

Let us consider first an example of a quality measure of approximations.

Accuracy of approximation. A rough set X can be characterized numerically by the following coefficient

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|},$$

called the *accuracy of approximation*, where $|X|$ denotes the cardinality of $X \neq \emptyset$ and B is a set of attributes. Obviously $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, X is *crisp* with respect to B (X is *exact* with respect to B), and otherwise, if $\alpha_B(X) < 1$, X is *rough* with respect to B (X is *vague* with respect to B).

Rough membership function. In classical set theory either an element belongs to a set or it does not. The corresponding membership function is the characteristic function of the set, i.e., the function takes values 1 and 0, respectively. In the case of rough sets the notion of membership is different. The *rough membership function* quantifies the degree of relative overlap between the set X and the equivalence class to which x belongs. It is defined as follows:

$$\mu_X^B(x): U \to [0,1] \text{ and } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}.$$

The rough membership function can be interpreted as a frequency–based estimate of $\Pr(y \in X \mid u)$, the conditional probability that object y belongs to set X, given the information signature $u = Inf_B(x)$ of object x with respect to attributes B. The value $\mu_X^B(x)$ measures degree of inclusion of $\{y \in U : Inf_B(x) = Inf_B(y)\}$ in X.

Positive region and its measure. If $X_1, \ldots, X_{r(d)}$ are decision classes of $\mathcal{A}$, then the set $\underline{B}X_1 \cup \ldots \cup \underline{B}X_{r(d)}$ is called the *B–positive region of* $\mathcal{A}$ and is denoted by $POS_B(d)$. The number $|POS_B(d)|/|U|$ measures a degree of inclusion of the partition defined by attributes from B into the partition defined by the decision.

Dependencies in a degree. Another important issue in data analysis is discovering dependencies among attributes. Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C. In other words, D depends totally on C, if there exists a functional dependency between values of D and C. Dependency can be formally defined as follows.

Let D and C be subsets of A. We will say that D *depends on* C in a *degree* k $(0 \leq k \leq 1)$, denoted $C \Rightarrow_k D$, if

$$k = \gamma(C,D) = \frac{|POS_C(D)|}{|U|},$$

where $POS_C(D) = POS_C(d_D)$.

Obviously

$$\gamma(C,D) = \sum_{X \in U/D} \frac{|\underline{C}(X)|}{|U|}.$$

If $k = 1$ we say that D *depends totally* on C, and if $k < 1$, we say that D *depends partially* (to a *degree* k) on C. $\gamma(C,D)$ describes the closeness of the partition U/D and its approximation with respect to conditions from C.

The coefficient k expresses the ratio of all elements of the universe which can be properly classified to blocks of the partition U/D by employing attributes C. It will be called the *degree of the dependency.*

Inclusion and closeness in a degree. Instead of classical exact set inclusion inclusion in a degree is often used in the process of deriving knowledge from data. Well known measure of inclusion of two nonempty sets $X, Y \subseteq U$ is described by $|X \cap Y|/|X|$ [2], [46]; their closeness can be defined by

$$min\left(|X \cap Y|/|X|, |X \cap Y|/|Y|\right).$$

2 Searching for knowledge

We have pointed out that rough set approach has been introduced by Z. Pawlak [41] to deal with vague or imprecise concepts. More generally it is an approach for deriving knowledge from data and for reasoning about knowledge derived from data. Searching for knowledge is usually guided by some constraints [23]. A wide class of such constraints can be expressed in rough set setting or its generalizations (like rough mereology [46], or granular computing [51]). Knowledge derived from data by rough set approach consists of different constructs. Among them are basic for rough set approach constructs, called reducts, different kinds of rules (like decision rules or association rules) dependencies, patterns (templates) or classifiers. The reducts are of special importance because all other constructs can be derived from different kinds of reducts using rough set approach. Searching strategies for reducts are based on Boolean (propositional) reasoning [4] because constraints (e.g. related to discernibility of objects) are expressible by propositional formulas. Moreover, using Boolean reasoning data models with the minimum description length [53], [28] can be induced because they correspond to some constructs of Boolean functions called prime implicants (or their approximations). Searching for knowledge can be performed in the language close to data or in a language with more abstract concepts what is closely related to problems of feature selection and feature extraction in Machine Learning or Pattern Recognition [28]. Let us also mention that data models derived from data by using rough set approach are controlled using statistical test procedures (for more details see, e.g., [11], [10]).

In the paper we present illustrative examples showing how the above outlined general scheme is used for deriving knowledge from data.

Finally, we would like to mention that extensions of rough sets like rough mereology [46] or granular computing [51] have been developed for extracting knowledge and reasoning about knowledge related to more complex data models like those in distributed environment or related to qualitative reasoning (e.g., spatial reasoning [54]).

Now, it will be important to make some remarks on Boolean reasoning because the most methods discussed later are based on generation of reducts using Boolean reasoning.

Boolean reasoning

The combination of rough set approach with Boolean Reasoning [4] has created a powerful methodology allowing to formulate and efficiently solve searching problems for different kinds of reducts and their approximations.

The idea of Boolean reasoning is based on the construction for a given problem P of a corresponding Boolean function f_P with the following property: the solutions for the problem P can be recovered from prime implicants of f_P. An implicant of a Boolean function f is any conjunction of literals (variables or their negations) such that if the values of these literals are true under an arbitrary valuation v of variables then the value of the function f under v is also true. A prime implicant is a minimal implicant.

Searching strategies for data models under a given partition of objects are based, using rough set approach, on discernibility and Boolean reasoning (see e.g., [35], [32],[58], [64], [65], [48], [49]). This process covers also tuning of parameters like thresholds used to extract relevant partition (or covering), to measure the degree of inclusion (or closeness) of sets, or the parameters measuring the quality of approximation.

It is necessary to deal with Boolean functions of large size to solve real-life problems. However, a successful methodology based on the discernibility of objects and Boolean reasoning has been developed for computing of many important for applications constructs like reducts and their approximations, decision rules, association rules, discretization of real value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data as well as conflict resolution or negotiation. Reducts are also basic tools for extracting from data functional dependencies or functional dependencies in a degree (for references see the papers and bibliography in [58], [37], [48], [49]).

Most of the problems related to generation of the above mentioned constructs are of high computational complexity (i.e., they are NP-complete or NP-hard). This is also showing that most of the problems related to, e.g., feature selection, pattern extraction from data have intrinsic high computational complexity. However, using developed methodology based on discernibility and Boolean reasoning it was possible to discover efficient heuristics returning suboptimal solutions of the problems.

The reported results of experiments on many data sets are very promising. They show very good quality of solutions (expressed by the classification quality of unseen objects and time necessary for solution construction) generated by the heuristics in comparison with other methods reported in literature. Moreover, for large data sets the decomposition methods based on patterns called templates have been developed (see e.g., [35], [32]) as well as a method to deal with large relational databases (see e.g., [31]). The first one is based on decomposition of large data into regular sub-domains which are of size feasible for developed methods. We will discuss this method later. The second, (see e.g., [31]) has shown that Boolean reasoning methodology can be extended to large relational data bases. The main idea is based on observation that relevant Boolean variables for very large formula (corresponding to analyzed relational data base) can be discovered by analyzing some statistical information. This statistical information can be efficiently extracted from large data bases.

Another interesting statistical approach is based on different sampling strategies. Samples are analyzed using the developed strategies and stable constructs for sufficiently large number of samples are considered as relevant for the whole table. This approach has been successfully used for generating different kinds of so called dynamic reducts (see e.g., [3]). It has been used for example for generation of so called dynamic decision rules. Experiments on different data sets have proved that these methods are promising for large data sets.

Our approach is strongly related to propositional reasoning [55] and further progress in propositional reasoning will bring further progress in developing of the discussed methods. It is important to note that the methodology allows to construct heuristics having a very important *approximation property* which can be formulated as follows: expressions (i.e., implicants) generated by heuristics *close* to prime implicants define approximate solutions for the problem [55].

In the sequel we will discuss in more details different kinds of reducts and their applications for deriving different forms of knowledge from data.

2.1 Reducts in information systems and decision systems

We start from reducts of information systems. Given an $\mathcal{A} = (U, A)$, a *reduct* is a minimal set of attributes $B \subseteq A$ such that $IND_{\mathcal{A}}(B) = IND_{\mathcal{A}}(A)$. In other words, a reduct is a minimal set of attributes from A that preserves the original classification defined by the set A of attributes. Finding a minimal reduct is NP-hard [59]; one can also show that for any m there exists an information system with m attributes having an exponential number of reducts. There exist fortunately good heuristics that compute sufficiently many reducts in an acceptable time.

Let $\mathcal{A}$ be an information system with n objects. The *discernibility matrix* of $\mathcal{A}$ is a symmetric $n \times n$ matrix with entries c_{ij} as given below. Each entry consists of the set of attributes upon which objects x_i and x_j differ.

