Available online at www.sciencedirect.com
ScienceDirect

Information Sciences 177 (2007) 41–73

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Rough sets and Boolean reasoning

Zdzisław Pawlak, Andrzej Skowron *

Institute of Mathematics, Warsaw University, ul. Banacha 2, 02-097 Warsaw, Poland

Received 24 February 2006; received in revised form 7 June 2006; accepted 7 June 2006

Commemorating the life and work of Zdzisław Pawlak.

Abstract

In this article, we discuss methods based on the combination of rough sets and Boolean reasoning with applications in pattern recognition, machine learning, data mining and conflict analysis.

Keywords: Boolean reasoning; Approximate Boolean reasoning; (In)discernibility; Rough sets; Reducts; Decision rules; Classifiers; Discretization; Symbolic value grouping; Association rules; Conflict analysis

> Data! data! data!
> Sir Artur Conan Doyle (1994). The Adventures of Sherlock Holmes
> Penguin Books, London

1. Introduction: discernibility and Boolean reasoning

The reader is referred to [92] for the notation used in this survey article.

Tasks collected under the labels of data mining, knowledge discovery, decision support, pattern classification, and approximate reasoning require tools aimed at discovering *templates* (*patterns*) in data and classifying them into certain *decision classes*. Templates are in many cases most frequent sequences of events, most probable events, regular configurations of objects, the decision rules of high quality, standard reasoning schemes. Tools for discovery and classification of templates are based on *reasoning schemes* rooted in various paradigms [25]. Such patterns can be extracted from data by means of methods based, e.g., on Boolean reasoning and discernibility (see Section 2 and [15]).

Discernibility relations belong to the most important relations considered in rough set theory. The ability to discern between perceived objects is important for constructing many entities like reducts, decision rules or decision algorithms. In the classical rough set approach, a discernibility relation $DIS(B) \subseteq U \times U$, where

[*] Corresponding author.
E-mail address: skowron@mimuw.edu.pl (A. Skowron).

0020-0255/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2006.06.007

$B \subseteq A$ is a subset of attributes of an information system (U, A), is defined by $xDIS(B)y$ if and only if *non*-$(xI(B)y)$, where $I(B)$ is the B-indiscernibility relation [92]. However, this is, in general, not the case for the generalized approximation spaces. One can define indiscernibility by $x \in I(y)$ and discernibility by $I(x) \cap I(y) = \emptyset$ for any objects x, y, where $I(x) = B(x), I(y) = B(y)$ in the case of the indiscernibility relation, and $I(x), I(y)$ are neighborhoods of objects not necessarily defined by the equivalence relation in a more general case [91].

The idea of Boolean reasoning is based on 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 decoded from prime implicants of the Boolean function f_P. Let us mention that to solve real-life problems it is necessary to deal with Boolean functions having large number of variables.

A successful methodology based on discernibility of objects and Boolean reasoning has been developed for computing of many entities important for applications, 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 (see Section 2).

Most of the problems related to generation of the mentioned above entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in the literature (e.g., with respect to the classification quality of unseen objects). Moreover, these heuristics are very efficient from the point of view of time necessary for computing of solutions. Many of these methods are based on discernibility matrices (see Section 2). Note that it is possible to compute the necessary information about these matrices using[1] information encoded in decision systems (e.g., sorted in preprocessing [5,78,134]) directly, which significantly improves the efficiency of algorithms.

It is important to note that the methodology makes it possible to construct heuristics having a very important *approximation property* which can be formulated as follows: Expressions generated by heuristics, i.e., implicants *close* to prime implicants define approximate solutions for the problem.

In Section 2, we discuss applications of methods based on rough sets and Boolean reasoning in machine learning, pattern recognition, and data mining. Section 3 is dedicated to conflict analysis based on different aspects of discernibility and indiscernibility.

2. Rough set methods for machine learning, pattern recognition, and data mining

In supervised machine learning paradigm [28,47,62,63], a learning algorithm is given a training data set, usually in the form of a decision system $\mathscr{A} = (U, A, d)$,[2] prepared by an expert. Every such decision system classifies elements from U into decision classes. The purpose of the algorithm is to return a set of decision rules together with matching procedure and conflict resolution strategy, called a classifier, which makes it possible to classify unseen objects, i.e., objects that are not described in the original decision table. In this section, we provide a number of rough set methods that can be used in construction of classifiers. For more information the reader is referred, e.g., to [2,10,17,18,22,24,26,29–46,53–56,58–60,64,65,67,76,3,83–88,93,97–104,106,107,111,112,122,124–128,130,132,133,135], and for papers on hierarchical learning and ontology approximation, e.g., to [7,11–13,77,80,79,108,113,115,116].

Most of the techniques discussed below are based on computing prime implicants for computing different kinds of reducts. Unfortunately, they are computationally hard. However, many heuristics have been developed which turned out to be very promising. The results of experiments on many data sets, reported in the literature, show a very good quality of classification of unseen objects using these heuristics. A variety of methods for computing reducts and their applications can be found in [5,51,57,88,99,100,106,107,112,114,117,118,135,136]. The fact that the problem of finding a minimal reduct of a given information system is NP-hard was proved in [114].

[1] That is, without the necessity of generation and storing of the discernibility matrices.

[2] For simplicity, we consider decision systems with one decision.

As we mentioned, there exists a number of good heuristics that compute sufficiently many reducts in an acceptable time. Moreover, a successful methodology, based on different reducts, has been developed for solution of many problems like attribute selection, decision rule generation, association rule generation, discretization of real-valued attributes, and symbolic value grouping. For further readings the reader is referred to [10,105,127] (attribute selection); [74,68,69,72,109] (discretization); [70,71] (discretization of data stored in relational databases); and [73] (reduct approximation and association rules).

Many of these methods are based on discernibility matrices defined in this section. It is possible to compute the necessary information about these matrices using information or decision systems (e.g., sorted in preprocessing [5,78]) directly what significantly improves the efficiency of algorithms.

The results presented in this section have been implemented, e.g., in the RSES and ROSETTA software systems (see http://logic.mimuw.edu.pl/~rses/ for RSES and http://rosetta.lcb.uu.se/general/ for ROSETTA), see also [4,5,9,14,48]. Sections 2.1–2.6 are based on a chapter of the book [23]. For links to other rough set software systems such as GROBIAN, KDD-R, LERS, ROSE2, ROSECON the reader is referred to http://rsds.wsiz.rzeszow.pl.

2.1. Reducts in information and decision systems

A crucial concept in the rough set approach to machine learning is that of a reduct. In fact, the term "reduct" corresponds to a wide class of concepts. What typifies all of them is that they are used to reduce information (decision) systems by removing redundant attributes. In this section, we consider three kinds of reducts which will be used in the remainder of this article.

Given an information system $\mathscr{A} = (U, A)$, a *reduct* is a minimal set (w.r.t. inclusion) of attributes $B \subseteq A$ such that $I(B) = I(A)$, where $I(B)$, $I(A)$ are the indiscernibility relations defined by B and A, respectively [89,92]. The intersection of all reducts is called a *core*.

Intuitively, a reduct is a minimal set of attributes from A that preserves the original classification defined by A. Reducts are extremely valuable in applications. Unfortunately, finding a minimal reduct is NP-hard in the general case. One can also show that, for any m, there is an information system with m attributes having an exponential (w.r.t. m) number of reducts. Fortunately, there are reasonably good heuristics which allow one to compute sufficiently many reducts in an acceptable amount of time.

To provide a general method for computing reducts, we will use the following constructs.

Let $\mathscr{A} = (U, A)$ be an information system with n objects. The *discernibility matrix* of $\mathscr{A}$ is an $n \times n$ matrix with elements c_{ij} consisting of the set of attributes from A on which objects x_i and x_j differ, i.e.,

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

A *discernibility function* $f_{\mathscr{A}}$ for $\mathscr{A}$ is a propositional formula of m Boolean variables, $a_1^*, \ldots, a_m^*$, corresponding to the attributes $a_1, \ldots, a_m$, defined by

$$f_{\mathscr{A}}(a_1^*, \ldots, a_m^*) = \bigwedge_{1 \leqslant j < i \leqslant m} \; \bigvee_{c \in c_{ij}^*, c_{ij} \neq \emptyset} c, \tag{2}$$

where $c_{ij}^* = \{a^* : a \in c_{ij}\}$. In the sequel, we write a_i instead of a_i^*, for simplicity.

The discernibility function $f_{\mathscr{A}}$ describes constraints which must hold to preserve discernibility between all pairs of discernible objects from $\mathscr{A}$. It requires keeping at least one attribute from each non-empty element of the discernibility matrix corresponding to any pair of discernible objects.

It can be shown [114] that for any information system $\mathscr{A} = (U, A)$ the set of all prime implicants of $f_{\mathscr{A}}$ determines the set of all reducts of $\mathscr{A}$.

Example 1. Consider the information system $\mathscr{A}$ whose associated information table is provided in Table 1. The discernibility matrix for $\mathscr{A}$ is presented in Table 2. (The letters s, c and h stand for *Speed*, *Color* and *Humidity*, respectively.) The discernibility function for the information system $\mathscr{A}$ is then given by

$$f_{\mathscr{A}}(s, c, h) \equiv (c \vee h) \wedge (s \vee c) \wedge (s \vee c \vee h).$$

Table 1
The information table considered in Example 1

Object	*Speed*	*Color*	*Humidity*
car1	medium	green	high
car2	medium	yellow	low
car3	high	blue	high

Table 2
The discernibility matrix for the information table provided in Table 1

$\mathscr{M}(\mathscr{A})$	car1	car2	car3
car1		c, h	s, c
car2	c, h		s, c, h
car3	s, c	s, c, h	

The prime implicants of $f_{\mathscr{A}}(s, c, h)$ can be computed in order to derive the reducts for $\mathscr{A}$:

$$f_{\mathscr{A}}(s, c, h) \equiv (c \vee h) \wedge (s \vee c) \wedge (s \vee c \vee h) \equiv (c \vee h) \wedge (s \vee c) \equiv c \vee (h \wedge s).$$

The prime implicants of $f_{\mathscr{A}}(s, c, h)$ are c and $h \wedge s$. Accordingly, there are two reducts of $\mathscr{A}$, namely $\{Color\}$ and $\{Humidity, Speed\}$.

The second type of reduct used in this article are the *decision-relative reducts* for decision systems.

In terms of decision tables, $\partial_A(x)$, called the generalized decision function, is the mapping on U such that for any object x it specifies all rows in the table whose attribute values are the same as for x, and then collects the decision values from each row [92]. A *decision-relative reduct* of $\mathscr{A} = (U, A, d)$ is a minimal (w.r.t. inclusion) non-empty set of attributes $B \subseteq A$ such that $\partial_B = \partial_A$. Intuitively, the definition states that B allows us to classify exactly the same objects, as belonging to equivalence classes U/∂_A, as A. In terms of decision tables, the columns associated with the attributes $A \backslash B$ may be removed without affecting the classification power of the original table.

To compute decision-relative reducts, we extend the definitions of discernibility matrix and discernibility function in the following straightforward manner. Let $\mathscr{A} = (U, A, d)$ be a consistent decision system (i.e., $\partial_A(x)$ consists of exactly one decision for any $x \in U$) and let $\mathscr{M}(\mathscr{A}) = [c_{ij}]$ be the discernibility matrix of the information system (U, A). We construct a new matrix, $\mathscr{M}'(\mathscr{A}) = [c'_{ij}]$, where

$$c'_{ij} = \begin{cases} \emptyset, & \text{if and only if } d(x_i) = d(x_j), \\ c_{ij}, & \text{otherwise.} \end{cases}$$

$\mathscr{M}'(\mathscr{A})$ is called the *decision-relative discernibility matrix* of $\mathscr{A}$. The *decision-relative discernibility function* $f^r_{\mathscr{A}}$ for $\mathscr{A}$ is constructed from the decision-relative discernibility matrix for $\mathscr{A}$ in the same way as a discernibility function is constructed from a discernibility matrix. Then it can be shown [114], that the set of all prime implicants of $f^r_{\mathscr{A}}$ determines the set of all decision-relative reducts of the consistent decision system $\mathscr{A}$.

Example 2. Consider the decision table associated with a decision system $\mathscr{A}$ as represented in Table 3.

The discernibility matrix for $\mathscr{A}$ is the same as the one given in Table 2, and the decision-relative discernibility matrix for $\mathscr{A}$ is provided in Table 4.

Table 3
The decision table considered in Example 2

Object	*Speed*	*Color*	*Humidity*	*Danger*
car1	medium	green	high	no
car2	medium	yellow	small	no
car3	high	blue	high	yes

Table 4
The decision-relative discernibility matrix corresponding to the decision system shown in Table 3

$\mathcal{M}^r(\mathcal{A})$	car1	car2	car3
car1			s, c
car2			s, c, h
car3	s, c	s, c, h	

Table 5
$\{Speed\}$-reduction of the decision system $\mathcal{A}$

Objects	*Speed*	*Danger*
car1, car2	medium	no
car3	high	yes

Using the decision-relative discernibility matrix, we can compute the decision-relative discernibility function for $\mathcal{A}$:

$$f^r_{\mathcal{A}}(s,c,h) \equiv (s \vee c) \wedge (s \vee c \vee h) \equiv (s \vee c).$$

The set of all prime implicants of $f^r_{\mathcal{A}}(s,c,h)$ is $\{s,c\}$. Therefore, there are two decision-relative reducts of $\mathcal{A}$, namely $\{Speed\}$ and $\{Color\}$.

To each decision-relative reduct B of a decision system $\mathcal{A}$, we assign a new decision system, called the *B-reduction* of $\mathcal{A}$. The details are as follows. Let $\mathcal{A} = (U, A, d)$ be a consistent decision system and suppose that B is a decision-relative reduct of $\mathcal{A}$. A *B-reduction* of $\mathcal{A}$ is a decision system $\mathcal{A}^* = (V, B, d)$, where[3]

- $V = \{[x]_B : x \in U\}$;
- $a([x]_B) = a(x)$, for each $a \in B$ and each $[x]_B \in V$;
- $d([x]_B) = d(x)$, for each $[x]_B \in V$.

Let $\mathcal{A}^*$ be the $\{Speed\}$-reduction of the decision system $\mathcal{A}$. The decision table associated with $\mathcal{A}^*$ is provided in Table 5.

The above defined method for decision relative reducts computation can be easily extended to inconsistent decision systems.

Observe that another kind of reducts can be obtained by using the discernibility requirement relative to the positive regions, i.e., $POS_A(d) = POS_B(d)$ instead of $\partial_B = \partial_A$. Certainly, for inconsistent decision systems the former requirement is less restrictive than the latter.

