Rough Sets: A Tutorial

Jan Komorowski [1,4], **Lech Polkowski** [3,4], **Andrzej Skowron** [2]

[1] *Department of Computer and Information Science*
Norwegian University of Science and Technology (NTNU)
7034 Trondheim, Norway,
janko@idi.ntnu.no

[2] *Institute of Mathematics, Warsaw University*
Banacha 2, 02-097 Warszawa, Poland
skowron@mimuw.edu.pl

[3] *Institute of Mathematics, Warsaw University of Technology*
Pl. Politechniki 1, 00-665 Warszawa, Poland
polk@mimuw.edu.pl

[4] *Polish-Japanese Institute of Information Technology*
Koszykowa 86, 02-008 Warszawa, Poland

Abstract

A rapid growth of interest in rough set theory [297] and its applications can be lately seen in the number of international workshops, conferences and seminars that are either directly dedicated to rough sets, include the subject in their programs, or simply accept papers that use this approach to solve problems at hand. A large number of high quality papers on various aspects of rough sets and their applications have been published in recent years as a result of this attention. The theory has been followed by the development of several software systems that implement rough set operations. In Sect. 1.12 we present a list of software systems based on rough sets. Some of the toolkits, provide advanced graphical environments that support the process of developing and validating rough set classifiers. Rough sets are applied in many domains, such as, for instance, medicine, finance, telecommunication, vibration analysis, conflict resolution, intelligent agents, image analysis, pattern recognition, control theory, process industry, marketing, etc.

Several applications have revealed the need to extend the traditional rough set approach. A special place among various extensions is taken by the approach that replaces indiscernibility relation based on equivalence with a tolerance relation.

In view of many generalizations, variants and extensions of rough sets a uniform presentation of the theory and methodology is in place. This tutorial paper is intended to fulfill these needs. It introduces basic notions and illustrates them with simple examples. It discusses methodologies for analyzing data and surveys applications. It also presents an introduction to logical, algebraic and topological aspects and major extensions to standard rough sets. It finally glances at future research.

Keywords: approximate reasoning, soft computing, indiscernibility, lower and upper approximations, rough sets, boundary region, positive region, rough membership function, decision rules, dependencies to a degree, patterns, feature extraction and selection, rough mereology.

Contents

List of Figures

List of Tables

Introduction

Rough set theory was developed by Zdzisław Pawlak [292, 297, 205] in the early 1980's. It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesize approximation of concepts from the acquired data. We show that first in the traditional approach and later how it evolves towards "information granules" under tolerance relation.

The purpose of developing such definitions may be twofold. In some instances, the aim may be to gain insight into the problem at hand by analyzing the constructed model, i.e. the structure of the model is itself of interest. In other applications, the transparency and explainability features of the model may be of secondary importance and the main objective is to construct a classifier that classifies unseen objects well. A logical calculus on approximate notions is equally important. It is based on the concept of "being a part to a degree" and is known as rough mereology (see e.g. [335, 337, 338, 339, 340, 347, 407, 342, 343]).

The overall modeling process typically consists of a sequence of several sub-steps that all require various degrees of tuning and fine-adjustments. In order to perform these functions, an environment to interactively manage and process data is required. An important feature of rough sets is that the theory is followed by practical implementations of toolkits that support interactive model development. Several software systems based on rough sets exist. For a list of these systems see Sect. 1.12.

The article consists of two parts. In Part I we discuss:

- classical rough set theory (Sections 1.1 to 1.5),
- the modeling process using rough sets which includes feature selection, feature extraction (by discretization, symbolic attribute value grouping, searching for relevant hyperplanes), rule synthesis and validation, (Sect. 1.7),
- some extensions to classical rough set approach (Sect. 1.8),
- some introductory information on algebraic and logical aspects of rough sets (Sect. 1.9),
- some relationships with other approaches (Sect. 1.10),
- a list of applications of rough sets (Sect. 1.11)
- a list of software systems that implement rough set methods (Sect. 1.12),
- and, finally, some conclusions including also considerations on future research.

In Part II we overview rough mereology developed as a tool for synthesis of objects satisfying a given specification to a satisfactory degree. The main goal of this approach is to develop methodology for construction of calculus on approximate concepts.

The tutorial attempts to address the needs of a broad readership. By combining informal introductions of each topic with simple examples, it should be accessible to all readers with interest in data analysis: from undergraduate students in computer science, to engineers, medical informatics scientists, to financial analysts, to social science researchers, etc. Since every informal exposition is followed by precise definitions, the tutorial is also an authoritative source for graduate students and researchers in the subject.

Chapter 1

Rough sets

1.1 Information Systems

A data set is represented as a table, where each row represents a case, an event, a patient, or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object; the attribute may be also supplied by a human expert or user. This table is called an *information system*. More formally, it is a pair $\mathcal{A} = (U, A)$, where U is a non-empty finite set of *objects* called the *universe* and A is a non-empty finite set of *attributes* such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set* of a.

Example 1.1.1 A very simple information system is shown in Tab. 1.1. There are seven cases or objects, and two attributes (*Age* and Lower Extremity Motor Score *LEMS*).

	Age	*LEMS*
x_1	16-30	50
x_2	16-30	0
x_3	31-45	1-25
x_4	31-45	1-25
x_5	46-60	26-49
x_6	16-30	26-49
x_7	46-60	26-49

Table 1.1: An example information system.

The reader will easily notice that cases x_3 and x_4 as well as x_5 and x_7 have exactly the same values of conditions. The cases are (pairwise) *indiscernible* using the available attributes. □

In many applications there is an outcome of classification that is known. This *a posteriori* knowledge is expressed by one distinguished attribute called decision attribute; the process is known as supervised learning. Information systems of this kind are called *decision systems*. A decision system is any information system of the form $\mathcal{A} = (U, A \cup \{d\})$, where $d \notin A$ is the *decision attribute*. The elements of A are called *conditional attributes* or simply *conditions*. The decision attribute may take several values though binary outcomes are rather frequent.

Example 1.1.2 A small example decision table can be found in Tab. 1.2. The table has the same seven cases as in the previous example, but one decision attribute (*Walk*) with two possible outcomes has been added.

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

Table 1.2: *Walk*: An example decision table

The reader may again notice that cases x_3 and x_4 as well as x_5 and x_7 still have exactly the same values of conditions, but the first pair has a different outcome (different value of the decision attribute) while the second pair also has the same outcome. □

The definitions to be synthesized from decision tables will be of the rule form "if *Age* is 16-30 and *LEMS* is 50 then *Walk* is Yes". Among the possible properties of the constructed rule sets, minimality (of the left hand side lengths of the rules) is one of the important issues. This is studied in the next section.

1.2 Indiscernibility

A decision system (i.e. a decision table) expresses all the knowledge about the model. This table may be unnecessarily large in part because it is redundant in at least two ways. The same or indiscernible objects may be represented several times, or some of the attributes may be superfluous. We shall look into these issues now.

The notion of equivalence is recalled first. A binary relation $R \subseteq X \times X$ which is reflexive (i.e. an object is in relation with itself xRx), symmetric (if xRy then yRx) and transitive (if xRy and yRz then xRz) is called an equivalence relation. The *equivalence class* of an element $x \in X$ consists of all objects $y \in X$ such that xRy.

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

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

$IND_{\mathcal{A}}(B)$ is called the *B-indiscernibility relation.* If $(x, x') \in IND_{\mathcal{A}}(B)$, then objects x and x' are indiscernible from each other by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted $[x]_B$. The subscript $\mathcal{A}$ in the indiscernibility relation is usually omitted if it is clear which information system is meant.

Some extensions of standard rough sets do not require transitivity to hold. See, for instance, [408]. Such a relation is called tolerance or similarity. This case will be discussed in Sect. 1.8.

Example 1.2.1 Let us illustrate how a decision table such as Tab. 1.2 defines an indiscernibility relation. The non-empty subsets of the conditional attributes are $\{Age\}$, $\{LEMS\}$ and $\{Age, LEMS\}$.

If we consider, for instance, $\{LEMS\}$, objects x_3 and x_4 belong to the same equivalence class and are indiscernible. (By the same token, x_5, x_6 and x_7 belong to another indiscernibility class.) The relation IND defines three partitions of the universe.

$$IND(\{Age\}) \;\; = \;\; \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\}$$

$$\begin{aligned} IND(\{LEMS\}) &= \{\{x_1\},\{x_2\},\{x_3,x_4\},\{x_5,x_6,x_7\}\} \\ IND(\{Age, LEMS\}) &= \{\{x_1\},\{x_2\},\{x_3,x_4\},\{x_5,x_7\},\{x_6\}\} \end{aligned}$$

□

1.3 Set Approximation

An equivalence relation induces a partitioning of the universe (the set of cases in our example). These partitions can be used to build new subsets of the universe. Subsets that are most often of interest have the same value of the outcome attribute. It may happen, however, that a concept such as "*Walk*" cannot be defined in a crisp manner. For instance, the set of patients with a positive outcome cannot be defined crisply using the attributes available in Tab. 1.2. The "problematic" patients are objects x_3 and x_4. In other words, it is not possible to induce a crisp (precise) description of such patients from the table. It is here that the notion of rough set emerges. Although we cannot define those patients crisply, it is possible to delineate the patients that certainly have a positive outcome, the patients that certainly do not have a positive outcome and, finally, the patients that belong to a boundary between the certain cases. If this boundary is non-empty, the set is rough. These notions are formally expressed as follows.

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

The objects in $\underline{B}X$ can be with certainty classified as members of X on the basis of knowledge in B, while the objects in $\overline{B}X$ can be only classified as possible members of X on the basis of knowledge in B. The set $BN_B(X) = \overline{B}X \Leftrightarrow \underline{B}X$ is called the *B-boundary region of X*, and thus consists of those objects that we cannot decisively classify into X on the basis of knowledge in B. The set $U \Leftrightarrow \overline{B}X$ is called the *B-outside region of X* and consists of those objects which can be with certainty classified as do not belonging to X (on the basis of knowledge in B). A set is said to be *rough* (respectively *crisp*) if the boundary region is non-empty (respectively empty)[1].

Example 1.3.1 The most common case is to synthesize definitions of the outcome (or decision classes) in terms of the conditional attributes. Let $W = \{x \mid \mathit{Walk}(x) = \text{Yes}\}$, as given by Tab. 1.2. We then obtain the approximation regions $\underline{A}W = \{x_1, x_6\}$, $\overline{A}W = \{x_1, x_3, x_4, x_6\}$, $BN_A(W) = \{x_3, x_4\}$ and $U \Leftrightarrow \overline{A}W = \{x_2, x_5, x_7\}$. It follows that the outcome *Walk* is rough since the boundary region is not empty. This is shown in Fig. 1.1. □

One can easily show the following properties of approximations:

(1) $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$

(2) $\underline{B}(\emptyset) = \overline{B}(\emptyset) = \emptyset, \underline{B}(U) = \overline{B}(U) = U$

(3) $\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$

(4) $\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$

[1] The letter B refers to the subset B of the attributes A. If another subset were chosen, e.g. $F \subseteq A$, the corresponding names of the relations would have been *F-boundary region, F-lower- and F-upper approximations.*

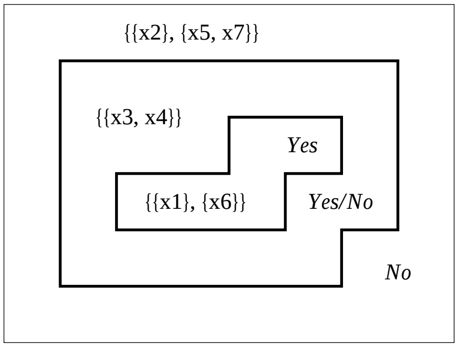


Figure 1.1: Approximating the set of walking patients using the two conditional attributes *Age* and *LEMS*. Equivalence classes contained in the corresponding regions are shown.

(5) $X \subseteq Y$ implies $\underline{B}(X) \subseteq \underline{B}(Y)$ and $\overline{B}(X) \subseteq \overline{B}(Y)$

(6) $\underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y)$

(7) $\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$

(8) $\underline{B}(\Leftrightarrow X) = \Leftrightarrow \overline{B}(X)$

(9) $\overline{B}(\Leftrightarrow X) = \Leftrightarrow \underline{B}(X)$

(10) $\underline{B}(\underline{B}(X)) = \overline{B}(\underline{B}(X)) = \underline{B}(X)$

(11) $\overline{B}(\overline{B}(X)) = \underline{B}(\overline{B}(X)) = \overline{B}(X)$

where $\Leftrightarrow X$ denotes $U \Leftrightarrow X$.

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

One can define the following four basic classes of rough sets, i.e., four categories of vagueness:

a) X is *roughly B-definable*, iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) \neq U$,

b) X is *internally B-undefinable*, iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) \neq U$,

c) X is *externally B-undefinable,* iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) = U$,

d) X is *totally B-undefinable,* iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) = U$.

The intuitive meaning of this classification is the following.

X is roughly *B-definable* means that with the help of B we are able to decide for some elements of U that they belong to X and for some elements of U that they belong to $\Leftrightarrow X$.

X is internally B-undefinable means that using B we are able to decide for some elements of U that they belong to $\Leftrightarrow X$, but we are unable to decide for any element of U whether it belongs to X.

X is externally B-undefinable means that using B we are able to decide for some elements of U that they belong to X, but we are unable to decide for any element of U whether it belongs to $\Leftrightarrow X$.

X is totally B-undefinable means that using B we are unable to decide for any element of U whether it belongs to X or $\Leftrightarrow X$.

Rough set can be also characterized numerically by the following coefficient

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

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

1.4 Reducts

In the previous section we investigated one natural dimension of reducing data which is to identify equivalence classes, i.e. objects that are indiscernible using the available attributes. Savings are to be made since only one element of the equivalence class is needed to represent the entire class. The other dimension in reduction is to keep only those attributes that preserve the indiscernibility relation and, consequently, set approximation. The rejected attributes are redundant since their removal cannot worsen the classification. There is usually several such subsets of attributes and those which are minimal are called reducts. Computing equivalence classes is straightforward. Finding a minimal reduct (i.e. reduct with a minimal cardinality of attributes among all reducts) is NP-hard [409]. One can also show that the number of reducts of an information system with m attributes may be equal to

$$\binom{m}{\lfloor m/2 \rfloor}$$

This means that computing reducts it is a non-trivial task that cannot be solved by a simple minded increase of computational resources. It is, in fact, one of the bottlenecks of the rough set methodology. Fortunately, there exist good heuristics (e.g. [533, 534]) based on genetic algorithms that compute sufficiently many reducts in often acceptable time, unless the number of attributes is very high.

Example 1.4.1 Consider the following decision system (defined in Tab. 1.3): $\mathcal{A}' = (U, \{$ *Diploma*, *Experience*, *French*, *Reference*$\} \cup \{$*Decision*$\})$. Let us consider only the conditional attributes i.e. an information system

$$\mathcal{A} = (U, \{\mathit{Diploma}, \mathit{Experience}, \mathit{French}, \mathit{Reference}\}.$$

For simplicity, each equivalence class contains one element. It appears that there is a minimal set of attributes $\{$*Experience*, *Reference*$\}$ which discerns objects in the same way as the full set of considered objects. The reader may check that the indiscernibility relation using the full set of attributes and the set $\{$*Experience*,*Reference*$\}$ is the same. The actual construction of minimal sets of attributes with such property will be soon revealed. □

Given an information system $\mathcal{A} = (U, A)$ the definitions of these notions are as follows. A *reduct* of $\mathcal{A}$ is a minimal set of attributes $B \subseteq A$ such that $\mathit{IND}_{\mathcal{A}}(B) = \mathit{IND}_{\mathcal{A}}(A)$. In other words, a reduct is a minimal set of attributes from A that preserves the partitioning of the universe and hence the ability to perform classifications as the whole attribute set A does.

Let $\mathcal{A}$ be an information system with n objects. The *d*iscernibility matrix of $\mathcal{A}$ is a symmetric $n \times n$ matrix with entries c_{ij} as given below.

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

	Diploma	*Experience*	*French*	*Reference*	*Decision*
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

Table 1.3: *Hiring*: An example of an unreduced decision table.

Each entry thus consists of the set of attributes upon which objects x_i and x_j differ.

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

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

where $c_{ij}^* = \{a^* \mid a \in c_{ij}\}$. The set of all prime implicants[2] of $f_{\mathcal{A}}$ determines the set of all reducts of $\mathcal{A}$.

Example 1.4.2 The discernibility function for the information system $\mathcal{A}$ defined in Tab. 1.3 is:

$$\begin{aligned} f_{\mathcal{A}}(d, e, f, r) \quad = \quad & (e{\vee}r)(d{\vee}e{\vee}r)(d{\vee}e{\vee}r)(d{\vee}r)(d{\vee}e)(e{\vee}f{\vee}r)(d{\vee}e{\vee}f) \\ & (d{\vee}r)(d{\vee}e)(d{\vee}e)(d{\vee}e{\vee}r)(e{\vee}f{\vee}r)(d{\vee}f{\vee}r) \\ & (d{\vee}e{\vee}r)(d{\vee}e{\vee}r)(d{\vee}e{\vee}r)(d{\vee}e{\vee}f)(f{\vee}r) \\ & (e)(r)(d{\vee}f{\vee}r)(d{\vee}e{\vee}f{\vee}r) \\ & (e{\vee}r)(d{\vee}e{\vee}f{\vee}r)(d{\vee}e{\vee}f{\vee}r) \\ & (d{\vee}f{\vee}r)(d{\vee}e{\vee}f) \\ & (d{\vee}e{\vee}r) \end{aligned}$$

where each parenthesized tuple is a conjunction in the Boolean expression, and where the one-letter Boolean variables correspond to the attribute names in an obvious way. After simplification, the function is $f_{\mathcal{A}}(d, e, f, r) = er$. (The notation er is a shorthand for $e \wedge r$.)

Let us also notice that each row in the above discernibility function corresponds to one column in the discernibility matrix. This matrix is symmetrical with the empty diagonal. So, for instance, the last but one row says that the sixth object (more precisely, the sixth equivalence class) may be discerned from the seventh one by any of the attributes *Diploma*, *French* or *Reference* and by any of *Diploma*, *Experience* or *French* from the eight one. □

If we instead construct a Boolean function by restricting the conjunction to only run over column k in the discernibility matrix (instead of over all columns), we obtain the so-called *k-relative discernibility function.* The set of all prime implicants of this function determines

[2] An implicant of a Boolean function f is any conjunction of literals (variables or their negations) such that if the values of these literals are true under an arbitrary valuation v of variables then the value of the function f under v is also true. A prime implicant is a minimal implicant. Here we are interested in implicants of monotone Boolean functions only i.e. functions constructed without negation.

the set of all *k-relative reducts* of $\mathcal{A}$. These reducts reveal the minimum amount of information needed to discern $x_k \in U$ (or, more precisely, $[x_k] \subseteq U$) from all other objects.

Using the notions introduced above, the problem of supervised learning, (i.e., the problem where the outcome of the classification is known), is to find the value of the decision d that should be assigned to a new object which is described with the help of the conditional attributes. We often require the set of attributes used to define the object to be minimal. For the example Tab. 1.3 it appears that $\{Experience, Reference\}$ and $\{Diploma, Experience\}$ are two minimal sets of attributes that uniquely define to which decision class an object belongs. The corresponding discernibility function is relative to the decision. The notions are now formalized.

Let $\mathcal{A} = (U, A \cup \{d\})$ be given. The cardinality of the image $d(U) = \{k \mid d(x) = k, x \in U\}$ is called the *rank of d* and is denoted by $r(d)$. Let us further assume that the set V_d of values of decision d is equal to $\{v_d^1, \ldots, v_d^{r(d)}\}$.

Example 1.4.3 Quite often the rank is two, e.g., $\{\text{Yes}, \text{No}\}$ or $\{\text{Accept}, \text{Reject}\}$. It can be an arbitrary number, however. For instance in the *Hiring* example, we could have rank three if the decision had values in the set $\{\text{Accept}, \text{Hold}, \text{Reject}\}$. □

The decision d determines a partition $CLASS_{\mathcal{A}}(d) = \{X_{\mathcal{A}}^1, \ldots, X_{\mathcal{A}}^{r(d)}\}$ of the universe U, where $X_{\mathcal{A}}^k = \{x \in U \mid d(x) = v_d^k\}$ for $1 \leq k \leq r(d)$. $CLASS_{\mathcal{A}}(d)$ is called the *classification of objects in $\mathcal{A}$ determined by the decision d.* The set $X_{\mathcal{A}}^i$ is called the *i-th decision class of $\mathcal{A}$.* By $X_{\mathcal{A}}(u)$ we denote the decision class $\{x \in U \mid d(x) = d(u)\}$, for any $u \in U$.

Example 1.4.4 There are two decision classes in each of the running example decision systems, i.e., $\{\text{Yes}, \text{No}\}$ and $\{\text{Accept}, \text{Reject}\}$, respectively. The partitioning of the universe for the *Walk* table is $U = X^{\text{Yes}} \cup X^{\text{No}}$, where $X^{\text{Yes}} = \{x_1, x_4, x_6\}$ and $X^{\text{No}} = \{x_2, x_3, x_5, x_7\}$. For the *Hiring* table we have $U = X^{\text{Accept}} \cup X^{\text{Reject}}$, where $X^{\text{Accept}} = \{x_1, x_4, x_6, x_7\}$ and $X^{\text{Reject}} = \{x_2, x_3, x_5, x_8\}$. The notation X^{Yes} and X^{No} is a shorthand for X^1 and X^2, respectively. □

If $X_{\mathcal{A}}^1, \ldots, X_{\mathcal{A}}^{r(d)}$ are the decision classes of $\mathcal{A}$, then the set $\underline{B}X_1 \cup \ldots \cup \underline{B}X_{r(d)}$ is called the *B-positive region of $\mathcal{A}$* and is denoted by $POS_B(d)$.