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \text{ for } i, j = 1, ..., n.$$

A *discernibility function* $f_{\mathcal{A}}$ for an information system $\mathcal{A}$ is a Boolean function of m Boolean variables $a_1^*, ..., a_m^*$ (corresponding to the attributes $a_1, ..., a_m$) defined by

$$f_{\mathcal{A}}(a_1^*, ..., a_m^*) = \bigwedge \left\{ \bigvee c_{ij}^* \mid 1 \leq j \leq i \leq n,\ c_{ij} \neq \emptyset \right\}$$

where $c_{ij}^* = \{a^* \mid a \in c_{ij}\}$. In the sequel we will write a_i instead of a_i^*.

The discernibility function $f_{\mathcal{A}}$ describes constraints which should be preserved if one would like to p-reserve discernibility between all pairs of discernible objects from $\mathcal{A}$. It requires to keep at least one attribute from each non-empty entry of the discernibility matrix, i.e., corresponding to any pair of discernible objects. One can show [59] that the sets of all minimal sets of attributes preserving discernibility between objects, i.e., reducts correspond to prime implicants of the discernibility function $f_{\mathcal{A}}$.

The intersection of all reducts is the so-called *core*.

In general, the decision is not constant on the indiscernibility classes. Let $\mathcal{A} = (U, A, d)$ be a decision system. The *generalized decision in* $\mathcal{A}$ is the function $\partial_A : U \longrightarrow \mathcal{P}(V_d)$ defined by $\partial_A(x) = \{i \mid \exists x' \in U\ x'\, IND(A)\, x \text{ and } d(x') = i\}$. A decision system $\mathcal{A}$ is called *consistent (deterministic)*, if $|\partial_A(x)| = 1$ for any $x \in U$, otherwise $\mathcal{A}$ is *inconsistent (non-deterministic)*. Any set consisting of all objects with the same generalized decision value is called the *generalized decision class.*

It is easy to see that a decision system $\mathcal{A}$ is consistent if, and only if, $POS_A(d) = U$. Moreover, if $\partial_B = \partial_{B'}$, then $POS_B(d) = POS_{B'}(d)$ for any pair of non-empty sets $B, B' \subseteq A$. Hence the definition of a decision-relative reduct: a subset $B \subseteq A$ is a *relative reduct* if it is a minimal set such that $POS_A(d) = POS_B(d)$. Decision-relative reducts may be found from a discernibility matrix: $M^d(\mathcal{A}) = (c_{ij}^d)$ assuming $c_{ij}^d = c_{ij} - \{d\}$ if $(|\partial_A(x_i)| = 1$ or $|\partial_A(x_j)| = 1)$ and $\partial_A(x_i) \neq \partial_A(x_j)$, $c_{ij}^d = \emptyset$, otherwise. Matrix $M^d(\mathcal{A})$ is called *the decision-relative discernibility matrix of* $\mathcal{A}$. Construction of *the decision-relative discernibility function* from this matrix follows the construction of the discernibility function from the discernibility matrix. One can observe [59] that the set of *prime implicants* of $f_M^d(\mathcal{A})$ defines the set of all *decision-relative reducts* of $\mathcal{A}$.

In some applications, instead of reducts we prefer to use their approximations called α-reducts, where $\alpha \in [0, 1]$ is a real parameter. For a given information system $\mathcal{A} = (U, A)$ the set of attributes $B \subseteq A$ is called α-reduct if B has nonempty intersection with at least $\alpha \cdot 100\%$ of nonempty sets $c_{i,j}$ of the discernibility matrix of $\mathcal{A}$.

2.2 Reducts and Boolean reasoning: Examples of applications

We will present examples showing how the rough set methods in combination with Boolean reasoning can be used for solving several KDD problems. A crucial for our approach are rough set constructs called reducts. They are (prime) implicants of suitably chosen Boolean functions expressing discernibility conditions which should be preserved during reduction.

Feature selection

Selection of relevant features is an important problem and has been extensively studied in Machine Learning and Pattern Recognition (see e.g., [28]). It is also a very active research area in the rough set community.

One of the first ideas [41] was to consider the *core* of the reduct set of the information system $\mathcal{A}$ as the source of relevant features. One can observe that relevant feature sets (in a sense used by the machine learning community) can be interpreted in most cases as the decision-relative reducts of decision systems obtained by adding appropriately constructed decisions to a given information system.

Another approach is related to dynamic reducts (for references see e.g., [48]). The attributes are considered relevant if they belong to dynamic reducts with a sufficiently high stability coefficient, i.e., they appear with sufficiently high frequency in random samples of a given information system. Several experiments (see [48]) show that the set of decision rules based on such attributes is much smaller than the set of all decision rules. At the same time the quality of classification of new objects increases or does not change if one only considers rules constructed over such relevant features.

The idea of attribute reduction can be generalized by introducing a concept of *significance of attributes* which enables to evaluate attributes not only in the two-valued scale *dispensable – indispensable* but also in the multi-value case by assigning to an attribute a real number from the interval [0,1] that expresses the importance of an attribute in the information table.

Significance of an attribute can be evaluated by measuring the effect of removing the attribute from an information table.

Let C and D be sets of condition and decision attributes, respectively, and let $a \in C$ be a condition attribute. It was shown previously that the number $\gamma(C, D)$ expresses the degree of dependency between attributes C and D, or the accuracy of the approximation of U/D by C. It may be now checked how the coefficient $\gamma(C, D)$ changes when attribute a is removed. In other words, what is the difference between $\gamma(C, D)$ and $\gamma((C - \{a\}, D)$. The difference is normalized and the significance of attribute a is defined by

$$\sigma_{(C,D)}(a) = \frac{(\gamma(C,D) - \gamma(C - \{a\}, D))}{\gamma(C,D)} =$$

$$= 1 - \frac{\gamma(C - \{a\}, D)}{\gamma(C,D)}.$$

Coefficient $\sigma_{C,D}(a)$ can be understood as a classification error which occurs when attribute a is dropped. The significance coefficient can be extended to sets of attributes as follows:

$$\sigma_{(C,D)}(B) = \frac{(\gamma(C,D) - \gamma(C - B, D))}{\gamma(C,D)} =$$

$$= 1 - \frac{\gamma(C - B, D)}{\gamma(C,D)}.$$

Another possibility is to consider as relevant the features that come from approximate reducts of sufficiently high quality.

Any subset B of C can be treated as an *approximate reduct* of C and the number

$$\varepsilon_{(C,D)}(B) = \frac{(\gamma(C,D) - \gamma(B,D))}{\gamma(C,D)} = 1 - \frac{\gamma(B,D)}{\gamma(C,D)},$$

is called an *error of reduct approximation.* It expresses how exactly the set of attributes B approximates the set of condition attributes C with respect to determining D.

Several other methods of reduct approximation based on measures different from positive region have been developed. All experiments confirm the hypothesis that by tuning the level of approximation the classification quality of new objects may be increased in most cases. It is important to note that it is once again possible to use Boolean reasoning to compute the different types of reducts and to extract from them relevant approximations.

Feature extraction

Non-categorical attributes must be discretized in a pre-processing step. The discretization step determines how coarsely we want to view the world. Discretization is a step that is not specific to the rough set approach. A majority of rule or tree induction algorithms require it in order to perform well. The search for appropriate cut-off points can be reduced to finding some minimal Boolean expressions called prime implicants.

Discretization can be treated as a searching for more coarser partitions of the universe still relevant for inducing concept description of high quality. We will also show that this basic problem can be reduced to computing of basic constructs of rough sets, namely reducts of some systems. Hence it follows that we can estimate the computational complexity of the discretization problems. Moreover, heuristics for computing reducts and prime implicants can be used here. The general heuristics can be modified to more optimal ones using konwledge about the problem e.g. natural order of the set of reals, etc. The discretization is only an illustrative example of many other problems with the same property.

The rough set community have been committed to constructing efficient algorithms for (new) feature extraction. Rough set methods combined with Boolean reasoning [4] lead to several successful approaches to feature extraction. The most successful methods are:

$\mathbf{A}$	a	b	d
u_1	0.8	2	1
u_2	1	0.5	0
u_3	1.3	3	0
u_4	1.4	1	1
u_5	1.4	2	0
u_6	1.6	3	1
u_7	1.3	1	1

(a)

$\Rightarrow$

$\mathbf{A}^{\mathbf{P}}$	a^P	b^P	d
u_1	0	2	1
u_2	1	0	0
u_3	1	2	0
u_4	1	1	1
u_5	1	2	0
u_6	2	2	1
u_7	1	1	1

(b)

Table 1: The discretization process: (a) The original decision system $\mathcal{A}$. (b) The **P**-discretization of $\mathcal{A}$, where $\mathbf{P} = \{(a, 0.9), (a, 1.5), (b, 0.75), (b, 1.5)\}$

- discretization techniques,
- methods of partitioning of nominal attribute value sets and
- combinations of the above methods.