The last type of reduct, considered in this section, is used in applications where approximations to reducts are prefered to standard reducts. For example, approximate reducts for decision-relative reducts are making it possible to generate approximate decision rules. In the case of approximate reducts we relax the requirement for the discernibility preserving. Instead of preserving the discernibility for all entries of the discernibility matrix where it is necessary we preserve it to a degree, i.e., in a number of entries characterized by a coefficient α. Such reducts are called α-reducts, where α is a real number from the interval $[0,1]$. More formal definition of approximate reducts is the following:

Let $\mathcal{A} = (U, A, d)$ be a decision system and let $\mathcal{M}(\mathcal{A})$ be the discernibility matrix of $\mathcal{A}$. Assume further that n is the number of non-empty sets in $\mathcal{M}(\mathcal{A})$. A set of attributes $B \subseteq A$ is called an *α-reduct* if and only if $\frac{m}{n} \geqslant \alpha$, where m is the number of sets that have a non-empty intersection with B.

The reader is referred to [75,89,117,118] for information on various types of approximate reducts. Additionally, [8,73,101,119] provide approximation criteria based on discernibility and, therefore, related to Boolean reasoning principles.

[3] Recall that $[x]_B$, where $x \in U$, denotes the equivalence class of the relation $I(B)$ which contains x.

2.2. Attribute selection

In the supervised machine learning approach, a learning algorithm is provided with training data. In the context of rough set machine learning techniques, training data is provided in the form of training decision systems, or their equivalent representations as decision tables.

Since the conditional attributes of a specific decision table are typically extracted from large sets of unstructured data, it is often the case that some of the attributes are irrelevant for the purpose of classification. Such attributes should be removed from the table if possible. The *attribute selection problem* is the problem of choosing a relevant subset of attributes, while removing the irrelevant ones.

A natural solution of the attribute selection problem is to assume that the intersection of the decision-relative reducts of a training decision table is the source of the relevant attributes. Unfortunately, there are two problems with this solution. Firstly, the intersection can be empty. Secondly, the number of attributes contained in all decision-relative reducts is usually small. Consequently, although these attributes perfectly characterize the training decision table, they are, in general, inadequate for providing a satisfactory classification of new objects not occurring in the training data.

To deal with the attribute selection problem, it is often reasonable to use various approximations of decision-relative reducts.

Let $\mathscr{A} = (U, A, d)$ be a consistent decision system. Any subset B of A is called an *approximate reduct* of $\mathscr{A}$. The number

$$\varepsilon_{A,\{d\}}(B) = \frac{\gamma(A,\{d\}) - \gamma(B,\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(B,\{d\})}{\gamma(A,\{d\})}, \tag{3}$$

is called an *error of reduct approximation*.[4]

The error of reduct approximation expresses exactly how the set of attributes B approximates the set of condition attributes A with respect to determination of d. Note that $\varepsilon_{A,\{d\}}(B) \in [0,1]$, where 0 indicates no error, and the closer $\varepsilon_{A,\{d\}}(B)$ is to 1, the greater is the error. The reader is referred, e.g., to [73,119] for more information on approximate reducts.

There are two general approaches to attribute selection: an open-loop approach and a closed-loop approach. Methods based on the open-loop approach are characterized by the fact that they do not use any feedback information about classifier quality for attribute selection. In contrast, the methods based on the closed-loop approach use feedback information as criteria for attribute selection.

A number of attribute selection algorithms have been proposed in the machine learning literature, but they will not be considered here since our focus is on rough set based techniques. Rough set techniques which attempt to solve the attribute selection problem are typically based on the closed-loop approach and consist of the following basic steps[5]:

(1) Decision-relative reducts are extracted from a training decision table. The attributes contained in these reducts (or in their intersection) are viewed as potentially relevant.
(2) Using the specific machine learning algorithm, a classifier based on the chosen attributes is constructed.
(3) The classifier is then tested on a new set of training data; if its performance is unsatisfactory (w.r.t. some measure), a new set of attributes is constructed by extracting approximate additional reducts for the initial training table, and the process is repeated.

Reducts need not be the only source of information used in the selection of attributes. The rough set approach offers another interesting possibility. The main idea is to generalize the notion of attribute reduction by introducing the concept of *significance of attributes*. This measure enables attributes to be evaluated using a multi-valued scale which assigns a real number from the interval $[0, 1]$ to an attribute. This number, expressing

[4] Recall that the coefficient $\gamma(X, Y)$ expresses the degree of dependency between sets of attributes X and Y (see [92]).

[5] There are public domain software packages, for instance the RSES system (for references see, e.g., [14] and http://logic.mimuw.edu.pl/~rses/), which offer software that may be used to solve the attribute selection problem.

the importance of an attribute in a decision system, is evaluated by measuring the effect of removing the attribute from the table.

The *significance of an attribute a* in a decision table $\mathscr{A} = (U, A, d)$ is defined by

$$\sigma_{A,\{d\}}(a) = \frac{\gamma(A,\{d\}) - \gamma(A - \{a\},\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(A - \{a\},\{d\})}{\gamma(A,\{d\})}. \tag{4}$$

Assume that $B \subseteq A$. The significance coefficient can be extended to sets of attributes as follows,

$$\sigma_{(A,\{d\})}(B) = \frac{\gamma(A,\{d\}) - \gamma(A - B,\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(A - B,\{d\})}{\gamma(A,\{d\})}. \tag{5}$$

The coefficient $\sigma_{A,\{d\}}(B)$, can be understood as a classification error which occurs when the attributes $a \in B$ are removed from the decision system. Note that $\sigma_{A,\{d\}}(B) \in [0, 1]$, where 0 indicates that removal of attributes in B causes no error, and the closer $\sigma_{A,\{d\}}(B)$ is to 1, the greater the error is.

Remark 3. In this section, we have mainly concentrated on the case, where the attributes are selected from the set of attributes of the input decision system. In some cases it might be useful to replace some attributes by a new one.

For example, if one considers a concept of a safe distance between vehicles, then attributes, say *VS* standing for "vehicle speed" and *SL* standing for "speed limit", can be replaced by an attribute *DIF* representing the difference *SL–VS*. In fact, the new attribute better corresponds to the concept of safe distance than the pair (VS, SL).

2.3. Value set reduction

Consider a decision system with a large number of attribute values. There is a very low probability that a new object will be properly recognized by matching its attribute value vector with any of the rows in the decision table associated with the decision system. So, in order to construct a high quality classifier, it is often necessary to reduce the cardinality of the value sets of specific attributes in a training decision table. The task of reducing the cardinality of value sets is referred to as the *value set reduction problem*.

In this section, two methods of value set reduction are considered:

(1) discretization, used for real value attributes, and
(2) symbolic attribute value grouping, used for symbolic attributes.

2.3.1. Discretization

A discretization replaces value sets of conditional real-valued attributes with intervals. The replacement ensures that a consistent decision system is obtained (assuming a given consistent decision system) by substituting original values of objects in the decision table by the unique names of the intervals comprising these values. This substantially reduces the size of the value sets of real-valued attributes.

The use of discretization is not specific to the rough set approach to machine learning. In fact, a majority of rule or tree induction algorithms require it for a good performance.

Let $\mathscr{A} = (U, A, d)$ be a consistent decision system. Assume $V_a = [l_a, r_a) \subset \mathbb{R}$,[6] for any $a \in A$, and $l_a < r_a$. A pair (a, c), where $a \in A$ and $c \in V_a$, is called a *cut* on V_a.

Any attribute $a \in A$ defines a sequence of real numbers $v_1^a < v_2^a < \cdots < v_{k_a}^a$, where $\{v_1^a, v_2^a, \ldots, v_{k_a}^a\} = \{a(x) : x \in U\}$. The *set of basic cuts* on a, written B_a, is specified by

$$B_a = \{(a, (v_1^a + v_2^a)/2), (a, (v_2^a + v_3^a)/2), \ldots, (a, (v_{k_a-1}^a + v_{k_a}^a)/2)\}.$$

The set $\cup_{a \in A} B_a$ is called the *set of basic cuts* on $\mathscr{A}$.

[6] $\mathbb{R}$ denotes the set of real numbers.

Example 4. Consider a consistent decision system $\mathscr{A}$ and the associated decision table presented in Table 6(a). We assume that the initial value domains for the attributes a and b are

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

The sets of values of a and b for objects from U are

$$a(U) = \{0.8, \; 1.0, \; 1.3, \; 1.4, \; 1.6\};$$
$$b(U) = \{0.5, \; 1.0, \; 2.0, \; 3.0\}.$$

By definition, the sets of basic cuts for a and b are

$$B_a = \{(a,0.9), \; (a,1.15), \; (a,1.35), \; (a,1.5)\};$$
$$B_b = \{(b,0.75); \; (b,1.5); \; (b,2.5)\}.$$

Using the idea of cuts, decision systems with real-valued attributes can be discretized. For a decision system $\mathscr{A} = (U,A,d)$ and $a \in A$, let

$$C_a = \{(a,c_1^a),(a,c_2^a),\ldots,(a,c_k^a)\},$$

be any set of cuts of a. Assume that $c_1^a < c_2^a < \cdots < c_k^a$. The set of cuts $C = \cup_{a\in A} C_a$ defines a new decision system $\mathscr{A}^C = (U,A^C,d)$, called the *C-discretization of* $\mathscr{A}$, where

- $A^C = \{a^C : a \in A\}$;
- $a^C(x) = \begin{cases} 0, & \text{if and only if } a(x) < c_1^a, \\ i, & \text{if and only if } a(x) \in [c_i^a, c_{i+1}^a), \text{ for } i \in \{1,\ldots,k-1\}, \\ k+1, & \text{if and only if } a(x) > c_k^a. \end{cases}$

Example 5 (Example 4 continued). Let $C = B_a \cup B_b$. It is easy to check that the C-discretization of $\mathscr{A}$ is the decision system whose decision table is provided in Table 6(b).

Since a decision system can be discretized in many ways, a natural question arises how to evaluate various possible discretizations.

A set of cuts C is called $\mathscr{A}$*-consistent*, if $\partial_{\mathscr{A}} = \partial_{\mathscr{A}^C}$, where $\partial_{\mathscr{A}}$ and $\partial_{\mathscr{A}^C}$ are generalized decision functions for $\mathscr{A}$ and $\mathscr{A}^C$, respectively. An $\mathscr{A}$-consistent set of cuts C is $\mathscr{A}$*-irreducible* if C' is not $\mathscr{A}$-consistent for any $C' \subset C$. The $\mathscr{A}$-consistent set of cuts C is $\mathscr{A}$*-optimal* if $card(C) \leqslant card(C')$, for any $\mathscr{A}$-consistent set of cuts C'.

As easily observed, the set of cuts considered in Example 5 is $\mathscr{A}$-consistent. However, as we shall see in Example 6, it is neither optimal nor irreducible.

Since the purpose of the discretization process is to reduce the size of individual value sets of attributes, we are primarily interested in optimal sets of cuts. These are extracted from the basic sets of cuts for a given decision system.

Let $\mathscr{A} = (U,A,d)$ be a consistent decision system where $U = \{u_1,\ldots,u_n\}$. Recall that any attribute $a \in A$ defines a sequence $v_1^a < v_2^a < \cdots < v_{k_a}^a$, where $\{v_1^a, v_2^a, \ldots, v_{k_a}^a\} = \{a(x) : x \in U\}$. Let $ID(\mathscr{A})$ be the set of pairs

Table 6
The discretization process: (a) the original decision system $\mathscr{A}$ considered in Example 4 (b) the C-discretization of $\mathscr{A}$ considered in Example 5

$\mathscr{A}$	a	b	d		$\mathscr{A}^C$	a^C	b^C	d
(a)					(b)			
u_1	0.8	2.0	1		u_1	0	2	1
u_2	1.0	0.5	0		u_2	1	0	0
u_3	1.3	3.0	0		u_3	2	3	0
u_4	1.4	1.0	1	⇒	u_4	3	1	1
u_5	1.4	2.0	0		u_5	3	2	0
u_6	1.6	3.0	1		u_6	4	3	1
u_7	1.3	1.0	1		u_7	2	1	1

(i,j) such that $i < j$ and $d(u_i) \neq d(u_j)$. We now construct a propositional formula, called the *discernibility formula* of $\mathscr{A}$, as follows:

(1) To each interval of the form $[v_k^a, v_{k+1}^a)$, $a \in A$ and $k \in \{1, \ldots, n_a - 1\}$, we assign a Boolean variable denoted by p_k^a. The set of all these variables is denoted by $V(\mathscr{A})$.
(2) We first construct a family of formulas

$$\{B(a,i,j) : a \in A \quad \text{and} \quad (i,j) \in ID(\mathscr{A})\},$$

where $B(a,i,j)$ is a disjunction of all elements from the set

$$\left\{p_k^a : \left[v_k^a, v_{k+1}^a\right) \subseteq \left[\min\{a(u_i), a(u_j)\}, \max\{a(u_i), a(u_j)\}\right)\right\}.$$

(3) Next, we construct a family of formulas

$$\{C(i,j) : i,j \in \{1,\ldots,n\}, i<j \quad \text{and} \quad (i,j) \in ID(\mathscr{A})\},$$

where $C(i,j) = \bigvee_{a \in A} B(a,i,j)$.
(4) Finally, the discernibility formula for $\mathscr{A}$, $D(\mathscr{A})$, is defined as

$$D(\mathscr{A}) = \bigwedge C(i,j),$$

where $i < j$ and $(i,j) \in ID(\mathscr{A})$ and $C(i,j) \not\equiv$ False.

Any non-empty set $S = \{p_{k_1}^{a_1}, \ldots, p_{k_r}^{a_r}\}$ of Boolean variables from $V(\mathscr{A})$ uniquely defines a set of cuts, $C(S)$, given by

$$C(S) = \{(a_1, (v_{k_1}^{a_1} + v_{k_1+1}^{a_1})/2), \cdots, (a_r, (v_{k_r}^{a_r} + v_{k_r+1}^{a_r})/2)\}.$$

Then we have the following properties:

Let $\mathscr{A} = (U, A, d)$ be a consistent decision system. For any non-empty set $S \subseteq V(\mathscr{A})$ of Boolean variables, the following two conditions are equivalent:

(1) The conjunction of variables from S is a prime implicant of the discernibility formula for $\mathscr{A}$.
(2) $C(S)$ is an $\mathscr{A}$-irreducible set of cuts on $\mathscr{A}$.

Let $\mathscr{A} = (U, A, d)$ be a consistent decision system. For any non-empty set $S \subseteq V(\mathscr{A})$ of Boolean variables, the following two conditions are equivalent:

(1) The conjunction of variables from S is a minimal (w.r.t. to length) prime implicant of the discernibility formula for $\mathscr{A}$.
(2) $C(S)$ is an $\mathscr{A}$-optimal set of cuts on $\mathscr{A}$.

Example 6 (Example 5 continued).

$$ID(\mathscr{A}) = \{(1,2),\ (1,3),\ (1,5),\ (2,4),\ (2,6),\ (2,7)\ (3,4),\ (3,6),\ (3,7),\ (4,5),\ (5,6),\ (5,7)\}.$$

(1) We introduce four Boolean variables, p_1^a, p_2^a, p_3^a, p_4^a, corresponding respectively to the intervals

$[0.8, 1.0),\ [1.0, 1.3),\ [1.3, 1.4),\ [1.4, 1.6)$

of the attribute a, and three Boolean variables, p_1^b, p_2^b, p_3^b, corresponding respectively to the intervals

$[0.5, 1.0),\ [1.0, 2.0),\ [2, 3.0)$

of the attribute b.