Example 1.4.5 A quick check, left to the reader, reveals that $\underline{A}X^{\text{Yes}} \cup \underline{A}X^{\text{No}} \neq U$ while $\underline{A}X^{\text{Accept}} \cup \underline{A}X^{\text{Reject}} = U$. This is related to the fact that for the decision system in Tab. 1.2 a unique decision cannot be made for objects x_3 and x_4 while in case of the other table all decisions are unique. □

This important property of decision systems is formalized as follows. Let $\mathcal{A} = (U, A \cup \{d\})$ be a decision system. The *generalized decision in $\mathcal{A}$* is the function $\partial_A : U \nrightarrow \mathcal{P}(V_d)$ defined by $\partial_A(x) = \{i \mid \exists x' \in U \ x' \ IND(A) \, x \text{ and } d(x) = i\}$. A decision table $\mathcal{A}$ is called *consistent (deterministic)* if $|\partial_A(x)| = 1$ for any $x \in U$, otherwise $\mathcal{A}$ is *inconsistent (non-deterministic).*

It is easy to see that a decision table $\mathcal{A}$ is consistent if, and only if, $POS_A(d) = U$. Moreover, if $\partial_B = \partial_{B'}$, then $POS_B(d) = POS_{B'}(d)$ for any pair of non-empty sets $B, B' \subseteq A$.

Example 1.4.6 The *A-positive region of $\mathcal{A}$* in the *Walk* decision system is a proper subset of U, while in the *Hiring* decision system the corresponding set is equal to the universe U. The first system is non-deterministic, the second one - deterministic. □

We have introduced above the notion of *k-relative* discernibility function. Since the decision attribute is so significant, it is useful to introduce a special definition for its case. Let $\mathcal{A} = (U, A \cup \{d\})$ be a consistent decision table and let $M(\mathcal{A}) = (c_{ij})$ be its discernibility matrix. We construct a new matrix $M^d(\mathcal{A}) = (c_{ij}^d)$ assuming $c_{ij}^d = \emptyset$ if $d(x_i) = d(x_j)$ and $c_{ij}^d = c_{ij} \Leftrightarrow \{d\}$, otherwise. Matrix $M^d(\mathcal{A})$ is called *the decision-relative discernibility matrix of* $\mathcal{A}$. Construction of *the decision-relative discernibility function* from this matrix follows the construction of the discernibility function from the discernibility matrix. It has been shown [409] that the set of *prime implicants* of $f_M^d(\mathcal{A})$ defines the set of all *decision-relative reducts* of $\mathcal{A}$.

Example 1.4.7 The *Hiring* decision table in Tab. 1.4 is now used to illustrate the construction of the corresponding decision-relative discernibility matrix and function. The rows are reordered for convenience putting the accepted objects in the top rows. The corresponding discernibility matrix in Tab. 1.5 is symmetrical and the diagonal is empty, and so are all the entries for which the decisions are equal.

	Diploma	*Experience*	*French*	*Reference*	*Decision*
x_1	MBA	Medium	Yes	Excellent	Accept
x_4	MSc	High	Yes	Neutral	Accept
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_5	MSc	Medium	Yes	Neutral	Reject
x_8	MCE	Low	No	Excellent	Reject

Table 1.4: *Hiring*: The reordered decision table.

	$[x_1]$	$[x_4]$	$[x_6]$	$[x_7]$	$[x_2]$	$[x_3]$	$[x_5]$	$[x_8]$
$[x_1]$	$\emptyset$							
$[x_4]$	$\emptyset$	$\emptyset$						
$[x_6]$	$\emptyset$	$\emptyset$	$\emptyset$					
$[x_7]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$				
$[x_2]$	e, r	d, e	d, e, r	e, f, r	$\emptyset$			
$[x_3]$	d, e, r	d, e, r	d, e, r	d, e, f	$\emptyset$	$\emptyset$		
$[x_5]$	d, r	e	e, r	d, e, f, r	$\emptyset$	$\emptyset$	$\emptyset$	
$[x_8]$	d, e, f	d, e, f, r	d, e, f	d, e, r	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Table 1.5: *Hiring*: The decision-relative discernibility matrix.

The resulting simplified decision-relative discernibility function is $f_M^d(\mathcal{A}) = ed \vee er$. From the definition of the decision-relative matrix it follows that selecting one column of the indiscernibility matrix, e.g., corresponding to $[x_1]$, and simplifying it gives a minimal function that discerns $[x_1]$ from objects belonging to the corresponding decision class from objects belonging to the other decision classes. For example, the first column gives a Boolean function $(e{\vee}r)(d{\vee}e{\vee}r)(d{\vee}r)(d{\vee}e{\vee}f)$ which after simplification becomes $ed \vee rd \vee re \vee rf$. The reader can check that, for instance, "if *Reference* is Excellent and *French* is Yes then *Decision* is Accept" is indeed the case for x_1. It is rather illuminating to notice that if there is any other object for which "*Reference* is Excellent" and "*French* is Yes" hold, then the decision will also be "Accept". Indeed, this is the case for x_6. □

If a Boolean function such as in the case of k-relative discernibility function is constructed by restricting the conjunction to run only over these entries of the column that corresponds to objects with a decision different from the decision on x_k then the (k, d)*–relative discernibility function* is obtained. Decision rules with minimal descriptions of their left hand sides may be constructed from prime implicants of these functions (see Sect. 1.7.3).

Example 1.4.8

Figures 1.2 to 1.5 display these four types of indiscernibility. It is possible to consider other kinds of reducts, e.g. reducts that preserve the positive region and then use the same Boolean reasoning method to compute these reducts.

□

1.5 Rough Membership

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

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

The rough membership function can be interpreted as a frequency-based estimate of $\Pr(x \in X \mid u)$, the conditional probability that object x belongs to set X, given knowledge u of the information signature of x with respect to attributes B, i.e. $u = Inf_B(x)$ (see e.g. [531], [312], [310], [541]).

The formulae for the lower and upper set approximations can be generalized to some arbitrary level of precision $\pi \in (\frac{1}{2}, 1]$ by means of the rough membership function [549], as shown below.

$$\underline{B}_\pi X = \{x \mid \mu_X^B(x) \geq \pi\}$$
$$\overline{B}_\pi X = \{x \mid \mu_X^B(x) > 1 \Leftrightarrow \pi\}$$

Note that the lower and upper approximations as originally formulated are obtained as a special case with $\pi = 1.0$.

Approximations of concepts are constructed on the basis of background knowledge. Obviously, concepts are also related to unseen so far objects. Hence it is very useful to define parameterized approximations with parameters tuned in the searching process for approximations of concepts. This idea is crucial for construction of concept approximations using rough set methods.

Rough sets can thus approximately describe sets of patients, events, outcomes, etc. that may be otherwise difficult to circumscribe.

1.6 Dependency of Attributes

Another important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C. In other

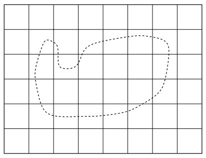


Figure 1.2: Not relative to a particular case (or object) and not relative to the decision attribute. The full indiscernibility relation is preserved. Reducts of this type are minimal attribute subsets that enable us to discern all cases from each other, up to the same degree as the full set of attributes does.

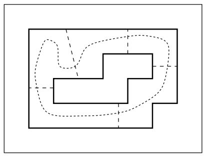


Figure 1.3: Not relative to a particular case (or object) but relative to the decision attribute. All regions with the same value of the generalized decision ∂_A are preserved. Reducts of this type are minimal conditional attribute subsets $B \subseteq A$ that for all cases enable us to make the same classifications as the full set of attributes does, i.e. $\partial_A = \partial_B$.

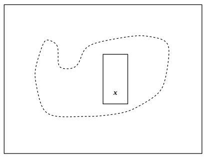


Figure 1.4: Relative to case (or object) x but not relative to the decision attribute. Reducts of this type are minimal conditional attribute subsets that enable us to discern case x from all other cases up to the same degree as the full set of conditional attributes does.

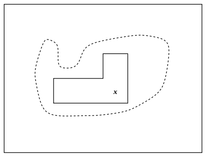


Figure 1.5: Relative to case (or object) x and relative to the decision attribute. Our ability to discern case x from cases with different generalized decision than x is preserved. Reducts B of this type are minimal conditional attribute subsets that enable us to determine the outcome of case x, up to the same degree as the full set of attributes does, i.e. $\partial_A(x) = \partial_B(x)$.

words, D depends totally on C, if there exists a functional dependency between values of D and C.

Formally dependency can be defined in the following way. Let D and C be subsets of A.

We will say that D *depends on* C in a *degree* k $(0 \leq k \leq 1)$, denoted $C \Rightarrow_k D$, if

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

where

$$POS_C(D) = \bigcup_{X \in U/D} \underline{C}(X),$$

called a *positive region* of the partition U/D with respect to C, is the set of all elements of U that can be uniquely classified to blocks of the partition U/D, by means of C.

Obviously

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

If $k = 1$ we say that D *depends totally* on C, and if $k < 1$, we say that D *depends partially* (in a *degree* k) on C.

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

It can be easily seen that if D depends totally on C then $IND(C) \subseteq IND(D)$. This means that the partition generated by C is finer than the partition generated by D. Let us notice that the concept of dependency discussed above corresponds to that considered in relational databases.

Summing up: D is *totally* (*partially*) dependent on C, if employing C *all* (*possibly some*) elements of the universe U may be uniquely classified to blocks of the partition U/D.

1.7 Concept Approximation Construction: The Modeling Process

One of the main goals of machine learning, pattern recognition, knowledge discovery and data mining as well as of fuzzy sets and rough sets is to synthesize approximations of target concepts (e.g. decision classes) from the background knowledge (represented e.g. in the form of decision tables). It is usually only possible to search for approximate descriptions of target concepts due to incomplete knowledge about them (e.g. positive and negative examples of concept objects are given).

Approximate descriptions of concepts are constructed from some primitive concepts. It is furthermore well known that target concept descriptions defined directly by Boolean combinations of descriptors of the form $a = v$ (when a is and attribute and $a \in V_a$ are often not of good approximation quality. Feature selection and feature extraction problems are often approached by searching for relevant primitive concepts and are well known approaches in machine learning, KDD and other areas as (see, e.g. [147, 210, 92]).

In the case of feature selection relevant features are sought among the given features e.g. among descriptors $a = v$ where a is a relevant attribute. In Sect. 1.7.1 rough set-based methods for feature selection are briefly discussed.

The feature extraction problem is implemented as a search for some new features that are more relevant for classification and are defined (in some language) by means of the existing features.

These new features can be e.g. of the form $a \in [0.5, 1)$ or $2a + 3b > 0.75$. Their values on a given object are computed from given values of conditional attributes on the object. The new features are often binary taking value 1 on a given object iff the specified condition is true on this object. In the case of symbolic value attributes we look for new features like $a \in \{\text{French, English, Polish}\}$ with value 1 iff a person speaks any of these languages. The important issues in feature extraction are problems of discretization of real value attributes, grouping of symbolic (nominal) value attributes, searching for new features defined by hyperplanes or more complex surfaces defined over existing attributes. In Section 1.7.2 discretization based on rough set and Boolean reasoning approach is discussed. Some other approaches to feature extraction that are based on Boolean reasoning are also discussed. All cases of feature extraction problem mentioned above may be described in terms of searching for relevant features in a particular language of features. Boolean reasoning plays the crucial role of an inference engine for feature selection problems.

Feature extraction and feature selection are usually implemented in a pre-processing stage of the whole modeling process. There are some other aspects related to this stage of modeling such as, for instance, elimination of noise from the data or treatment of missing values. More information related to these problems can be found in [344, 345] and in the bibliography included in these books.

In the next stage of the synthesis of target concept approximations descriptions of the target concepts are constructed from the extracted relevant features (relevant primitive concepts) by applying some operations. In the simplest case when Boolean connectives $\vee$ and $\wedge$ are chosen these descriptions form the so-called decision rules. In Sect. 1.7.3 we give a short introduction to methods for decision rule synthesis that are based on rough set methods and Boolean reasoning. Two main cases of decision rules are discussed: exact (deterministic) and approximate (non-deterministic) rules. More information on decision rule synthesis and using rough set approach the reader may find in [344, 345] and in the bibliography included in these books.

Finally, it is necessary to estimate the quality of constructed approximations of target concepts. Let us observe that the "building blocks" from which different approximations of target concepts are constructed may be inconsistent on new, so far unseen objects (i.e. some objects from the same class may be classified to disjoint concepts). This creates a necessity to develop methods for resolving these inconsistencies. The quality of target concept approximations can be considered acceptable if the inconsistencies may be resolved by using these methods. In Sect. 1.7.4 some introductory comments on this problem are presented and references to rough set methods that resolve conflicts among different decision rules by voting for the final decision are given.

1.7.1 Significance of Attributes and Approximate Reducts

One of the first ideas [297] was to consider as relevant features those in the *core* of an information system, i.e. features that belong to the intersection of all reducts of the information system. It can be easily checked that several definitions of relevant features that are used by machine learning community [4] can be interpreted by choosing a relevant decision system corresponding to the information system.

Another approach is related to dynamic reducts (see e.g. [19]) i.e. conditional attribute sets appearing “sufficiently often” as reducts of samples of the original decision table. The attributes belonging to the “majority” of dynamic reducts are defined as relevant. The value thresholds for "sufficiently often" and "majority" needs to be tuned for the given data. Several of the reported experiments show that the set of decision rules based on such attributes is much smaller than the set of all decision rules and the quality of classification of new objects is increasing or at least not significantly decreasing if only rules constructed over such relevant features are considered.

It is also possible to consider as relevant features those from some approximate reducts of sufficiently high quality. As it follows from the considerations concerning reduction of attributes, they can be not equally important and some of them can be eliminated from an information table without loosing information contained in the table. The idea of attribute reduction can be generalized by an introduction of the concept of *significance of attributes*, which enables an evaluation of attributes not only by a two-valued scale, *dispensable – indispensable*, but by associating with an attribute a real number from the [0,1] closed interval; this number expresses the importance of the attribute in the information table.

Significance of an attribute a in a decision table $\mathcal{A} = (U, C \cup D)$ (with the decision set D) can be evaluated by measuring the effect of removing of an attribute $a \in C$ from the attribute set C on the positive region defined by the table $\mathcal{A}$. As shown previously, the number $\gamma(C, D)$ expresses the degree of dependency between attributes C and D, or accuracy of approximation of U/D by C. We can ask how the coefficient $\gamma(C, D)$ changes when an attribute a is removed, i.e., what is the difference between $\gamma(C, D)$ and $\gamma((C \Leftrightarrow \{a\}, D)$. We can normalize the difference and define the significance of an attribute a as

$$\sigma_{(C,D)}(a) = \frac{(\gamma(C, D) \Leftrightarrow \gamma(C \Leftrightarrow \{a\}, D))}{\gamma(C, D)} = 1 \Leftrightarrow \frac{\gamma(C \Leftrightarrow \{a\}, D)}{\gamma(C, D)},$$

Thus the coefficient $\sigma(a)$ can be understood as the error of classification which occurs when attribute a is dropped. The significance coefficient can be extended to the set of attributes as follows:

$$\sigma_{(C,D)}(B) = \frac{(\gamma(C, D) \Leftrightarrow \gamma(C \Leftrightarrow B, D))}{\gamma(C, D)} = 1 \Leftrightarrow \frac{\gamma(C \Leftrightarrow B, D)}{\gamma(C, D)},$$

denoted by $\sigma(B)$, if C and D are understood, where B is a subset of C.

If B is a reduct of C, then $\sigma(C \Leftrightarrow B) = 0$, i.e., removing any reduct complement from the set of conditional attributes enables to make decisions with certainty, whatsoever.

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

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

denoted simply as $\varepsilon(B)$, will be called an *error of reduct approximation.* It expresses how exactly the set of attributes B approximates the set of condition attributes C (relatively to D).

The concept of approximate reduct (with respect to the positive region) is a generalization of the reduct concept. A minimal subset B of condition attributes C, such that $\gamma(C, D) = \gamma(B, D)$, or $\varepsilon_{(C,D)}(B) = 0$ is a reduct (preserving the positive region). The idea of an approximate reduct can be useful in those cases when a smaller number of condition attributes is preferred over the accuracy of classification on training data. This can allow to increase the classification accuracy on testing data. The error level of reduct approximation should be tuned for a given data set to achieve this effect.

Section 1.7.3 introduces several other methods of reduct approximation that are based on other measures than positive region. Experiments show that by tuning the approximation level one can, in most cases, increase the classification quality of new objects. It is important to note once again that Boolean reasoning may be used to compute these different types of reducts and to extract relevant approximations from them (see e.g. [420].

1.7.2 Feature Extraction Methods

The discretization step determines how coarsely we want to view the world. For instance, temperature, which is usually measured in real numbers, can be discretized into two, three or more, but finitely many, intervals. Another example could be heart-beat rate at rest. Although the parameter is already expressed as discrete value (i.e. a natural number), medical doctors will usually not distinguish among, say 68 or 72 beats per minute, and classify it as normal. On the other hand, 48 to 56 beats per second is considered low, (but normal for a trained long-distance runner) while 120 to 140 beats will be very fast and abnormal unless it is the rate for a fetus in a certain digestional stage. One can easily see that the selection of appropriate intervals and partitioning of attribute value sets is a complex problem and its complexity can grow exponentially in the number of attributes to be discretized. Discretization is a step that is not specific to the rough set approach but that most rule or tree induction algorithms currently require for them to perform well.

A number of successful approaches to the problem of finding effective heuristics for real value attributes quantization (discretization) has been proposed by machine learning, pattern recognition and KDD researchers see, e.g. [46, 68, 91, 232, 287, 354].

The rough set community has been also committed to constructing efficient algorithms for new feature extraction, in particular the efforts have been focused on discretization and symbolic attribute value grouping (see e.g. [179, 180, 46, 245, 182, 246, 248, 238]).

Applications of rough set methods combined with Boolean reasoning [28] have been developed for extraction of new features from data tables under an assumption that these features belong to a predefined set.

The most successful among these methods are:

- discretization techniques (see e.g. [245, 246, 238, 247, 239, 248]),
- methods of partitioning (grouping) of nominal (symbolic) attribute value sets (see e.g. [238, 248, 242, 243, 244]) and
- combinations of the above methods (see e.g. [242, 243, 244]).

Searching for new features expressed by multi-modal formulae (see e.g. [15, 16, 17]) needs also to be mentioned here as a successful method for feature extraction.

The results reported in the above cited papers show that the discretization problems and symbolic value partition problems are of high computational complexity (i.e. NP-complete or NP-hard). This clearly justifies the need to design efficient heuristics.

Discretization.

We will now concentrate on the basic discretization methods based on the rough set and Boolean reasoning approaches. In the discretization of a decision table $\mathcal{A} = (U, A \cup \{d\})$, where $V_a = [v_a, w_a)$ is an interval of reals, we search for a partition P_a of V_a for any $a \in A$. Any partition of V_a is defined by a sequence of the so-called cuts $v_1 < v_2 < ... < v_k$ from V_a. Hence, any family of partitions $\{P_a\}_{a \in A}$ can be identified with a set of cuts. In the discretization process we search for a set of cuts satisfying some natural conditions.

Example 1.7.1 Let us consider a (consistent) decision system (Tab. 1.6 (a)) with two conditional attributes a and b and seven objects $u_1, ..., u_7$. The values of attributes on these objects and the values of decision d are presented in Tab. 1.6. Geometrical interpretation of objects and decision classes are shown in Fig.1.6.

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

(a)

$$\Longrightarrow$$

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

(b)

Table 1.6: Discretization process. (a) The original decision system $\mathcal{A}$. (b) **P**-discretization of $\mathcal{A}$, where $\mathbf{P} = \{(a, 0.9),\ (a, 1.5),\ (b, 0.75),\ (b, 1.5)\}$

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

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

The sets of values of a and b on objects from U are given by

$$\begin{aligned} a(U) &= \{0.8, 1, 1.3, 1.4, 1.6\}\,; \\ b(U) &= \{0.5, 1, 2, 3\}\,, \end{aligned}$$

respectively.

We will describe a discretization process that returns a partition of the value sets of conditional attributes into intervals. The partition is done in such a way that if the name of the interval containing an arbitrary object is substituted for any object instead of its original value in $\mathcal{A}$ a consistent decision system is also obtained. In this way the size of the value attribute sets in a decision system is reduced.

In our example the following intervals for condition attributes are obtained:
$[0.8.1)$; $[1, 1.3)$; $[1.3, 1.4)$; $[1.4, 1.6)$ for a;
$[0.5, 1)$; $[1, 2)$; $[2, 3)$ for b
defined by objects in decision system. The reader may notice that we do not consider intervals $[0, 0.5)$, $[1.6, 2)$ for a and $[0, 0.5)$, $[3, 4)$ for b. The reason for that will be clear later.

The idea of cuts is used now. Cuts are pairs (a, c) where $c \in V_a$. We will restrict our considerations for cuts defined by the middle points of the intervals defined above. The following cuts are obtained (see Fig. 1.7):
$(a, 0.9)$; $(a, 1.15)$; $(a, 1.35)$; $(a, 1.5)$;
$(b, 0.75)$; $(b, 1.5)$; $(b, 2.5)$.

Any cut defines a new conditional attribute with binary values. For example, the attribute corresponding to the cut $(a, 1.2)$ is equal to 0 if $a(x) < 1.2$, otherwise is equal to 1. Hence, objects positioned on different sides of the straight line $a = 1.2$ are discerned by this cut. The reader may now see why some of the above mentioned intervals have been eliminated from our considerations: cuts positioned in these intervals will not discern any pair of objects in the table.