Searching for new features expressed by multi-modal formulae can be mentioned here. Structural objects can be interpreted as models (so called Kripke models) of such formulas and the problem of searching for relevant features reduces to construction of multimodal formulas expressing properties of the structural objects discerning objects or sets of objects [36].

For more details the reader is referred to the bibliography in [49].

The reported results show that discretization problems and symbolic value partition problems are of high computational complexity (i.e. NP-complete or NP-hard) which clearly justifies the importance of designing efficient heuristics. The idea of discretization is illustrated with a simple example.

Example 2.1 Let us consider a (consistent) decision system (see Tab. 1(a)) with two conditional attributes a and b and seven objects $u_1, ..., u_7$. The values of the attributes of these objects and the values of the decision d are presented in Tab. 1.

The sets of possible values of a and b are defined by:

$$V_a = [0, 2) \, ; V_b = [0, 4) \, .$$

The sets of values of a and b for objects from U are respectively given by:

$$\begin{array}{rcl} a(U) & = & \{0.8, 1, 1.3, 1.4, 1.6\} \text{ and} \\ b(U) & = & \{0.5, 1, 2, 3\} \end{array}$$

□

A discretization process produces a partition of the value sets of the conditional attributes into intervals. The partition is done so that a consistent decision system is obtained from a given consistent decision system by a substitution of any object's original value in $\mathcal{A}$ by the (unique) name of the interval(s) in which it is contained. In this way the size of the value sets of the attributes may be reduced. If a given decision system is not consistent one can transform it to the consistent decision system by taking the generalized decision instead of the original one. Next it is possible to apply the above method. It will return cuts with the following property: regions bounded by them consist of objects with the same generalized decision. Certainly, one can consider also *soft (impure)* cuts and induce the relevant cuts on their basis (see the bibliography in [48]).

Example 2.2 The following intervals are obtained in our example system:

$$[0.8, 1); \; [1, 1.3); \; [1.3, 1.4); \; [1.4, 1.6) \text{ for } a);$$

$$[0.5, 1); \; [1, 2); \; [2, 3) \text{ for } b).$$

The idea of cuts can be introduced now. Cuts are pairs (a, c) where $c \in V_a$. Our considerations are restricted to cuts defined by the middle points of the above intervals. In our example the following cuts are obtained:

$$(a, 0.9); \; (a, 1.15); \; (a, 1.35); \; (a, 1.5);$$

$$(b, 0.75); \; (b, 1.5); \; (b, 2.5).$$

Any cut defines a new conditional attribute with binary values. For example, the attribute corresponding to the cut $(a, 1.2)$ is equal to 0 if $a(x) < 1.2$; otherwise it is equal to 1. □

Any set P of cuts defines a new conditional attribute a_P for any a. Given a partition of the value set of a by cuts from P one can put the unique names for the elements of these partition.

Example 2.3 Let

$$P = \{(a, 0.9), (a, 1.5), (b, 0.75), (b, 1.5)\}$$

be the set of cuts. These cuts glue together the values of a smaller then 0.9, all the values in interval $[0.9, 1.5)$ and all the values in interval $[1.5, 4)$. A similar construction can be repeated for b. The values of the new attributes a_P and b_P are shown in Tab. 1 (b). □

The next natural step is to construct a set of cuts with a minimal number of elements. This may be done using Boolean reasoning.

Let $\mathcal{A} = (U, A, d)$ be a decision system where $U = \{x_1, x_2, \ldots, x_n\}, A = \{a_1, \ldots, a_k\}$ and $d : U \longrightarrow \{1, \ldots, r\}$. We assume $V_a = [l_a, r_a) \subset \Re$ to be a real interval for any $a \in A$ and $\mathcal{A}$ to be a consistent decision system. Any pair (a, c) where $a \in A$ and $c \in \Re$ will be called a *cut on* V_a. Let $\mathbf{P}_a = \{[c_0^a, c_1^a), [c_1^a, c_2^a), \ldots, [c_{k_a}^a, c_{k_a+1}^a)\}$ be a partition of V_a (for $a \in A$) into subintervals for some integer k_a, where $l_a = c_0^a < c_1^a < c_2^a < \ldots < c_{k_a}^a < c_{k_a+1}^a = r_a$ and $V_a = [c_0^a, c_1^a) \cup [c_1^a, c_2^a) \cup \ldots \cup [c_{k_a}^a, c_{k_a+1}^a)$. It follows that any partition $\mathbf{P}_a$ is uniquely defined and is often identified with the set of cuts

$$\{(a, c_1^a), (a, c_2^a), \ldots, (a, c_{k_a}^a)\} \subset A \times \Re.$$

Given $\mathcal{A} = (U, A, d)$ any set of cuts $\mathbf{P} = \bigcup_{a \in A} \mathbf{P}_a$ defines a new decision system $\mathcal{A}^{\mathbf{P}} = (U, A^{\mathbf{P}}, d)$ called $\mathbf{P}$-*discretization of* $\mathcal{A}$, where $A^{\mathbf{P}} = \{a^{\mathbf{P}} : a \in A\}$ and $a^{\mathbf{P}}(x) = i \Leftrightarrow a(x) \in [c_i^a, c_{i+1}^a)$ for $x \in U$ and $i \in \{0, .., k_a\}$.

Two sets of cuts $\mathbf{P}'$ and $\mathbf{P}$ are equivalent, written $\mathbf{P}' \equiv_{\mathbf{A}} \mathbf{P}$, iff $\mathcal{A}^{\mathbf{P}} = \mathcal{A}^{\mathbf{P}'}$. The equivalence relation $\equiv_{\mathcal{A}}$ has a finite number of equivalence classes. Equivalent families of partitions will be not discerned in the sequel.

The set of cuts $\mathbf{P}$ is called $\mathcal{A}$-*consistent* if $\partial_A = \partial_{A^{\mathbf{P}}}$, where ∂_A and $\partial_{A^{\mathbf{P}}}$ are generalized decisions of $\mathcal{A}$ and $\mathcal{A}^{\mathbf{P}}$, respectively. An $\mathcal{A}$-consistent set of cuts $\mathbf{P}^{irr}$ is $\mathcal{A}$-*irreducible* if $\mathbf{P}$ is not $\mathcal{A}$-consistent for any $\mathbf{P} \subset \mathbf{P}^{irr}$. The $\mathcal{A}$-consistent set of cuts $\mathbf{P}^{opt}$ is $\mathcal{A}$-*optimal* if $card\,(\mathbf{P}^{opt}) \leq card\,(\mathbf{P})$ for any $\mathcal{A}$-consistent set of cuts $\mathbf{P}$.

It can be shown that the decision problem of checking if for a given decision system $\mathcal{A}$ and an integer k there exists an irreducible set of cuts $\mathbf{P}$ in $\mathcal{A}$ such that $card(\mathbf{P}) < k$ is NP-complete. The problem of searching for an optimal set of cuts $\mathbf{P}$ in a given decision system $\mathcal{A}$ is NP-hard.

Despite these complexity bounds it is possible to devise efficient heuristics that return semi-minimal sets of cuts. The simplest huristics is based on Johnson's strategy. The strategy is first to look for a cut discerning a maximal number of object pairs and then to eliminate all already discerned object pairs. This procedure is repeated until all object pairs to be discerned are discerned. It is interesting to note that this heuristics can be realized by computing the minimal relative reduct of the corresponding decision system. The *"MD heuristic"* is analogous to Johnson's approximation algorithm. It may be formulated as follows:

ALGORITHM: MD-heuristics (A semi-optimal family of partitions)

S1. *Construct table* $\mathcal{A}^* = (U^*, A^*, d)$ *from* $\mathcal{A} = (U, A)$ *where* U^* *is the set of pairs* (x, y) *of objects to be discerned by* d *and* A^* *consists of attribute* c^* *for any cut* c *and* c^* *is defined by* $c^*(x, y) = 1$ *if and only if* c *discerns* x *and* y (*i.e.,* x, y *are in different half-spaces defined by* c); *set* $\mathcal{B}= \mathcal{A}^*$;

S2. *Choose a column from* $\mathcal{B}$ *with the maximal number of occurrences of 1's;*

S3. *Delete from* $\mathcal{B}$ *the column chosen in Step 2 (S2) and all rows marked with 1 in this column;*

S4. *If* $\mathcal{B}$ *is non-empty then go to Step 2 (S2) else Stop.*

This algorithm searches for a cut which discerns the largest number of pairs of objects (MD-heuristic). Then the cut c is moved from A^* to the resulting set of cuts $\mathbf{P}$; and all pairs of objects discerned by c are removed from U^*. The algorithm continues until U^* becomes empty.