(2) The following are the formulas $B(a,i,j)$ and $B(b,i,j)$, where $i<j$ and $(i,j)\in ID(\mathscr{A})$:

$$\begin{array}{ll}
B(a,1,2)\equiv p_1^a & B(b,1,2)\equiv p_1^b\vee p_2^b\\
B(a,1,3)\equiv p_1^a\vee p_2^a & B(b,1,3)\equiv p_3^b\\
B(a,1,5)\equiv p_1^a\vee p_2^a\vee p_3^a & B(b,1,5)\equiv \textsc{False}\\
B(a,2,4)\equiv p_2^a\vee p_3^a & B(b,2,4)\equiv p_1^b\\
B(a,2,6)\equiv p_2^a\vee p_3^a\vee p_4^a & B(b,2,6)\equiv p_1^b\vee p_2^b\vee p_3^b\\
B(a,2,7)\equiv p_2^a & B(b,2,7)\equiv p_1^b\\
B(a,3,4)\equiv p_3^a & B(b,3,4)\equiv p_2^b\vee p_3^b\\
B(a,3,6)\equiv p_3^a\vee p_4^a & B(b,3,6)\equiv \textsc{False}
\end{array}$$

$$\begin{array}{ll}
B(a,3,7)\equiv \textsc{False} & B(b,3,7)\equiv p_2^b\vee p_3^b\\
B(a,4,5)\equiv \textsc{False} & B(b,4,5)\equiv p_2^b\\
B(a,5,6)\equiv p_4^a & B(b,5,6)\equiv p_3^b\\
B(a,5,7)\equiv p_3^a & B(b,5,7)\equiv p_2^b.
\end{array}$$

(3) The following are the formulas $C(i,j)$, where $i<j$ and $(i,j)\in ID(\mathscr{A})$:

$$\begin{array}{ll}
C(1,2)\equiv p_1^a\vee p_1^b\vee p_2^b & C(1,3)\equiv p_1^a\vee p_2^a\vee p_3^b\\
C(1,5)\equiv p_1^a\vee p_2^a\vee p_3^a & C(2,4)\equiv p_2^a\vee p_3^a\vee p_1^b\\
C(2,6)\equiv p_2^a\vee p_3^a\vee p_4^a\vee p_1^b\vee p_2^b\vee p_3^b & C(2,7)\equiv p_2^a\vee p_1^b\\
C(3,4)\equiv p_3^a\vee p_2^b\vee p_3^b & C(3,6)\equiv p_3^a\vee p_4^a\\
C(3,7)\equiv p_2^b\vee p_3^b & C(4,5)\equiv p_2^b\\
C(5,6)\equiv p_4^a\vee p_3^b & C(5,7)\equiv p_3^a\vee p_2^b.
\end{array}$$

(4) The discernibility formula for $\mathscr{A}$ is then given by

$$\begin{aligned}
D(\mathscr{A})\equiv{}&(p_1^a\vee p_1^b\vee p_2^b)\wedge(p_1^a\vee p_2^a\vee p_3^b)\wedge(p_1^a\vee p_2^a\vee p_3^a)\wedge(p_2^a\vee p_3^a\vee p_1^b)\wedge(p_2^a\vee p_3^a\vee p_4^a\vee p_1^b\vee p_2^b\vee p_3^b)\\
&\wedge(p_2^a\vee p_1^b)\wedge(p_3^a\vee p_2^b\vee p_3^b)\wedge(p_3^a\vee p_4^a)\wedge(p_2^b\vee p_3^b)\wedge p_2^b\wedge(p_4^a\vee p_3^b)\wedge(p_3^a\vee p_2^b).
\end{aligned}$$

The prime implicants of the formula $D(\mathscr{A})$ are

$p_2^a\wedge p_4^a\wedge p_2^b$
$p_2^a\wedge p_3^a\wedge p_2^b\wedge p_3^b$
$p_3^a\wedge p_1^b\wedge p_2^b\wedge p_3^b$
$p_1^a\wedge p_4^a\wedge p_1^b\wedge p_2^b$.

Suppose we take the prime implicant $p_1^a\wedge p_4^a\wedge p_1^b\wedge p_2^b$. Its corresponding set of cuts is

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

The decision table for the C-discretization of $\mathscr{A}$ is provided in Table 7.

Observe that the set of cuts corresponding to the prime implicant $p_2^a\wedge p_4^a\wedge p_2^b$ is $\{(a,1.15),(a,1.5),(b,1.5)\}$. Thus C is not an optimal set of cuts.

The problem of searching for an optimal set of cuts P in a given decision system $\mathscr{A}$ is NP-hard. However, it is possible to devise efficient heuristics which, in general, return reasonable sets of cuts. One of them, called MD-heuristics, is presented below.

We say that a cut (a,c) discerns objects x and y if and only if $a(x)<c\leqslant a(y)$ or $a(y)<c\leqslant a(x)$.

Let n be the number of objects and let k be the number of attributes of a decision system $\mathscr{A}$. It can be shown that the best cut can be found in $O(kn)$ steps using $O(kn)$ space only.

Table 7
The C-discretization considered in Example 6

$\mathscr{A}^C$	a^C	b^C	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

Example 7. Consider the decision table with the associated decision system $\mathscr{A}$, provided in Table 6 from Example 4. The associated information table for the information system $\mathscr{A}^*$ is presented in Table 8.

Under the assumption that columns with maximal number of 1's are chosen from left to right (if many such columns exist in a given step), the set of cuts returned by the algorithm is $\{(a,1.35),(b,1.5),(a,1.15),(a,1.5)\}$. However, as shown in Example 6, it is not an optimal set of cuts.

```
INPUT: a decision system $\mathscr{A} = (U, A, d)$
OUTPUT: a set of cuts $\mathscr{C}$
  (1) Set $\mathscr{C}$ to $\emptyset$.
  (2) Let $\cup_{a \in A} C_a$ be the set of basic cuts on $\mathscr{A}$.
  (3) Construct an information table $\mathscr{A}^* = (U^*, A^*)$ such that
      • $U^*$ is the set of pairs $(u_i, u_j)$ of objects discerned by $d$ (in $\mathscr{A}$) such that $i < j$;
      • $A^* = \cup_{a \in A} C_a$, where for each $c \in A^*$,
            $c(x,y) = \begin{cases} 1, & \text{if and only if } c \text{ discerns } x \quad \text{and} \quad y \text{ (in } \mathscr{A}), \\ 0, & \text{otherwise.} \end{cases}$
  (4) Choose a column from $\mathscr{A}^*$ with the maximal number of occurrences of 1's; add the cut corresponding to this column to $\mathscr{C}$; delete the column from $\mathscr{A}^*$, together with all rows marked with 1 in it.
  (5) If $A^*$ is non-empty, then go to step 4 else stop.
```

2.3.2. Symbolic attribute value grouping

Symbolic attribute value grouping is a technique for reducing the cardinality of value sets of symbolic attributes. Let $\mathscr{A} = (U, A, d)$ be a decision system. Any function $c_a : V_a \to \{1, \ldots, m\}$, where $m \leqslant card(V_a)$, is called a *clustering function* for V_a. The *rank of* c_a, denoted by $rank(c_a)$, is the value $card(\{c_a(x) | x \in V_a\})$.

Table 8
The information table for the information system $\mathscr{A}^*$

$\mathscr{A}^*$	$(a,0.9)$	$(a,1.15)$	$(a,1.35)$	$(a,1.5)$	$(b,0.75)$	$(b,1.5)$	$(b,2.5)$
(u_1,u_2)	1	0	0	0	1	1	0
(u_1,u_3)	1	1	0	0	0	0	1
(u_1,u_5)	1	1	1	0	0	0	0
(u_2,u_4)	0	1	1	0	1	0	0
(u_2,u_6)	0	1	1	1	1	1	1
(u_2,u_7)	0	1	0	0	1	0	0
(u_3,u_4)	0	0	1	0	0	1	1
(u_3,u_6)	0	0	1	1	0	0	0
(u_3,u_7)	0	0	0	0	0	1	1
(u_4,u_5)	0	0	0	0	0	1	0
(u_5,u_6)	0	0	0	1	0	0	1
(u_5,u_7)	0	0	1	0	0	1	0

For $B \subseteq A$, a family of clustering functions $\{c_a\}_{a\in B}$ is *B-consistent* if and only if

$$\forall a \in B\ [c_a(a(u)) = c_a(a(u'))],$$

implies

$$(u,u') \in I(B) \cup I(\{d\}), \text{for any pair } (u,u') \in U.$$

The notion of B-consistency has the following intuitive interpretation: If two objects are indiscernible w.r.t. clustering functions for value sets of attributes from B, then they are indiscernible either by the attributes from B or by the decision attribute.

We consider the following problem, called the *symbolic value partition grouping problem*:

> Given a decision system $\mathscr{A} = (U,A,d)$, where $U = \{u_1,\ldots,u_k\}$, and a set of attributes $B \subseteq A$, search for a B-consistent family $\{c_a\}_{a\in B}$ of clustering functions such that $\sum_{a\in B} rank(c_a)$ is minimal.

In order to solve this problem, we apply the following steps:

(1) Introduce a set of new Boolean variables[7]:

$$\{a_v^{v'} :\ a \in B \text{ and } v,v' \in V_a \text{ and } v \neq v'\}.$$

We extract a subset S of this set such that $a_v^{v'} \in S$ implies that $v' < v$ w.r.t. some arbitrary linear order $<$ on the considered domain.

(2) Construct matrix $\mathscr{M} = [c_{ij}]_{i,j=1,\ldots,k}$ as follows:

$$c_{ij} = \{a_v^{v'} \in S :\ v' = a(u_i) \text{ and } v = a(u_j) \text{ and } d(u_i) \neq d(u_j)\}.$$

It is easily seen that in the case of a binary decision, the matrix can be reduced by placing objects corresponding to the first decision in rows and those corresponding to the second decision in columns. We call such a matrix a *reduced discernibility matrix.*

(3) Using the reduced matrix, $\mathscr{M}'$, obtained in the previous step, construct the function

$$\bigwedge_{c_{ij}\in\mathscr{M}'} \left(\bigvee_{c\in c_{ij}, c_{ij}\neq\emptyset} c \right).$$

(4) Compute the shortest prime implicant I of the constructed function.

(5) Using I, construct, for each attribute $a \in B$, an undirected graph $\Gamma_a = \langle V_a^\Gamma, E_a^\Gamma \rangle$, where

- $V_a^\Gamma = \{a_v \mid v \in V_a\}$;
- $E_a^\Gamma = \{(a_x,a_y) \mid x,y \in U \text{ and } a(x) \neq a(y)\}$. Note that using I one can construct E_a^Γ due to the equality $E_a^\Gamma = \{(a_v,a_{v'}) : a_v^{v'} \text{ occurs in } I\}$.

(6) Find a minimal coloring of vertices for Γ_a.[8] The coloring defines a partition of V_a^Γ by assuming that all vertices of the same color belong to the same partition set and no partition contains vertices with different colors. Partition sets are named using successive natural numbers.
The clustering function for V_a is $c_a(a_v) = i$, provided that a_v is a member of the ith partition set.

Remark 8. In practical implementations, one does not usually construct the matrix $\mathscr{M}$ explicitly, as required in Steps (2)–(3) above. Instead, prime implicants are directly extracted from the original decision system.

It should be emphasized that in Step (4) above, there can be many different shortest prime implicants and in Step (6) there can be many different colorings of the obtained graphs. Accordingly, one can obtain many

[7] The introduced variables serve to discern between pairs of objects w.r.t. an attribute a.

[8] The colorability problem is solvable in polynomial time for $k = 2$, but remains NP-complete for all $k \geqslant 3$. But, similarly to discretization, one can apply some efficient search heuristics for generating (sub-) optimal partitions.

substantially different families of clustering functions resulting in different classifiers. In practice, one often generates a number of families of clustering functions, tests them against data and chooses the best one.

Using the construction above to generate a family of partitions, it is usually possible to obtain a substantially smaller decision table, according to the following definition.

Let $\mathscr{A} = (U, A, d)$ be a decision system and $B \subseteq A$. Any family of clustering functions $c = \{c_a\}_{a \in B}$ specifies a new decision system $\mathscr{A}^c = (U, A^c, d)$ called the *c-reduction of* $\mathscr{A}$ w.r.t. B, where $A^c = \{a^c : a \in B\}$ and $a^c(x) = c_a(a(x))$.

Example 9. Consider the decision table provided in Table 9. The goal is to solve the symbolic value partition problem for $B = A$. One then has to perform the following steps:

(1) Introduce new Boolean variables a_v^u, b_x^w, for all $u, v \in V_a, u < v$ and $w, x \in V_b, w < x$.
(2) The reduced matrix $\mathscr{M}'$ is presented in Table 10.
(3) The required Boolean function is given by

$$b_{b_4}^{b_1} \wedge b_{b_4}^{b_2} \wedge (a_{a_2}^{a_1} \vee b_{b_4}^{b_3}) \wedge (a_{a_3}^{a_1} \vee b_{b_4}^{b_1}) \wedge (a_{a_2}^{a_1} \vee b_{b_2}^{b_1}) \wedge a_{a_2}^{a_1} \wedge b_{b_3}^{b_2} \wedge (a_{a_3}^{a_2} \vee b_{b_2}^{b_1}) \wedge a_{a_2}^{a_1} \wedge (a_{a_2}^{a_1} \vee b_{b_2}^{b_1}) \wedge b_{b_3}^{b_1}$$
$$\wedge a_{a_3}^{a_2} \wedge (a_{a_4}^{a_1} \vee b_{b_2}^{b_1}) \wedge a_{a_4}^{a_1} \wedge (a_{a_4}^{a_2} \vee b_{b_3}^{b_2}) \wedge (a_{a_4}^{a_3} \vee b_{b_2}^{b_1}) \wedge (a_{a_3}^{a_1} \vee b_{b_4}^{b_1}) \wedge (a_{a_3}^{a_1} \vee b_{b_4}^{b_2}) \wedge (a_{a_3}^{a_2} \vee b_{b_4}^{b_3}) \wedge b_{b_4}^{b_1}$$
$$\wedge (a_{a_2}^{a_1} \vee b_{b_5}^{b_1}) \wedge (a_{a_2}^{a_1} \vee b_{b_5}^{b_2}) \wedge b_{b_5}^{b_3} \wedge (a_{a_3}^{a_2} \vee b_{b_5}^{b_1}).$$

(4) The shortest prime implicant for the function is

$$I \equiv a_{a_2}^{a_1} \wedge a_{a_3}^{a_2} \wedge a_{a_4}^{a_1} \wedge a_{a_4}^{a_3} \wedge b_{b_4}^{b_1} \wedge b_{b_4}^{b_2} \wedge b_{b_3}^{b_2} \wedge b_{b_3}^{b_1} \wedge b_{b_5}^{b_3}.$$

(5) The graphs corresponding to a and b are shown in Fig. 1.