Any set P of cuts defines a new conditional attribute a_P for any a. One should consider a partition of the value set of a by cuts from P and put the unique names for the elements of these partition. Lets take the following set of cuts: $P = \{(a, 0.9),(a, 1.5),(b, 0.75),(b, 1.5)\}$. This set assigns all values of a less then 0.9 to the interval named 0, all values in the all values in the interval $[0.9, 1.5)$ to the interval 1 and all values from $[1.5, 4)$ to the interval 2. An analogous

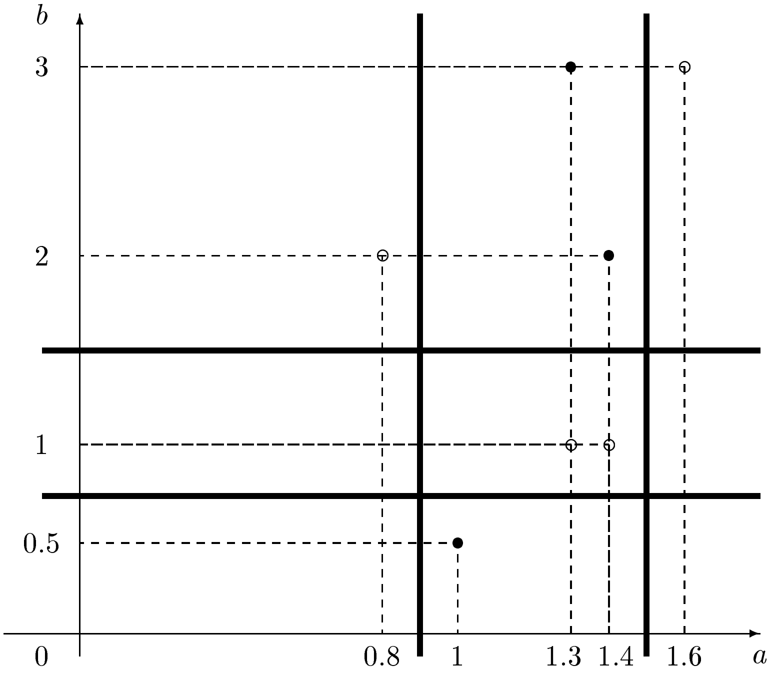


Figure 1.6: A geometrical representation of data and cuts.

construction is done for b. The values of the new attributes a_P and b_P are shown in Tab. 1.6 (b).

The next natural question is: How to construct a set of cuts with a minimal number of elements discerning all pairs of objects to be discerned? We will show that this can be done using Boolean reasoning.

Let us introduce a Boolean variable corresponding to any attribute a and any interval determined by a. In our example the set of Boolean variables defined by $\mathcal{A}$ is equal to

$$VB(A) = \left\{p_1^a, p_2^a, p_3^a, p_4^a, p_1^b, p_2^b, p_3^b\right\};$$

where $p_1^a \sim [0.8;1)$ of a (i.e. p_1^a corresponds to the interval $[0.8;1)$ of attribute a); $p_2^a \sim [1;1.3)$ of a; $p_3^a \sim [1.3;1.4)$ of a; $p_4^a \sim [1.4;1.6)$ of a; $p_1^b \sim [0.5;1)$ of b; $p_2^b \sim [1;2)$ of b; $p_3^b \sim [2;3)$ of b (see Fig. 1.7).

Let us recall that a valuation of propositional variables is any function from the set of propositional variables into $\{0,1\}$. Now one can easily observe that there is a one-to-one correspondence between the set of valuations of propositional variables defined above for a given $\mathcal{A}$ and the set of cuts in $\mathcal{A}$. The rule is as follows: (i) for any cut choose the interval containing it and next the propositional variable corresponding to it; (ii) for any propositional variable choose a cut in the interval corresponding to the variable. For example, the set of cuts $\mathbf{P} = \{(a,0.9),(a,1.5),(b,0.75),(b,1.5)\}$ corresponds to the valuation assigning 1 to the propositional variables: p_1^a, p_4^a, p_1^b, p_2^b only. Having this correspondence, we will say that the Boolean formula built from the propositional variables is satisfied by a given set of cuts iff it is satisfied by the valuation corresponding to that set (i.e. taking value 1 only on variables corresponding to cuts from this set).

Now, using our example, we will show how to built a Boolean formula $\Phi^{\mathcal{A}}$, called the discernibility formula for a given $\mathcal{A}$ and with the following property: *the set of prime implicants*

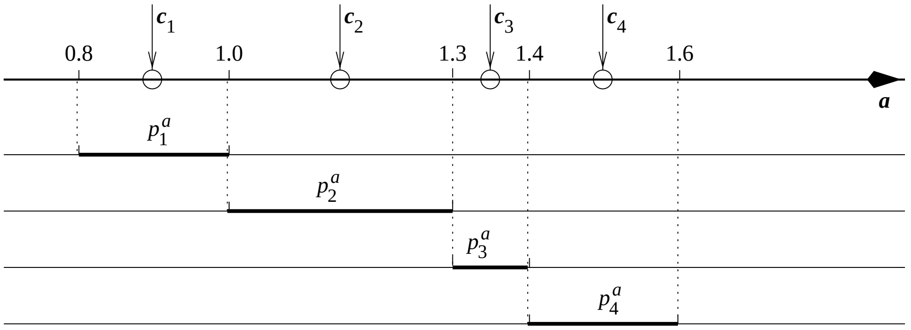


Figure 1.7: The set of cuts $(a,c_1),(a,c_2),(a,c_3),(a,c_4)$ on a, the set of propositional variables p_1^a, p_2^a, p_3^a, p_4^a and the set of intervals corresponding to these variables in $\mathcal{A}$ (see Example 1.7.1)

of $\Phi^{\mathcal{A}}$ defines uniquely the family of all minimal set of cuts discerning objects in $\mathcal{A}$. Moreover, any valuation satisfying this formula determines the set of cuts discerning all object pairs to be discerned.

Having in mind the discernibility matrix for $\mathcal{A}$, one can see that we should choose at least one cut on one of the attributes appearing in the entry (x_i, x_j) of the discernibility matrix of $\mathcal{A}$ for any objects x_i and x_j discernible by conditional attributes which have different decisions.

The discernibility formulae $\psi(i,j)$ for different pairs (u_i, u_j) of discernible objects from $U \times U$ with different decisions have the following form:

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

For example, formula $\psi(5,6)$ is true on the set of cuts if there exists a cut $p_1 = (a,c)$ on V_a in this set such that $c \in [1.4, 1.6)$ or a cut $p_2 = (b,c)$ on V_b that $c \in [2,3)$.

The discernibility formula $\Phi^{\mathcal{A}}$ in CNF form is given by taking all the above conditions:

$$\begin{aligned}
\Phi^{\mathcal{A}} = \; & \left(p_1^a \vee p_1^b \vee p_2^b\right) \wedge \left(p_1^a \vee p_2^a \vee p_3^b\right) \wedge \left(p_1^a \vee p_2^a \vee p_3^a\right) \wedge \left(p_2^a \vee p_3^a \vee p_1^b\right) \\
& \wedge p_2^b \wedge \left(p_2^a \vee p_2^b \vee p_3^b\right) \wedge \left(p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b\right) \wedge \left(p_3^a \vee p_4^a\right) \\
& \wedge \left(p_4^a \vee p_3^b\right) \wedge \left(p_2^a \vee p_1^b\right) \wedge \left(p_2^b \vee p_3^b\right) \wedge \left(p_3^a \vee p_2^b\right).
\end{aligned}$$

Transforming formula $\Phi^{\mathcal{A}}$ to the DNF form we obtain four prime implicants:

$$\begin{aligned}
\Phi^{\mathcal{A}} = \; & \left(p_2^a \wedge p_4^a \wedge p_2^b\right) \vee \left(p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b\right) \\
& \vee \left(p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b\right) \vee \left(p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b\right).
\end{aligned}$$

If we decide to take e.g. the last prime implicant $S = \left\{p_1^a, p_4^a, p_1^b, p_2^b\right\}$, we obtain the following set of cuts

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

The new decision system $\mathcal{A}^{\mathbf{P}(S)}$ is represented in Table 1.6 (b). □

A more formal description of the discretization problem is now presented.

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

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

Two sets of cuts $\mathbf{P}', \mathbf{P}$ are equivalent, i.e. $\mathbf{P}' \equiv_{\mathcal{A}} \mathbf{P}$, iff $\mathcal{A}^{\mathbf{P}} = \mathcal{A}^{\mathbf{P}'}$. The equivalence relation $\equiv_{\mathcal{A}}$ has a finite number of equivalence classes. In the sequel we will not discern between equivalent families of partitions.

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

One can show [245] that the decision problem of checking if for a given decision system $\mathcal{A}$ and an integer k there exists an irreducible set of cuts $\mathbf{P}$ in $\mathcal{A}$ such that $card(\mathbf{P}) < k$ is NP-complete. The problem of searching for an optimal set of cuts $\mathbf{P}$ in a given decision system $\mathcal{A}$ is NP-hard.

However, one can construct efficient heuristics returning semi-minimal sets of cuts [245, 246, 250, 238, 242, 243, 244, 326]. Here we discuss the simplest one based on the Johnson strategy. Using this strategy one can look for a cut discerning the maximal number of object pairs (with different decisions), next one can eliminate all already discerned object pairs and repeat the procedure until all object pairs to be discerned are discerned. It is intersecting to note that this can be realized by computing the minimal relative reduct of the corresponding decision system.

Again we will explain this idea using our example.

From a given decision system one can construct a new decision system $\mathcal{A}^*$ having as objects all pairs of objects from $\mathcal{A}$ with different decision values, so all object pairs to be discerned. We are adding one more object *new* on which all constructed new conditional attributes have value 0 and on which the decision value is also 0. The new decision is equal to 1 on all other objects in the new decision system. The set of condition attributes in the new decision system $\mathcal{A}^*$ is equal to the set of all attributes defined by all cuts (or all propositional variables considered above). These attributes are binary. The value of the new attribute corresponding to a cut (a, c) on the pair (u_i, u_j) is equal to 1 iff this cut is discerning objects (u_i, u_j) (i.e. $min(a(u_i), a(u_j)) < c < max(a(u_i), a(u_j)))$ and 0 otherwise. One can formulate this condition in another way. The value of the new attribute corresponding to the propositional variable p_s^a on the pair (u_i, u_j) is equal to 1 iff the interval corresponding to p_s^a is included in $[min(a(u_i), a(u_j)),\ max(a(u_i), a(u_j))]$ and 0 otherwise.

The resulting new decision system $\mathcal{A}^*$ is shown in Tab. 1.7.

Objects in $\mathcal{A}^*$ are all pairs (x_i, x_j) discernible by the decision d. One more object is included, namely *new* with all values of attributes equal to 0. This allows formally to keep the condition: *"at least one occurrence of 1 (for conditional attributes) appears in any row for any subset of columns corresponding to any prime implicant"*.

The relative reducts of this table correspond exactly to the prime implicants of the function

$\mathcal{A}^*$	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b	d^*
(u_1, u_2)	1	0	0	0	1	1	0	1
(u_1, u_3)	1	1	0	0	0	0	1	1
(u_1, u_5)	1	1	1	0	0	0	0	1
(u_4, u_2)	0	1	1	0	1	0	0	1
(u_4, u_3)	0	0	1	0	0	1	1	1
(u_4, u_5)	0	0	0	0	0	1	0	1
(u_6, u_2)	0	1	1	1	1	1	1	1
(u_6, u_3)	0	0	1	1	0	0	0	1
(u_6, u_5)	0	0	0	1	0	0	1	1
(u_7, u_2)	0	1	0	0	1	0	0	1
(u_7, u_3)	0	0	0	0	0	1	1	1
(u_7, u_5)	0	0	1	0	0	1	0	1
new	0	0	0	0	0	0	0	0

Table 1.7: Decision system $\mathcal{A}^*$ constructed from $\mathcal{A}$

$\Phi^{\mathcal{A}}$ (for the proof see e.g. [238]).

Our *"MD heuristic"* is based on searching for a cut with maximal number of object pairs discerned by this cut [245], [242]. The idea is analogous to the Johnson approximation algorithm and can be formulated as follows:

ALGORITHM MD-heuristic (Semi-optimal family of partitions**)**

Step 1. *Construct the table $\mathcal{A}^*$ from $\mathcal{A}$ and erase the last row (i.e. a "new" element) from $\mathcal{A}^*$; set $\mathcal{B} := \mathcal{A}^*$;*

Step 2. *Choose a column from $\mathcal{B}$ with the maximal number of occurrences of 1's;*

Step 3. *Delete from $\mathcal{B}$ the column chosen in Step 2 and all rows marked in this column by 1;*

Step 4. ***If** $\mathcal{B}$ is non-empty then go to Step 2 **else** Stop.*

In our example the algorithm is choosing first p_2^b next p_2^a and finally p_4^a. Hence $S = \{p_2^a, p_4^a, p_2^b\}$ and the resulting set of cuts $P = \{(a, 1.15), (a, 1.5), (b, 1.5)\}$. Fig 1.8 is showing the constructed set of cuts (marked by bold lines).

The algorithm based on Johnson's strategy described above is searching for a cut which discerns the largest number of pairs of objects (MD-heuristic). Then we move the cut c from $\mathcal{A}^*$ to the resulting set of cuts $\mathbf{P}$ and remove from U^* all pairs of objects discerned by c. Our algorithm is continued until $U^* = \{new\}$. Let n be the number of objects and let k be the number of attributes of decision system $\mathcal{A}$. Then $card\,(A^*) \leq (n \Leftrightarrow 1)\,k$ and $card\,(U^*) \leq \frac{n(n-1)}{2}$. It is easy to observe that for any cut $c \in A^*$ we need $O\left(n^2\right)$ steps to find the number of all pairs of objects discerned by c. Hence the straightforward realization of this algorithm requires $O\left(kn^2\right)$ of memory space and $O(kn^3)$ steps to determine one *cut*, so it is not feasible in practice. The MD-heuristic presented in [241] determines the best cut in $O\left(kn\right)$ steps using $O\left(kn\right)$ space only. This heuristic is very efficient with respect to the time necessary for decision rules generation as well as with respect to the quality of unseen object classification. (see e.g [238, 246, 242]).

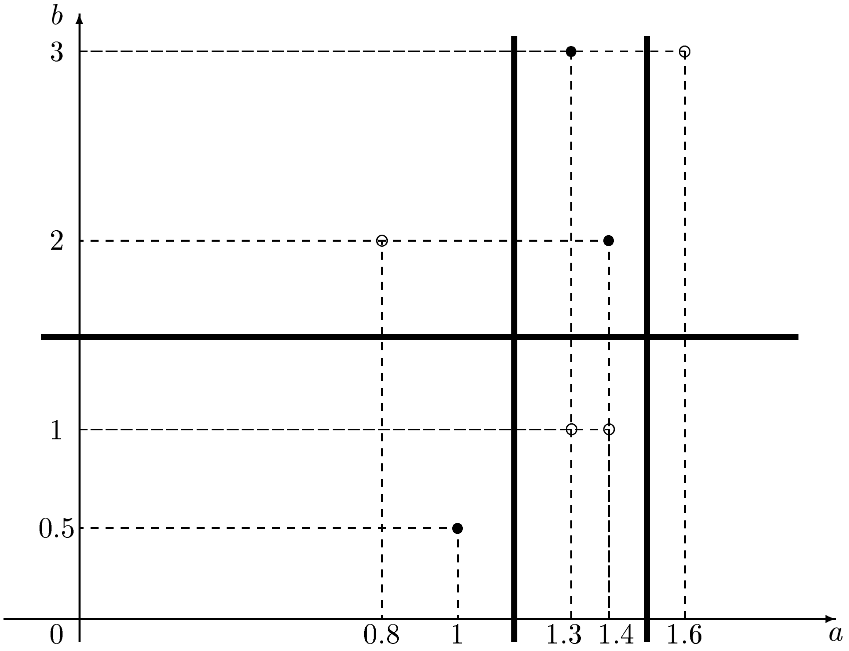


Figure 1.8: The minimal set of cuts of $\mathcal{A}$

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

One can extend the presented approach (see e.g. [248], [242], [243], [243],) to the case of symbolic (nominal, qualitative) attributes as well as to the case when in a given decision system nominal and numeric attribute appear. The received heuristics are of very good quality.

Features Defined by Oblique Hyperplanes.

Let us consider the decision table $\mathcal{A} = (U, A \cup \{d\})\,; U = \{u_1, u_2, \ldots, u_n\}\,; A = \{a_1, \ldots, a_k\}$; $d(U) = \{1, \ldots, r\}$.

Each of the points from U belongs to one of the decision classes $C_1, C_2, ..., C_r$ defined by the decision d i.e. $C_i = \{u \in U : d(u) = i\}$ for $i \in \{1, \ldots, r\}$. We denote the cardinality of classes C_i by $n_i = card\,(C_i)$.

Let $conflict(\mathcal{A})$ denote the number of pairs of objects with different decisions; this value can be computed as follows:

$$conflict\,(\mathcal{A}) = \frac{1}{2} card\,\{(u_i, u_j) \in U \times U : (d\,(u_i) \neq d\,(u_j))\}$$

Each hyperplane in $\mathcal{R}^k$:

$$H = \{(x_1, x_2, ..., x_k) \in \mathcal{R}^k : \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_k x_k + \alpha_{k+1} = 0\}$$
$$\text{for some } \alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1} \in \mathcal{R}$$

splits C_i into two subclasses defined by:

$$C_i^L\,(H) = \{u \in C_i : \alpha_1 a_1(u) + \alpha_2 a_2(u) + ... + \alpha_k a_k(u) + \alpha_{k+1} < 0\};$$

which corresponds to the left part of C_i relative to H; and

$$C_i^R\,(H) = \{u \in C_i : \alpha_1 a_1(u) + \alpha_2 a_2(u) + \cdots + \alpha_k a_k(u) + \alpha_{k+1} \geq 0\};$$

which corresponds to the right part of C_i relative to H.

In the concept of discretization one can apply the following measure to estimate the quality of cuts (or hyperplanes parallel to the axes):

$$award(H) = \sum_{i \neq j} card\left(C_i^R(H)\right) \cdot card\left(C_j^L(H)\right)$$

If $award(H) > award(H')$ then the number of pairs of objects in different decision discernible classes by the hyperplane H is greater than the corresponding number defined by the hyperplane H'. This measure has been applied in the MD-heuristic as the number of discernible of object pairs.

Let $L_i = card(C_i^{L,H})$ and $R_i = card(C_i^{R,H})$ for $i = 1, \ldots, r$ and let $R = R_1 + \cdots + R_r$; $L = L_1 + \cdots + L_r$. Then we have the following equality:

$$award(H) = \sum_{i \neq j} (L_i \cdot R_j) = L \cdot R \Leftrightarrow \sum_{i=1}^{r} (L_i \cdot R_i)$$

Using those notations, the quality of a given hyperplane H can be estimated by applying other measures:

1. Sum Minority

$$Sum\_Minority\,(H) = \min\left\{|L_i|\right\}_{i=1}^{r} + \min\left\{|R_i|\right\}_{i=1}^{r} \tag{1.1}$$

2. Max Minority

$$Max\_Minority\,(H) = \max\left\{\min\left\{|L_i|\right\}_{i=1}^{r}, \min\left\{|R_i|\right\}_{i=1}^{r}\right\}$$

3. Sum Impurity

$$Sum\_impurity\,(H) = \sum_{i=1}^{r} |L_i| \cdot (i \Leftrightarrow avg_L)^2 + \sum_{i=1}^{r} |R_i| \cdot (i \Leftrightarrow avg_R)^2$$

where $avg_L = \frac{\sum_{i=1}^{r} i \cdot |L_i|}{\sum_{i=1}^{r} |L_i|}$ and $avg_R = \frac{\sum_{i=1}^{r} i \cdot |R_i|}{\sum_{i=1}^{r} |R_i|}$ are averages of object decision values on the left side and the right side of H (respectively). Hence, the Sum-Impurity measure is a sum of variations on both sides of H and it is minimal if H separates the set of objects correctly.

These measures can be treated as *energies* of a given hyperplane and usually we want to get the hyperplane with lowest energy.

In general the optimization problems based on different measures such as *award* are either NP-complete or NP-hard depending on the formalization of problems [238].

We report the results of experiments with $award(H)$ only. We would like to point out that the application of the penalty function can be an effective tool in searching for relevant features not only defined by hyperplanes.

We can also define the function $penalty(H)$ as follows:

$$penalty(H) = \sum_{i=1}^{r} card\left(C_i^{U,H}\right) \cdot card\left(C_i^{L,H}\right) = \sum_{i=1}^{r} (L_i \cdot R_i)$$

One can apply the function *award* to measure the quality of hyperplanes not necessarily parallel to the axes i.e.

$$power_1(H) = award(H).$$

More advanced functions like:

$$\begin{aligned} power_2(H) &= \frac{w_1 \cdot award(H)}{penalty(H) + w_2}; \\ power_3(H) &= w_1 \cdot award(H) \Leftrightarrow w_2 \cdot penalty(H). \end{aligned}$$

can also be used.

Searching for Optimal Oblique Hyperplanes.

There are numerous methods for searching for optimal hyperplanes[120, 233]. The presented methods are based on various heuristics like "simulated annealing" [120], "randomized induction" [233] but all of them are designed for one chosen measure.