Let n be the number of objects and let k be the number of attributes of decision system $\mathcal{A}$. The following inequalities hold: $card\,(A^*) \leq (n-1)\,k$ and $card\,(U^*) \leq \frac{n(n-1)}{2}$. It is easy to observe that for any cut $c \in A^*$ $O\left(n^2\right)$ steps are required in order to find the number of all pairs of objects discerned by c. A straightforward realization of the algorithm therefore requires $O\left(kn^2\right)$ of memory space and $O(kn^3)$ steps in order to determine one cut *cut*. This approach is clearly impractical. However, it is possible to observe that in the process of searching for the set of pairs of objects discerned by currently analyzed cut from an increasing sequence of cuts one can use information about such set of pairs of objects computed for the previously considered cut. The MD-heuristic using this observation [30] determines the best cut (for a given attribute) in $O\,(kn)$ steps using $O\,(kn)$ space only. This heuristic is reported to be very efficient with respect to the time necessary for decision rules generation as well as with respect to the quality of unseen object classification.

Let us observe that in the considered case of discretization the new features are of the form $a \in V$, where $V \subseteq V_a$ and V_a is the set of the values of attribute a.

We report some results of experiments on data sets using this heuristic. We would like to comment for example on the result of classification received by application of this heuristic to Shuttle data (Table 3). The result concerning classification quality is the same as

Names	**Nr of class.**	**Train. table**	**Test. table**	**Best results**
Australian	2	690×14	CV5	85.65%
Glass	7	214×9	CV5	69.62%
Heart	2	270×13	CV5	82.59%
Iris	3	150×4	CV5	96.00%
Vehicle	4	846×19	CV5	69.86%
Diabetes	2	768×8	CV5	76.04%
SatImage	6	4436×36	2000	90.06%
Shuttle	6	43500×7	14500	99.99%

Table 2: Data tables stored in the UC Irvine Repository

Data tables	**Diagonal cuts**		**Hyperplanes**	
	#cuts	quality	#cuts	quality
Australian	18	79.71%	16	82.46%
Glass	14±1	67.89%	12	70.06%
Heart	11±1	79.25%	11±1	80.37%
Iris	7±2	92.70%	6±2	96.7%
Vehicle	25	59.70%	20±2	64.42%
Diabetes	20	74.24%	19	76.08%
SatImage	47	81.73%	43	82.90%
Shuttle	15	99.99%	15	99.99%

Table 3: Results of experiments on Machine Learning data.

the best result reported in [27] but the time is of order better than for the best result from [27]. In the table we present also the results of experiments with heuristic searching for features defined by oblique hyperplanes. This heuristic has been developed using genetic algorithm allowing to tune the position of hyperplane to the optimal one [30]. In this way one can implement propositional reasoning using some background knowledge about the problem.

In experiments we have chosen several data tables with real value attributes from the U.C. Irvine repository. For some tables, taking into account the small number of their objects, we have adopted the approach based on five-fold cross-validation ($CV-5$). The obtained results (Table 3) can be compared with those reported in [9, 27] (Table 2). For predicting decisions on new cases we apply only decision rules generated either by the decision tree (using hyperplanes) or by rules generated in parallel with discretization.

For some tables the classification quality of our algorithm is better than that of the C4.5 or Naive -Bayes induction algorithms [52] even when used with different discretization methods [9, 27, 15].

Comparing this method with the other methods reported in [27], we can conclude that our algorithms have the shortest runtime and a good overall classification quality (in many cases our results were the best in comparison to many other methods reported in literature).

We would like to stress that inducing of the minimal number of the relevant cuts is equivalent to computing of the minimal reduct of decision system constructed from the discussed above system $\mathcal{A}^*$ [30]. This in turn, as we have shown, is equivalent to the problem of computing of minimal prime implicants of Boolean functions. This is only illustration of a wide class of basic problems of Machine Learning, Pattern Recognition and KDD which can be reduced to problems of relevant reduct computation.

Our next illustrative example concerns symbolic (nominal, qualitative) attribute value grouping. We also present some experimental results of heuristics based on the developed methods in case of mixed nominal and numeric attributes.

In case of symbolic value attribute (i.e., without preassumed order on values of given attributes) the problem of searching for new features of the form $a \in V$ is, in a sense, from practical point of view more complicated than the for real value attributes. However, it is possible to develop efficient heuristics for this case using Boolean reasoning.

Let $\mathcal{A} = (U, A, d)$ be a decision table. Any function $P_a : V_a \to \{1, \ldots, m_a\}$ (where $m_a \leq |V_a|$) is called *a partition of* V_a. The *rank of* P_{a_i} is the value $rank\left(P_{a_i}\right) = \left|P_{a_i}\left(V_{a_i}\right)\right|$. The family of partitions $\{P_a\}_{a\in B}$ is *consistent with* B ($B - consistent$) iff the condition $[(u,u') \notin IND(B)$ and $d(u) \neq d(u')$ implies $\exists_{a\in B}[P_a(a(u)) \neq P_a(a(u'))]]$ holds for any $(u,u') \in U$. It means that if two objects u, u' are discerned by B and d, then they must be discerned by partition attributes defined by $\{P_a\}_{a\in B}$. We consider the following optimization problem

PARTITION PROBLEM: SYMBOLIC VALUE PARTITION PROBLEM:

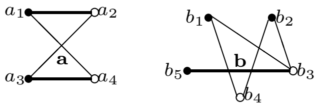
Given a decision table $\mathcal{A} = (U, A, d)$ and a set of attributes $B \subseteq A$, search for the minimal $B-$ *consistent* family of partitions (i.e., such $B-$ *consistent* family $\{P_a\}_{a\in B}$ that $\sum_{a\in B} rank\left(P_a\right)$ is minimal).

To discern between pairs of objects we will use new binary features $a_v^{v'}$ (for $v \neq v'$) defined by $a_v^{v'}(x,y) = 1$ iff $a(x) = v \neq v' = a(y)$. One can apply the Johnson heuristic for the new matrix with these attributes to

$\mathcal{A}$	**a**	**b**	**d**
u_1	a_1	b_1	0
u_2	a_1	b_2	0
u_3	a_2	b_3	0
u_4	a_3	b_1	0
u_5	a_1	b_4	1
u_6	a_2	b_2	1
u_7	a_2	b_1	1
u_8	a_4	b_2	1
u_9	a_3	b_4	1
u_{10}	a_2	b_5	1

$\mathcal{M}(\mathcal{A})$	u_1	u_2	u_3	u_4
u_5	$\mathbf{b}_{b_4}^{b_1}$	$\mathbf{b}_{b_4}^{b_2}$	$\mathbf{a}_{a_2}^{a_1}$, $\mathbf{b}_{b_4}^{b_3}$	$\mathbf{a}_{a_3}^{a_1}$, $\mathbf{b}_{b_4}^{b_1}$
u_6	$\mathbf{a}_{a_2}^{a_1}$, $\mathbf{b}_{b_2}^{b_1}$	$\mathbf{a}_{a_2}^{a_1}$	$\mathbf{b}_{b_3}^{b_2}$	$\mathbf{a}_{a_3}^{a_2}$, $\mathbf{b}_{b_2}^{b_1}$
u_7	$\mathbf{a}_{a_2}^{a_1}$	$\mathbf{a}_{a_2}^{a_1}$, $\mathbf{b}_{b_2}^{b_1}$	$\mathbf{b}_{b_3}^{b_1}$	$\mathbf{a}_{a_3}^{a_2}$
u_8	$\mathbf{a}_{a_4}^{a_1}$, $\mathbf{b}_{b_2}^{b_1}$	$\mathbf{a}_{a_4}^{a_1}$	$\mathbf{a}_{a_4}^{a_2}$, $\mathbf{b}_{b_3}^{b_2}$	$\mathbf{a}_{a_4}^{a_3}$, $\mathbf{b}_{b_2}^{b_1}$
u_9	$\mathbf{a}_{a_3}^{a_1}$, $\mathbf{b}_{b_4}^{b_1}$	$\mathbf{a}_{a_3}^{a_1}$, $\mathbf{b}_{b_4}^{b_2}$	$\mathbf{a}_{a_3}^{a_2}$, $\mathbf{b}_{b_4}^{b_3}$	$\mathbf{b}_{b_4}^{b_1}$
u_{10}	$\mathbf{a}_{a_2}^{a_1}$, $\mathbf{b}_{b_5}^{b_1}$	$\mathbf{a}_{a_2}^{a_1}$, $\mathbf{b}_{b_5}^{b_2}$	$\mathbf{b}_{b_5}^{b_3}$	$\mathbf{a}_{a_3}^{a_2}$, $\mathbf{b}_{b_5}^{b_1}$

Figure 1: The decision table and the discernibility matrix



$\Rightarrow$

$\mathbf{a}^{P_a}$	$\mathbf{b}^{P_b}$	**d**
1	**1**	0
2	**2**	0
1	**2**	1
2	**1**	1

Figure 2: Coloring of attribute value graphs and the reduced table.

search for minimal set of new attributes that discerns all pairs of objects from different decision classes. After extracting of these sets, for each attribute a_i we construct a graph $\Gamma_a = \langle V_a, E_a \rangle$ where E_a is defined as the set of all new attributes (propositional variables) found for the attribute a. Any vertex coloring of Γ_a defines a partition of V_a. The colorability problem is solvable in polynomial time for $k = 2$, but remains NP-complete for all $k \geq 3$. But, similarly to discretization, one can apply some efficient heuristic searching for optimal partition.