Table 9
The decision table considered in Example 9

$\mathscr{A}$	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

Table 10
The reduced matrix corresponding to the decision table provided in Table 9

$\mathscr{M}'$	u_1	u_2	u_3	u_4
u_5	$b_{b_4}^{b_1}$	$b_{b_4}^{b_2}$	$a_{a_2}^{a_1}, b_{b_4}^{b_3}$	$a_{a_3}^{a_1}, b_{b_4}^{b_1}$
u_6	$a_{a_2}^{a_1}, b_{b_2}^{b_1}$	$a_{a_2}^{a_1}$	$b_{b_3}^{b_2}$	$a_{a_3}^{a_2}, b_{b_2}^{b_1}$
u_7	$a_{a_2}^{a_1}$	$a_{a_2}^{a_1}, b_{b_2}^{b_1}$	$b_{b_3}^{b_1}$	$a_{a_3}^{a_2}$
u_8	$a_{a_4}^{a_1}, b_{b_2}^{b_1}$	$a_{a_4}^{a_1}$	$a_{a_4}^{a_2}, b_{b_3}^{b_2}$	$a_{a_4}^{a_3}, b_{b_2}^{b_1}$
u_9	$a_{a_3}^{a_1}, b_{b_4}^{b_1}$	$a_{a_3}^{a_1}, b_{b_4}^{b_2}$	$a_{a_3}^{a_2}, b_{b_4}^{b_3}$	$b_{b_4}^{b_1}$
u_{10}	$a_{a_2}^{a_1}, b_{b_5}^{b_1}$	$a_{a_2}^{a_1}, b_{b_5}^{b_2}$	$b_{b_5}^{b_3}$	$a_{a_3}^{a_2}, b_{b_5}^{b_1}$

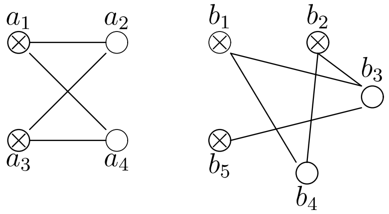

Fig. 1. Coloring of attribute value graphs constructed in Example 9.

Table 11
The reduced table corresponding to graphs shown in Fig. 1

a^c	b^c	d
1	1	0
2	2	0
1	2	1
2	1	1

(6) The graphs are 2-colored, as shown in Fig. 1, where nodes marked by $\otimes$ are colored black and the other nodes are colored white. These colorings generate the following clustering functions:

$$\begin{aligned}c_a(a_1) &= c_a(a_3) = 1\\ c_a(a_2) &= c_a(a_4) = 2\\ c_b(b_1) &= c_b(b_2) = c_b(b_5) = 1\\ c_b(b_3) &= c_b(b_4) = 2.\end{aligned}$$

Given these clustering functions, one can construct a new decision system (see Table 11).

Observe that discretization and symbolic attribute value grouping can be simultaneously used in decision systems including both real-value and symbolic attributes.

2.4. Minimal decision rules

In this section, techniques for constructing minimal rules for decision systems will be considered.

Given a decision table $\mathscr{A}$, a *minimal decision rule* (w.r.t. $\mathscr{A}$) is a rule which is True in $\mathscr{A}$ and which becomes False in $\mathscr{A}$ if any elementary descriptor from the left-hand side of the rule is removed.[9]

The minimal number of elementary descriptors in the left-hand side of a minimal decision rule defines the largest subset of a decision class. Accordingly, information included in the conditional part of any minimal decision rule is sufficient for predicting the decision value of all objects satisfying this part of the rule. The conditional parts of minimal decision rules define the largest object sets relevant for approximating decision classes. The conditional parts of minimal decision rules can be computed using prime implicants.

To compute the set of all minimal rules w.r.t. to a decision system $\mathscr{A} = (U, A, d)$, we proceed as follows, for any object $x \in U$:

(1) Construct a decision-relative discernibility function f_x^r by considering the row corresponding to object x in the decision-relative discernibility matrix for $\mathscr{A}$.
(2) Compute all prime implicants of f_x^r.

[9] A decision rule $\varphi \Rightarrow \psi$ is True in $\mathscr{A}$ if and only if $\|\varphi\|_{\mathscr{A}} \subseteq \|\psi\|_{\mathscr{A}}$.

(3) On the basis of the prime implicants, create minimal rules corresponding to x. To do this, consider the set $A(I)$ of attributes corresponding to propositional variables in I, for each prime implicant I, and construct the rule:

$$\left(\bigwedge_{a\in A(I)}(a=a(x))\right)\Rightarrow d=d(x).$$

The following example illustrates the idea.

Example 10. Consider the decision system $\mathscr{A}$ whose decision table is provided in Table 12. Table 12 contains the values of conditional attributes of vehicles (L, W, C, standing for *Length*, *Width*, and *Color*, respectively), and a decision attribute S standing for *Small* which allows one to decide whether a given vehicle is small.

This system has exactly one decision-relative reduct consisting of attributes L and W. The $\{L, W\}$-reduction of $\mathscr{A}$ as shown in Table 13.

Table 13 results in the following set of non-minimal decision rules:

$$(L=7.0)\wedge(W=\mathsf{large})\Rightarrow(S=\mathsf{no})$$
$$(L=4.0)\wedge(W=\mathsf{medium})\Rightarrow(S=\mathsf{yes})$$
$$(L=5.0)\wedge(W=\mathsf{medium})\Rightarrow(S=\mathsf{no})$$
$$(L=4.5)\wedge(W=\mathsf{medium})\Rightarrow(S=\mathsf{no})$$
$$(L=4.0)\wedge(W=\mathsf{large})\Rightarrow(S=\mathsf{no}).$$

To obtain the minimal decision rules, we apply the construction provided above, for $x\in\{1,\ldots,7\}$.

(1) The decision-relative discernibility functions $f_1^r,\ldots,f_7^r$ are constructed on the basis of the reduced discernibility matrix shown in Table 14:

$$f_1^r\equiv(L\vee W)\wedge(L\vee W\vee C)\equiv(L\vee W)$$
$$f_2^r\equiv(L\vee W\vee C)\wedge(L\vee W\vee C)\equiv(L\vee W\vee C)$$
$$f_3^r\equiv(L\vee W)\wedge(L\vee W\vee C)\wedge(L\vee C)\wedge L\wedge(W\vee C)$$
$$\equiv(L\wedge W)\vee(L\wedge C)$$

Table 12
Decision table considered in Example 10

Object	*L*	*W*	*C*	*S*
1	7.0	large	green	no
2	7.0	large	blue	no
3	4.0	medium	green	yes
4	4.0	medium	red	yes
5	5.0	medium	blue	no
6	4.5	medium	green	no
7	4.0	large	red	no

Table 13
$\{L, W\}$-reduction considered in Example 10

Objects	*L*	*W*	*S*
1, 2	7.0	large	no
3, 4	4.0	medium	yes
5	5.0	medium	no
6	4.5	medium	no
7	4.0	large	no

Table 14
Reduced decision-relative discernibility matrix from Example 10

	3	4
1	L, W	L, W, C
2	L, W, C	L, W, C
5	L, C	L, C
6	L	L, C
7	W, C	W

$$
\begin{aligned}
f_4^r &\equiv (L \vee W \vee C) \wedge (L \vee W \vee C) \wedge (L \vee C) \wedge (L \vee C) \wedge W \\
&\equiv (L \wedge W) \vee (C \wedge W) \\
f_5^r &\equiv (L \vee C) \wedge (L \vee C) \equiv (L \vee C) \\
f_6^r &\equiv L \wedge (L \vee C) \equiv L \\
f_7^r &\equiv (W \vee C) \wedge W \equiv W.
\end{aligned}
$$

(2) The following prime implicants are obtained from formulas $f_1^r, \ldots, f_7^r$:

$$
\begin{aligned}
&f_1^r : L, W \\
&f_2^r : L, W, C \\
&f_3^r : L \wedge W, L \wedge C \\
&f_4^r : L \wedge W, C \wedge W \\
&f_5^r : L, C \\
&f_6^r : L \\
&f_7^r : W.
\end{aligned}
$$

(3) Based on the prime implicants, minimal decision rules are created for objects $1, \ldots, 7$. For instance, from prime implicants L and W corresponding to f_1^r, the following minimal decision rules are generated based on object 1:

$$
\begin{aligned}
&(L = 7.0) \Rightarrow (S = \mathsf{no}) \\
&(W = \mathsf{large}) \Rightarrow (S = \mathsf{no}).
\end{aligned}
$$

On the basis of object 3 and prime implicants $L \wedge W$ and $L \wedge C$ for f_3^r we obtain the following rules:

$$
\begin{aligned}
&(L = 4.0) \wedge (W = \mathsf{medium}) \Rightarrow (S = \mathsf{yes}) \\
&(L = 4.0) \wedge (C = \mathsf{green}) \Rightarrow (S = \mathsf{yes}).
\end{aligned}
$$

Similarly, minimal decision rules can easily be obtained for all other formulas.

In practice, the number of minimal decision rules can be large. One then tries to consider only subsets of these rules or to drop some conditions from minimal rules.

Remark 11. The main challenge in inducing rules from decision systems lies in determining which attributes should be included into the conditional parts of the rules. Using the strategy outlined above, the minimal rules are computed first. Their conditional parts describe the largest object sets with the same generalized decision value in a given decision system. Although such minimal decision rules can be computed, this approach can result in a set of rules of unsatisfactory classification quality. Such rules might appear too general or too specific for classifying new objects. This depends on the data analyzed. Techniques have been developed for the further tuning of minimal rules.

Table 15
Training data considered in Section 2.5

Object	*SL*	*VS*	*W*	*AD*	*Distance*
1	70	60	rain	3.0	small
2	70	70	sun	5.0	medium
3	50	60	rain	5.0	small
4	50	60	sun	9.0	medium
5	30	15	rain	9.0	large
6	30	10	sun	5.0	large
7	70	60	rain	15.0	large
8	50	40	rain	15.0	large

2.5. Example: learning of concepts

Given that one has all the techniques described in the previous sections at one's disposal, an important task is to induce definitions of concepts from training data, where the representation of the definition is as efficient and of high quality as possible. These definitions may then be used as classifiers for the induced concepts.

Let us concentrate on the concept of *Distance* between cars on the road. The rough relation $Distance(x, y, z)$ denotes the approximate distance between vehicles x and y, where $z \in \{\text{small}, \text{medium}, \text{large}, \text{unknown}\}$. Below we simplify the definition somewhat, and consider $Distance(x, z)$ which denotes that the distance between x and the vehicle directly preceding x is z.[10] Assume that sample training data has been gathered in a decision table which is provided in Table 15, where[11]

- *SL* stands for the "speed limit" on a considered road segment;
- *VS* stands for the "vehicle speed";
- *W* stands for "weather conditions";
- *AD* stands for "actual distance" between a given vehicle and its predecessor on the road.

For the sake of simplicity, we concentrate on generating rules to determine whether the distance between two objects is small.

On the basis of the training data, one can compute a discernibility matrix. Since we are interested in rules for the decision small only, it suffices to consider a simplified discernibility matrix with columns labelled by objects 1 and 3, as these are the only two objects, where the corresponding decision is small. The resulting discernibility matrix is shown in Table 16.

The discernibility matrix gives rise to the following discernibility functions:

$$\begin{aligned} f_1 &\equiv (VS \vee W \vee AD) \wedge (SL \vee W \vee AD) \wedge (SL \vee VS \vee AD) \\ &\quad \wedge (SL \vee VS \vee W \vee AD) \wedge AD \wedge (SL \vee VS \vee AD) \\ &\equiv AD \\ f_3 &\equiv (SL \vee VS \vee W) \wedge (W \vee AD) \wedge (SL \vee VS \vee AD) \\ &\quad \wedge (SL \vee VS \vee W) \wedge (SL \vee AD) \wedge (VS \vee AD) \\ &\equiv (W \wedge AD) \vee (SL \wedge AD) \vee (VS \wedge AD) \vee (SL \wedge VS \wedge W). \end{aligned}$$

Based on the discernibility functions, one can easily find prime implicants and obtain the following rules for the decision small[12]:

[10] In fact, here we consider a distance to be small if it causes a dangerous situation, and to be large if the situation is safe.

[11] Of course, real-life sample data would consist of hundreds or thousands of examples.

[12] In practical applications one would have to discretize *AD* before extracting rules.

Table 16
Discernibility matrix of Table 15 for decision small

Object	1	3
2	VS, W, AD	SL, VS, W
4	SL, W, AD	W, AD
5	SL, VS, AD	SL, VS, AD
6	SL, VS, W, AD	SL, VS, W
7	AD	SL, AD
8	SL, VS, AD	VS, AD

Table 17
Information table considered in Example 12

Customer	*Bread*	*Milk*	*Jam*	*Beer*
1	yes	yes	no	no
2	yes	yes	yes	yes
3	yes	yes	yes	no
4	no	yes	yes	no

$$\begin{aligned}&(AD = 3.0) \Rightarrow (Distance = \text{small}) \qquad (6)\\&(W = \text{rain}) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small})\\&(SL = 50) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small})\\&(VS = 60) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small})\\&(SL = 50) \wedge (VS = 60) \wedge (W = \text{rain}) \Rightarrow (Distance = \text{small}).\end{aligned}$$

There have been also developed methods for approximation of compound concepts based on rough sets, hierarchical learning, and ontology approximation (see, e.g., [7,11–13,77,80,79,108,113,115,116]).

2.6. Association rules

In this section [73,75], we show how rough set techniques can be used to extract association rules from information systems. Association rules playing an important role in the field of data mining, provide associations among attributes.[13] A real number from the interval $[0, 1]$ is assigned to each rule and provides a measure of the confidence of the rule. The following example will help to illustrate this.

Example 12. Consider the information table provided in Table 17.

Each row in the table represents items bought by a customer. For instance, customer 1 bought bread and milk, whereas customer 4 bought milk and jam. An association rule that can be extracted from the above table is: *a customer who bought bread also bought milk*. This is represented by

$$(Bread = \text{yes}) \Rightarrow (Milk = \text{yes}).$$

Since all customers who bought bread actually bought milk too, the confidence of this rule is 1. Now consider the rule

$$(Bread = \text{yes}) \wedge (Milk = \text{yes}) \Rightarrow (Jam = \text{yes})$$

stating that a customer who bought bread and milk, bought jam as well. Since three customers bought both bread and milk and two of them bought jam, the confidence of this rule is 2/3.