Let us fix an axis (e.g. the axis x_1 corresponding to the attribute a_1) and a positive integer b. In any two-dimensional plane $\mathbf{L}(x_1, x_i)$ we choose vectors $v_1^i, v_2^i, ..., v_{2^b}^i$ (not parallel to x_1) defined by:

$$v_j^i = \left[\alpha_j^i, 0, \ldots, 0, \overset{i\text{-th position}}{1}, 0, \ldots, 0\right] \text{ for } i = 2, .., k \text{ and } j = 1, .., 2^b$$

These vectors can be chosen e.g. by applying one of the following steps:

1. Random choice of 2^b values: $\alpha_1^i, \alpha_2^i, \ldots, \alpha_{2^b}^i$.
2. The values $\alpha_1^i, \alpha_2^i, \ldots, \alpha_{2^b}^i$ are chosen in such a way that all the angles between successive vectors are equal i.e. $\alpha_j^i = ctg\left(j \frac{\pi}{1+2^l}\right)$
3. The sequence $\alpha_1^i, \alpha_2^i, \ldots, \alpha_{2^b}^i$ is chosen to be an arithmetical progression (e.g. $\alpha_j^i = j \Leftrightarrow 2^{l-1}$).

Let us observe that any vector from the list $v_1^i, v_2^i, ..., v_{2^b}^i$ can be represented by b bits, namely the vector v_j^i is encoded by the b-bit string $binary(j)$ being the binary representation of the integer j.

We consider a set of vectors of the form $\left\{v_{j_2}^2, v_{j_3}^3, ..., v_{j_k}^k\right\}$:

$$\begin{aligned} v_{j_2}^2 &= \left[\alpha_{j_2}^2, 1, 0, 0, \ldots, 0\right] \\ v_{j_3}^3 &= \left[\alpha_{j_3}^3, 0, 1, 0, \ldots, 0\right] \\ &\ldots\ldots. \\ v_{j_k}^k &= \left[\alpha_{j_k}^k, 0, 0, 0, \ldots, 1\right]. \end{aligned}$$

It is easy to note that for any set of indexes $j_2, j_3, ..., j_k \in \left\{1, ..., 2^b\right\}$ the vectors $v_{j_2}^2$, $v_{j_3}^3$, ..., $v_{j_k}^k$ are linearly independent in the vector space $\mathcal{R}^k$.

Let $\mathbf{L} = Lin(v_{j_2}^2, v_{j_3}^3, ..., v_{j_k}^k)$ be the linear subspace generated by the vectors $v_{j_2}^2$, $v_{j_3}^3$, ..., $v_{j_k}^k$. Hence $rank(\mathbf{L}) = k \Leftrightarrow 1$. To define a hyperplane spanned over these vectors must fix a point in $\mathcal{R}^k$. Since vectors of the form $v_{j_i}^i$ are not parallel to the axis x_1, we can choose this point on the axis x_1 i.e. a point of the form $\mathbf{P}_0 = (p, 0, 0, ..., 0)$.

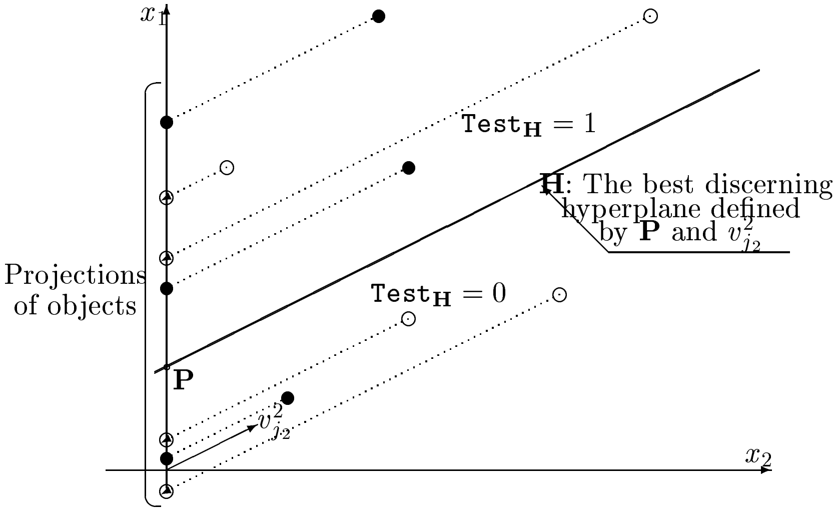

x_1
$\mathtt{Test}_{\mathbf{H}} = 1$
$\mathbf{H}$: The best discerning hyperplane defined by $\mathbf{P}$ and $v^2_{j_2}$
Projections of objects
$\mathtt{Test}_{\mathbf{H}} = 0$
$\mathbf{P}$
$v^2_{j_2}$
x_2

Figure 1.9: Interpretation of the `Test` and `Projection` functions in two-dimensional space.

Now we define a hyperplane defined by $\left(\mathbf{P}_0, v^2_{j_2}, v^3_{j_3}, ..., v^k_{j_k}\right)$ as follows:

$$\begin{aligned} H &= \mathbf{P}_0 \oplus \mathbf{L} = \left\{\mathbf{P} \in \mathcal{R}^k : \overleftrightarrow{\mathbf{P}_0\mathbf{P}} \in \mathbf{L}\right\} \\ &= \left\{(x_1, x_2, \ldots, x_k) \in \mathcal{R}^k : [x_1 \Leftrightarrow p, x_2, ..., x_k] = b_2 v^2_{j_2} + b_3 v^3_{j_3} + \cdots + b_k v^k_{j_k}\right\} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for some } b_2, ..., b_k \in \mathcal{R} \\ &= \left\{(x_1, x_2, \ldots, x_k) \in \mathcal{R}^k : x_1 \Leftrightarrow p = \alpha^2_{j_2} x_2 + \alpha^3_{j_3} x_3 + \cdots + \alpha^k_{j_k} x_k\right\} \\ &= \left\{(x_1, x_2, \ldots, x_k) \in \mathcal{R}^k : x_1 \Leftrightarrow \alpha^2_{j_2} x_2 \Leftrightarrow \alpha^3_{j_3} x_3 \Leftrightarrow \cdots \Leftrightarrow \alpha^k_{j_k} x_k \Leftrightarrow p = 0\right\} \end{aligned}$$

Having the hyperplane H defined by $\left(\mathbf{P}_0, v^2_{j_2}, v^3_{j_3}, ..., v^k_{j_k}\right)$ we can check in time $O(k)$ on which side of the hyperplane H a given object $u \in U$ is (Figure 1.9). The corresponding function $Test_H : U \to \{0, 1\}$ is defined by

```
FUNCTION          Test($u$):$\{0,1\}$

     If ($a_1(u) \Leftrightarrow \alpha^2_{j_2} a_2(u) \Leftrightarrow \alpha^3_{j_3} a_3(u) \Leftrightarrow \cdots \Leftrightarrow \alpha^k_{j_k} a_k(u) \geq p$) then
          Test:= 1;
     Else Test := 0;
```

Assuming that vectors $v^2_{j_2}, v^3_{j_3}, ..., v^k_{j_k}$ are given, we can find the projection of any object $u \in U$ onto x_1 parallel to $\mathbf{L} = Lin(v^2_{j_2}, v^3_{j_3}, ..., v^k_{j_k})$ in time $O(k)$.

```
FUNCTION          Projection($u$):Real;

     Projection := $f_1(u) \Leftrightarrow \alpha^2_{j_2} f_2(u) \Leftrightarrow \alpha^3_{j_3} f_3(u) \Leftrightarrow \cdots \Leftrightarrow \alpha^k_{j_k} f_k(u)$;
```

In the search for the optimal set of separating hyperplanes we will use a genetic algorithm. Now we describe our learning algorithm. This is an algorithm searching for hyperplanes which should discern between all pairs of objects discernible by the generalized decision in a given decision table. In case of a consistent decision table the algorithm will be continued until the hyperplanes cut the space $\mathcal{R}^k$ into regions containing objects from one decision class only. In general, this condition is replaced by the equation $\partial_{\mathbf{B}} = \partial_{\mathbf{A}}$, where $\mathbf{B} = (U, B \cup \{d\})$ is a decision table defined by replacing the set A of attributes of a given decision table $\mathbf{A} = (U, A \cup \{d\})$ by the set B of test functions defined by hyperplanes constructed as follows.

ALGORITHM: Hyperplane Extraction from data.

```
Step 1  Initialize a new table $\mathbf{B} = (U, B \cup \{d\})$ such that $B = \emptyset$;
Step 2  (Search for the best hyperplane)
            for $i := 1$ to $k$ do
                Search for the best hyperplane $H_i$ attached to the axis
                  $x_i$ using a genetic algorithm;
            $H$ := Best hyperplane from the set $\{H_1, H_2, ..., H_k\}$;
Step 3  $B := B \cup \{Test_H\}$;
Step 4  If $\partial_{\mathbf{B}} = \partial_{\mathbf{A}}$ then Stop else goto Step2.
```

Now we will describe the search for the best hyperplane attached to x_1.

Chromosomes : Any chromosome consists of a bit vector of length $b(k \Leftrightarrow 1)$ containing $(k \Leftrightarrow 1)$ fragments of length b. The $i \Leftrightarrow th$ fragment $(i = 1, 2, .., k \Leftrightarrow 1)$ corresponds to one of the vectors of the form v_j^{i+1}. Let us consider two examples of chromosomes (assuming $b = 4$):

$$\begin{array}{rcccccccc}
chr_1 & = & 0010 & 1110 & ... & 0100 & ... & 1010 \\
 & & 1 & 2 & ... & i & ... & k \Leftrightarrow 1 \\
chr_2 & = & 0000 & 1110 & ... & 1000 & ... & 0101 \\
 & & 1 & 2 & ... & i & ... & k \Leftrightarrow 1
\end{array}$$

Operators: The genetic operators are defined as follows:

1. Mutation and Selection are defined in a standard way [208]. Mutation of chr_1 is realized in two steps; first one block, e.g. the i^{th}, is randomly chosen, and then its contents (in our example "0100") are randomly changed into a new block, e.g. "1001". The described example of mutation changes the chromosome chr_1 into chr_1', where:

$$\begin{array}{rcccccccc}
chr_1' & = & 0010 & 1110 & ... & \underline{1001} & ... & 1010 & . \\
 & & 1 & 2 & ... & i & ... & k \Leftrightarrow 1 &
\end{array}$$

2. Crossover is done by exchanging entire chromosome fragments corresponding to one vector. The result of a crossover of two chromosomes is realized in two steps as well; first the block position i is randomly chosen, and next, the contents of the $i \Leftrightarrow th$ blocks of two chromosomes are exchanged. For example, if crossover is performed on chr_1, chr_2 and the $i \Leftrightarrow th$ block position is randomly chosen then we obtain their offspring:

$$\begin{array}{rcccccccc}
chr_1' & = & 0010 & 1110 & ... & \underline{1000} & ... & 1010 \\
 & & 1 & 2 & ... & i & ... & k \Leftrightarrow 1 \\
chr_2' & = & 0000 & 1110 & ... & \underline{0100} & ... & 0101 \\
 & & 1 & 2 & ... & i & ... & k \Leftrightarrow 1
\end{array}$$

Fitness function: The fitness of any chromosome is equal to the number of object pairs in different decision classes discerned by the best hyperplane among hyperplanes parallel to all the vectors encoded in the chromosome. It is computed as follows:

In the first step a procedure searching for the best hyperplane is executed and, the point of intersection of the hyperplane with a fixed axis is returned as an output. The procedure searches for projections of all objects on the distinguished axis. The projections are parallel to all vectors defined by the chromosome and they are calculated by the procedure *Projection.* Next the set of projection points is sorted and the hyperplane is located at the best position between these projections. The best position is defined by the best hyperplane containing this point. The hyperplane quality and, in consequence, the fitness of the chromosome can be calculated using different measures introduced in previous section.

The most time-consuming step in our algorithm is to compute the fitness function values. The value of this function is equal to the number of pairs discerned by the best hyperplane among hyperplanes parallel to all the vectors encoded in the chromosome. Searching for this hyperplane is realized in time $O(nk + n\log n)$ where n is the number of objects and k is the number of conditional attributes.

Searching for Optimal Set of Surfaces.

In the previous section we considered a method for searching for semi-optimal hyperplanes. Below, we present a natural way to generate a semi-optimal set of *high degree surfaces (curves)* by applying the existing methods for hyperplanes.

Let us note that any i^{th} *degree surface in* $\mathcal{R}^k$ can be defined as follows:

$$S = \left\{ (x_1, \ldots, x_k) \in \mathcal{R}^k : P\left(x_1, \ldots, x_k\right) = 0 \right\}$$

where $P\left(x_1, \ldots, x_k\right)$ is an arbitrary i^{th} degree polynomial over k variables.

Any i^{th} degree polynomial is a linear combination of monomials, each of degree not greater than i. By $\eta\left(i,k\right)$ we denote the number of k-variable monomials of degrees $\leq i$. Then, instead of searching for i^{th} degree surfaces in the k-dimensional affine real space $\mathcal{R}^k$, we can search for hyperplanes in the space $\mathcal{R}^{\eta(i,k)}$.

It is easy to see that the number of j^{th} degree monomials built from k variables is equal to $\left(\begin{array}{c} j+k \Leftrightarrow 1 \\ k \Leftrightarrow 1 \end{array}\right)$. Then we have

$$\eta\left(i,k\right) = \sum_{j=1}^{i} \left(\begin{array}{c} j+k \Leftrightarrow 1 \\ k \Leftrightarrow 1 \end{array}\right) = O\left(k^i\right). \tag{1.2}$$

As we can see, by applying the above surfaces we have more chances to discern objects in different decision classes with a smaller number of ”cuts”. This is because higher degree surfaces are more flexible than normal cuts. This fact can be shown by applying the VC (Vapnik-Chervonenkis) dimension to the corresponding set of functions [506].

To search for an optimal set of i^{th} degree surfaces discerning objects in different decision classes of a given decision table $\mathbf{A} = (U, A \cup \{d\})$ we can construct a new decision table $\mathbf{A}^i = (U, A^i \cup \{d\})$ where A^i is the set of all monomials of degree $\leq i$ built on attributes in A. Any hyperplane found for the decision table $\mathbf{A}^i$ is a surface in original decision table $\mathbf{A}$. The cardinality of A^i is estimated by the formula 1.2.

Hence for a better solution we pay having a dramatic increase of space and time complexity.

Let us assume, for instance, that in the Example 1.7.1 we want to search for an optimal second degree curve. First, we have to construct the new decision table $\mathbf{A}^2 = (U, A^2 \cup \{d\})$ where $A^2 = \{a, b, a^2, ab, b^2\}$.

$\mathbf{A}^2$	a	b	a^2	ab	b^2	H^*	d
u_1	0.8	2	0.64	1.6	4	0.36	1
u_2	1	0.5	1	0.5	0.25	-0.25	0
u_3	1.3	3	1.69	3.9	9	-0.14	0
u_4	1.4	1	1.96	1.4	1	1.24	1
u_5	1.4	2	1.96	2.8	4	-0.36	0
u_6	1.6	3	2.56	4.8	9	1.24	1
u_7	1.3	1	1.69	1.3	1	0.26	1

Table 1.8: The decision table $\mathbf{A}^2$

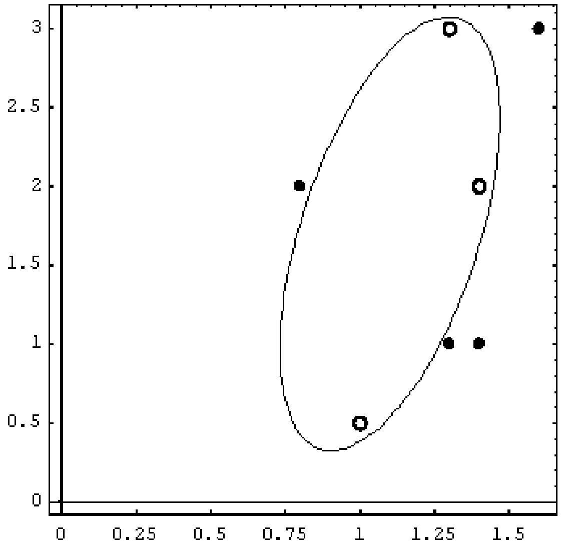


Figure 1.10: Geometrical presentation of the best surface.

There is a lot of hyperplanes which discern the set of objects $C_0 = \{u_2, u_3, u_5\}$ from the set of objects $C_1 = \{u_1, u_4, u_6, u_7\}$. For the best hyperplane among them:

$$H^* = \Leftrightarrow 24a + b + 14a^2 \Leftrightarrow 4ab + b^2 + 11$$

we can define decision rules:

$$\begin{array}{l} \textbf{if} \;\; \left(\Leftrightarrow 24a\,(u) + b\,(u) + 14\,[a\,(u)]^2 \Leftrightarrow 4a\,(u)\,b\,(u) + [b\,(u)]^2 + 11 < 0\right) \\ \textbf{then} \;\; (d\,(u) = 0) \;\; \textbf{else} \;\; (d\,(u) = 1) \end{array}$$

The geometrical interpretation of H^* in Euclidean space $\mathcal{R}^2$ is an ellipse presented in Figure 1.10.

Experimental Results.

We include the results reporting the results of some tests performed on different data tables. We have used several techniques for testing the quality of decision algorithms (sets of rules and decision trees).

In the simplest strategy called *"train-and-test"* the original data table has been randomly split into a training table and a testing table (usually the testing table is $\approx 20 \Leftrightarrow 40\%$ of the

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

Table 1.9: Data tables stored in the UC Irvine Repository

entire original decision table). We first generate a *decision algorithm* (i.e. a set of decision rules or a decision tree) for the training table and next we check its quality on the testing table. We denote by N the number of objects of the testing table and by N_1 the number of objects in this table properly recognized by the decision algorithm. The classification quality (accuracy rate) of learning algorithm is defined by $q = \frac{N_1}{N} 100\%$.

The second approach is known in Machine Learning as an $m \Leftrightarrow$*fold cross validation* ($CV \Leftrightarrow m$) [521] where m is an arbitrary positive integer. In $CV \Leftrightarrow m$ approach we divide a given data table into m equal parts. Every part is tested via the decision algorithm obtained from the $(m \Leftrightarrow 1)$ remaining parts. The classification quality of a learning algorithm is equal to the average value of all classification qualities i.e. $q = \frac{1}{n} \sum_{i=1}^{m} n_i$ where n_i is the number of objects properly recognized in the i^{th} test.

We present the results of experiments performed on different data by the two methods (rules generated from parallel cuts and decision tree generated from oblique hyperplanes). We also present an empirical analysis of decision trees for different data.

In the first series of experiments we have chosen several data tables with real value attributes from the U.C. Irvine repository [231]. For some tables, taking into account the small number of their objects, we have adopted the approach based on five-fold cross-validation ($CV \Leftrightarrow 5$). The obtained results (Table 1.10) can be compared with those reported in [68, 211] (Table 1.9). For predicting decisions on new cases we apply only decision rules generated either by the decision tree (using hyperplanes) or by rules generated in parallel with discretization.

For some tables the classification quality of our algorithm is better than that of the C4.5 or Naive -Bayes induction algorithms [355] even when used with different discretization methods [68, 211, 46].

Comparing this method with the other methods reported in [211], we can conclude that our algorithms have the shortest runtime and a good overall classification quality (in many cases our results were the best as compared to many other methods reported in the literature).

Our second example describes the classification quality of decision rules generated from data table with about 700 rows containing description of experiments with model of inverted pendulum. Data have been received from dr Mrózek (Institute of Theoretical and Applied Informatics, PAS). Two real value conditional attributes are included in the table (i.e. $\theta(t), \dot{\theta}(t)$) and the decision attribute has values: ”LEFT”, ”RIGHT”, ”NO”. The quality of classification increased when the list of conditional attributes had been extended to 5 attributes: $\theta(t)$, $\dot{\theta}(t)$, $decision\,(t \Leftrightarrow 1)$, $\theta(t \Leftrightarrow 1)$, $\dot{\theta}(t \Leftrightarrow 1)$.

The next experiment (see Table 1.11) concerns the autopilot data table containing 31864 rows describing plane states in discrete moments of time. The data table has 6 conditional

Data tables	**Diagonal cuts**			**Oblique Hyperplanes**			
	#cuts	#rules	quality	#cuts	#rules	depth of DT	quality
Australian	18	190±10	79.71%	16	180±10	9.5	82.46%
Glass	14±1	90±9	67.89%	12	80±5	7.8	70.06%
Heart	11±1	85±10	79.25%	11±1	80±10	7.6	80.37%
Iris	7±2	11±2	92.70%	6±2	6±2	4.52	96.7%
Vehicle	25	400±10	59.70%	20±2	380±5	10.51	64.42%
Diabetes	20	315	74.24%	19	300	9.27	76.08%
SatImage	47	1354	81.73%	43	1235	12.26	82.90%
Shuttle	15	47	99.99%	15	47	7.34	99.99%

Table 1.10: Results of experiments on Machine Learning data.

Data tables	*Nr of attributes*	*Nr of classes*	*Training set*	*Testing set*	*Number of rules*	*Quality of classification*
Inverted Pendulum 1	2	3	300	373	51	$88.47 \pm 4,02\%$
	2	3	400	273	56	$92.31 \pm 1.46\%$
Inverted Pendulum 2	5	3	300	372	42	$93.54 \pm 1.61\%$
	5	3	400	272	44	$93.01 \pm 2.59\%$
Aircraft	6	7	12000	19864	668	87.41%
	6	7	19864	12000	2056	97.38%
	6	7	21864	10000	2221	97.15%

Table 1.11: Results of experiments on inverted pendulum and autopilot data

attributes with real values and one decision attribute representing the angle of the rudder. The problem is to predict this angle with error not exceeding 1° for cases with known values of 6 other attributes.