Let us consider an example of decision table presented in Figure 1 and (a reduced form) of its discernibility matrix (Figure 1).

From the Boolean function $f_{\mathcal{A}}$ with Boolean variables of the form $\mathbf{a}_{v_1}^{v_2}$ one can find the shortest prime implicant: $\mathbf{a}_{a_2}^{a_1} \wedge \mathbf{a}_{a_3}^{a_2} \wedge \mathbf{a}_{a_4}^{a_1} \wedge \mathbf{a}_{a_4}^{a_3} \wedge \mathbf{b}_{b_4}^{b_1} \wedge \mathbf{b}_{b_4}^{b_2} \wedge \mathbf{b}_{b_3}^{b_2} \wedge \mathbf{b}_{b_3}^{b_1} \wedge \mathbf{b}_{b_5}^{b_3}$ which can be represented by graphs (see Figure 2).

We can color vertices of those graphs as it is shown in Figure 2. The colors are corresponding to the partitions:

$$\begin{aligned} P_{\mathbf{a}}(a_1) &= P_{\mathbf{a}}(a_3) = 1; \\ P_{\mathbf{a}}(a_2) &= P_{\mathbf{a}}(a_4) = 2 \\ P_{\mathbf{b}}(b_1) &= P_{\mathbf{b}}(b_2) = P_{\mathbf{b}}(b_5) = 1; \\ P_{\mathbf{b}}(b_3) &= P_{\mathbf{b}}(b_4) = 2. \end{aligned}$$

Names of Tables	Classification accuracies			
	S-ID3	C4.5	MD	MD-G
Australian	78.26	85.36	83.69	84.49
Breast (L)	62.07	71.00	69.95	69.95
Diabetes	66.23	70.84	71.09	76.17
Glass	62.79	65.89	66.41	69.79
Heart	77.78	77.04	77.04	81.11
Iris	96.67	94.67	95.33	96.67
Lympho	73.33	77.01	71.93	82.02
Monk-1	81.25	75.70	100	93.05
Monk-2	69.91	65.00	99.07	99.07
Monk-3	90.28	97.20	93.51	94.00
Soybean	100	95.56	100	100
TicTacToe	84.38	84.02	97.7	97.70
Average	78.58	79.94	85.48	87.00

Table 4: Quality comparison of various decision tree methods. Abbreviations: MD: MD-heuristic; MD-G: MD-heuristic with symbolic value partition

At the same time one can construct the new decision table (Figure 2).

One can extend the presented approach (see e.g., [33]) to the case when in a given decision system nominal as well as numeric attributes appear. The received heuristics are of very good quality. Experiments for classification methods (see [33]) have been carried over decision systems using two techniques called *"train-and-test"* and *"n-fold-cross-validation"*. In Table 4 some results of experiments obtained by testing the proposed methods: MD (using only discretization based on MD-heurisctic with Johnson approximation strategy [30], [58]) and MD-G (using discretization and symbolic value grouping [32], [58]) for classification quality on some data tables from the "UC Irvine repository" are shown. The results reported in [12] are summarized in columns labeled by S-ID3 and C4.5 in Table 4). Let us note that the heuristics MD and MD-G are also very efficient with respect to the time complexity.

Decision rules

Reducts serve the purpose of inducing *minimal* decision rules. Any such rule contains the minimal number of descriptors in the conditional part so that their conjunction defines the largest subset of a generalized decision class (decision class, if the decision table is deterministic). Hence, information included in conditional part of any minimal rule is sufficient for prediction of the generalized decision value for all objects satisfying this part. The conditional parts of minimal rules define largest object sets relevant for generalized decision classes approximation. It turns out that the conditional parts of minimal rules can be computed (by using Boolean reasoning) as so called reducts relative to objects or local reducts (see e.g., [57], [3]). Once the reducts have been computed, the conditional parts of rules are easily constructed by laying the reducts over the original decision system and reading off the values. In the discussed case the generalized decision value is preserved during the reduction. One can consider stronger constraints which should be preserved. For example, in [62] the constraints are described by probability distributions corresponding to information signatures of objects. Again the same methodology can be used to compute the reducts corresponding to these constraints.

The main challenge in inducing rules from decision systems lies in determining which attributes should be included in the conditional part of the rule. Using the outlined above strategy first the minimal rules are computed. Their conditional parts describe largest object sets (definable by conjunctions of descriptors) with the same generalized decision value in a given decision system. Hence, they create the largest sets still relevant for defining the decision classes (or sets of decision classes when the decision system is inconsistent). Although such minimal decision rules can be computed, this approach can result in set of rules of not satisfactory classification quality. Such detailed rules will be overfit and they will poorly classify unseen cases. Shorter rules should rather be synthesized. Although they will not be perfect on the known cases there is a good chance that they will be of high quality when classifying new cases. They can be constructed by computing approximations of the above mentioned reducts. Approximations of reducts received by drooping some descriptors from the conditional parts of minimal rules define larger sets, not purely included in decision classes but included in a satisfactory degree. It means that these shorter descriptions can be more relevant for decision class (concept) approximation than the exact reducts. Hence, e.g., one can expect that when by dropping the descriptor from the conditional part we receive the description of the object set almost included in the approximated decision class than this descriptor is a good candidate for dropping.

Several other strategies have been implemented. Methods of boundary region thinning [72] are based, e.g., on the idea that sets of objects included in decision classes in satisfactory degree can be treated as parts of the lower approximations of decision classes. Hence the lower approximations of decision classes are enlarged and decision rules generated for them are usually stronger (e.g., they are supported by more examples). The degree of inclusion is tuned experimentally to achieve, e.g., high classification quality of new cases. One can also adopt an idea of dynamic reducts for decision rule generation.

For estimation of the quality of decision classes approximation global measures based on the positive region [57] or entropy [11] are used.

When a set of rules has been induced from a decision system containing a set of training examples, they can be used to classify new objects. However, to resolve conflict between different decision rules recognizing new objects one should develop strategies for resolving conflicts between them when they are voting for different decisions (see the bibliography in [48] and [49]). Recently [66], it has been shown that rough set methods can be used to learn from data the strategy for conflict resolving between decision rules when they are classifying new objects contrary to existing methods using some fixed strategies.

α-reducts and association rules

In this section we discuss a relationship between association rules [2] and approximations of reducts being basic constructs of rough sets [57], [58], [34].

We consider formulas called *templates* being conjunction of descriptors. The templates will be denoted by $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$ and descriptors by D with or without subscripts. By $support_{\mathcal{A}}(\mathbf{T})$ is denoted the cardinality of $\|\mathbf{T}\|_{\mathcal{A}}$ and by $confidence_{\mathcal{A}}(\mathbf{P} \to \mathbf{Q})$ is denoted the number $support_{\mathcal{A}}(\mathbf{P} \wedge \mathbf{Q})/support_{\mathcal{A}}(\mathbf{P})$.

The, mentioned above, reduct approximations are descriptions of the object sets matched by templates. They describe these sets in an approximate sense expressed by coefficients called support and confidence.

There are two main steps of many developed association rule generation methods for given information system $\mathcal{A}$ and parameters of support s and confidence c:

1. Extraction from data as many as possible templates $\mathbf{T} = D_1 \wedge D_2 ... \wedge D_k$ such that $support_{\mathcal{A}}(\mathbf{T}) \geq s$ and $support_{\mathcal{A}}(\mathbf{T} \wedge D) < s$ for any descriptor D different from descriptors of $\mathbf{T}$ (i.e., generation of maximal templates among those supported by more than s objects);

2. Searching for a partition $\mathbf{T} = \mathbf{P} \wedge \mathbf{Q}$ for any of generated template $\mathbf{T}$ satisfying the following conditions:

   (a) $support_{\mathcal{A}}(\mathbf{P}) < \frac{support_{\mathcal{A}}(\mathbf{T})}{c}$

(b) **P** has the shortest length among templates satisfying the previous condition.

The second step can be solved using rough set methods and Boolean reasoning approach.