We now formalize this approach to confidence measures for association rules. Recall that by a template we mean a conjunction of elementary descriptors, i.e., expressions of the form $a = v$, where a is an attribute and $v \in V_a$. For an information system $\mathscr{A}$ and a template T we denote by $support_{\mathscr{A}}(T)$ the number of objects

[13] Association between attributes are also studied using association reducts [117].

satisfying T. Let $\mathscr{A}$ be an information system and $T = D_1 \wedge \cdots \wedge D_m$ be a template. By an *association rule generated from* T, we mean any expression of the form

$$\bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in Q} D_j,$$

where $\{P, Q\}$ is a partition of $\{D_1, \ldots, D_m\}$. By a *confidence of an association rule* $\phi \equiv \bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in Q} D_j$ we mean the coefficient

$$confidence_{\mathscr{A}}(\phi) = \frac{support_{\mathscr{A}}(D_1 \wedge \cdots \wedge D_m)}{support_{\mathscr{A}}\left(\bigwedge_{D_i \in P} D_i\right)}.$$

There are two basic steps used in methods aimed at generating association rules. (Below s and c stand for support and confidence thresholds w.r.t. a given information system $\mathscr{A}$, respectively.)

(1) Generate as many templates $T = D_1 \wedge \cdots \wedge D_k$ as possible, such that $support_{\mathscr{A}}(T) \geqslant s$ and $support_{\mathscr{A}}(T \wedge D_i) < s$, for any descriptor D_i different from all descriptors $D_1, \ldots, D_k$.

(2) Search for a partition $\{P, Q\}$ of T, for each T generated in the previous step, satisfying
(a) $support_{\mathscr{A}}(P) < \frac{support_{\mathscr{A}}(T)}{c}$.
(b) P has the shortest length among templates satisfying (a).
Every such partition leads to an association rule of the form $P \Rightarrow Q$ whose confidence is greater than c.

The second step, crucial to the process of extracting association rules, can be solved using rough set methods.

Let $T = D_1 \wedge D_2 \wedge \cdots \wedge D_m$ be a template such that $support_{\mathscr{A}}(T) \geqslant s$. For a given confidence threshold $c \in [0, 1]$, the association rule $\phi \equiv P \Rightarrow Q$ is called *c-irreducible* if $confidence_{\mathscr{A}}(P \Rightarrow Q) \geqslant c$ and for any association rule $\phi' \equiv P' \Rightarrow Q'$ such that P' is a sub-formula of P, we have

$$confidence_{\mathscr{A}}(P' \Rightarrow Q') < c.$$

The problem of searching for c-irreducible association rules from a given template is equivalent to the problem of searching for α-reducts in a decision table, for some $\alpha \in [0, 1]$ (see Section 2.1).

Let $\mathscr{A}$ be an information system and $T = D_1 \wedge D_2 \wedge \cdots \wedge D_m$ be a template. By a *characteristic table* for T w.r.t. $\mathscr{A}$, we understand a decision system $\mathscr{A}|_T = (U, A|_T, d)$, where

(1) $A|_T = \{a_{D_1}, a_{D_2}, \ldots, a_{D_m}\}$ 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 a template T, i.e.,

$$d(u) = \begin{cases} 1, & \text{if the object } u \text{ satisfies } T, \\ 0, & \text{otherwise.} \end{cases}$$

The following property provides the relationship between association rules and approximations of reducts.

For a given information system $\mathscr{A} = (U, A)$, a template $T = D_1 \wedge D_2 \wedge \cdots \wedge D_m$ and a set of descriptors $P \subseteq \{D_1, \ldots, D_m\}$, the association rule

$$\bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in \{D_1, \ldots, D_m\} - P} D_j,$$

is

(1) a 1-irreducible association rule from T if and only if $\bigcup_{D_i \in P}\{a_{D_i}\}$ is a decision-relative reduct of $\mathscr{A}|_T$;
(2) a c-irreducible association rule from T if and only if $\bigcup_{D_i \in P}\{a_{D_i}\}$ is an α-reduct of $\mathscr{A}|_T$, where

$$\alpha = 1 - \left[\left(\frac{1}{c} - 1\right) \Big/ \left(\frac{|U|}{support_{\mathscr{A}}(T)} - 1\right)\right].$$

The problem of searching for the shortest association rules is NP-hard.

The following example illustrates the main ideas used in the searching method for association rules.

Example 13. Consider the information table $\mathscr{A}$ with 18 objects and 9 attributes presented in Table 18.

Consider the template

$$T = (a_1 = 0) \wedge (a_3 = 2) \wedge (a_4 = 1) \wedge (a_6 = 0) \wedge (a_8 = 1). \tag{7}$$

It is easily seen that $support_{\mathscr{A}}(T) = 10$. The new constructed decision table $\mathscr{A}|_T$ is presented in Table 19.

The reduced discernibility matrix $\mathscr{A}|_T$ is provided in Table 20, where for simplicity, the second column represents, in fact, 10 columns with identical contents, labeled by $u_2, u_3, u_4, u_8, u_9, u_{10}, u_{13}, u_{15}, u_{16}, u_{17}$, respectively. Given the discernibility matrix, one can easily compute the discernibility function $\mathscr{A}|_T$ for $\mathscr{A}|_T$:

$$\begin{aligned} f_{\mathscr{A}_T}(a_{D_1}, a_{D_2}, a_{D_3}, a_{D_4}, a_{D_5}) \equiv\ & (a_{D_2} \vee a_{D_4} \vee a_{D_5}) \wedge (a_{D_1} \vee a_{D_3} \vee a_{D_4}) \wedge (a_{D_2} \vee a_{D_3} \vee a_{D_4}) \\ & \wedge (a_{D_1} \vee a_{D_2} \vee a_{D_3} \vee a_{D_4}) \wedge (a_{D_1} \vee a_{D_3} \vee a_{D_5}) \wedge (a_{D_2} \vee a_{D_3} \vee a_{D_5}) \\ & \wedge (a_{D_3} \vee a_{D_4} \vee a_{D_5}) \wedge (a_{D_1} \vee a_{D_5}), \end{aligned}$$

where D_i denotes the ith conjunct of (7).

The discernibility function has the following prime implicants: $a_{D_3} \wedge a_{D_5}$, $a_{D_4} \wedge a_{D_5}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_3}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_4}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_5}$, $a_{D_1} \wedge a_{D_3} \wedge a_{D_4}$. This gives rise to the reducts: $\{a_{D_3}, a_{D_5}\}$, $\{a_{D_4}, a_{D_5}\}$, $\{a_{D_1}, a_{D_2}, a_{D_3}\}$, $\{a_{D_1}, a_{D_2}, a_{D_4}\}$, $\{a_{D_1}, a_{D_2}, a_{D_5}\}$, $\{a_{D_1}, a_{D_3}, a_{D_4}\}$. Thus, there are six association rules with confidence 1, i.e., 1-irreducible:

$$\begin{aligned} & D_3 \wedge D_5 \Rightarrow D_1 \wedge D_2 \wedge D_4 \\ & D_4 \wedge D_5 \Rightarrow D_1 \wedge D_2 \wedge D_3 \\ & D_1 \wedge D_2 \wedge D_3 \Rightarrow D_4 \wedge D_5 \\ & D_1 \wedge D_2 \wedge D_4 \Rightarrow D_3 \wedge D_5 \\ & D_1 \wedge D_2 \wedge D_5 \Rightarrow D_3 \wedge D_4 \\ & D_1 \wedge D_3 \wedge D_4 \Rightarrow D_2 \wedge D_5. \end{aligned}$$

Table 18
Information table $\mathscr{A}$ considered in Example 13

$\mathscr{A}$	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

Table 19
Decision table $\mathscr{A}|_T$ considered in Example 13

$\mathscr{A}\|_T$	a_{D_1} $(a_1 = 0)$	a_{D_2} $(a_3 = 2)$	a_{D_3} $(a_4 = 1)$	a_{D_4} $(a_6 = 0)$	a_{D_5} $(a_8 = 1)$	d
u_1	1	0	1	0	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	0
u_6	1	0	0	0	1	0
u_7	0	0	0	0	1	0
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	0
u_{12}	1	0	0	1	0	0
u_{13}	1	1	1	1	1	1
u_{14}	1	1	0	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	0

Table 20
Reduced discernibility matrix for $\mathscr{A}|_T$ from Example 13

$\mathscr{M}(\mathscr{A}\|_T)$	$u_2, u_3, u_4, u_8, u_9 u_{10}, u_{13}, u_{15}, u_{16}, u_{17}$
u_1	$a_{D_2}, a_{D_4}, a_{D_5}$
u_5	$a_{D_1}, a_{D_3}, a_{D_4}$
u_6	$a_{D_2}, a_{D_3}, a_{D_4}$
u_7	$a_{D_1}, a_{D_2}, a_{D_3}, a_{D_4}$
u_{11}	$a_{D_1}, a_{D_3}, a_{D_5}$
u_{12}	$a_{D_2}, a_{D_3}, a_{D_5}$
u_{14}	$a_{D_3}, a_{D_4}, a_{D_5}$
u_{18}	a_{D_1}, a_{D_5}

For confidence 0.9, we look for α-reducts for the decision table $\mathscr{A}|_T$, where

$$\alpha = 1 - \left(\frac{1}{0.9} - 1\right) \Big/ \left(\frac{18}{10} - 1\right) \approx 0.86.$$

Hence, we look for a set of descriptors that covers at least $\lceil (18 - 10) * \alpha \rceil = \lceil 8 * 0.86 \rceil = 7$ elements of the discernibility matrix $\mathscr{M}(\mathscr{A}|_{\mathbf{T}})$. 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 intersections with exactly 7 members of the discernibility matrix $\mathscr{M}(\mathscr{A}|_T)$. Consequently, the 0.9-irreducible association rules obtained from those sets are the following:

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

The technique illustrated by this example can be applied to find useful dependencies between attributes in complex application domains. In particular, one could use such dependencies in constructing robust classifiers conforming to the laws of the underlying reality.

2.7. Approximate Boolean reasoning: discretization of large data sets stored in relational databases

In this section (see [70,71]), we discuss an application of approximate Boolean reasoning to efficient searching for cuts in large data sets stored in relational databases. Searching for relevant cuts is based on simple statistics which can be efficiently extracted from relational databases. This additional statistical knowledge is making it possible to perform the searching based on Boolean reasoning much more efficient. It can be shown that the extracted cuts by using such reasoning are quite close to optimal.

Searching algorithms for optimal partitions of real-valued attributes, defined by cuts, have been intensively studied. The main goal of such algorithms is to discover cuts which can be used to synthesize decision trees or decision rules of high quality w.r.t. some quality measures (e.g., quality of classification of new unseen objects, quality defined by the decision tree height, support and confidence of decision rules).

In general, all those problems are hard from computational point of view (e.g., the searching problem for minimal and consistent set of cuts is NP-hard). In consequence, numerous heuristics have been developed for approximate solutions of these problems. These heuristics are based on approximate measures estimating the quality of extracted cuts. Among such measures *discernibility measures* are relevant for the rough set approach.

We outline an approach for solution of a searching problem for optimal partition of real-valued attributes by cuts, assuming that the large data table is represented in a relational database. In such a case, even the linear time complexity w.r.t. the number of cuts is not acceptable because of the time needed for one step. The critical factor for time complexity of algorithms solving that problem is the number of SQL queries of the form

```
SELECT COUNT FROM aTable WHERE (anAttribute BETWEEN value1 AND value2) AND (additional condition)
```

necessary to construct partitions of real-valued attribute sets. We assume the answer time for such queries does not depend on the interval length.[14] Using a straightforward approach to optimal partition selection (w.r.t. a given measure), the number of necessary queries is of order O(N), where N is the number of preassumed cuts. By introducing some optimization measures, it is possible to reduce the size of searching space. Moreover, using only O($\log N$) simple queries, suffices to construct a partition very close to optimal.

Let $\mathscr{A} = (U, A, d)$ be a decision system with real-valued condition attributes. Any cut (a, c), where $a \in A$ and c is a real number, defines two disjoint sets given by

$$U_L(a,c) = \{x \in U : a(x) \leqslant c\};$$
$$U_R(a,c) = \{x \in U : a(x) > c\}.$$

If both $U_L(a,c)$ and $U_R(a,c)$ are non-empty, then c is called a *cut on attribute a*. The cut (a,c) discerns a pair of objects x, y if either $a(x) < c \leqslant a(y)$ or $a(y) < c \leqslant a(x)$.

Let $\mathscr{A} = (U, A, d)$ be a decision system with real-valued condition attributes and decision classes X_i, for $i = 1, \ldots, r(d)$. A *quality of a cut* (a,c), denoted by $W(a,c)$, is defined by

$$W(a,c) = \sum_{i \neq j}^{r(d)} L_i(a,c) * R_j(a,c) = \left(\sum_{i=1}^{r(d)} L_i(a,c)\right) * \left(\sum_{i=1}^{r(d)} R_i(a,c)\right) - \sum_{i=1}^{r(d)} L_i(a,c) * R_i(a,c), \tag{8}$$

where $L_i(a,c) = card(X_i \cap U_L(a,c))$ and $R_i(a,c) = card(X_i \cap U_R(a,c))$, for $i = 1, \ldots, r(d)$.

In the sequel, we will be interested in finding cuts maximizing the function $W(a,c)$.

The following definition will be useful. Let $\mathscr{C}_a = \{(a,c_1), \ldots, (a,c_N)\}$ be a set of cuts on attribute a, over a decision table $\mathscr{A}$ and assume $c_1 < c_2 \cdots < c_N$. By a *median of the* ith *decision class*, denoted by $Median(i)$, we

[14] This assumption is satisfied in some existing database management systems.

mean the minimal index j for which the cut $(a, c_j) \in \mathcal{C}_a$ minimizes the value $|L_i(a, c_j) - R_i(a, c_j)|$,[15] where L_i and R_i are defined before.

One can use only $O(r(d)*\log N)$ SQL queries to determine the medians of decision classes by using the well-known binary search algorithm.

Then one can show that the quality function $W_a(i) \overset{\text{def}}{=} W(a, c_i)$, for $i = 1, \ldots, N$, is increasing in $\{1, \ldots, min\}$ and decreasing in $\{max, \ldots, N\}$, where min and max are defined by

$$min = \min_{1 \leqslant i \leqslant N} Median(i);$$

$$max = \max_{1 \leqslant i \leqslant N} Median(i).$$

In consequence, the search space for maximum of $W(a, c_i)$ is reduced to $i \in [min, max]$.

Now, one can apply the divide and conquer strategy to determine the best cut, given by $c_{Best} \in [c_{min}, c_{max}]$, w.r.t. the chosen quality function. First, we divide the interval containing all possible cuts into k intervals. Using some heuristics, one then predict the interval which most probably contains the best cut. This process is recursively applied to that interval, until the considered interval consists of one cut. The problem which remains to be solved is how to define such approximate measures which could help us to predict the suitable interval.