Feature Extraction by Grouping of Symbolic Values.

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

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

PARTITION PROBLEM: SYMBOLIC VALUE PARTITION PROBLEM:

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

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

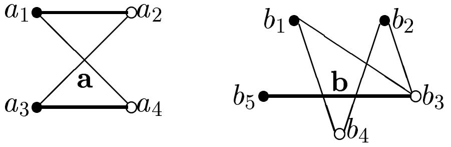
Figure 1.11: The decision table and the discernibility matrix

$\mathcal{A}$	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{d}$
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

$\Longrightarrow$

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

Figure 1.12: Coloring of attribute value graphs and the reduced table.



$\Longrightarrow$

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

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

Let us consider an example of decision table presented in Figure 1.11 and (a reduced form) of its discernibility matrix (Figure 1.11). From the Boolean function $f_{\mathcal{A}}$ with Boolean variables of the form a_{v1}^{v2} one can find the shortest prime implicant: $\mathbf{a}_{a_2}^{a_1} \wedge \mathbf{a}_{a_3}^{a_2} \wedge \mathbf{a}_{a_4}^{a_1} \wedge \mathbf{a}_{a_4}^{a_3} \wedge \mathbf{b}_{b_4}^{b_1} \wedge \mathbf{b}_{b_4}^{b_2} \wedge \mathbf{b}_{b_3}^{b_2} \wedge \mathbf{b}_{b_3}^{b_1} \wedge \mathbf{b}_{b_5}^{b_3}$ which can be treated as graphs presented in the Figure 1.12. We can color vertices of those graphs as it is shown in Figure 1.12. The colors are corresponding to the partitions: $P_{\mathbf{a}}(a_1) = P_{\mathbf{a}}(a_3) = 1$; $P_{\mathbf{a}}(a_2) = P_{\mathbf{a}}(a_4) = 2$; $P_{\mathbf{b}}(b_1) = P_{\mathbf{b}}(b_2) = P_{\mathbf{b}}(b_5) = 1$; $P_{\mathbf{b}}(b_3) = P_{\mathbf{b}}(b_4) = 2$. At the same time one can construct the new decision table (Figure 1.12).

Feature Extraction by Discretization and Symbolic Value Grouping.

One can extend the presented approach (see e.g. [243]) to the case when in a given decision system nominal and numeric attribute appear. The received heuristics are of very good quality.

Experiments for classification methods (see [243]) have been carried over decision systems using two techniques called *"train-and-test"* and *"n-fold-cross-validation"*. In Table 1.12 some results of experiments obtained by testing the proposed methods MD (using only discretization based on MD-heurisctic using Johnson approximation strategy) and MD-G (using discretization and symbolic value grouping) for classification quality on well known data tables from the "UC Irvine repository" are shown. The results reported in [96] are summarized in columns labeled

Table 1.12: The quality comparison between decision tree methods. MD: MD-heuristics; MD-G: MD-heuristics with symbolic value partition

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

by S-ID3 and C4.5 in Table 1.12). It is interesting to compare those results with regard both to the classification quality. Let us note that the heuristics MD and MD-G are also very efficient with respect to the time complexity.

In case of real value attributes one can search for features in the feature set containing the characteristic functions of half-spaced determined by hyperplanes or parts of spaces defined by more complex surfaces in multidimensional spaces. In [246], [238], [244] genetic algorithms have been applied in searching for semi-optimal hyperplanes or second order surfaces. The reported results are showing substantial increase in the quality of classification of unseen objects but we pay for that spending more time in searching for the semi-optimal hyperplanes.

In all of these cases one can use a general "board game" determined by the corresponding discernibility matrix in searching for optimal, in a sense, features and apply the following general scheme. For each entry of the discernibility matrix for discernible objects x and y one should consider the set of all formulas (from a considered language of features) discerning these objects. From the discernibility matrix the Boolean function(s) is (are) constructed, in a standard way [409], with the following property: the prime implicants of these functions determine the problem solutions. Using this general scheme one can invent much easier efficient heuristics searching for semi-prime implicants, and hence semi-optimal solutions, because they can be extracted by manipulation on Boolean formulas with a simple structure. The experimental results are supporting this claim (see e.g. [242], [243]). One of the possible strategy in searching for semi-optimal solutions is to search for short prime implicants because using *the minimum description length principle*, one can expect that from them the decision algorithms with high quality of unseen object classification can be built.

Boolean reasoning can also be used as a tool to measure the complexity of approximate solution of a given problem. As a complexity measure of a given problem one can consider the complexity of the corresponding to that problem Boolean function (represented by the number of variables, number of clauses, etc.).

1.7.3 Decision Rule Synthesis

The reader has certainly realized that the reducts (of all the various types) can be used to synthesize *minimal* decision rules. Once the reducts have been computed, the rules are easily constructed by overlaying the reducts over the originating decision table and reading off the values.

Example 1.7.2 Given the reduct $\{\mathit{Diploma}, \mathit{Experience}\}$ in the Tab. 1.4, the rule read off the first object is "if *Diploma* is MBA and *Experience* is Medium then *Decision* is Accept". □

We shall make these notions precise.

Let $\mathcal{A} = (U, A \cup \{d\})$ be a decision system and let $V = \bigcup\{V_a \,|\, a \in A\} \cup V_d$. Atomic formulae over $B \subseteq A \cup \{d\}$ and V are expressions of the form $a = v$; they are called *descriptors* over B and V, where $a \in B$ and $v \in V_a$. The set $\mathcal{F}(B, V)$ of formulae over B and V is the least set containing all atomic formulae over B and V and closed with respect to the propositional connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation).

Let $\varphi \in \mathcal{F}(B, V)$. $\| \varphi_{\mathcal{A}} \|$ denotes the meaning of φ in the decision table $\mathcal{A}$ which is the set of all objects in U with the property φ. These sets are defined as follows:

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

The set $\mathcal{F}(B, V)$ is called the set of *conditional formulae of* $\mathcal{A}$ and is denoted $\mathcal{C}(B, V)$.

A *decision rule* for $\mathcal{A}$ is any expression of the form $\varphi \Rightarrow d = v$, where $\varphi \in \mathcal{C}(B, V)$, $v \in V_d$ and $\| \varphi_{\mathcal{A}} \| \neq \emptyset$. Formulae φ and $d = v$ are referred to as the *predecessor* and the *successor* of decision rule $\varphi \Rightarrow d = v$.

Decision rule $\varphi \Rightarrow d = v$ is *true* in $\mathcal{A}$ if, and only if, $\| \varphi_{\mathcal{A}} \| \subseteq \| d = v_{\mathcal{A}} \|$; $\| \varphi_{\mathcal{A}} \|$ is the set of objects *matching* the decision rule; $\| \varphi_{\mathcal{A}} \| \cap \| d = v_{\mathcal{A}} \|$ is the set of objects *supporting* the rule.

Example 1.7.3 Looking again at Tab. 1.4, some of the rules are, for example:

$$\begin{array}{lll}
\mathit{Diploma} = \text{MBA} & \wedge\, \mathit{Experience} = \text{Medium} & \Rightarrow \mathit{Decision} = \text{Accept} \\
\mathit{Experience} = \text{Low} & \wedge\, \mathit{Reference} = \text{Good} & \Rightarrow \mathit{Decision} = \text{Reject} \\
\mathit{Diploma} = \text{MSc} & \wedge\, \mathit{Experience} = \text{Medium} & \Rightarrow \mathit{Decision} = \text{Accept}
\end{array}$$

The first two rules are true in Tab. 1.4 while the third one is not true in that table. □

Let us assume that our decision table is consistent. One can observe that by computing (k, d)–relative reducts for $x_k \in U$ it is possible to obtain the decision rules with minimal number of descriptors on their left hand sides among rules true in $\mathcal{A}$. It is enough for any such prime implicant to create the left hand side of the rule as follows: construct a conjunction of all descriptors $a = v$ where a is in prime implicant and v is the value of a on x_k.

Several numerical factors can be associated with a synthesized rule. For example, the support of a decision rule is the number of objects that match the predecessor of the rule. Various frequency-related numerical quantities may be computed from such counts like the *accuracy coefficient* equal to

$$\frac{|\| \varphi_{\mathcal{A}} \| \cap \| d = v_{\mathcal{A}} \||}{|\| \varphi_{\mathcal{A}} \||}$$

(see e.g. [109], [210], [14]).

The main challenge in inducing rules from decision tables lies in determining which attributes should be included in the conditional part of the rule. Although we can compute minimal decision rules, this approach results in rules that may contain noise or other peculiarities of the data set. Such detailed rules will be over-fit and will poorly classify unseen cases. More general, i.e. shorter rules should be rather synthesized which are not perfect on known cases (influenced by noise) but can be of high quality on new cases. Several strategies implementing this idea have been implemented. They are based on different measures like boundary region thinning (see e.g. [549], [403]), preserving up to a given threshold the positive region (see e.g. [403]), entropy (see [420], [417]). One can also use reduct approximations, i.e. attribute subsets that in a sense "almost" preserve e.g. the indiscernibility relation. One way of computing approximations is first to compute reducts for some random subsets of the universe of a given decision system and next to select the most stable reducts, i.e. reducts that occur in most of the subsystems. These reducts, called *dynamic reducts*, are usually inconsistent for the original table, but the rules synthesized from them are more tolerant to noise and other abnormalities; they perform better on unseen cases since they cover more general patterns in the data [14], [19]. Another approach is related to searching for patterns almost included in the decision classes combined with decomposition of decision tables into regular domains (see e.g. [240, 243, 250, 252, 253, 534]). One can also search for default rules. For a presentation of generating default rules see [218, 216, 217] and [136] who investigate synthesis of default rules or normalcy rules and some implementations of heuristics that search for such reducts.

One particularly successful method based on the re-sampling approach is called dynamic reducts. It is implemented in the Rosetta system [282].

For a systematic overview of rule synthesis see e.g. [444], [14], [109], [403].

1.7.4 Rule Application

When a set of rules have been induced from a decision table containing a set of training examples, they can be inspected to see if they reveal any novel relationships between attributes that are worth pursuing for further research. Furthermore, the rules can be applied to a set of unseen cases in order to estimate their classificatory power.

Several application schemes can be envisioned. Let us consider one of the simplest which has shown to be useful in practice.

1. When a rough set classifier is presented with a new case, the rule set is scanned to find applicable rules, i.e. rules whose predecessors match the case.

2. If no rule is found (i.e. no rule "fires"), the most frequent outcome in the training data is chosen.

3. If more than one rule fires, these may in turn indicate more than one possible outcome.

4. A voting process is then performed among the rules that fire in order to resolve conflicts and to rank the predicted outcomes. A rule casts as many votes in favor of its outcome as its associated support count. The votes from all the rules are then accumulated and divided by the total number of votes cast in order to arrive at a numerical measure of certainty for each outcome. This measure of certainty is not really a probability, but may be interpreted as an approximation to such, if the model is well calibrated.

For a systematic overview of rule application methods see e.g. [430], [14], [109, 110].

1.8 Rough Sets and Tolerance Relations

We discuss in this section extensions of rough sets based on tolerance relations but we would like to mention that many other generalizations have been studied like abstract approximation spaces [41], [207], [412], (see also Sect. 1.9); nondeterministic information systems (see e.g. [200], [291], [278], [350], [277]); recently developed extensions of rough set approach to deal with preferential ordering on attributes (criteria) in multi-criteria decision making [106], [104], [105]; an extension based on reflexive relations (as models for object closeness, only) [437]; extensions of rough set methods for incomplete information systems [176], [177]; formal languages approximations [146], [289], [290]; neighborhood systems [184], [185], [186]; extensions of rough sets for distributed systems and multi–agent systems (see e.g. [357], [358], [357], [343]). For discussion of other possible extensions see [343].

Tolerance relations provide an attractive and general tool for studying indiscernibility phenomena. The importance of those phenomena had been noticed by Poincare and Carnap. Studies have led to the emergence of such approaches to indiscernibility in rough set community.

We present only some examples of problems related to an extension of rough sets by using tolerance relations instead of equivalence relations as a model for indiscernibility. More details the reader can find e.g. in [30, 40, 41, 99, 124, 140, 151, 268, 269, 274, 275, 276, 166, 167, 168, 184, 191, 203, 253, 252, 257, 348, 344, 345, 349, 302, 350, 403, 408, 406, 412, 436, 437, 447, 448, 475, 502, 503, 535, 537, 539, 538, 541, 555, 556]. Let us also note that there are many interesting results on relationships between similarity and fuzzy sets (see e.g. [542, 72, 73, 75, 76, 77, 87, 88, 123, 385, 394, 395, 501]). Problems of similarity relations are also related to problems of clustering (see e.g. [63, 524]).

We call a relation $\tau \subseteq X \times U$ a *tolerance relation* on U if (i) τ is *reflexive:* $x\tau x$ for any $x \in U$ (ii) τ is *symmetric*: $x\tau y$ implies $y\tau x$ for any pair x, y of elements of U.
The pair (U, τ) is called a *tolerance space*. It leads to a metric space with the distance function

$$d_\tau(x, y) = \min\{k : \exists_{x_0, x_1, \ldots, x_k} x_0 = x \wedge x_k = y \wedge (x_i \tau x_{i+1} \text{ for } i = 0, 1, \ldots, k \Leftrightarrow 1)\}$$

Sets of the form $\tau(x) = \{y \in U : x\tau\, y\}$ are called *tolerance sets*.

One can easily generalize the definitions of the lower and upper approximations of sets by substituting tolerance classes for equivalence classes of the indiscernibility relation. We obtain the following formulae for the τ- approximations of a given subset X of the universe U:

$\underline{\tau}X = \{x \in U : \ \tau(x) \subseteq X\}$ and $\overline{\tau}X = \{x \in U : \ \tau(x) \cap X \neq \emptyset\}$.

However, one can observe that when we are dealing with tolerances we have a larger class of definable sets than in case of equivalence relations. For example one could take as primitive definable sets the tolerance classes of some iterations of tolerance relations or the equivalence classes of the relation defined from the tolerance relation τ by: $x IND_\tau y$ iff $dom_\tau(x) = dom_\tau(y)$ where $dom_\tau(x) = \cap\{\tau(z) : x \in \tau(z)\}$. Moreover, the presented above definition of the set approximations is not unique. For example in [41] approximations of sets have been defined which are more close in a sense to X than the classical ones. They can be defined as follows:

$$\tau_* X = \{x \in U : \ \exists y (x\tau y \& \tau(y) \subseteq X)\}$$

and

$$\tau^* X = \{x \in U : \ \forall y (x\tau y \Rightarrow \tau(y) \cap X \neq \emptyset)\}.$$

One can check that $\underline{\tau}X \subseteq \tau_* X \subseteq X \subseteq \tau^* X \subseteq \overline{\tau}X$

This approximations are closely related to the Brouwerian ortho–complementation (see [41]). One can take for any set X as its ortho-complementation the set $X^{\#} = \{x \in U : \forall h \in X(\neg(x\tau h))\} \subseteq X^c$ where $X^c = U \Leftrightarrow X$ and to find formulas (see [41]) expressing the new approximations using this kind of complementation. Let us observe that the condition $\neg(x\tau h)$ inside of the above formula can be interpreted as the discernibility condition for x, h.

Hence in the process of learning of the concept approximations we have more possibilities for approximation definition, approximation tuning and primitive definable sets choosing when we deal with tolerance relations than in case of equivalence relations. However, we pay for this because it is harder from computational point of view to search for relevant approximations in this larger space.

There has been made a great effort to study properties of logical systems based on similarity relations (see e.g. [542, 73, 75, 77, 88, 257, 268, 269, 151, 275, 276, 350, 394, 395, 448, 502, 503, 505, 538].

There is a great need for algorithmic tools suitable for relevant tolerance relation discovery from data, to tune the parameters of these relations or set approximations to obtain approximations of analyzed concepts of satisfactory quality. Recently, results in this direction have been reported (see e.g. [99, 253, 252, 436, 437, 167]) with promising experimental results for extracting patterns from data. Tolerance relations can be interpreted as graphs and several problems of searching for relevant patterns in data are strongly related to graph problems (see e.g. [253, 252]. These problems are NP-complete or NP-hard however several efficient heuristics have been developed to extract relevant patterns from data. Practitioners will look very much for logical systems helping to infer relevant tolerance relations and this is a challenge for logicians.

Let us mention some examples [253], [240].

One important aspect occurring in any algorithm constructing tolerance classes is a *homogeneous ratio* ρ of tolerance classes used for covering the training set. For a tolerance class $[x]_{\tau_x}$, the homogeneous ratio is defined by $\rho([x]_{\tau_x}) = \frac{card\{y\in[x]_{\tau_x}:d(y)=d(x)\}}{card\{[x]_{\tau_x}\}}$. The classifier is called *precise*, if its homogeneous ratio $\rho = 100\%$, and it is called *approximate*, if $\rho < 100\%$. The classifiers with high homogeneous ratio show usually a high classification accuracy but they show a low recognition ratio. In Table 1.13, we present the quality of classifiers obtained as the result of searching for the optimal relation among relations in the conjunctive form [240]. One can compare the recognition ratio of precise and approximate classifiers.

The quality of classifiers defined by semi-optimal relation in the conjunctive form [240] is presented in Table 1.14.

In [240] several other parameterized classes of similarity relations are distinguished and searching methods for semi-optimal relations are presented. In [240] it is reported that the semi-optimal relations extracted from the class of relations in the combination form show advantage for 5 data sets: Australian credits, Lymphography, Heart and Monk1 and Monk3 [231]. For the rest of data tables the conjunctive form appears to be better.

An interesting comparison of k-NN strategy [211, 210] with methods based searching for semi-optimal similarity relations is reported in [240]. Additionally two functions are used for distance measuring: HD (according to Hamming distance) and ED (according to Euclidean distance) [240]. The former function is applied to relations in the conjunctive form and the latter one is applied to relations in the combination form. A new object is classified using objects in the closest tolerance class.

In Table 1.15 the quality of conjunctive and combination form relations obtained using "closeness" strategy is presented and compared with the k-NN strategy. In these experiments the approximate classifiers are used.

Data	**Precise classifiers**		**Approx. classifiers**		
sets	**Acc.(%)**	**Rec.(%)**	ρ	**Acc.(%)**	**Rec.(%)**
Australian	87.47	65.07	0.80	85.25	91.15
Breast	77.30	74.86	0.70	75.20	91.24
Diabetes	75.70	70.10	0.65	74.47	82.81
Heart	83.70	80.33	0.75	82.10	90.5
Iris	96.67	95.50	0.90	96.67	100.00
Lympho	77.50	60.01	0.70	76.50	85.50
Soybean(S)	98.00	70.00	0.85	97.75	90.25
Tic-tac-toe	98.10	92.20	0.90	96.65	91.20
SatImage	87.65	70.30	0.75	85.25	87.00
Monk1	91.20	70.64	0.85	95.93	90.00
Monk2	87.00	75.00	0.80	85.00	82.33
Monk3	95.00	65.00	0.85	91.00	85.12

Table 1.13: The quality of matched tolerance class strategy. Classifiers are defined by conjunctive form relation (**Acc.**: accuracy of classification; **Rec.**: Recognition percent).

Data	**Precise classifiers**		**Approximate classifiers**		
sets	**Acc.(%)**	**Rec.(%)**	ρ	**Acc.(%)**	**Rec.(%)**
Australian	**88.47**	70.07	0.8	87.25	**91.15**
Breast	**76.20**	80.86	0.7	75.20	**90.20**
Diabetes	74.70	73.10	65	**75.20**	**85.50**
Heart	**84.70**	82.33	0.8	82.15	**87.5**
Iris	95.00	91.50	90	**95.70**	**100.00**
Lympho	**83.50**	75.01	0.75	82.50	**85.50**
Soybean(S)	95.00	70.00	0.85	**96.75**	**90.25**
Tic-tac-toe	95.10	85.20	0.90	**96.65**	**91.20**
SatImage	**85.65**	72.30	0.85	83.25	**85.00**
Monk1	**95.20**	74.64	0.8	94.93	**92.00**
Monk2	**85.00**	79.00	0.85	89.10	**87.33**
Monk3	94.00	72.00	0.85	**95.00**	**80.12**

Table 1.14: The quality of "matching" strategy. Classifiers are defined by conjunctive form relation.(**Acc.**: accuracy of classification; **Rec.**: Recognition percent)

Data sets	k-NN	Conjunctive form	Combination form
Australian	**86.00**	80.72	82.46
Breast	**74.30**	71.6	74.00
Diabetes	68.70	68.03	**74.30**
Heart	82.00	79.62	**83.40**
Iris	95.10	**96.67**	95.70
Lympho	77.00	81.01	**83.70**
Soybean(S)	**98.00**	97.53	93.11
Tic-tac-toe	98.10	**98.43**	97.00
SatImage	**90.10**	85.65	83.90

Table 1.15: The quality of the "closest" strategy. Classifiers are defined by conjunction and combination form relation.

Let us recall some previous observations related to concept approximations.

The lower and upper approximations are only examples of the possible approximations. In terminology of machine learning they are approximations of subsets of objects known from training sample. However, when one would like to deal with approximations of subsets of all objects (including also new i.e. unseen so far objects) some techniques have been proposed to construct set approximations suitable for such applications. The best known among them is the technique called the boundary region thinning related to the variable precision rough set approach [549]; another technique is used in tuning of decision rules. For instance, achieving better quality on new objects classification by introducing some degree of inconsistency on training objects. This technique is analogous to the well known techniques for decision tree pruning. The discussed approaches can be characterized in the following way: parameterized approximations of sets are defined and by tuning these parameters better approximations of sets or decision rules are obtained. Some of the above methods can be extended to tune concept approximations defined by tolerance relations. Further research in this direction will certainly lead to new interesting results.