Let $\mathbf{T} = D_1 \wedge D_2 \wedge \ldots \wedge D_m$ be a template with $support_{\mathcal{A}}(\mathbf{T}) \geq s$. For a given confidence threshold $c \in (0;1)$ the decomposition $\mathbf{T} = \mathbf{P} \wedge \mathbf{Q}$ is called c-irreducible if $confidence_{\mathcal{A}}(\mathbf{P} \rightarrow \mathbf{Q}) \geq c$ and for any decomposition $\mathbf{T} = \mathbf{P}' \wedge \mathbf{Q}'$ such that $\mathbf{P}'$ is a subtemplate of $\mathbf{P}$, we have $confidence_{\mathcal{A}}(\mathbf{P}' \rightarrow \mathbf{Q}') < c$.

Now we are going to explain that problem of searching for c-irreducible association rules from the given template is equivalent to the problem of searching for local α-reducts (for some α) from a decision table. The last problem is a well known problem in rough set theory.

Let us define a new decision table $\mathcal{A}|_{\mathbf{T}} = (U, A|_{\mathbf{T}}, d)$ from the original information system $\mathcal{A}$ and the template $\mathbf{T}$ by

1. $A|_{\mathbf{T}} = \{a_{D_1}, a_{D_2}, ..., a_{D_m}\}$ is a set of attributes corresponding to the descriptors of **T** such that $a_{D_i}(u) = \begin{cases} 1 & \text{if the object } u \text{ satisfies } D_i, \\ 0 & \text{otherwise.} \end{cases}$

2. the decision attribute d determines if the object satisfies template **T**, i.e., $d(u) = \begin{cases} 1 & \text{if the object } u \text{ satisfies } \mathbf{T}, \\ 0 & \text{otherwise.} \end{cases}$

The following facts [58], [34] describe the relationship between association rules and approximations of reducts.

For the given information table $\mathcal{A} = (U, A)$, the template **T**, the set of descriptors **P**. The implication $\left(\bigwedge_{D_i \in \mathbf{P}} D_i \longrightarrow \bigwedge_{D_j \notin \mathbf{P}} D_j\right)$ is

1. 100%-irreducible association rule from **T** if and only if **P** is a reduct in $\mathcal{A}|_{\mathbf{T}}$.

2. c-irreducible association rule from **T** if and only if **P** is an α-reduct of $\mathcal{A}|_{\mathbf{T}}$, where $\alpha = 1 - (\frac{1}{c} - 1)/(\frac{n}{s} - 1)$, n is the total number of objects from U and $s = support_{\mathcal{A}}(\mathbf{T})$.

One can show, that the problem of searching for the shortest α-reducts is NP-hard [34]. From the above facts it follows that extracting association rules from data is strongly related to extraction from the data reduct approximations [34] being basic constructs of rough sets.

$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
u_1	0	1	1	1	80	2	2	2	3
u_2	0	1	2	1	81	0	aa	1	aa
u_3	0	2	2	1	82	0	aa	1	aa
u_4	0	1	2	1	80	0	aa	1	aa
u_5	1	1	2	2	81	1	aa	1	aa
u_6	0	2	1	2	81	1	aa	1	aa
u_7	1	2	1	2	83	1	aa	1	aa
u_8	0	2	2	1	81	0	aa	1	aa
u_9	0	1	2	1	82	0	aa	1	aa
u_{10}	0	3	2	1	84	0	aa	1	aa
u_{11}	0	1	3	1	80	0	aa	2	aa
u_{12}	0	2	2	2	82	0	aa	2	aa
u_{13}	0	2	2	1	81	0	aa	1	aa
u_{14}	0	3	2	2	81	2	aa	2	aa
u_{15}	0	4	2	1	82	0	aa	1	aa
u_{16}	0	3	2	1	83	0	aa	1	aa
u_{17}	0	1	2	1	84	0	aa	1	aa
u_{18}	1	2	2	1	82	0	aa	2	aa

$\mathcal{A}\|_{\mathbf{T}}$	D_1	D_2	D_3	D_4	D_5	d
	$a_1 = 0$	$a_3 = 2$	$a_4 = 1$	$a_6 = 0$	$a_8 = 1$	
u_1	1	0	1	0	0	
u_2	1	1	1	1	1	1
u_3	1	1	1	1	1	1
u_4	1	1	1	1	1	1
u_5	0	1	0	0	1	
u_6	1	0	0	0	1	
u_7	0	0	0	0	1	
u_8	1	1	1	1	1	1
u_9	1	1	1	1	1	1
u_{10}	1	1	1	1	1	1
u_{11}	1	0	1	1	0	
u_{12}	1	0	0	1	0	
u_{13}	1	1	1	1	1	1
u_{14}	1	1	0	0	0	
u_{15}	1	1	1	1	1	1
u_{16}	1	1	1	1	1	1
u_{17}	1	1	1	1	1	1
u_{18}	0	1	1	1	0	

Table 5: The example of information table $\mathcal{A}$ and template **T** support by 10 objects and the new decision table $\mathcal{A}|_{\mathbf{T}}$ constructed from $\mathcal{A}$ and template **T**

The following example illustrates the main idea of our method. Let us consider the following information table $\mathcal{A}$ with 18 objects and 9 attributes.

Assume that the template

$$\mathbf{T} = (a_1 = 0) \wedge (a_3 = 2) \wedge (a_4 = 1) \wedge (a_6 = 0) \wedge (a_8 = 1)$$

has been extracted from the information table $\mathcal{A}$. One can see that $support(\mathbf{T}) = 10$ and $length(\mathbf{T}) = 5$. The new constructed decision table $\mathcal{A}|_{\mathbf{T}}$ is presented in Table 5. The discernibility function for $\mathcal{A}|_{\mathbf{T}}$ can be described as follows

$$\begin{aligned} f(D_1, D_2, D_3, D_4, D_5) = (D_2 \vee D_4 \vee D_5) \wedge (D_1 \vee D_5) \\ \wedge (D_1 \vee D_3 \vee D_4) \wedge (D_2 \vee D_3 \vee D_4) \\ \wedge (D_1 \vee D_2 \vee D_3 \vee D_4) \wedge (D_1 \vee D_3 \vee D_5) \\ \wedge (D_2 \vee D_3 \vee D_5) \wedge (D_3 \vee D_4 \vee D_5) \end{aligned}$$

After simplification we obtain six reducts corresponding to the prime implicants:
$f(D_1, D_2, D_3, D_4, D_5) = (D_3 \wedge D_5) \vee (D_4 \wedge D_5) \vee (D_1 \wedge$

$\mathcal{M}(\mathcal{A}\vert_{\mathbf{T}})$	u_2, u_3, u_4, u_8, u_9 $u_{10}, u_{13}, u_{15}, u_{16}, u_{17}$
u_1	$D_2 \vee D_4 \vee D_5$
u_5	$D_1 \vee D_3 \vee D_4$
u_6	$D_2 \vee D_3 \vee D_4$
u_7	$D_1 \vee D_2 \vee D_3 \vee D_4$
u_{11}	$D_1 \vee D_3 \vee D_5$
u_{12}	$D_2 \vee D_3 \vee D_5$
u_{14}	$D_3 \vee D_4 \vee D_5$
u_{18}	$D_1 \vee D_5$

$= \ _{100\%} \Longrightarrow$

$D_3 \wedge D_5 \Rightarrow D_1 \wedge D_2 \wedge D_4$
$D_4 \wedge D_5 \Rightarrow D_1 \wedge D_2 \wedge D_3$
$D_1 \wedge D_2 \wedge D_3 \Rightarrow D_4 \wedge D_5$
$D_1 \wedge D_2 \wedge D_4 \Rightarrow D_3 \wedge D_5$
$D_1 \wedge D_2 \wedge D_5 \Rightarrow D_3 \wedge D_4$
$D_1 \wedge D_3 \wedge D_4 \Rightarrow D_2 \wedge D_5$

$= \ _{90\%} \Longrightarrow$

$D_1 \wedge D_2 \Rightarrow D_3 \wedge D_4 \wedge D_5$
$D_1 \wedge D_3 \Rightarrow D_3 \wedge D_4 \wedge D_5$
$D_1 \wedge D_4 \Rightarrow D_2 \wedge D_3 \wedge D_5$
$D_1 \wedge D_5 \Rightarrow D_2 \wedge D_3 \wedge D_4$
$D_2 \wedge D_3 \Rightarrow D_1 \wedge D_4 \wedge D_5$
$D_2 \wedge D_5 \Rightarrow D_1 \wedge D_3 \wedge D_4$
$D_3 \wedge D_4 \Rightarrow D_1 \wedge D_2 \wedge D_5$

Table 6: The simplified version of discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$ and association rules

$D_2 \wedge D_3) \vee (D_1 \wedge D_2 \wedge D_4) \vee (D_1 \wedge D_2 \wedge D_5) \vee (D_1 \wedge D_3 \wedge D_4)$ for the decision table $\mathcal{A}|_{\mathbf{T}}$. Thus, we have found from $\mathbf{T}$ six association rules with (100%)-confidence.