Let us consider a simple probabilistic model. Let (a, c_L), (a, c_R) be two cuts such that $c_L < c_R$ and $i = 1, \ldots, r(d)$. For any cut (a, c) satisfying $c_L < c < c_R$, we assume that $x_1, \ldots, x_{r(d)}$, where $x_i = card(X_i \cap U_L(a, c) \cap U_R(a, c))$ are independent random variables with uniform distribution over sets $\{0, \ldots, M_1\}, \ldots, \{0, \ldots, M_{r(d)}\}$, respectively, that

$$M_i = M_i(a, c_L, c_R) = card(X_i \cap U_L(a, c_R) \cap U_R(a, c_L)).$$

Under these assumptions the following fact holds. For any cut $c \in [c_L, c_R]$, the mean $E(W(a, c))$ of quality $W(a, c)$, is given by

$$E(W(a,c)) = \frac{W(a, c_L) + W(a, c_R) + conflict((a, c_L), (a, c_R))}{2}, \tag{9}$$

where $conflict((a, c_L), (a, c_R)) = \sum_{i \neq j} M_i * M_j$.

In addition, the standard deviation of $W(a, c)$ is given by

$$D^2(W(a,c)) = \sum_{i=1}^{n} \left[\frac{M_i(M_i + 2)}{12} \left(\sum_{j \neq i} (R_j(a, c_R) - L_j(a, c_L)) \right)^2 \right]. \tag{10}$$

Formulas (9) and (10) can be used to construct a predicting measure for the quality of the interval $[c_L, c_R]$:

$$Eval([c_L, c_R], \alpha) = E(W(a,c)) + \alpha \sqrt{D^2(W(a,c))}, \tag{11}$$

where the real parameter $\alpha \in [0, 1]$ can be tuned in a learning process.

To determine the value $Eval([c_L, c_R], \alpha)$, we need to compute the numbers

$$L_1(a, c_L), \ldots, L_{r(d)}(a, c_L), M_1, \ldots, M_{r(d)}, R_1(a, c_R), \ldots, R_{r(d)}(a, c_R).$$

This requires $O(r(d))$ SQL queries of the form

```
SELECT COUNT
FROM DecTable
WHERE (attribute_a BETWEEN value_1 AND value_2) AND (dec = i).
```

Hence, the number of queries required for running this algorithm is

$$O(r(d) k \log_k N).$$

[15] The minimization means that $|L_i(a, c_j) - R_i(a, c_j)| = \min_{1 \leqslant k \leqslant N} |L_i(a, c_k) - R_i(a, c_k)|$.

In practice, we set $k = 3$, since the function $f(k) = r(d)k\log_k N$ over positive integers is taking minimum for $k = 3$.

Numerous experiments on different data sets have shown that the proposed solution allows one to find a cut which is very close to the optimal one. For more details the reader is referred to the literature (see [70,71]).

3. Conflicts

Knowledge discovery in databases considered in the previous sections reduces to searching for functional dependencies in the data set. In this section, we will discuss another kind of relationship in the data – not dependencies, but conflicts. Formally, the conflict relation can be seen as a negation (not necessarily classical) of indiscernibility relation which was used as a basis of rough set theory. Thus indiscernibility and conflict are closely related from logical point of view. It turns out that the conflict relation can be used to the conflict analysis study. Conflict analysis and resolution play an important role in business, governmental, political and lawsuits disputes, labor-management negotiations, military operations, and others. To this end many mathematical formal models of conflict situations have been proposed and studied, e.g., [16,19,20,27,49,50,52,61,66,81,90,129]. Various mathematical tools, e.g., graph theory, topology, differential equations have been used for that purpose. Needless to say, game theory can also be considered as a mathematical model of conflict situations. In fact, there is no "universal" theory of conflicts yet, and mathematical models of conflict situations are strongly domain dependent. In this paper, we are going to present yet another approach to conflict analysis – based on some ideas of rough set theory – along the lines of [90]. The considered model is simple enough for easy computer implementation and seems to be adequate for many real life applications.

3.1. Basic concepts of conflict theory

In this section, we give definitions of basic concepts of the proposed approach in lines with [90].

Let us assume that we are given a finite, non-empty set Ag called the *universe*. Elements of Ag will be referred to as *agents*. Let a *voting function* $v : Ag \to \{-1, 0, 1\}$ be given, assigning to every agent one of the number $-1, 0$ or 1, which represents his opinion, view, voting result, etc. about some discussed issue, and meaning *against, neutral* and *favorable*, respectively.

Voting functions correspond to situations. Hence, let us assume there is given a set of situations U and a set of voting functions $Voting\_Fun$ as well as a *conflict function* $Conflict : U \to Voting\_Fun$. Any pair $S = (s, v)$, where $s \in U$ and $v = Conflict(s)$ will be called a *conflict situation*.

In order to express relations between agents from Ag, defined by a given voting function v we define three basic binary relations in Ag^2 : *conflict, neutrality*, and *alliance*.

To this end, we first define the following auxiliary function:

$$\phi_v(ag, ag') = \begin{cases} 1, & \text{if } v(ag)v(ag') = 1 \quad \text{or} \quad ag = ag', \\ 0, & \text{if } v(ag)v(ag') = 0 \quad \text{and} \quad ag \neq ag', \\ -1, & \text{if } v(ag)v(ag') = -1. \end{cases} \tag{12}$$

This means that if $\phi_v(ag, ag') = 1$, then agents ag and ag' have the same opinion about issue v (are *allied* on v); $\phi_v(ag, ag') = 0$ means that at least one of agents ag or ag' has neutral approach to issue v (is *neutral* on v), and $\phi_v(ag, ag') = -1$, means that both agents have different opinions about issue v (are in *conflict* on v).

In what follows we will define three basic binary relations R_v^+, $R_v^0, R_v^- \subseteq Ag^2$ called *alliance, neutrality* and *conflict* relations, respectively, and defined by

$$\begin{aligned} &R_v^+(ag, ag') \quad \text{iff} \quad \phi_v(ag, ag') = 1 \\ &R_v^0(ag, ag') \quad \text{iff} \quad \phi_v(ag, ag') = 0 \\ &R_v^-(ag, ag') \quad \text{iff} \quad \phi_v(ag, ag') = -1. \end{aligned} \tag{13}$$

It is easily seen that the alliance relation has the following properties:

$$\begin{aligned}&R_v^+(ag, ag)\\ &R_v^+(ag, ag') \quad \text{implies} \quad R_v^+(ag', ag)\\ &R_v^+(ag, ag') \quad \text{and} \quad R_v^+(ag', ag'') \quad \text{implies} \quad R_v^+(ag, ag''),\end{aligned} \tag{14}$$

i.e., R_v^+ is an *equivalence* relation. Each equivalence class of alliance relation will be called a *coalition* with respect to v. Let us note that the last condition in (14) can be expressed as "a friend of my friend is my friend".

For the conflict relation we have the following properties:

$$\begin{aligned}&\text{not } R_v^-(ag, ag)\\ &R_v^-(ag, ag') \quad \text{implies} \quad R_v^-(ag', ag)\\ &R_v^-(ag, ag') \quad \text{and} \quad R_v^-(ag', ag'') \quad \text{implies} \quad R_v^+(ag, ag'')\\ &R_v^-(ag, ag') \quad \text{and} \quad R_v^+(ag', ag'') \quad \text{implies} \quad R_v^-(ag, ag'').\end{aligned} \tag{15}$$

The last two conditions in (15) refer to well known sayings "an enemy of my enemy is my friend" and "a friend of my enemy is my enemy".

For the neutrality relation we have:

$$\begin{aligned}&\text{not } R_v^0(ag, ag)\\ &R_v^0(ag, ag') = R_v^0(ag', ag).\end{aligned} \tag{16}$$

Let us observe that there are no coalitions in the conflict and neutrality relations.

We have $R_v^+ \cup R_v^0 \cup R_v^- = Ag^2$ because for any pair of agents $(ag, ag') \in Ag^2$, $\Phi_v(ag, ag') = 1$ or $\Phi_v(ag, ag') = 0$ or $\Phi_v(ag, ag') = -1$, so $(ag, ag') \in R_v^+$ or $(ag, ag') \in R_v^-$ or $(ag, ag') \in R_v^-$. All the three relations R_v^+, R_v^0 , R_v^- are pairwise disjoint, i.e., every pair of objects (ag, ag') belongs to exactly one of relation (is in conflict, is allied or is neutral).

With every conflict situation $S = (s, v)$ we will associate a *conflict graph*

$$G_S = (R_v^+, R_v^0, R_v^-). \tag{17}$$

An example of a conflict graph is shown in Fig. 2. Solid lines denote conflicts, whereas dotted line – alliance.[16] Clearly, B, C, and D form a coalition.

A conflict degree $Con(S)$ of the conflict situation $S = (s, v)$ is defined by

$$Con(S) = \frac{\sum_{\{(ag,ag'):\ \phi_v(ag,ag')=-1\}} |\phi_v(ag, ag')|}{2\lceil \frac{n}{2} \rceil \times (n - \lceil \frac{n}{2} \rceil)}, \tag{18}$$

where $n = card(Ag)$.

One can consider a more general case of a conflict function viz., a mapping of the form *Conflict*: $U \to Voting\_Fun^k$, where k is a positive integer. Then, a conflict situation is any pair $S = (s, (v_1, \ldots, v_k))$, where $(v_1, \ldots, v_k) = Conflict(s)$, and the conflict degree in S can be defined by

$$Con(S) = \frac{\sum_{i=1}^{k} Con(S_i)}{k}, \tag{19}$$

where $S_i = (s, v_i)$ for $i = 1, \ldots, k$. Each function v_i is called a voting function on the ith issue in s.

3.2. An example

In this section, we will illustrate the ideas presented above by means of a very simple tutorial example using concepts presented previously. We consider a conflict situation $S = (s, v)$, where the domain ag of the voting function v is defined by $Ag = \{(1, A), \ldots, (240, A), (241, B), \ldots, (280, B), (281, C), \ldots, (340, C), (341, D), \ldots, (500, D)\}$ and $v(1, A) = \cdots = v(200, A) = 1$, $v(201, A) = \cdots = v(230, A) = 0$, $v(231, A) = \cdots = v(240, A) = -1$, $v(241, B) = \cdots =$

[16] For simplicity, the neutrality is not shown explicitly in the graph.

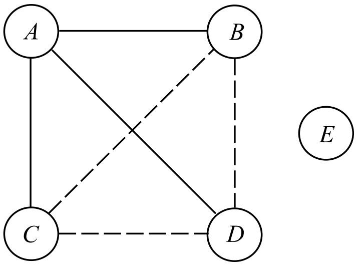

Fig. 2. Exemplary conflict graph.

$v(255, B) = 1$, $v(256, B) = \cdots = v(280, B) = -1$, $v(281, C) = \cdots = v(300, C) = 0$, $v(301, C) = \cdots = v(340, C) = -1$, $v(341, D) = \cdots = v(365, D) = 1$, $v(366, D) = \cdots = v(400, D) = 0$, $v(401, D) = \cdots = v(500, D) = -1$.

This conflict situation is presented in Table 21, where 1 and -1 are abbreviated by '+', '−', respectively. The maximal coalitions in this conflict situation are $v^{-1}(\{1\})$ and $v^{-1}(\{-1\})$.

If one would like to keep only party name, then Table 21 can be represented as it is shown in Table 22. This table presents a decision table in which the only condition attribute is *Party*, whereas the decision attribute is *Voting*. The table describes voting results in a parliament containing 500 members grouped in four political parties denoted A, B, C and D. Suppose the parliament discusses certain issue (e.g., membership of the country in European Union) and the voting result is presented in column *Voting*, where $+$, 0 and $-$ denoted *yes*, *abstention* and *no*, respectively. The column *support* contains the number of voters for each option.

Table 21
Conflict situation with agents (*Member, Party*) and the voting function *Voting*

(*Member, Party*)	*Voting*
(1, A)	+
...	...
(200, A)	+
(201, A)	0
...	...
(230, A)	0
(231, A)	−
...	...
(240, A)	−
(241, B)	+
...	...
(255, B)	+
(256, B)	−
...	...
(280, B)	−
(281, C)	0
...	...
(300, C)	0
(301, C)	−
...	...
(340, C)	−
(341, D)	+
...	...
(365, D)	+
(366, D)	0
...	...
(400, D)	0
(401, D)	−
...	...
(500, D)	−

Table 22
Decision table with one condition attribute *Party* and the decision *Voting*

Fact	*Party*	*Voting*	*Support*
1	*A*	+	200
2	*A*	0	30
3	*A*	−	10
4	*B*	+	15
5	*B*	−	25
6	*C*	0	20
7	*C*	−	40
8	*D*	+	25
9	*D*	0	35
10	*D*	−	100

Table 23
The certainty and coverage factors for Table 22

Fact	*Strength*	*Certainty*	*Coverage*
1	0.40	0.83	0.83
2	0.06	0.13	0.35
3	0.02	0.04	0.06
4	0.03	0.36	0.06
5	0.05	0.63	0.14
6	0.04	0.33	0.23
7	0.08	0.67	0.23
8	0.05	0.16	0.10
9	0.07	0.22	0.41
10	0.20	0.63	0.57

The strength, the certainty and the coverage factors for Table 22 are given in Table 23. The certainty and coverage factors have now a natural interpretation for the considered conflict situation.

From the certainty factors we can conclude, for example, that

- 83.3% of party *A* voted *yes*;
- 12.5% of party *A abstained*;
- 4.2% of party *A* voted *no*.

From the coverage factors we can get, for example, the following explanation of voting results:

- 83.3% *yes* votes came from party *A*;
- 6.3% *yes* votes came from party *B*;
- 10.4% *yes* votes came from party *C*.

3.3. Conflicts and rough sets

There are strong relationships between the approach to conflicts presented in Section 3.1 and the rough set approach. In this section, we discuss examples of such relationships. The approach presented in this section seems to be very promising for solving problems related to conflict resolution and negotiations (see, e.g., [49,50,52,129]).

The application of rough sets can bring new results in the area related to conflict resolution and negotiations between agents because this makes it possible to introduce approximate reasoning about vague concepts into the area.

Now, we would like to outline this possibility.

First, let us observe that any conflict situation $S = (s, V)$, where $V = (v_1, \ldots, v_k)$ and each v_i is defined on the set of agents $Ag = \{ag_1, \ldots, ag_n\}$, can be treated as an information system $\mathscr{A}(S)$ with the set of objects Ag and the set of attributes $\{v_1, \ldots, v_k\}$. The discernibility degree between agents ag and ag' in S can be defined by

$$disc_S(ag, ag') = \frac{\sum_{\{i:\ \phi_{v_i}(ag,ag')=-1\}} |\phi_{v_i}(ag, ag')|}{k}, \tag{20}$$

where $ag, ag' \in Ag$ and $\cdot$ denotes the absolute value.

Now, one can consider reducts of $\mathscr{A}(S)$ relative to the discernibility degrees defined by $disc_S$. For example, one can consider agents $ag, ag^{'}$ as discernible if

$$disc_S(ag, ag') \geqslant tr,$$

where tr is a given threshold.[17] Any reduct $R \subseteq V$ of S is a minimal set of voting functions preserving on R to a degree at least tr the discernibility in voting between any two agents which are discernible on V to degree at least tr. All voting functions from V–R are dispensable with respect to such a preservation of discernibility degrees between agents.