One extension of rough set approach is based on recently developed rough mereology ([335, 337, 338, 339, 340, 337, 341, 342, 343]). The relations to be a part to a degree (discovered from data) are defining tolerance relations (defining so called rough inclusions) used to measure the closeness of approximated concepts. Tolerance relations play an important role in the process of schemes construction defining approximations of target concepts by some primitive ones. Contrary to classical approaches these schemes are "derived" from data by applying some algorithmic methods. The reader can look for more details in the section of the paper on rough mereological approach.

Tolerance relations can be defined from information systems or decision tables. Hence the reduction problems of information necessary to define tolerances relations arise (see e.g. [411, 412, 447, 448, 406]). We will briefly present an idea of this approach. By a *tolerance information system* [412] we understand a triple $\mathcal{A}' = (U, A, \tau)$ where $\mathcal{A}' = (U, A)$ is an information system and τ is a tolerance relation on *information vectors* $Inf_B(x) = \{(a, a(x)) : a \in B\}$ where $x \in U$, $B \subseteq A$. In particular, a tolerance information system can be realized as a pair $(\mathcal{A}, D)$ where $\mathcal{A} = (U, A)$ is an information system, while $D = (D_B)_{B \subseteq A}$ and $D_B \subseteq \mathit{INF}(B) \times \mathit{INF}(B)$ is a relation, called *the discernibility relation*, satisfying the following conditions:

(i) $\mathit{INF}(B) \times \mathit{INF}(B) \Leftrightarrow D_B$ is a tolerance relation;

(ii) $((u \Leftrightarrow v) \cup (v \Leftrightarrow u) \subseteq (u_0 \Leftrightarrow v_0) \cup (v_0 \Leftrightarrow u_0))$ & $u D_B v \to u_0 D_B v_0$ for any $u, v, u_0, v_0 \in \mathit{INF}(B)$ i.e. D_B is monotonic with respect to the discernibility property;

(iii) $\mathit{non}(u D_C v)$ implies $\mathit{non}(u|B\ D_B\ v|B)$ for any $B \subseteq C$ and $u, v \in \mathit{INF}(C)$

where $\mathit{INF}(B) = \{Inf_B(x) : x \in U\}$ and if $u \in \mathit{INF}(C)$ and $B \subseteq C \subseteq A$ then $u|B = \{(a, w) \in u : a \in B\}$ i.e. $u|B$ is the restriction of u to B. A (B, D_B)-tolerance τ_B is defined by

$$y \tau_B x \quad \text{iff} \quad \text{non}(Inf_B(x) D_B Inf_B(y)).$$

A (B, D_B)*-tolerance function* $I[B, D_B] : U \Leftrightarrow\rightarrow \mathcal{P}(U)$ is defined by $I[B, D_B](x) = \tau_B(x)$ for any $x \in U$.

The set $I[B, D_B](x)$ is called *the tolerance set of* x. The relation $\mathit{INF}(B) \times \mathit{INF}(B) \Leftrightarrow D_B$ expresses similarity of objects in terms of accessible information about them. The set $\mathit{RED}(\mathcal{A}, D)$ is defined by

$$\{B \subseteq A : I[A, D_A] = I[B, D_B] \quad \text{and} \quad I[A, D_A] \neq I[C, D_C] \quad \text{for any} \quad C \subset B\}$$

Elements of $RED(\mathcal{A}, D)$ are called *tolerance reducts of* $(\mathcal{A}, D)$ (or, *tolerance reducts*, in short). It follows from the definition that the tolerance reducts are minimal attribute sets preserving (A, D_A) - tolerance function. The tolerance reducts of $(\mathcal{A}, D)$ can be constructed in an analogous way as reducts of information systems. The problem of minimal tolerance reduct computing is NP-hard [412]. However again some efficient heuristics for computing semi-minimal reducts can be constructed. The method can be extended for computing so called relative tolerance reducts and other objects [448]. It is possible to apply Boolean reasoning to the object set reduction in tolerance information systems. This is based on the notion of an absorbent [475]. A subset $Y \subseteq X$ is an *absorbent* for a tolerance relation τ (τ-*absorbent*, in short) if and only if for each $x \in X$ there exists $y \in Y$ such that $x\tau y$. The problem of minimal absorbent construction for a given tolerance information system can be easily transformed to the problem of minimal prime implicant finding for a Boolean function corresponding to this system. Hence, again the problem of minimal absorbent construction is NP-hard so efficient heuristics have been constructed to find sub-minimal absorbents for tolerance information systems.

The presented methods of information reduction in tolerance information systems create some step towards practical applications. However, more research in this direction should still be done.

Further progress in investigations on tolerance information systems will have impact on applications of rough sets in many areas like granular computing, case based reasoning, process control, scaling continuous decisions etc.

We have discussed in this section some problems related to rough set approach based on tolerance approach. We have pointed out some interesting problems to be investigated.

1.9 Algebraic and Logical Aspects of Rough Sets

We have discussed in previous sections a representation scheme for a wide class of problems including problems from such areas like decision support [409], [297], machine learning, data mining [92], or conflict resolution in multi-agent systems [343]. On the basis of the representation scheme we construct (monotone) Boolean functions with the following property: their prime implicants [28](minimal valuations satisfying propositional formulas) are directly corresponding to the problem solutions (compare the George Boole idea from 1848 discussed, e.g., in [28]). In all these cases the implicants close to prime implicants define approximate solutions for considered problems (compare the discussion on Challenge_9 in [398]).

The general scheme for the problem encoding is the following.

First, a finite family $CONS$ of consistent sets is specified. It describe the set of considered(observed) *situations* (cases, objects).

Next, we specify a family $CONFLICT$ consisting pairs $\{u, u'\}$ of *conflicting situations* from $CONS$ i.e. sets whose union $u \cup u'$ is inconsistent with assumed axioms (in more advanced cases some degrees of inconsistency can be considered but this problem will be not discussed in the paper). The conflicting pairs correspond to pairs of situations which are *discernible.* The discernibility of two situations means (in the case discussed in the paper) that it is not true that the situations are *indiscernible.* The indiscernibility notion is a fundamental notion of rough set theory [297]. The reader can look for the roots of discussions on indiscernibility problems to the Leibnitz works. The discernibility has been discussed in many papers dedicated to different applications (compare e.g. [141], [409]). In this way we specify pairs $(CONS, CONFLICT)$, called *knowledge bases.* In the simplest cases considered in the paper they can be represented by data tables called *information systems* or *decision tables* [297].

Let us note that there are also involved in our considerations some simple axioms (which should be considered as non-logical axioms) describing e.g that in a given situation the attribute value is uniquely specified and that the logical sum of propositional variables corresponding to all its possible values is *true*.

Now we are ready to describe a general scheme for considered problems. Instances of our problems are defined by knowledge bases. For any considered problem we consider a family of transformations of knowledge bases into knowledge bases assuming also that a *quality measure* is defined on these transformations (with non-negative real values). For any given instance of the problem we are searching for *optimal* or *semi-optimal* transformations (with respect to a given measure) preserving the consistency of situations as well as all (or some) conflicting pairs.

The searching process for the optimal (semi-optimal) transformation for a given instance problem can be implemented as Boolean (propositional) reasoning applied to the so called *discernibility function* constructed from a problem description in the form of knowledge base. The discernibility function is responsible for preserving conflicting pairs. This function can be derived easily from so called *discernibility matrices* [409] (see also [141] where such matrices are used for solving problems in combinational digital circuits area). The prime implicants [520] of this function are explicitly defining the solutions for the problem [28]. Moreover, the implicants close to the prime implicants define approximate solutions for the problem.

The general encoding scheme presented above is a form of abstraction of our investigations in the last several years and presented earlier for some specific cases. It clarifies from the logical point of view the general nature of the reasoning strategies developed in the project.

We have presenting several examples of applications of this general scheme of reasoning based on propositional formulas. We show that the experimental results of propositional calculus on knowledge bases are very promising. We illustrate this point of view comparing the experimental results reported from the literature with those obtained using the discussed approach.

We will now present a short introduction related to other logical and algebraic aspects of rough sets.

One of the basic algebraic problem related to rough sets can be characterized as follows.

Let Σ be a class of information systems, , – a class of algebraic structures and e – a mapping form Σ into , . We say that Σ is *e-dense* in , if for any algebra Alg from , there exists and information system $\mathcal{A}$ in Σ such that Alg is isomorphic to $e(\mathcal{A})$ (or a sub-algebra of $e(\mathcal{A})$). If Σ is *e-dense* in , then we say that the representation theorem for Σ (relatively to e and ,) holds.

From this definition it follows that to formulate the representation theorem first one should choose the mapping e and the class , . They should be chosen as "natural" for the considered class of information systems. The mapping e endows the information systems with a natural algebraic structure. We will show some examples of natural algebraic structures for information systems to give the reader some flavor of the research going on. The reader interested in study of algebraic characterizations of rough sets should refer to [276, 275, 284, 80] and papers cited in these articles.

Let us recall that a definable set in an information system $\mathcal{A}$ is any union of discernibility classes of $IND(A)$. The first observation is that the set $DE(\mathcal{A})$ of all definable sets in $\mathcal{A}$ endowed with set theoretical operations: union, intersection and complementation forms a Boolean algebra with the empty set as 0 and the universe U as 1. The equivalence classes of the indiscernibility relation are the only atoms of this Boolean algebra. Let us note that definability of sets in incomplete information systems (i.e. attributes are partial functions on objects) has also been investigated [34].

For any information system $\mathcal{A} = (U, A)$ one can define the family $RS(\mathcal{A})$ of *rough sets* i.e.

pairs $(\underline{A}X, \overline{A}X)$ where $X \subseteq U$. Hence two questions arise. How to characterize the set of all rough sets in a given information system? What are the "natural" algebraic operations on rough sets?

To answer the first question let us assign to any rough set $(\underline{A}X, \overline{A}X)$ the pair $(\underline{A}X, BN_A X)$. One can easily see that the boundary region $BN_A X = \overline{A}X \Leftrightarrow \underline{A}X$ does not contain any singleton discernibility class (i.e. a class with one object only). Let us consider the set $\mathbf{Z}$ of all pairs (Y, Z) from $\mathcal{P}(U) \times \mathcal{P}(U)$ such that for some $X \subseteq U$ we have $Y = \underline{A}X$ and $Z = BN_A X$. One can observe that the set $\mathbf{Z}$ can be characterized as the set of all pairs of definable sets in $\mathcal{A}$ which are disjoint and the second element of any pair does not contain any singleton indiscernibility class of $IND(A)$.

From the point of view of algebraic operations one can choose another representation of rough sets. Let us recall that the lower approximation of a given set X is the set of all objects which can be with certainty classified as belonging to X on the basis of knowledge encoded in A and the set theoretical complement of the upper approximation of X is the set of all objects in U which can be with certainty rejected as belonging to X on the basis of knowledge encoded in A. Hence to any rough set $(\underline{A}X, \overline{A}X)$ in $\mathcal{A}$ one can assign a pair $(\underline{A}X, U \Leftrightarrow \overline{A}X)$. It happens that one can define some "natural" operations on such pairs of sets. First one easily will guess that $(\emptyset, U)$ corresponds to the smallest rough set and $(U, \emptyset)$ corresponds to the largest rough set. To define operations on such representations of rough sets let us imagine that we have two experts able to deliver answers about objects (observed through "glasses" of A) if they belong to some concepts i.e. subsets of U. Can we now define an approximate fusion of this concepts? There are several possibilities. We can treat as the lower approximation of the concept (representing concepts of two agents) the intersection of the lower approximations of two concepts using a rule: if both experts classify with certainty the observed object to their concepts we will treat this object as belonging with certainty to a concept being a fusion of those two. We will reject the observed object as belonging to the upper approximation of the concept being the fusion of two concepts if at least one of the experts will reject it with certainty as belonging to the corresponding concept. Hence we obtain the following definition of the algebraic operation on considered representations of rough sets: $(X_1, X_2) \wedge (Y_1, Y_2) = (X_1 \cap Y_1, X_2 \cup Y_2)$. The reader can immediately find interpretation for another operation: $(X_1, X_2) \vee (Y_1, Y_2) = (X_1 \cup Y_1, X_2 \cap Y_2)$. Let us consider one more example. How we can built a model for the complementation of a concept observed by an expert on the basis of his judgments? We again have several possibilities. The first model is the following: if the expert is classifying with certainty an observed object as belonging to a concept then we are rejecting it with certainty as belonging to the concept but if the expert is rejecting with certainty an observed object as belonging to a concept we are classifying it with certainty to the concept. Hence we have the following definition of one argument negation operation $\sim$: $\sim (X_1, X_2) = (X_2, X_1)$. However, now the reader will observe that there are some other possibilities to build a model for the complement of the concept to which the expert is referring e.g. by assuming $\neg(X_1, X_2) = (U \Leftrightarrow X_1, X_1)$ or $\div(X_1, X_2) = (X_2, U \Leftrightarrow X_2)$. The defined operations are not random operations. We are now very close (still the operation corresponding to implication should be defined properly!) to examples of known algebras, like Nelson or Heyting algebras, intensively studied in connection with different logical systems. The reader can find formal analysis of relationships of rough sets with Nelson, Heyting, Lukasiewicz, Post or double Stone algebras e.g. in [284] and in particular, the representation theorems for rough sets in different classes of algebras. Let us also note that the properties of defined negation operations are showing that they correspond to well known negations studied in logic: strong (constructive) negation or weak (intuitionistic) negation.

Algebraic structures relevant for construction of generalized approximation spaces have been also investigated e.g. in [41]. In [41] it is shown that the general structure of po-sets augmented with two sub-posets consisting of "inner definable" elements and "outer definable" elements is sufficient to define inner and outer approximation maps producing the best approximation from the bottom (lower approximation) and from the top (upper approximation) of any element with respect to the poset. By imposing De Morgan law it is received a duality between inner approximation space and outer approximation space. This class of De Morgan structures includes degenerate and quasi Brouwer-Zadeh posets, which are generalizations of topological spaces and preclusivity spaces, respectively. In the former case the approximable concepts are described as points of posets whereas in the later case the approximable concepts are described by subsets of a given universe. The classical, Pawlak approach coincides with the class of all clopen topologies or with the class of all preclusivity spaces induced from equivalence relations.

There is another research direction based on information systems [276]. The aim is to study information algebras and information logics corresponding to information systems. First, so called information frames are defined. They are relational structures consisting parameterized families of binary relations over the universe of objects. These relations are e.g indiscernibility relations corresponding to different subsets of attributes. Many other interesting frames can be found e.g. in [276]. If a frame is extended by adding e.g. set theoretical operations new algebraic structure called (concrete) information algebra is received. The information algebras in the abstract form are Boolean algebras augmented with some parameterized families of operations reflecting relevant properties of frames and in consequence of information systems. The main problems studied are related to the representation theorems for information algebras as well as to construction and properties of logical systems with semantics defined by information algebras [276].

An attempt to define rough algebras derived from rough equality is presented e.g. in [11].

For more readings on algebraic aspects of (generalized) approximation spaces the reader is referred to [107], [128], [365], [370], [371], [369], [129], [48], [504], [519], [40], [517].

There is a number of results on logics reflecting rough set aspects (for the bibliography see [345], [276]). Among these logics there are propositional as well as predicate logics. They have some new connectives (usually modal ones) reflecting different aspects of approximations. On semantical level they are allowing to express e.g. how the indiscernibility classes (or tolerance classes) interact with interpretations of formulas in a given model M. For example, in case of necessity connective the meaning $(\Box\alpha)_M$ of the formula α in the model M is the lower approximation of α_M, in case of possibility connective $(\langle\rangle\alpha)_M$ it is the upper approximation of α_M, i.e. the interpretation of α in M. Many other connectives have been introduced and logical systems with these connectives have been characterized. For example in predicate logic one can consider also rough quantifiers [170]. The results related to the completeness of axiomatization, decidability as well as expressibility of these logical systems are typical results. More information on rough logic the reader can find in [276], in particular in [12] a review of predicate rough logic is presented. Many results on information logics, in particular characterization theorems, can be found e.g. in [505].

Some relationships of rough algebras with many-valued logics have been shown e.g. in [11], [267]. For example in [11] soundness and completeness of 3-valued Łukasiewicz logic with respect to rough semantics has been proven. The rough semantics has been defined by rough algebras [11] (i.e. a special kind of topological quasi-Boolean algebra) [518]. Relationships of rough sets with 4-valued logic are presented in [267] and with quantum logic in [41].

We would like to mention several attempts to use rough set logics for reasoning about knowledge (see e.g. [372], [368], [373], [379], [380], [381], [304]).

Properties of dependencies in information systems have been studied by many researchers see e.g. [262], [263], [33], [83], [130], [264], [377], [378], [265], [515], [375], [376].

Finally we would like to mention a research direction related to so called rough mereological approach for approximate synthesis of objects satisfying a given specification to a satisfactory degree. We will discuss some aspects of this approach in Part II of this tutorial. Let us note here that one of the perspective for applied logic is to look for algorithmic methods of extracting logical structures from data e.g. relational structures corresponding to relevant feature extraction [413], default rules (approximate decision rules see e.g. [218]), connectives for uncertainty coefficients propagation and schemes of approximate reasoning. This is very much related to rough mereological approach and is crucial for many applications, in particular in knowledge discovery and data mining [92], calculi on information granules and computing with words [543], [544].

For more readings on logical aspects of rough sets the reader is referred to [276], [66], [12], [237], [80], [9], [284], [343], [505], [79], [150], [347], [311], [65], [195], [236], [339], [504],[235], [410], [502], [503], [367], [272], [273], [271], [365], [90], [364], [374], [89], [268], [270], [278].

1.10 Relationships with Other Approaches

Some interesting results on relationships of rough sets with other approaches to reasoning under uncertainty have been reported. In this section we point out on applications of rough sets in decision analysis, data mining and knowledge discovery, we present a comparison of some experimental results received by applying some machine learning techniques and rough set methods, we discuss some relationships of rough sets and fuzzy sets, we present some consequences of relationships of rough set approach with the Dempster-Shafer theory of evidence and finally we overview some hybrid methods and systems.

There have been studied also relationships of rough sets with other approaches e.g. with mathematical morphology (see e.g. [341, 330, 331, 332, 334]), statistical and probabilistic methods (see e.g. [293], [531], [449], [169], [322],[328], [540], [81], [82], [84]), concept analysis (see e.g. [101], [143], [283], [285]).

1.10.1 Decision analysis

Decision analysis is a discipline providing various tools for modeling decision situation in view of explaining them or prescribing actions increasing the coherence between the possibilities offered by the situation, and goals and value systems of the agents involved. Mathematical decision analysis consists in building a functional or a relational model. The functional model has been extensively used within the framework of multi–attribute utility theory. The relational is known e.g. in the form of an out ranking relation or a fuzzy relation (see [309], [386], [387], [388], [389], [10]).

Both modeling and explanation/prescription stages are also crucial operations et elaboration of a systematic and rational approach to modeling and solving complex decision problems [2], [309].

Rough set approach proved to be a useful tool for solving problems in decision analysis in particular in the analysis of multi–criteria decision problems related to:

(i) multi–criteria sorting problems;

(ii) multi–criteria, multi–sorting problems;

(iii) multi–criteria description of objects.

The case (i) can be described as decision problems related to the decision table with one decision. One can expect the following results from the rough set analysis of decision table: (i) evaluation of importance of particular attributes; (ii) construction of minimal subsets of independent attributes which can not be eliminated without disturbing the ability of approximating the sorting decisions; (iii) computing the relevant attributes i.e. core of the attribute set; (iv) elimination of redundant attributes from the decision table; (v) generation of sorting rules from the reduced decision table; they involve the relevant attributes only and explain a decision policy of the agent (decision maker or expert) in particular how to solve conflicts between decision rules voting for different decision when new objects are matched by these rules (see e.g. [430], see also 1.7.4). The multi–criteria sorting problems represents the largest class of decision problems to which the rough set approach has been successfully used. The applications concern many domains (see Sect. 1.11).

In the case (ii) we deal with decision tables with more than one decision (received from different agents). Using rough set methods one can measure the degree of consistency of agents, detect and explain discordant and concordant parts of agent's decision policies, evaluate the degree of conflict among the agents, and construct the preference models (sorting rules) expressed in common terms (conditional attributes) in order to facilitate a mutual understanding of the agents [309].

In the case (iii) the primary objective is to describe a decision situation. The rough set approach to the decision situation description is especially well suited when minimal descriptions in terms of attributes is of primary concern. Another important problem analyzed by rough set methods is conflict analysis [302]. If agents are not explicitly represented in the information system one can look for discovery of dependencies among conditional attributes interpreted as consequences of decisions represented by objects. Again, rough set methodology can be used to solve this type of problems [309].

For more readings on rough set approach to decision analysis see e.g. [430], [433], [434], [432], [300], [303], [308], [309].

Let us also note that some extensions of rough set approach have been proposed for dealing with preferential ordering of attributes (criteria) (see e.g. [104]).

1.10.2 Rough Sets and Data Mining

Rough set theory has proved to be useful in Data Mining and Knowledge Discovery. It constitutes a sound basis for data mining applications. The theory offers mathematical tools to discover hidden patterns in data. It identifies partial or total dependencies (i.e. cause–effect relations) in data bases, eliminates redundant data, gives approach to null values, missing data, dynamic data and others. The methods of data mining in very large data bases using rough sets have been developed.

There are some important steps in the synthesis of approximations of concepts related to the construction of (i) relevant primitive concepts from which approximations of more complex concepts will be constructed; (ii) (closeness) similarity measures between concepts; (iii) operations for construction of more complex concepts from primitive ones.