If $c = 90\%$ it means that we would like to find α-reducts for the decision table $\mathcal{A}|_{\mathbf{T}}$, where $\alpha = 1 - \frac{\frac{1}{c}-1}{\frac{n}{s}-1} = 0.86$. Hence we would like to search for a set of descriptors that covers at least $\lceil (n-s)(\alpha) \rceil = \lceil 8 \cdot 0.86 \rceil = 7$ elements of the discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$. One can see that the following sets of descriptors: $\{D_1, D_2\}$, $\{D_1, D_3\}$, $\{D_1, D_4\}$, $\{D_1, D_5\}$, $\{D_2, D_3\}$, $\{D_2, D_5\}$, $\{D_3, D_4\}$ have nonempty intersection with exactly 7 members of the discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$. In Table 6 we present all association rules corresponding to those sets. Heuristics searching for α-reducts are discussed e.g. in [34].

Decomposition of large data tables

Several methods based on rough sets have been developed to deal with large data tables, e.g., to generate strong decision rules for them. We will discuss one of the methods based on decomposition of tables by using patterns, called templates, describing regular sub-domains of the universe (e.g., they describe large number of customers having large number of common features).

Long templates with large support are preferred in many Data Mining tasks. Several quality functions can be used to compare templates. For example they can be defined by $quality^1_{\mathcal{A}}(\mathbf{T}) = support_{\mathcal{A}}(\mathbf{T}) + length(\mathbf{T})$ and $quality^2_{\mathcal{A}}(\mathbf{T}) = support_{\mathcal{A}}(\mathbf{T}) \times length(\mathbf{T})$. Problems of high quality templates generation (by using different optimization criteria) are of high computational complexity. However, efficient heuristics have been developed for solving them (see e.g., [2, 71]), [32]).

Extracted from data templates are used to decompose large data tables. In consequence the decision tree is built with internal nodes labeled by the extracted from data templates, and outgoing from them edges by 0 (false) and 1 (true). Any leaf is labeled by a subtable (subdomain) consisting of all objects from the original table matching all templates or their complements appearing on the path from the root of the tree to the leaf. The process of decomposition is continued until the size of subtables attached to leaves is feasible for existing algorithms (e.g., decision rules for them can be generated efficiently) based on rough set methods. The reported experiments are showing that such decomposition returns interesting patterns of regular subdomains of large data tables (for references see [32], [35], [48] and [49]).

It is also possible to search for patterns that are almost included in the decision classes, i.e., default rules [29]. For a presentation of generating default rules see the bibliography in [48] and [49].

Conclusions

We have shown that rough set theory constitutes a sound basis for KDD.

There has been done a substantial progress in developing rough set methods for KDD (like methods for extraction from data rules, partial or total dependencies, methods for elimination of redundant data, methods dealing with missing data, dynamic data and others reported e.g., in [6], [7], [8], [16], [18], [24], [29], [30], [37], [48], [49], [50], [74]). New methods for extracting patterns from data (see e.g., [21], [35], [29]), [20], [44]), decomposition of decision systems (see e.g., [35]) as well as a new methodology for data mining in distributed and multiagent systems (see e.g., [47]) have been reported. Recently, rough set based methods have been proposed for data mining in very large relational data bases.

There are numerous areas of successful applications of rough set software systems (see [49] and `http://www.idi.ntnu.no/~aleks/rosetta/` for the ROSETTA system). Many interesting case studies are reported (for references see e.g., [48, 49], [37] and the bibliography in these books, in particular [7], [16], [20], [67], [74]).

We would like to mention some generalizations of rough set approach like rough mereological approach (see e.g., [51], [46]). The inclusion relation $x\mu_r y$ with the intended meaning *x is a part of y in a degree r* has been taken as the basic notion of the rough mereology being a generalization of the Leśniewski mereology. Rough mereology offers a methodology for synthesis and analysis of objects in distributed environment of intelligent agents, in particular, for synthesis of objects satisfying a given specification in satisfactory degree, i.e., objects sufficiently close to standard objects (prototypes) satisfying the specification. Moreover, rough mereology has been recently used [47] for developing foundations of the *information granule calculus*, an attempt towards formalization of the Computing with Words paradigm, recently formulated by Lotfi Zadeh [68], [69]. Let us also note that one of the prospects for rough mereological applications is to look for algorithmic methods of extracting logical structures from data such as, for instance, finding relational structures corresponding to relevant feature extraction, synthesizing default rules (approximate decision rules), constructing connectives for uncertainty coefficients propagation and synthesizing schemes of approximate reasoning creating a higher level knowledge extracted from data (e.g. qualitative schemes of reasoning). The development of such methods is crucial for further progress in many applications. It is also one of the central issues of KDD [13].

Several other generalizations of rough sets have been investigated and some of them have been used for real life data analysis (see e.g., [72], [5], [39], [14], [22], [38], [25], [56], [47]).

Finally, we would like to point out that the algebraic and logical aspects of rough sets have been intensively studied since the beginning of rough set theory. The reader interested in that topic is referred to the bibliography in [48].

Acknowledgment. This work has been supported by the Wallenberg Foundation, by the ESPRIT-CRIT 2 project #20288, and by grant from the State Committee for Scientific Research (KBN) of the Republic of Poland.

References

[1] T. Ågotnes, J. Komorowski, T. Loken (1999), *Taming large rule models in rough set approaches,* Proceedings of the 3rd European Conference of Principles and Practice of Knowledge Discovery in Databases, September 15-18, 1999, Prague, Czech Republic, Lecture Notes in Artificial Intelligence **1704**, Springer-Verlag, Berlin, pp. 193-203.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Verkano(1996), *Fast discovery of association rules*, Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (Eds.): Advances in Knowledge Discovery and Data Mining, The AAAI Press/The MIT Press 1996, pp. 307-328.

[3] J. G. Bazan (1998). *A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision system,* in: [48], pp. 321–365.

[4] F. M. Brown (1990), *Boolean Reasoning*, Kluwer Academic Publishers, Dordrecht 1990.

[5] G. Cattaneo (1998), *Abstract approximation spaces for rough theories*, In: Polkowski and Skowron [48], pp. 59–98.

[6] J. Cios, W. Pedrycz, R.W. Swiniarski (1998), *Data Mining in Knowledge Discovery*, Academic Publishers (in press).

[7] A. Czyżewski (1998), *Soft processing of audio signals.* In: Polkowski and Skowron [49], pp. 147–165.

[8] J. Deogun, V. Raghavan, A. Sarkar, H. Sever (1997), *Data mining: Trends in research and development*, In: Lin and Cercone [24], pp. 9–45.

[9] J. Dougherty, R. Kohavi, M. Sahami (1995), *Supervised and unsupervised discretization of continuous features,* In: Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA.

[10] I. Duentsch, G. Gediga (1997): *Statistical evaluation of rough set dependency analysis,* International Journal of Human-Computer Studies **46**, pp. 589-604.

[11] I. Duentsch, G. Gediga (2000): *Rough set data analysis,* in: Encyclopedia of Computer Science and Technology, Marcel Dekker (to appear).

[12] J. Friedman, R. Kohavi, Y. Yun (1996), *Lazy Decision Trees*, Proc. of AAAI-96, pp. 717–724.

[13] U. Fayyad, G. Piatetsky-Shapiro, G. (Eds.) (1996), *Advances in knowledge discovery and data mining*, MIT/AAAI Press.

[14] S. Greco, B. Matarazzo, R. Słowinski (1998), *Rough Approximation of a Preference Relation in a Pairwise Comparison Table.* In: Polkowski and Skowron [49] pp. 13–36.

[15] M. R. Chmielewski, J. W. Grzymala -Busse (1994), *Global discretization of attributes as preprocessing for machine learning,* Proceedings of the Third International Workshop on Rough Sets and Soft Computing (RSSC'94), San Jose State University, San Jose, California, USA, November 10–12, pp. 294–301.

[16] J.W. Grzymała–Busse (1998), *Applications of the rule induction system LERS.* In: Polkowski and Skowron [48], pp. 366–375.

[17] P. J. Huber (1981), *Robust statistics*, Wiley, New York.

[18] J. Komorowski, J. Żytkow (Eds.) (1997), *The First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97).* June 25–27, Trondheim, Norway, Lecture Notes in Artificial Intelligence **1263**, Springer-Verlag, Berlin pp. 1–396.

[19] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron (1999), *Rough sets: A tutorial*, in: S.K. Pal and A. Skowron (eds.), Rough fuzzy hybridization: A new trend in decision–making, Springer-Verlag, Singapore, pp. 3-98.

[20] W. Kowalczyk (1998), *Rough data modelling, A new technique for analyzing data.* In: Polkowski and Skowron [48], pp. 400–421.

[21] K. Krawiec, R. Słowiński, and D. Vanderpooten (1998), *Learning decision rules from similarity based rough approximations.* In: Polkowski and Skowron [49], pp. 37–54.