Reducts of the information system $\mathscr{A}_T(S)$ with the universe of objects equal to $\{v_1, \ldots, v_k\}$ and attributes defined by agents and voting functions by $ag(v) = v(ag)$, for $ag \in Ag$ and $v \in V$, can be considered in an analogous way. The discernibility degree between voting functions can be defined, e.g., by

$$disc(v, v') = |Con(S_v) - Con(S_{v'})|, \tag{21}$$

and it can be used to measure the difference between voting functions v and v'.

Any reduct R of $\mathscr{A}_T(S)$ is a minimal set of agents preserving on R to a degree at least tr the discernibility degree between any two voting functions which are discernible on Ag to degree at least tr.

In our next example, we extend the model of conflict by adding a set A of (condition) attributes used to describe the situations in terms of values of attributes from A. The set of given situations is denoted by U. In this way, we have defined an information system (U, A). Let us assume that there is also given a set of agents Ag. Each agent $ag \in Ag$ has access to a subset $A_{ag} \subseteq A$ of condition attributes. Moreover, we assume that $A = \cup_{ag \in Ag} A_{ag}$. We assume that there is also defined a decision attribute d on U such that $d(s)$ is a conflict situation $S = (s, V)$, where $V = (v_1, \ldots, v_k)$. Observe that $S = (s, V)$ can be represented by a matrix

$$[v_i(ag_j)]_{i=1,\ldots,n;j=1,\ldots,k},$$

where $v_i(ag_j)$ is the result of voting by jth agent on the ith issue. Such a matrix is a compound decision[18] corresponding to s. For the constructed decision system (U, A, d) one can use, e.g., the function given by (19) to measure conflict degrees of situations from U. Let us mention one more kind of reducts which have a natural interpretation in conflict analysis. Such reducts preserve the discernibility between any two A-discernible situations such that the absolute value of the difference between corresponding to them conflict degrees is at least equal tr, where tr is a given threshold.

The described decision table can also be used in conflict resolution. We would like to illustrate this possibility. First, let us recall some notation. For $B \subseteq A$, we denote by $Inf_B(s)$ the B-signature of the situation s, i.e., the set $\{(a, a(s)) : a \in A\}$. Let $INF(B) = \{Inf_B(s) : s \in U\}$. Let us also assume, that for each agent $ag \in Ag$ there is given a similarity relation $\tau_{ag} \subseteq INF(A_{ag}) \times INF(A_{ag})$. In terms of these similarity relations one can consider the problem of conflict resolution relative to a given threshold tr in a given situation s described by $Inf_A(s)$. This is a searching problem for a situation s', if such a situation exists, satisfying the following conditions:

(1) $Inf_A(s')|A_{ag} \in \tau_{ag}(Inf_{A_{ag}}(s))$, where $\tau_{ag}(Inf_{A_{ag}}(s))$ is the tolerance class of $Inf_{A_{ag}}(s)$ with respect to τ_{ag} and $Inf_A(s')|A_{ag}$ denotes the restriction of $Inf_A(s')$ to A_{ag}.

[17] To compute such reducts one can follow the method presented in [114], assuming that any entry of the discernibility matrix corresponding to (ag, ag') with $disc(ag, ag') < tr$ is empty and the remaining entries are families of all minimal subsets of V on which the discernibility between (ag, ag') is at least equal to tr [21].

[18] For references to other papers on compound decision the reader is referred, e.g., to [6,8].

(2) $Inf_A(s')$ satisfies given local constraints (e.g., specifying coexistence of local situations [21,96,123]) and given global constraints (e.g., specifying quality of global situations [21]).
(3) The conflict degree in the conflict situation $d(s')$,[19] measured by means of the chosen conflict measure,[20] is at most tr.

In searching for conflict resolution one can apply methods based on Boolean reasoning (see Section 2 and [21]).

We have proposed changes in the acceptability by agents to be expressed by similarity relations. Observe that in real-life applications, these similarities are more compound than it was suggested above, i.e., they are not defined directly by sensory concepts describing situations. However, they are often specified by high level concepts (see, e.g., [49,91,116]). These high level concepts can be vague, and they are linked with the sensory concepts describing situations by means of a hierarchy of other vague concepts. Approximation of vague concepts in the hierarchy and dependencies between them (see [91]) makes it possible to approximate the similarity relations. This allows us to develop searching methods for acceptable value changes of sensory concepts preserving similarities (constraints) specified over high level vague concepts. One can also introduce some costs of changes of local situations by agents and search for new situations accessible under minimal or sub-minimal costs.

Using the rough set approach to conflict resolution and negotiations between agents, one can also consider more advanced models in which actions and plans performed by agents or their teams are involved in negotiations and conflict resolution. This is one of many interesting directions for further research study of the relationships between rough sets and conflicts.

4. Conclusions

We have discussed the methodology based on discernibility and Boolean reasoning for efficient computation of different entities including reducts and decision rules. We have also outlined a promising research direction based on approximate Boolean reasoning. Exemplary applications of the developed methods used for solving problems in pattern recognition, machine learning, and data mining are included.

We have also presented an approach to conflict analysis based on rough sets. This approach is based on properties of (in)discernibility.

Acknowledgement

The research of Andrzej Skowron has been supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

Many thanks to Professors James Peters, Dominik Ślęzak, and Anna Gomolińska for their incisive comments and for suggesting many helpful ways to improve this article.

References

[1] J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (Eds.), Third International Conference on Rough Sets and Current Trends in Computing (RSCTC'2002), Malvern, PA, October 14–16, 2002, Lecture Notes in Artificial Intelligence, vol. 2475, Springer-Verlag, Heidelberg, 2002.
[2] A. An, Y. Huang, X. Huang, N. Cercone, Feature selection with rough sets for web page classification, in: Peters et al. [95], pp. 1–13.
[3] M. Banerjee, S.K. Pal, Roughness of a fuzzy set, Information Sciences 93 (3–4) (1996) 235–246.
[4] J. Bazan, R. Latkowski, M. Szczuka, DIXER – Distributed executor for rough set exploration system, in: Ślęzak et al. [121], pp. 362–371.
[5] J. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak, J. Wróblewski, Rough set algorithms in classification problems, in: Polkowski et al. [98], pp. 49–88.

[19] Let us observe that s' is not necessarily in U. In such a case, the value $d(s')$ should be predicted by the classifier induced from (U, A, d).
[20] For example, one can consider (19).

[6] J. Bazan, A. Osmólski, A. Skowron, D. Ślęzak, M. Szczuka, J. Wróblewski, Rough set approach to the survival analysis, in: Alpigini et al. [1], pp. 522–529.

[7] J. Bazan, A. Skowron, On-line elimination of non-relevant parts of complex objects in behavioral pattern identification, in: Pal et al. [82], pp. 720–725.

[8] J. Bazan, A. Skowron, D. Ślęzak, J. Wróblewski, Searching for the complex decision reducts: the case study of the survival analysis, in: N. Zhong, Z.W. Ras, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems, 14th International Symposium on Methodologies for Intelligent Systems (ISMIS 2003), Maebashi City, Japan, October 28–31, 2003, Lecture Notes in Artificial Intelligence, vol. 2871, Springer, Heidelberg, 2003, pp. 160–168.

[9] J. Bazan, M. Szczuka, M. Wojna, M. Wojnarski, On the evolution of rough set exploration system, in: S. Tsumoto, R. Słowiński, J. Komorowski, J. Grzymała-Busse (Eds.), Proceedings of the 4th International Conference on Rough Sets and Current Trends in Computing (RSCTC'2004), Uppsala, Sweden, June 1–5, 2004, Lecture Notes in Artificial Intelligence, vol. 3066, Springer-Verlag, Heidelberg, 2004, pp. 592–601.

[10] J.G. Bazan, A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables, in: Polkowski and Skowron [99], pp. 321–365.

[11] J.G. Bazan, H.S. Nguyen, J.F. Peters, A. Skowron, M. Szczuka, Rough set approach to pattern extraction from classifiers, in: A. Skowron, M. Szczuka (Eds.), Proceedings of the Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS 2003, April 12–13, 2003, Electronic Notes in Computer Science, vol. 82 (4), Elsevier, Amsterdam, Netherlands, 2003, pp. 20–29. Available from: <http://www.elsevier.nl/locate/entcs/volume82.html>.

[12] J.G. Bazan, J.F. Peters, A. Skowron, Behavioral pattern identification through rough set modelling, in: Ślęzak et al. [121], pp. 688–697.

[13] J.G. Bazan, A. Skowron, Classifiers based on approximate reasoning schemes, in: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, M. Szczuka (Eds.), Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004), Advances in Soft Computing, Springer, Heidelberg, 2005, pp. 191–202.

[14] J.G. Bazan, M. Szczuka, RSES and RSESlib – a collection of tools for rough set computations, in: Ziarko and Yao [137], pp. 106–113.

[15] F. Brown, Boolean Reasoning, Kluwer Academic Publishers, Dordrecht, 1990.

[16] J.L. Casti, Alternate Realities? Mathematical Models of Nature and Man, John Wiley & Sons, New York, NY, 1989.

[17] M.R. Chmielewski, J.W. Grzymała-Busse, Global discretization of continuous attributes as preprocessing for machine learning, International Journal of Approximate Reasoning 15 (4) (1996) 319–331.

[18] S.K. Choubey, J.S. Deogun, V.V. Raghavan, H. Sever, A comparison of feature selection algorithms in the context of rough classifiers, in: F. Petry (Ed.), International Conference on Fuzzy Systems (FUZZ-IEEE'1996), September 8–11, 1996, New Orleans, LA, vol. 2, IEEE Service Center, Pistcataway, NJ, 1996, pp. 1122–1128.

[19] C.H. Coombs, G.S. Avruin, The Structure of Conflicts, Lawrence Erlbaum, London, 1988.

[20] R. Deja, Conflict analysis, rough set methods and applications, in: Polkowski et al. [98], pp. 491–520.

[21] R. Deja, A. Skowron, On some conflict models and conflict resolution, Romanian Journal of Information Science and Technology 5 (1–2) (2002) 69–82.

[22] J. Deogun, V.V. Raghavan, A. Sarkar, H. Sever, Data mining: Trends in research and development, in: Lin and Cercone [57], pp. 9–46.

[23] P. Doherty, W. Łukaszewicz, A. Skowron, A. Szałas, Knowledge Engineering: A Rough Set Approach, Studies in Fuzziness and Soft Computing, vol. 202, Springer, Heidelberg, 2006.

[24] V. Dubois, M. Quafafou, Concept learning with approximation: rough version spaces, in: Alpigini et al. [1], pp. 239–246.

[25] R. Duda, P. Hart, R. Stork, Pattern Classification, John Wiley & Sons, New York, NY, 2002.

[26] I. Düntsch, G. Gediga, Rough Set Data Analysis: A Road to Non-invasive Knowledge Discovery, Methodos Publishers, Bangor, UK, 2000.

[27] M. Fedrizzi, J. Kacprzyk, H. Nurmi, How different are social choice functions: A rough sets approach, Quality and Quantity 30 (1996) 87–99.

[28] J.H. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, Heidelberg, 2001.

[29] G. Gediga, I. Düntsch, Rough approximation quality revisited, Artificial Intelligence 132 (2001) 219–234.

[30] G. Gediga, I. Düntsch, On model evaluation, indices of importance, and interaction values in rough set analysis, in: Pal et al. [87], pp. 251–276.

[31] G. Góra, A.G. Wojna, RIONA: A new classification system combining rule induction and instance-based learning, Fundamenta Informaticae 51 (4) (2002) 369–390.

[32] S. Greco, M. Inuiguchi, R. Słowiński, Fuzzy rough sets and multiple-premise gradual decision rules, International Journal of Approximate Reasoning 41 (2) (2006) 179–211.

[33] J.W. Grzymała-Busse, Selected algorithms of machine learning from examples, Fundamenta Informaticae 18 (1993) 193–207.

[34] J.W. Grzymała-Busse, Classification of unseen examples under uncertainty, Fundamenta Informaticae 30 (3–4) (1997) 255–267.

[35] J.W. Grzymała-Busse, A new version of the rule induction system LERS, Fundamenta Informaticae 31 (1) (1997) 27–39.

[36] J.W. Grzymała-Busse, Three strategies to rule induction from data with numerical attributes, in: Peters et al. [95], pp. 54–62.

[37] J.W. Grzymała-Busse, LERS – A data mining system, in: Maimon and Rokach [62], pp. 1347–1351.

[38] J.W. Grzymała-Busse, Rule induction, in: Maimon and Rokach [62], pp. 277–294.

[39] J.W. Grzymała-Busse, W.J. Grzymała-Busse, Handling missing attribute values, in: Maimon and Rokach [62], pp. 37–57.

[40] J.W. Grzymała-Busse, W. Ziarko, Data mining and rough set theory, Communications of the ACM 43 (2000) 108–109.

[41] J. Herbert, J.T. Yao, Time-series data analysis with rough sets, in: Proceedings of the 4th International Conference on Computational Intelligence in Economics and Finance (CIEF'2005), Salt Lake City, UT, July 21–26, 2005, pp. 908–911.

[42] X. Hu, N. Cercone, Learning in relational databases: a rough set approach, Computational Intelligence: An International Journal 11 (2) (1995) 323–338.

[43] X. Hu, N. Cercone, Data mining via discretization, generalization and rough set feature selection, Knowledge and Information Systems: An International Journal 1 (1) (1999) 33–60.

[44] X. Hu, N. Cercone, Discovering maximal generalized decision rules through horizontal and vertical data reduction, Computational Intelligence: An International Journal 17 (4) (2001) 685–702.

[45] D. Kim, Data classification based on tolerant rough set, Pattern Recognition 34 (8) (2001) 1613–1624.

[46] D. Kim, S.Y. Bang, A handwritten numeral character classification using tolerant rough set, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (9) (2000) 923–937.

[47] W. Kloesgen, J. Żytkow (Eds.), Handbook of Knowledge Discovery and Data Mining, Oxford University Press, Oxford, 2002.

[48] J. Komorowski, A. Øhrn, A. Skowron, Rosetta and other software systems for rough sets, in: W. Klösgen, J. Żytkow (Eds.), Handbook of Data Mining and Knowledge Discovery, Oxford University Press, 2000, pp. 554–559.

[49] R. Kowalski. A logic-based approach to conflict resolution. Report, Department of Computing, Imperial College, 2003, pp. 1–28. Available from: URL <http://www.doc.ic.ac.uk/~rak/papers/conflictresolution.pdf>.

[50] S. Kraus, Strategic Negotiations in Multiagent Environments, The MIT Press, Cambridge, MA, 2001.

[51] M. Kryszkiewicz, H. Rybinski, Computation of reducts of composed information systems, Fundamenta Informaticae 27 (2–3) (1996) 183–195.

[52] G. Lai, C. Li, K. Sycara, J.A. Giampapa. Literature review on multi-attribute negotiations. Technical Report CMU-RI-TR-04-66, 2004, pp. 1–35.