These problems can be solved by combining the classical rough set approach and recent extensions of rough set theory. Methods for solving problems arising in the realization of these steps are crucial for knowledge discovery and data mining (KDD) [92] as well.

There have been done in last years a substantial progress in developing rough set methods for data mining and knowledge discovery (see the cited in Sect. 1.11 cases and e.g. [243], [242], [252], [253], [240], [552], [47], [162], [163], [325], [327],[217], [344], [345], [244], [164], [534], [419], [444], [282], [191], [126], [67], [192], [125], [212], [149], [165], [248], [254], [255], [393], [246],

[247], [190]), [490], [251], [53], [161], [160], [250], [408], [418], [401], [415], [245], [488], [489], [550], [403], [551], [547]).

In particular new methods for extracting patterns from data (see e.g. [167], [253], [217], [240]), decomposition of decision tables (see e.g. [251], [418], [255], [253], [419]) as well as a new methodology for data mining in distributed and multi-agent systems (see e.g. [343] and Part II of this article) have been developed.

The reader can also consider the experimental results on data tables reported in this tutorial.

In Sect. 1.11 there are reported many successful case studies of data mining and knowledge discovery based on rough set methods and the reader can find there more references to papers on data mining.

Here, we consider an example showing some relationships between association rules and approximate reducts.

In some applications (see [403]), instead of reducts we prefer to use their approximations called α-reducts, where $\alpha \in [0,1]$ is a real parameter. The set of attributes $B \subset A$ is called α-reduct if

$$B \text{ is } \alpha\text{-reduct in } \mathcal{A} \quad \text{iff} \quad \frac{|\{c_{i,j} : B \cap c_{i,j} \neq \emptyset\}|}{|\{c_{i,j} : c_{i,j} \neq \emptyset\}|} \geq \alpha$$

One can show that for a given α, the problems of searching for shortest α-reducts and for all α-reducts are also NP-hard.

Given an information table $\mathcal{A} = (U, A)$. By *descriptors* we mean terms of the form $(a = v)$, where $a \in A$ is an attribute and $v \in V_a$ is a value in the domain of a (see [253]). The notion of descriptor can be generalized by using terms of form $(a \in S)$, where $S \subseteq V_a$ is a set of values. By *template* we mean the conjunction of descriptors, i.e. $\mathbf{T} = D_1 \wedge D_2 \wedge ... \wedge D_m$, where $D_1, ... D_m$ are either simple or generalized descriptors. We denote by $length(\mathbf{T})$ the number of descriptors being in $\mathbf{T}$. An object $u \in U$ is satisfying the template $\mathbf{T} = (a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m)$ if and only if $\forall_j a_{i_j}(u) = v_j$. Hence the template $\mathbf{T}$ describes the set of objects having the common property: "*the values of attributes $a_{j_1}, ..., a_{j_m}$ on these objects are equal to $v_1, ..., v_m$, respectively*". The *support* of $\mathbf{T}$ is defined by $support(\mathbf{T}) = |\{u \in U : u \text{ satisfies } \mathbf{T}\}|$. The long templates with large support are preferred in many Data Mining tasks.

Regarding on a concrete optimization function, problems of finding optimal large templates are known as being NP-hard with respect to the number of attributes involved into descriptors, or open problems (see e.g. [253]). Nevertheless, the large templates can be found quite efficiently by *Apriori* and *AprioriTid* algorithms (see [3, 545]). A number of other methods for large template generation has been proposed e.g. in [253].

Association rules and their generations can be defined in many ways (see [3]). Here, according to the presented notation, association rules can be defined as implications of the form $(\mathbf{P} \Rightarrow \mathbf{Q})$, where $\mathbf{P}$ and $\mathbf{Q}$ are different simple templates, i.e. formulas of the form

$$(a_{i_1} = v_{i_1}) \wedge \ldots \wedge (a_{i_k} = v_{i_k}) \Rightarrow (a_{j_1} = v_{j_1}) \wedge \ldots \wedge (a_{j_l} = v_{j_l}) \tag{1.3}$$

These implication can be called *generalized association rules*, because association rules are originally defined by formulas $\mathbf{P} \Rightarrow \mathbf{Q}$ where $\mathbf{P}$ and $\mathbf{Q}$ are the sets of items (i.e. goods or articles in stock market) e.g. $\{A, B\} \Rightarrow \{C, D, E\}$ (see [3]). One can see that this form can be obtained from 1.3 by replacing values on descriptors by 1 i.e.: $(A = 1) \wedge (B = 1) \Rightarrow (C = 1) \wedge (D = 1) \wedge (E = 1)$.

Usually, for a given information table $\mathcal{A}$, the quality of the association rule $\mathcal{R} = \mathbf{P} \Rightarrow \mathbf{Q}$ can be evaluated by two measures called *support and confidence* with respect to $\mathcal{A}$. The support of the rule $\mathcal{R}$ is defined by the number of objects from $\mathcal{A}$ satisfying the condition $(\mathbf{P} \wedge \mathbf{Q})$ i.e. $support(\mathcal{R}) = support(\mathbf{P} \wedge \mathbf{Q})$. The second measure – confidence of $\mathcal{R}$ – is the ratio between the support of $(\mathbf{P} \wedge \mathbf{Q})$ and the support of $\mathbf{P}$ i.e. $confidence(\mathcal{R}) = \frac{support(\mathbf{P}\wedge\mathbf{Q})}{support(\mathbf{P})}$. The following problem has been investigated by many authors (see e.g. [3, 545]): *For a given information table $\mathcal{A}$, an integer s, and a real number $c \in [0,1]$, find as many as possible association rules $\mathcal{R} = (\mathbf{P} \Rightarrow \mathbf{Q})$ such that $support(\mathcal{R}) \geq s$ and $confidence(\mathcal{R}) \geq c$.* All existing association rule generation methods consists of two main steps:

1. Generate as many as possible templates $\mathbf{T} = D_1 \wedge D_2 ... \wedge D_k$ such that $support(\mathbf{T}) \geq s$ and $support(\mathbf{T} \wedge D) < s$ for any descriptor D (i.e. maximal templates among those which are supported by more than s objects).

2. For any template $\mathbf{T}$, search for a partition $\mathbf{T} = \mathbf{P} \wedge \mathbf{Q}$ such that $support(\mathbf{P}) < \frac{support(\mathbf{T})}{c}$ and $\mathbf{P}$ is the smallest template satisfying this condition.

One can show ([249]) that the second steps can be solved using rough set methods and Boolean reasoning approach.

Let us assume that the template $\mathbf{T} = D_1 \wedge D_2 \wedge \ldots \wedge D_m$, which is supported by as least s objects, has been found. For a given confidence threshold $c \in (0;1)$ the decomposition $\mathbf{T} = \mathbf{P} \wedge \mathbf{Q}$ is called c-irreducible if $confidence(\mathbf{P} \Rightarrow \mathbf{Q}) \geq c$ and for any decomposition $\mathbf{T} = \mathbf{P}' \wedge \mathbf{Q}'$ such that $\mathbf{P}'$ is a sub-template of $\mathbf{P}$, $confidence(\mathbf{P}' \Rightarrow \mathbf{Q}') < c$.

One can prove that for any $c \in [0;1]$. The problem of searching for the shortest association rule from the template $\mathbf{T}$ for a given table $\mathcal{S}$ with the confidence limited by c (Optimal c-Association Rules Problem) is NP-hard.

For solving the presented problem, we show that the problem of searching for optimal association rules from the given template is equivalent to the problem of searching for local α-reducts for a decision table, which is a well known problem in Rough set theory. We construct the new decision table $\mathcal{A}|_{\mathbf{T}} = (U, A|_{\mathbf{T}} \cup d)$ from the original information table $\mathcal{A}$ and the template $\mathbf{T}$ as follows:

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

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

The following facts describe the relationship between the association rules problem and the reduct searching problem.
For a given information table $\mathcal{A} = (U, A)$, the template $\mathbf{T}$, the set of descriptors $\mathbf{P}$. The implication $\left(\bigwedge_{D_i \in \mathbf{P}} D_i \Rightarrow \bigwedge_{D_j \notin \mathbf{P}} D_j\right)$ is

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

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

Searching for minimal α-reducts is another well known problem in rough set theory. One can show, that the problem of searching for the all α-reducts as well as the problem of searching for shortest α-reducts is NP-hard. Great effort has been done to solve those problems.

The following example illustrates the main idea of our method. Let us consider the information table $\mathcal{A}$ with 18 objects and 9 attributes 1.16. Assume that the template $\mathbf{T} = (a_1 = 0) \wedge (a_3 = 2) \wedge (a_4 = 1) \wedge (a_6 = 0) \wedge (a_8 = 1)$ has been extracted from the information table $\mathcal{A}$. One can see that $support(\mathbf{T}) = 10$ and $length(\mathbf{T}) = 5$. The new constructed decision table $\mathcal{A}|_{\mathbf{T}}$ is presented in Table 1.16. The discernibility function for $\mathcal{A}|_{\mathbf{T}}$ can be described as follows

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

After its simplification we obtain six reducts: $f(D_1, D_2, D_3, D_4, D_5) = (D_3 \wedge D_5) \vee (D_4 \wedge D_5) \vee (D_1 \wedge D_2 \wedge D_3) \vee (D_1 \wedge D_2 \wedge D_4) \vee (D_1 \wedge D_2 \wedge D_5) \vee (D_1 \wedge D_3 \wedge D_4)$ for the decision table $\mathcal{A}|_{\mathbf{T}}$. Thus, we have found from $\mathbf{T}$ six association rules with (100%)-confidence. For $c = 90\%$, we would like to find α-reducts for the decision table $\mathcal{A}|_{\mathbf{T}}$, where $\alpha = 1 \Leftrightarrow \frac{\frac{1}{c} - 1}{\frac{n}{s} - 1} = 0.86$. Hence we would like to search for a set of descriptors that covers at least $\lceil (n \Leftrightarrow s)(\alpha) \rceil = \lceil 8 \cdot 0.86 \rceil = 7$ elements of discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$. One can see that the following sets of descriptors: $\{D_1, D_2\}$, $\{D_1, D_3\}$, $\{D_1, D_4\}$, $\{D_1, D_5\}$, $\{D_2, D_3\}$, $\{D_2, D_5\}$, $\{D_3, D_4\}$ have nonempty intersection with exactly 7 members of the discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$. In Table 1.17 we present all association rules achieved from those sets.

1.10.3 Comparison with Some Results in Machine Learning

Recently several comparison studies have been reported showing that the results received by using software systems based on rough set methods are fully comparable with those obtained by using other systems (see e.g. [109], [116], [110], [14], [243], [444], [163]) for object classifying. The reader can find some comparison of rough set methods with other systems in Tables 1.18, 1.19, 1.20.

Let us also recall methods recently reported in [243]. Table 1.12 presents the results of experiments obtained by using the methods based on rough sets and Boolean reasoning and the methods reported in [96]. One can compare those results with regard to the classification qualities. MD and MD-G heuristics are developed using rough set methods and Boolean reasoning.

The reader can also have a look on the other experimental results on data tables reported in this tutorial.

Several papers are comparing the results received by applying statistical methods and comparing them with the results received by rough set methods (see e.g. [238], [35], [508]).

In the future more research should be done to recognize proper areas for application of hybrid methods based on rough sets and statistical methods.

1.10.4 Rough Sets and Fuzzy Sets

Rough set theory and fuzzy set theory are complementary. It is natural to combine the two models of uncertainty (vagueness for fuzzy sets and coarseness for rough sets) in order to get a more accurate account of imperfect information [72]. The results concerning relationships between rough sets and fuzzy sets are presented e.g. in [72], [77], [23], [310], [37], [38], [39], [42], [43], [44], [187], [189], [209], [295], [299], [301], [516], [314], [351], [352], [396], [191], [536].

Rough set methods provide approximate descriptions of concepts and they can be used to construct approximate description of fuzzy concepts as well. This is very important for more compressed representation of concepts, rules, patterns in KDD because using fuzzy concepts

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

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

Table 1.16: An example of information table $\mathcal{A}$ and template $\mathbf{T}$ supported by 10 objects and the new decision table $\mathcal{A}|_{\mathbf{T}}$ constructed from $\mathcal{A}$ and template $\mathbf{T}$.

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

$=_{100\%} \Longrightarrow$

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

$=_{90\%} \Longrightarrow$

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

Table 1.17: The simplified version of discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$ and association rules.

Paradigm	System	Error rate		
		Monk 1	Monk 2	Monk 3
Neural Nets	Backpropagation	**0.000**	**0.000**	0.070
	Cascade Correlation	**0.000**	**0.000**	**0.030**
Decision Trees	Assistant Professional	**0.000**	0.190	**0.000**
	ID3	0.010	0.320	0.060
	ID3(no windowing)	0.170	0.310	0.040
	ID5R	0.180	0.310	0.050
	ID5R-hat	0.100	0.340	-
	IDL	0.030	0.430	-
	TDIDT	0.240	0.330	-
	PRISM	0.140	0.270	0.100
Relation Learning	mFOIL	**0.000**	0.310	**0.000**
Decision Rules	AQ14-NT	**0.000**	0.230	**0.000**
	AQR	0.040	0.200	0.130
	CN2	**0.000**	0.310	0.110
	AQ15	**0.000**	0.230	0.160
	AQ15-GA	**0.000**	0.130	0.000
	AQ17-DCI	**0.000**	**0.000**	**0.030**
	AQ17-FCLS	**0.000**	0.070	**0.030**
	AQ17-HCI	**0.000**	**0.000**	**0.000**
Rough Set	RSES-lib	**0.000**	**0.030**	**0.032**

Table 1.18: A comparison results for Monk's problems

System	Error rate		
	Lymphography	Breast cancer	Primary tumor
Assistant Professional	0.240	**0.220**	**0.560**
C4.5	0.230	-	0.600
AQ15	0.180-0.200	0.320-0.340	0.590-0.710
CN2	0.180	0.320-0.340	0.550
Naive LERS	0.380	0.490	0.790
New LERS	0.190	0.300	0.670
RSES-lib	**0.150**	**0.228**	**0.567**

Table 1.19: A comparison results for medical data

Paradigm	System	Error rate		
		Diabetes	Australian credit	Sat Image
Classical Statistical Methods	Discrim	**0.225**	0.141	0.171
	Quadisc	0.262	0.207	0.155
	Logdisc	**0.223**	0.141	0.163
Modern Statistical Techniques	SMART	**0.232**	0.158	0.159
	ALLOC80	0.301	0.201	**0.132**
	k-NN	0.324	0.181	**0.094**
	CASTLE	0.258	0.148	0.194
	NaiveBay	0.262	0.151	0.287
Neural Nets	Backpropagation	**0.248**	0.154	0.139
	Kohonen	0.273	-	0.179
	LVQ	0.272	0.197	**0.105**
	DIPOL92	**0.224**	0.141	**0.111**
	RBF	0.243	0.145	**0.121**
Decision Trees	CART	**0.255**	0.145	0.138
	IndCART	0.271	0.152	0.138
	C4.5	0.270	0.155	0.150
	NewID	0.289	0.181	0.150
	AC^2	0.276	0.181	0.157
	ITrule	**0.245**	0.137	-
	Cal5	**0.250**	0.131	0.151
	BayTree	0.271	0.171	0.147
Decision Rules	CN2	0.289	0.204	0.150
Rough Set	RSES-lib	**0.255**	**0.130**	**0.132**

Table 1.20: A comparison results for the StatLog's data

one can describe these items in a more compact way. Moreover, these descriptions can be more suitable for communication with human being.

In rough set theory approximations of sets are defined relatively to a given background knowledge represented by data tables (information systems, decision tables) with the set of attributes A.

The rough membership function μ_X^B where $X \subseteq U$ and $B \subseteq A$ can be used to define approximations and the boundary region of a set, as shown below:

$$\begin{aligned} \underline{B}(X) &= \{x \in U : \mu_X^B(x) = 1\}, \\ \overline{B}(X) &= \{x \in U : \mu_X^B(x) > 0\}, \\ BN_B(X) &= \{x \in U : 0 < \mu_X^B(x) < 1\}. \end{aligned}$$

The rough membership function has the following properties [310]:

a) $\mu_X^B(x) = 1$ iff $x \in \underline{B}(X)$,

b) $\mu_X^B(x) = 0$ iff $x \in \Leftrightarrow\overline{B}(X)$,

c) $0 < \mu_X^B(x) < 1$ iff $x \in BN_B(X)$,

d) If $IND(B) = \{(x,x) : x \in U\}$, then $\mu_X^B(x)$ is the characteristic function of X,

e) If $xIND(B)y$, then $\mu_X^B(x) = \mu_X^B(y)$,

f) $\mu_{U-X}^B(x) = 1 \Leftrightarrow \mu_X^B(x)$ for any $x \in U$,

g) $\mu_{X \cup Y}(x) \geq \max(\mu_X^B(x), \mu_Y^B(x))$ for any $x \in U$,

h) $\mu_{X \cap Y}^B(x) \leq \min(\mu_X^B(x), \mu_Y^B(x))$ for any $x \in U$,

i) If $\mathbf{X}$ is a family of pairwise disjoint sets of U, then $\mu_{\cup \mathbf{X}}^B(x) = \sum_{X \in \mathbf{X}} \mu_X^B(x)$ for any $x \in U$,

The above properties show clearly the difference between fuzzy and rough memberships. In particular properties g) and h) show that the rough membership can be regarded formally as a generalization of fuzzy membership. Let us observe that the "rough membership", in contrast to the "fuzzy membership", has probabilistic flavor.

It has been shown [310] that the formulae received from inequalities g) and h) by changing them into equalities are not true in general. This important observation is a simple consequence of the properties of set approximation: the calculus of set approximations in rough set theory is intensional. Namely, it is impossible to find a function independent from a given background knowledge that will allow to compute the values of the rough membership function for the intersection of sets (or union of sets) having only the values of the membership function for the argument sets. This property is showing some more differences between rough membership functions and fuzzy membership functions.

Rough set and fuzzy set approaches create many possibilities for hybridization. The number of reported results in this direction is continuously increasing. Let us mention some of them.

Combining rough sets and fuzzy sets allows to obtain rough approximations of fuzzy sets as well as approximations of sets by means of fuzzy similarity relations [DP4]. Let us consider the second case. Fuzzy rough sets (see e.g. [77]) are defined by membership function on the universe of objects U by

$$\mu_{\overline{S}(X)}(x) = sup_{\omega \in X} \mu_S(x, \omega)$$

$$\mu_{\underline{S}(X)}(x) = inf_{\omega \notin X}(1 \Leftrightarrow \mu_S(x,\omega))$$

where S is a fuzzy indistinguishability relation (fuzzy similarity relation) [77] and $x \in U$.

In this case we consider the fuzzy similarity relations instead of the (crisp) indiscernibility relations. In case of crisp indiscernibility (i.e. equivalence relation) relation we obtain

$$\mu_{\overline{S}(X)}(x) = 1 \text{ iff } x \in \overline{S}(X);$$

$$\mu_{\underline{S}(X)}(x) = 1 \text{ iff } x \in \underline{S}(X);$$

where $\overline{S}(X)$ and $\underline{S}(X)$ denote the upper and the lower approximations of X with respect to the indiscernibility relation S.

There are other interesting relationships of rough set with fuzzy sets (see e.g. [70], [71], [77]). For example, Ruspini's entailment can be explained in terms of rough deduction. In the rough set approach the indistinguishability notion is basic, while in Ruspini's fuzzy logic, it is the idea of closeness. It has been shown [72] that by introducing lower and upper approximations of fuzzy sets we come close to Caianiello's C-calculus [72]. Fuzzy rough sets are allowing to put together fuzzy sets and modal logic (see e.g. graded extensions of S5 system by Nakamura [77]). Rough aspects of fuzzy sets are also discussed in [13].

Rough–fuzzy hybridization methods give a tool for solving problems in KDD. In the sequel we will describe examples some tools and some research problems related to this topic.

The classical rough set approach is based on crisp sets. A generalization of rough set approach for handling different types of uncertainty has been proposed e.g. in [431]. It has been observed that the synthesized (extracted) features e.g. cuts [254], hyperplanes [238] can be tuned into more relevant features for classification when they are substituted by fuzzy cuts and fuzzy hyperplanes, respectively. This is related to the following property: points which are close to a cut or to a hyperplane can be hardly classified to half spaces corresponding to this cut or hyperplane because of e.g. noise influencing the position of points. The same idea can be extended to decision rules or pattern description [253]. Further investigations of techniques transforming crisp concepts (features) into fuzzy ones will certainly show more interesting results.

Let us mention another source for rough–fuzzy hybridization. These approaches can be characterized in the following way: parameterized approximations of sets are defined and by tuning these parameters approximations of fuzzy sets are received. Recently proposed shadowed sets for fuzzy sets [313] use this technique. Fuzzy membership function is represented by a family of parameterized functions with the same domain but only with three possible values. These functions correspond to the parameterized lower, upper and boundary region by a threshold determining the size of shadowed region. The size of this region can be tuned up in the process of learning.

One of the main problems in soft computing is to find methods allowing to measure the closeness of concept extensions. Rough set methods can also be used to measure the closeness of (fuzzy) concepts.

In classical rough set approach sets are represented by definable sets, i.e. unions of indiscernibility classes. Extensions of this approach have been proposed by several researchers (for references see e.g. [166], [253]). Instead of taking an equivalence relations as the indiscernibility relations the tolerance relation (or even more arbitrary binary relation [106]) is considered. This leads to a richer family of definable sets but it is harder (from computational complexity point of view) to construct concept approximations of high quality. Searching problems for optimal tolerance relations are NP-complete or NP-hard [253]. However, it has been possible to develop efficient heuristics searching for relevant tolerance relation(s) that allow to extract interesting patterns in data (see e.g. [166], [253]). The reported results are promising. A successful realization of this approach is possible because in the rough set approach relevant tolerance relations determining patterns can be extracted from the background knowledge represented in the form of data tables. The extracted patterns can be further fuzzyfied and applied in construction of concept approximation [240].