[22] M. Kryszkiewicz (1997), *Generation of rules from incomplete information systems.* In: Komorowski and Żytkow [18] pp. 156–166.

[23] P. Langley, H.A. Simon, G.L. Bradshaw, J.M. Żytkow (1987): Scientific Discovery, Computational Explorations of the Creative Processes, The MIT Press, Cambridge, Massachusetts.

[24] T. Y. Lin, N. Cercone (Eds.) (1997), *Rough sets and data mining. Analysis of imprecise data*, Kluwer Academic Publishers, Boston.

[25] T.Y. Lin (1989), *Granular computing on binary relations I, II.* In: Polkowski and Skowron [48], pp. 107–140.

[26] V.M. Marek, M. Truszczyński (1999), *Contributions to the theory of rough sets*, Fundamenta Informaticae **39(4)**, 1999, pp. 389-409.

[27] D. Michie, D.J. Spiegelhalter, C.C. Taylor (Eds.). *Machine learning, Neural and Statistical Classification.* Ellis Horwood, New York, 1994.

[28] T.M. Mitchell (1997), *Machine Learning*, Mc Graw-Hill, Portland.

[29] T. Mollestad and J. Komorowski (1998), *A Rough Set Framework for Propositional Default Rules Data Mining.* In: S.K. Pal and A. Skowron (Eds.) (1998), Rough – fuzzy hybridization: New trend in decision making, Springer–Verlag, Singapore.

[30] H. S. Nguyen (1997), *Discretization of Real Value Attributes, Boolean Reasoning Approach*, Ph.D. Dissertation, Warsaw University (1997), pp. 1–90.

[31] H. S. Nguyen (1999), *Efficient SQL-learning Method for Data Mining in Large Data Bases*, Proceedings pf the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99), 1999, pp. 806–811.

[32] S. H. Nguyen (2000), *Data regularity analysis and applications in data mining*, Ph.D. Dissertation, Warsaw University, 2000.

[33] H.S. Nguyen and S.H. Nguyen (1998), *Pattern extraction from data*, Fundamenta Informaticae **34**, pp. 129–144.

[34] H.S. Nguyen and S.H. Nguyen (1999), *Rough sets and association rule generation,* Fundamenta Informaticae **40/4** pp. .

[35] S. H. Nguyen, A. Skowron, P. Synak (1998), *Discovery of data patterns with applications to decomposition and classification problems.* In: Polkowski and Skowron [49], pp. 55–97.

[36] E. Orłowska (Ed.) (1998), *Incomplete Information, Rough Set Analysis*, Physica–Verlag, pp. 1–613.

[37] S. K. Pal, A. Skowron (1999), *Rough–fuzzy hybridization: New trend in decision making*, Springer–Verlag, Singapore (in print).

[38] G. Paun, L. Polkowski, A. Skowron (1996), *Parallel communicating grammar systems with negotiations.* Fundamenta Informaticae **28/3-4**, pp. 315–330.

[39] Z. Pawlak (1981), *Information systems – theoretical foundations.* Information Systems **6**, pp. 205–218.

[40] Z. Pawlak (1982), *Rough sets.* International Journal of Computer and Information Sciences **11**, pp. 341–356.

[41] Z. Pawlak (1991), *Rough Sets – Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht.

[42] Z. Pawlak and Z. Ras, (Eds.), (1996), *Proc. Ninth International Symposium on Methodologies for Intelligent Systems (ISMIS'96)*, June, Springer Verlag, LNAI.

[43] Z. Pawlak, A. Skowron (1999), *Rough set rudiments*, Bulletin of the International Rough Set Society **3/4**, pp. 181-185.

[44] Z. Piasta, A. Lenarcik (1998), *Rule induction with probabilistic rough classifiers.* Machine Learning (to appear).

[45] L. Polkowski (2000), *On synthesis of constructs for spatial reasoning via rough mereology*, Fundamenta Informaticae (to appear).

[46] L. Polkowski, A. Skowron (1996), *Rough mereology: A new paradigm for approximate reasoning*, International Journal of Approximate Reasoning **15/4**, pp. 333–365.

[47] L. Polkowski, A. Skowron (1998), *Rough sets: A perspective*, In: Polkowski and Skowron [48], pp. 31–58.

[48] L. Polkowski, A. Skowron (Eds.) (1998), *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, Physica-Verlag, Heidelberg.

[49] L. Polkowski, A. Skowron (Eds.) (1998), *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, Heidelberg.

[50] L. Polkowski, A. Skowron (Eds.) (1998), *Proc. First International Conference on Rough Sets and Soft Computing* – RSCTC'98, Warszawa, Poland, June 22–27, Springer-Verlag, LNAI **1424**.

[51] L. Polkowski, A. Skowron (1999), *Towards adaptive calculus of granules*, In: [70], **1**, pp. 201-227.

[52] J.R. Quinlan (1993), *C4.5. Programs for machine learning.* Morgan Kaufmann, San Mateo, CA.

[53] J. J. Rissanen (1978), *Modeling by Shortest Data Description, Automatica* **14**, pp. 465-471.

[54] J.F. Roddick J.F., M. Spiliopoulou (1999), *A bibliography of temporal, spatial, and temporal data mining research*, Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining **1/1**, pp. 34-38.

[55] B. Selman, H. Kautz and D. McAllester. *Ten challenges in propositional reasoning and search.* Proc. IJCAI'97, Japan.

[56] Z.W. Ras (1996), *Cooperative knowledge–based systems.* Journal of the Intelligent Automation Soft Computing **2/2** (special issue edited by T.Y. Lin), pp. 193–202

[57] A. Skowron (1995), *Synthesis of adaptive decision systems from experimental data.* In: A. Aamodt, J. Komorowski (eds), Proc. of the Fifth Scandinavian Conference on Artificial Intelligence (SCAI'95), May 1995, Trondheim, Norway, IOS Press, Amsterdam, pp. 220–238.

[58] A. Skowron, H.S. Nguyen (1999), *Boolean reasoning scheme with some applications in data mining,* Proceedings of the 3-rd European Conference on Principles and Practice of Knowledge Discovery in Databases, September 1999, Prague Czech Republic, Lecture Notes in Computer Science **1704**, pp. 107–115.

[59] A. Skowron, C. Rauszer (1992), *The Discernibility Matrices and Functions in Information Systems.* In: Słowiński [63], pp. 331–362.

[60] A. Skowron, J. Stepaniuk (1996), *Tolerance Approximation Spaces*, Fundamenta Informaticae **27**, pp. 245–253.

[61] A. Skowron, J. Stepaniuk, S. Tsumoto (1999), *Information Granules for Spatial Reasoning*, Bulletin of the International Rough Set Society **3/4**, pp. 147-154.

[62] D. Ślęzak (1998), *Approximate reducts in decision tables,* In: Proceedings of the Sixth International Conference, Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96) vol. 3, July 1–5, Granada, Spain, pp. 1159–1164.

[63] R. Słowiński, (Ed.) (1992), *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory*, Dordrecht, Kluwer Academic Publishers.

[64] R. Słowiński, D. Vanderpooten (1995), *Similarity relation as a basis for rough approximations.*

In: P. Wang (Ed.): Advances in Machine Intelligence & Soft Computing, Bookwrights, Raleigh NC (1997) pp. 17–33.

[65] R. Słowiński, D. Vanderpooten (1999), *A generalized definition of rough approximations based on similarity.* IEEE Trans. on Data and Knowledge Engineering (to appear).

[66] M. Szczuka (1999), *Symbolic and neural network methods for classifiers construction*, Ph.D. Dissertation, Warsaw University.

[67] S. Tsumoto (1998), *Modelling diagnostic rules based on rough sets.* In: Polkowski and Skowron [50], pp. 475–482.

[68] L.A. Zadeh (1996), *Fuzzy logic = computing with words*, IEEE Trans. on Fuzzy Systems **4**, pp. 103-111.

[69] L.A. Zadeh (1997), *Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic*, Fuzzy Sets and Systems **90**, pp. 111-127.

[70] L.A. Zadeh, J. Kacprzyk (eds.): Computing with Words in Information/Intelligent Systems vol.1-2, Physica-Verlag, Heidelberg, 1999.

[71] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li (1997), *New parallel algorithms for fast discovery of association rules.* In: Data Mining and Knowledge Discovery : An International Journal, special issue on Scalable High-Performance Computing for KDD **1/4**, pp. 343–373.

[72] W. Ziarko (1993), *Variable Precision Rough Set Model* , J. of Computer and System Sciences, 46, pp. 39–59.

[73] W. Ziarko (ed.) (1994), *Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93).* Workshops in Computing, Springer–Verlag & British Computer Society, London, Berlin.

[74] W. Ziarko (1998), *Rough sets as a methodology for data mining.* In: L. Polkowski A, Skowron (Eds.), Rough Sets in Knowledge Discovery 1: Methods Applications, Physica-Verlag, Heidelberg, pp. 554–576.