[53] R. Latkowski, On decomposition for incomplete data, Fundamenta Informaticae 54 (1) (2003) 1–16.

[54] R. Latkowski, Flexible indiscernibility relations for missing attribute values, Fundamenta Informaticae 67 (1–3) (2005) 131–147.

[55] J. Li, N. Cercone, A rough set based model to rank the importance of association rules, in: Ślęzak et al. [121], pp. 109–118.

[56] Y. Li, S.C.-K. Shiu, S.K. Pal, J.N.-K. Liu, A rough set-based case-based reasoner for text categorization, International Journal of Approximate Reasoning 41 (2) (2006) 229–255.

[57] T.Y. Lin, N. Cercone (Eds.), Rough Sets and Data Mining – Analysis of Imperfect Data, Kluwer Academic Publishers, Boston, USA, 1997.

[58] P. Lingras, Fuzzy – rough and rough – fuzzy serial combinations in neurocomputing, Neurocomputing 36 (1–4) (2001) 29–44.

[59] P. Lingras, Unsupervised rough set classification using gas, Journal of Intelligent Information Systems 16 (3) (2001) 215–228.

[60] P. Lingras, C. West, Interval set clustering of Web users with rough *K*-means, Journal of Intelligent Information Systems 23 (1) (2004) 5–16.

[61] Y. Maeda, K. Senoo, H. Tanaka, Interval density function in conflict analysis, in: Skowron et al. [110], pp. 382–389.

[62] O. Maimon, L. Rokach (Eds.), The Data Mining and Knowledge Discovery Handbook, Springer, Heidelberg, 2005.

[63] T.M. Mitchel, Machine Learning, McGraw-Hill Series in Computer Science, Boston, MA, 1999.

[64] P. Mitra, S. Mitra, S.K. Pal, Modular rough fuzzy mlp: evolutionary design, in: Skowron et al. [110], pp. 128–136.

[65] P. Mitra, S.K. Pal, M.A. Siddiqi, Non-convex clustering using expectation maximization algorithm with rough set initialization, Pattern Recognition Letters 24 (6) (2003) 863–873.

[66] A. Nakamura, Conflict logic with degrees, in: Pal and Skowron [88], pp. 136–150.

[67] M. Nakata, H. Sakai, Rough sets handling missing values probabilistically interpreted, in: Ślęzak et al. [120], pp. 325–334.

[68] H.S. Nguyen. Discretization of Real Value Attributes, Boolean Reasoning Approach. Ph.D. thesis, Warsaw University, Warsaw, Poland, 1997.

[69] H.S. Nguyen, From optimal hyperplanes to optimal decision trees, Fundamenta Informaticae 34 (1–2) (1998) 145–174.

[70] H.S. Nguyen, Efficient SQL-learning method for data mining in large data bases, in: T. Dean (Ed.), Sixteenth International Joint Conference on Artificial Intelligence IJCAI, Morgan-Kaufmann Publishers, Stockholm, Sweden, 1999, pp. 806–811.

[71] H.S. Nguyen, On efficient handling of continuous attributes in large data bases, Fundamenta Informaticae 48 (1) (2001) 61–81.

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

[73] H.S. Nguyen, S.H. Nguyen, Rough sets and association rule generation, Fundamenta Informaticae 40 (4) (1999) 383–405.

[74] H.S. Nguyen, A. Skowron, Quantization of real value attributes, in: Proceedings of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, NC, USA, 1995, pp. 34–37.

[75] H.S. Nguyen, D. Ślęzak, Approximate reducts and association rules – correspondence and complexity results, in: Skowron et al. [110], pp. 137–145.

[76] S.H. Nguyen, Regularity analysis and its applications in data mining, in: Polkowski et al. [98], pp. 289–378.

[77] S.H. Nguyen, J. Bazan, A. Skowron, H.S. Nguyen, Layered learning for concept synthesis, in: Peters and Skowron [94], pp. 187–208.

[78] S.H. Nguyen, H.S. Nguyen, Some efficient algorithms for rough set methods, in: Sixth International Conference on Information Processing and Management of Uncertainty on Knowledge Based Systems IPMU'1996, Granada, Spain, vol. III, 1996, pp. 1451–1456.

[79] T.T. Nguyen, Eliciting domain knowledge in handwritten digit recognition, in: Pal et al. [82], pp. 762–767.

[80] T.T. Nguyen, A. Skowron, Rough set approach to domain knowledge approximation, in: Wang et al. [131], pp. 221–228.

[81] H. Nurmi, J. Kacprzyk, M. Fedrizzi, Theory and methodology: Probabilistic, fuzzy and rough concepts in social choice, European Journal of Operational Research 95 (1996) 264–277.

[82] S.K. Pal, S. Bandoyopadhay, S. Biswas (Eds.), First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05) December 18–22, 2005, Indian Statistical Institute, Kolkata, Lecture Notes in Computer Science, vol. 3776, Springer-Verlag, Heidelberg, 2005.

[83] S.K. Pal, B. Dasgupta, P. Mitra, Rough self organizing map, Applied Intelligence 21 (2004) 289–299.

[84] S.K. Pal, P. Mitra, Case generation using rough sets with fuzzy representation, IEEE Transactions on Knowledge and Data Engineering 16 (3) (2004) 292–300.

[85] S.K. Pal, P. Mitra, Pattern Recognition Algorithms for Data Mining, CRC Press, Boca Raton, FL, 2004.

[86] S.K. Pal, W. Pedrycz, A. Skowron, R. Swiniarski (Eds.), Special Volume: Rough-neuro Computing, Neurocomputing, vol. 36, 2001.

[87] S.K. Pal, L. Polkowski, A. Skowron (Eds.), Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies, Springer-Verlag, Heidelberg, 2004.

[88] S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization: A New Trend in Decision-Making, Springer-Verlag, Singapore, 1999.

[89] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving, vol. 9, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

[90] Z. Pawlak, An inquiry into anatomy of conflicts, Journal of Information Sciences 109 (1998) 65–78.

[91] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences, in press, doi:10.1016/j.ins.2006.06.006.

[92] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences, in press, doi:10.1016/j.ins.2006.06.003.

[93] Z. Pawlak, L. Polkowski, A. Skowron, Rough sets and rough logic: a KDD perspective, in: Polkowski et al. [98], pp. 583–646.

[94] J.F. Peters, A. Skowron (Eds.), Transactions on Rough Sets I: Journal Subline, Lecture Notes in Computer Science, vol. 3100, Springer, Heidelberg, 2004.

[95] J.F. Peters, A. Skowron, D. Dubois, J.W. Grzymała-Busse, M. Inuiguchi, L. Polkowski (Eds.), Transactions on Rough Sets II. Rough Sets and Fuzzy Sets: Journal Subline, Lecture Notes in Computer Science, vol. 3135, Springer, Heidelberg, 2004.

[96] J.F. Peters, A. Skowron, Z. Suraj, An application of rough set methods in control design, Fundamenta Informaticae 43 (1–4) (2000) 269–290.

[97] J.F. Peters, Z. Suraj, S. Shan, S. Ramanna, W. Pedrycz, N.J. Pizzi, Classification of meteorological volumetric radar data using rough set methods, Pattern Recognition Letters 24 (6) (2003) 911–920.

[98] L. Polkowski, T.Y. Lin, S. Tsumoto (Eds.), Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems, Studies in Fuzziness and Soft Computing, vol. 56, Springer-Verlag/Physica-Verlag, Heidelberg, 2000.

[99] L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 1: Methodology and Applications, Studies in Fuzziness and Soft Computing, vol. 18, Physica-Verlag, Heidelberg, 1998.

[100] L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Studies in Fuzziness and Soft Computing, vol. 19, Physica-Verlag, Heidelberg, 1998.

[101] M. Quafafou, M. Boussouf, Generalized rough sets based feature selection, Intelligent Data Analysis 4 (1) (2000) 3–17.

[102] A. Roy, S.K. Pal, Fuzzy discretization of feature space for a rough set classifier, Pattern Recognition Letters 24 (6) (2003) 895–902.

[103] H. Sever, V.V. Raghavan, T.D. Johnsten, The status of research on rough sets for knowledge discovery in databases, in: S. Sivasundaram (Ed.), Proceedings of the Second Internationall Conference on Nonlinear Problems in Aviation and Aerospace (ICNPAA'1998), April 29–May 1, 1998, Daytona Beach, FL, vol. 2, Embry-Riddle Aeronautical University, Daytona Beach, FL, 1998, pp. 673–680.

[104] N. Shan, W. Ziarko, An incremental learning algorithm for constructing decision rules, in: W. Ziarko (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer Verlag, Berlin, 1994, pp. 326–334.

[105] A. Skowron, Synthesis of adaptive decision systems from experimental data, in: A. Aamodt, J. Komorowski (Eds.), Fifth Scandinavian Conference on Artificial Intelligence SCAI'1995, Frontiers in Artificial Intelligence and Applications, vol. 28, IOS Press, Trondheim, Norway, 1995, pp. 220–238.

[106] A. Skowron, Rough sets in KDD – plenary talk, in: Z. Shi, B. Faltings, M. Musen (Eds.), 16th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000), Publishing House of Electronic Industry, Beijing, 2000, pp. 1–14.

[107] A. Skowron, Rough sets and boolean reasoning, in: W. Pedrycz (Ed.), Granular Computing: An Emerging Paradigm, Studies in Fuzziness and Soft Computing, vol. 70, Springer-Verlag/Physica-Verlag, Heidelberg, 2001, pp. 95–124.

[108] A. Skowron, Approximate reasoning in distributed environments, in: N. Zhong, J. Liu (Eds.), Intelligent Technologies for Information Analysis, Springer, Heidelberg, 2004, pp. 433–474.

[109] A. Skowron, H.S. Nguyen, Boolean reasoning scheme with some applications in data mining, in: Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 1704, Springer Verlag, Berlin, 1999, pp. 107–115.

[110] A. Skowron, S. Ohsuga, N. Zhong (Eds.), Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC'99), Yamaguchi, November 9–11, 1999, Lecture Notes in Artificial Intelligence, vol. 1711, Springer-Verlag, Heidelbergy, 1999.

[111] A. Skowron, S.K. Pal (Eds.), Special volume: Rough sets, pattern recognition and data mining, Pattern Recognition Letters, vol. 24 (6), 2003.

[112] A. Skowron, Z. Pawlak, J. Komorowski, L. Polkowski, A rough set perspective on data and knowledge, in: W. Kloesgen, J. Żytkow (Eds.), Handbook of KDD, Oxford University Press, Oxford, 2002, pp. 134–149.

[113] A. Skowron, J. Peters, Rough sets: trends and challenges, in: Wang et al. [131], pp. 25–34 (plenary talk).

[114] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Słowiński (Ed.), Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, System Theory, Knowledge Engineering and Problem Solving, vol. 11, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 331–362.

[115] A. Skowron, J. Stepaniuk, Information granules and rough-neural computing, in: Pal et al. [87], pp. 43–84.

[116] A. Skowron, J. Stepaniuk, Ontological framework for approximation, in: Ślęzak et al. [120], pp. 718–727.

[117] D. Ślęzak, Association reducts: a framework for mining multi-attribute dependencies, in: M.-S. Hacid, N.V. Murray, Z.W. Ras, S. Tsumoto (Eds.), Foundations of Intelligent Systems, 15th International Symposium on Methodologies for Intelligent Systems (ISMIS 2005), Saratoga Springs, NY, May 25–28, 2005, Lecture Notes in Artificial Intelligence, vol. 3488, Springer, Heidelberg, 2005, pp. 354–363.

[118] D. Ślęzak, Various approaches to reasoning with frequency-based decision reducts: a survey, in: Polkowski et al. [98], pp. 235–285.

[119] D. Ślęzak, Approximate entropy reducts, Fundamenta Informaticae 53 (2002) 365–387.

[120] D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005), Regina, Canada, August 31–September 3, 2005, Part I, Lecture Notes in Artificial Intelligence, vol. 3641, Springer-Verlag, Heidelberg, 2005.

[121] D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005), Regina, Canada, August 31–September 3, 2005, Part II, Lecture Notes in Artificial Intelligence, vol. 3642, Springer-Verlag, Heidelberg, 2005.

[122] J. Stepaniuk, Knowledge discovery by application of rough set models, in: Polkowski et al. [98], pp. 137–233.

[123] Z. Suraj, Rough set methods for the synthesis and analysis of concurrent processes, in: Polkowski et al. [98], pp. 379–488.

[124] R. Swiniarski, Rough sets and principal component analysis and their applications. data model building and classification, in: Pal and Skowron [88], pp. 275–300.

[125] R. Swiniarski, An application of rough sets and Haar wavelets to face recognition, in: Ziarko and Yao [137], pp. 561–568.

[126] R. Swiniarski, L. Hargis, A new halftoning method based on error diffusion with rough set filterin, in: Polkowski and Skowron [100], pp. 336–342.

[127] R. Swiniarski, A. Skowron, Rough set methods in feature selection and extraction, Pattern Recognition Letters 24 (6) (2003) 833–849.

[128] R.W. Swiniarski, A. Skowron, Independent component analysis, principal component analysis and rough sets in face recognition, in: Peters and Skowron [94], pp. 392–404.

[129] K. Sycara, Multiagent systems, AI Magazine (1998) 79–92.

[130] S. Tsumoto, H. Tanaka, PRIMEROSE: probabilistic rule induction method based on rough sets and resampling methods, Computational Intelligence: An International Journal 11 (1995) 389–405.

[131] G. Wang, Q. Liu, Y. Yao, A. Skowron (Eds.), Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2003), Chongqing, China, May 26–29, 2003, Lecture Notes in Artificial Intelligence, vol. 2639, Springer-Verlag, Heidelberg, 2003.

[132] J. Wang, C. Jia, K. Zhao, Investigation on AQ11, ID3 and the principle of discernibility matrix, Journal of Computer Science and Technology 16 (1) (2001) 1–12.

[133] A. Wojna, Analogy based reasoning in classifier construction, in: J.F. Peters, A. Skowron (Eds.), Transactions on Rough Sets IV: Journal Subline, Lecture Notes in Computer Science, vol. 3700, Springer, Heidelberg, 2005, pp. 277–374.

[134] J. Wróblewski, Theoretical foundations of order-based genetic algorithms, Fundamenta Informaticae 28 (1996) 423–430.

[135] J. Wróblewski, Analyzing relational databases using rough set based methods, in: Eighth International Conference on Processing and Management of Uncertainty in Knowledge-Based Systems IPMU. Madrid, Spain, vol. I, 2000, pp. 256–262.

[136] J. Wróblewski, Adaptive aspects of combining approximation spaces, in: Pal et al. [87], pp. 139–156.

[137] W. Ziarko, Y. Yao (Eds.), Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000), Banff, Canada, October 16–19, 2000, vol. 2005, Lecture Notes in Artificial Intelligence, vol. 2005, Springer-Verlag, Heidelberg, 2001.