Rough set methods can be used to define fuzzy concepts approximately. In this case one should look for relevant α-cuts of the fuzzy set and treating these cuts as decision classes to find their approximations with respect to known conditional features. Once again problem of choosing relevant cuts is analogous to the problem of relevant feature extraction. From computational complexity point of view it is a hard problem and can be solved approximately by discovery of learning strategies. One can observe that the relevant cuts should be "well" approximated (i.e. new objects with high chance should be properly classified to them) as well as they should give together "good" approximation of the target fuzzy set.

The most general case is related to methods for approximate description of fuzzy concepts by fuzzy concepts. One can look at this issue as the searching problem for an approximate calculus on approximate concepts (information granules) [543], [544]. This calculus should allow to construct approximate descriptions of fuzzy concepts from approximate descriptions of known ones. One possible approach to solve this problem is to use fuzzy set methods based on t-norms and co-norms to define closeness of fuzzy concepts and to perform fusion of fuzzy concepts [69]. In practical applications there is a need to develop methods returning approximations of target concepts satisfying to a satisfactory degree given specifications (constraints) from approximations of some primitive (known) concepts. An approach to solve this problem has been recently proposed as rough mereology (see e.g. [339], [407], [343]). In this approach rules for propagation of uncertainty coefficients have to be learned form the available background knowledge represented by data tables. Another interesting property of this approach is that the construction of schemes for approximate description of the target concepts should be stable. This means that "small" changes in approximation quality of primitive concepts should give sufficiently "small" changes in approximation quality of constructed approximation of the target concepts. In [253] there are mentioned possible applications of this approach e.g. to the decomposition of large data tables.

Rough set approach combined with rough mereology can be treated as an inference engine for computing with words and granular computing [543], [544]. For example, the construction of satisfactory target fuzzy concept approximations from approximations of the input (primitive) fuzzy concepts can be realized in the following stages:

- first the fuzzy primitive (input) and the target (output) concept are represented by relevant families of cuts;
- next by using rough set methods the appropriate approximations of cuts are constructed in terms of available (conditions) measurable features (attributes) related to concepts;
- the approximations of input cuts obtained in stage 2 are used to construct schemes defining to a satisfactory degree the approximations of output cuts from approximated input cuts (and other sources of background knowledge) [339], [407], [343];
- the constructed family of schemes represents satisfactory approximation of the target concept by the input concepts; (in this step more compact descriptions of the constructed family of schemes can be created, if needed).

Progress in this direction seems to be crucial for further developments in soft computing and KDD.

1.10.5 Rough Sets and the Theory of Evidence

We only present one example of applications for the decision rule synthesis implied by the relationships between rough set methods and Dempster-Shafer's theory of evidence [400]. More details on the relationships between rough sets and Dempster-Shafer's theory the reader can find in [404]. In particular an interpretation of the Dempster-Shafer rule of combination by a simple operation on decision tables can be found in [404]. Some other aspects of relationships of rough sets and evidence theory are discussed in [530], [529], [537], [194], [193], [145].

In [404] it has been shown that one can compute a basic probability assignment (bpa) $m_{\mathcal{A}}$ for any decision table $\mathcal{A}$ assuming

$$m_{\mathcal{A}}(\emptyset) = 0 \quad \text{and} \quad m_{\mathcal{A}}(\theta) = \frac{|\{x \in U : \partial_A(x) = \theta\}|}{|U|}$$

where $\emptyset \neq \theta \subseteq \Theta_{\mathcal{A}} = \{i : d(x) = i \text{ for some } x \in U\}$.
Hence some relationships between belief functions $Bel_{\mathcal{A}}$ and $Pl_{\mathcal{A}}$ related to the decision table $\mathcal{A}$ can be shown [404]:

$$Bel_{\mathcal{A}}(\theta) = \frac{|\underline{A}\bigcup_{i\in\theta} X_i|}{|U|} \quad \text{and} \quad Pl_{\mathcal{A}}(\theta) = \frac{|\overline{A}\bigcup_{i\in\theta} X_i|}{|U|}$$

for any $\theta \subseteq \Theta_{\mathcal{A}}$.

The belief functions related to decision tables can be applied to generate strong approximate decision rules. One of the possible approach is to search for solutions of the following problem:

APPROXIMATION PROBLEM (AP)
INPUT: A decision table $\mathcal{A} = (U, A \cup \{d\})$, k – positive integer
and rational numbers $\varepsilon, tr \in (0, 1]$.
OUTPUT: Minimal (with respect to the inclusion) sets $B \subseteq A$ and $\theta \subseteq \Theta_{\mathcal{A}}$ of
cardinality at most k satisfying the following conditions:
(i) $|Pl_{\mathcal{A}|B}(\theta) \Leftrightarrow Bel_{\mathcal{A}|B}(\theta)| < \varepsilon$
(ii) $Bel_{\mathcal{A}|B}(\theta) > tr$.
where $\mathcal{A}|B = (U, B \cup \{d\})$.

The above conditions (i) and (ii) are equivalent to

$$|\overline{B}\bigcup_{i\in\theta} X_i \Leftrightarrow \underline{B}\bigcup_{i\in\theta} X_i| < \varepsilon|U| \quad \text{and} \quad |\underline{B}\bigcup_{i\in\theta} X_i| > tr|U|$$

respectively. Hence we are looking for "small" sets B and θ such that the boundary region (with respect to B) corresponding to $\cup_{i\in\theta} X_i$ is "small" (less than $\varepsilon|U|$) and the lower approximation of $\cup_{i\in\theta} X_i$ is "sufficiently large" (greater than $tr|U|$) so one can expect that the rules for this region will be strong. The solution for the above problem can be obtained by developing efficient heuristics.

1.10.6 Hybrid Methods and Systems

It is an experience of soft computing community that hybrid systems combining different soft computing techniques into one system can often improve the quality of the constructed system. This has also been claimed in case of rough set methods that combined with neural networks, genetic algorithms, statistical inference tools or Petri nets may give better solutions. A number of papers on hybrid systems showing the results which bear out this claim have been published. To be specific: adding statistical tools can improve the quality of decision rules induced by rough set methods (see e.g. [29]). Rough set based data reduction can be very useful in preprocessing of data input to neural networks. Several other methods for hybridization of rough sets and neural networks have been developed (see e.g. in [133], [134], [135], [462], [213], [465], [466], [471]). Decision algorithms synthesized by rough set methods can be used in designing neural networks (see e.g. [254], [256], [421], [469], [470], [471]). Rough set ideas can lead to new models of neurons (see e.g. [197], [198], [199], [471]). Optimization heuristics based on evolutionary programming can efficiently generate rough set constructs like reducts, patterns in data, decision rules (see e.g. [533], [255], [534]). Rough set methods can be useful in specifying concurrent systems from which corresponding Petri nets can be automatically generated (see e.g. [414], [415], [453]). Rough sets combined with fuzzy sets and Petri nets give an efficient method for designing clock information systems (see e.g. [318], [319]). Rough set approach to mathematical morphology leads to a generalization called analytical morphology ([336], [405], [341]), mathematical morphology of rough sets ([330], [331], [331], [334]) as well as to an idea of approximate compression of data ([333], [334]).

Moreover hybridization of rough set methods with classical methods like principal component analysis, Bayesian methods, 2D FFT (see e.g. [458], [457], [460], [461]) or wavelets (see e.g. [525]) leads to classifiers of better quality .

1.11 Applications and Case Studies

There have been developed different software systems based on rough set methods (see e.g. [345], [482] and Sect. 1.12). There are numerous areas of successful applications of rough set software systems. Many interesting case studies are reported. Let us mention some of them:

- **MEDICINE:**
  - Treatment of duodenal ulcer by HSV ([307], [94], [95], [435], [429]);
  - Analysis of data from peritoneal lavage in acute pancreatitis ([422], [423]);
  - Supporting of therapeutic decisions ([445]);
  - Knowledge acquisition in nursing ([31], [526], [110]);
  - Diagnosis of pneumonia patients ([320]);
  - Medical databases (e.g. headache, meningitis, CVD) analysis ([485], [487], [488], [500], [490], [491], [492], [493], [494], [498], [495], [496], [476], [477], [478], [479], [480], [481]);
  - Image analysis for medical applications ([222], [133], [135];)
  - Surgical wound infection ([139]);
  - Classification of histological pictures ([133]);
  - Preterm birth prediction ([527], [117], [528], [111], [112], [110]);

- Medical decision–making on board space station Freedom (NASA Johnson Space Center) ([110]);
- Verification of indications for treatment of urinary stones by extra–corporeal shock wave lithotripsy (ESWL) ([426]);
- Analysis of factors affecting the occurrence of breast cancer among women treated in US military facilities (reported by W. Ziarko);
- Analysis of factors affecting the differential diagnosis between viral and bacterial meningitis ([499], [552]);
- Developing an emergency room for diagnostic check list – A case study of appendicitis ([391]);
- Analysis of medical experience with urolithiasis patients treated by extra-corporeal shock wave lithotripsy ([424]);
- Diagnosing in progressive encephalopathy ([288], [512], [511]);
- Automatic detection of speech disorders ([62]);
- Rough set-based filtration of sound applicable to hearing prostheses ([58];
- Classification of tooth surfaces ([161], the EUFIT'96 competition);
- Discovery of attribute dependencies in experience with multiple injured patients ([427]);
- Modeling cardiac patient set residuals ([279]);
- Multistage analysis of therapeutic experience with acute pancreatitis ([425]);
- Breast cancer detection using electro-potentials ([461]);
- Analysis of medical data of patients with suspected acute appendicitis ([35]);
- Attribute reduction in a database for hepatic diseases ([473]);
- EEG signal analysis ([525]);

- **ECONOMICS, FINANCE AND BUSINESS:**
  - Evaluation of bankruptcy risk ([439], [438], [105]);
  - Company evaluation ([225]);
  - Bank credit policy ([225]);
  - Prediction of behavior of credit card holders ([522]);
  - Drafting and advertising budget of a company ([225]);
  - Customer behavior patterns ([327], [552]);
  - Response modeling in database marketing ([508]);
  - Analysis of factors affecting customer's income level ([552] also reported by Tu Bao Ho);
  - Analysis of factors affecting stock price fluctuation ([103]);
  - Discovery of strong predictive rules for stock market ([553], [18]);
  - Purchase prediction in database marketing ([507]);
  - Modeling customer retention ([165], [85]);
  - Temporal patterns ([160]);

  - Analysis of business databases ([323], [164], [325]);
  - Rupture prediction in a highly automated production system ([465]);

- **ENVIRONMENTAL CASES:**

  - Analysis of a large multi-species toxicity database ([142]);
  - Drawing premonitory factors for earthquakes by emphasizing gas geochemistry ([474]);
  - Control conditions on a polder ([384]);
  - Environmental protection ([108], [110]);
  - Global warming: influence of different variables on the earth global temperature ([113]);
  - Global temperature stability ([118], [110]);
  - Programming water supply systems ([6], [390], [106]);
  - Predicting water demands in Regina ([7], [6]);
  - Prediction of oceanic upwelling off the Mauretanian cost using sea surface temperature images, and real and model meteorological data (reported by I. Duentsch);
  - Prediction of slope-failure danger level from cases ([100]);

- **ENGINEERING:**

  - **Control:** The design and implementation of rough and rough–fuzzy controllers ([220], [554], [548], [221], [472], [223], [226], [329], [228], [229], [306], [230]), [509], [317], [224]);
  - **Signal and image analysis:**
    - ∗ Noise and distortion reduction in digital audio signal ([53], [54], [55], [58]) [56]);
    - ∗ Filtration and coding of audio ([60]);
    - ∗ Audio signal enhancement ([61]);
    - ∗ Recognition of musical sounds ([155]);
    - ∗ Detection and interpolation of impulsive distortions in old audio recordings ([51], [59]);
    - ∗ Subjective assessment of sound quality ([153], [154]);
    - ∗ Assessment of concert hall acoustics ([156]);
    - ∗ Classification of musical timbres and phrases ([152], [157], [158];
    - ∗ Mining musical databases ([159]);
    - ∗ Image analysis ([222], [133], [135]);
    - ∗ Converting a continuous tone image into a half-tone image using error diffusion and rough set methods ([468]);
    - ∗ Texture analysis ([256], [456], [458], [467], [460], [463]);
    - ∗ Voice recognition ([20], [57], [52], [62], [56]);
    - ∗ Classification of animal voices ([161], EUFIT'96 competition);
    - ∗ Handwritten digit recognition ([454], [16], [122], [17], [459]);
    - ∗ Optical character recognition ([50]);
    - ∗ Circuits synthesis and fault detection ([214], [215]);

    - Vibro-acoustic technical diagnostics ([258], [259], [260], [261], [201]);
    - Intelligent scheduling ([138]);
  - **Others:**
    - Preliminary wind-bracing in steel skeleton structure ([8]);
    - Technical diagnostics of mechanical objects ([446]);
    - Decision supporting for highly automated production system ([466]);
    - Estimation of important highway parameters ([197], [198]);
    - Time series analysis of highway traffic volumes ([197], [198]);
    - Real–time decision making ([416], [316]);
    - Material analysis ([131], [324]);
    - Power system security analysis ([178]);

- **INFORMATION SCIENCE:**
  - **Software engineering:**
    - Qualitative analysis of software engineering data ([392]);
    - Assessing software quality ([318]);
    - Software deployability ([315]);
    - Knowledge discovery form software engineering data([393]);
  - Information retrieval ([440], [441], [442], [99]);
  - Data mining from musical databases ([159]);
  - Analysis and synthesis of concurrent systems ([414], [415], [450], [451], [453]);
  - Integration RDMS and data mining tools using rough sets ([98],[255], [250]);
  - Rough set model of relational databases ([382], [21], [22], [190], [192]);
  - Cooperative knowledge base systems ([357], [358], [359], [360]);
  - Natural language processing ([115], [219], [116], [114], [110]);
  - Cooperative information system re–engineering ([450], [452]);

- **DECISION ANALYSIS:** (see cases and applications in this section);

- **SOCIAL SCIENCES, OTHERS:**
  - Conflict analysis ([294], [298], [302], [64]);
  - Social choice functions ([97], [266]);
  - Rough sets in librarian science ([399]);
  - Rough sets–based study of voter preference ([119]);
  - Analysis of test profile performance ([361]);
  - On–line prediction of volleyball game progress ([455], [464]);

- **MOLECULAR BIOLOGY:** Discovery of functional components of proteins from amino–acid sequences ([489], [497]);

- **CHEMISTRY: PHARMACY** Analysis of relationship between structure and activity of substances ([171], [172], [173], [174], [175]);

1.12 Software Systems

We enclose the list of software systems based on rough sets. The reader can find more details in [345]. It was possible to identify the following rough set systems for data analysis:

- Datalogic/R, http://ourworld.compuserve.com/homepages/reduct
- Grobian (Roughian),
  e-mail: I.Duentsch@ulst.ac.uk, ggediga@luce.psycho.Uni-Osnabrueck.DE
- KDD-R: Rough Sets-Based Data Mining System, e-mail: ziarko@cs.uregina.ca
- LERS—A Knowledge Discovery System , e-mail: jerzy@eecs.ukans.edu
- PRIMEROSE, e-mail: tsumoto@computer.org
- ProbRough — A System for Probabilistic Rough Classifiers Generation,
  e-mail: {zpiasta,lenarcik}@sabat.tu.kielce.pl
- Rosetta Software System, http://www.idi.ntnu.no/˜aleks/rosetta/
- Rough Family - Software Implementation of the Rough Set Theory,
  e-mail: Roman.Slowinski@cs.put.poznan.pl, Jerzy.Stefanowski@cs.put.poznan.pl
- RSDM: Rough Sets Data Miner, e-mail: {cfbaizan, emenasalvas}@.fi.upm.es
- RoughFuzzyLab - a System for Data Mining and Rough and Fuzzy Sets Based Classification, e-mail: rswiniar@saturn.sdsu.edu
- RSL – The Rough Set Library, ftp://ftp.ii.pw.edu.pl/pub/Rough/
- TAS: Tools for Analysis and Synthesis of Concurrent Processes using Rough Set Methods,
  e-mail: zsuraj@univ.rzeszow.pl
- Trance: a Tool for Rough Data Analysis, Classification, and Clustering,
  e-mail:wojtek@cs.vu.nl

Chapter 2

Rough Mereology: Approximate Synthesis of Objects

We would like to give here a brief account of the rough mereological approach to approximate reasoning [335, 337, 338, 339, 340, 347, 407, 342, 343]. We propose this formalization as a tool for solving multi–agent or distributed applications related to approximate reasoning and to calculi on information granules [544], [543].

2.1 Rough Mereology

Rough mereology offers the general formalism for the treatment of partial containment. Rough mereology can be regarded as a far - reaching generalization of mereology of Leśniewski [183]: it does replace the relation of being a (proper) part with a hierarchy of relations of being a part in a degree. The basic notion is the notion of a rough inclusion.

A real function $\mu(X, Y)$ on a universe of objects U with values in the interval $[0, 1]$ is called *a rough inclusion* when it satisfies the following conditions:

(A) $\mu(x, x) = 1$ for any x (meaning normalization);

(B) $\mu(x, y) = 1$ implies that $\mu(z, y) \geq \mu(z, x)$ for any triple x, y, z (meaning monotonicity);

(C) $\mu(x, y) = 1$ and $\mu(y, x) = 1$ imply $\mu(x, z) \geq \mu(y, z)$ for any triple x, y, z (meaning monotonicity);

(D) there is n such that $\mu(n, x) = 1$ for any x. An object n satisfying (D) is a μ-*null object.*

We let $x =_\mu y$ iff $\mu(x, y) = 1 = \mu(y, x)$ and $x \neq_\mu y$ iff $non(x =_\mu y)$.

We introduce other conditions for rough inclusion:

(E) **if** objects x, y have the property :

if $z \neq_\mu n$ and $\mu(z, x) = 1$

then there is $t \neq_\mu n$ with $\mu(t, z) = 1 = \mu(t, y)$

then it follows that: $\mu(x, y) = 1$.

(E) is an inference rule: it is applied to infer the relation of being a part from the relation of being a subpart.

(F) For any collection , of objects there is an object x with the properties:

(i) if $z \neq_{\mu} n$ and $\mu(z,x) = 1$ then there are $t \neq_{\mu} n$, $w \in$, such that

$$\mu(t,z) = \mu(t,w) = \mu(w,x) = 1;$$

(ii) if $w \in$, then $\mu(w,x) = 1$;

(iii) if y satisfies the above two conditions then $\mu(x,y) = 1$.

Any x satisfying F(i) is called a *set of objects* in , ; if, in addition, x satisfies F(ii,iii), then x is called the *class of objects* in , . These notions allow for representations of collections of objects as objects.

We interpret the formula: $\mu(x,y) = r$ as the statement: *x is a part of y in degree* at least r.

The formula $x = class(\mu_U)\{x_1, x_2, ..., x_k\}$ is interpreted as the statement that the object x is composed (designed, synthesized) from parts $x_1, x_2, ..., x_k$. In mereology of Leśniewski the notions of a part, an element, and a subset are all equivalent: one can thus interpret the formula $\mu(x,y) = r$ as the statement: x is an element (a subset) of y in degree r; if $y = class(\mu)$, , then $\mu(x,y) = r$ means that x is a member of the collection , to a degree r.
A standard choice of an appropriate measure can be based on the frequency count; the formal rendering of this idea is the *standard rough inclusion* function defined for two sets $X, Y \subseteq U$ by the formula

$$\mu(X,Y) = \frac{card(X \cap Y)}{card(X)}$$

when X is non-empty, 1 otherwise. This function satisfies all of the above axioms for rough inclusion.

Relations to fuzzy containment:

Fuzzy containment may be defined in a fuzzy universe U endowed with fuzzy membership functions μ_X, μ_Y by the formula :

$$\sigma(X \subseteq Y) = \inf_Z \{I(\mu_X(Z), \mu_Y(Z))\}$$

for a many - valued implication I. We quote a result in [339] which shows that rough inclusions generate a class of fuzzy containments stable under residual implications of the form $\overset{\Leftrightarrow}{\top}$ where $\top$ is a continuous t-norm [74] viz.: for any rough inclusion μ on U , the function

$$\sigma(X,Y) = \inf_Z \{\overset{\Leftrightarrow}{\top}(\mu(Z,X), \mu(Z,Y))\}.$$

is also a rough inclusion. The impact of this is that in models of rough mereology which implement σ as the model rough inclusion, we have the composition rule of the form:

$$\textbf{if } \sigma(x,y,r) \text{ and } \sigma(y,z,s) \textbf{ then } \sigma(x,z,\top(r,s)).$$

Hence we can develop an associative calculus of partial containment.

	hat	ker	pig
x_1	1	0	0
x_2	0	0	1
x_3	0	1	0
x_4	1	1	0

Table 2.1: The information system H

	x$_1$	x$_2$	x$_3$	x$_4$
x$_1$	1	0.33	0.33	0.66
x$_2$	0.33	1	0.33	0.00
x$_3$	0.33	0.33	1	0.66
x$_4$	0.66	0.00	0.66	1

Table 2.2: Initial rough inclusion for H

2.1.1 Rough Inclusions from Information Systems

Rough inclusions can be generated from a given information system $\mathcal{A}$; for instance, for a given partition $P = \{A_1, \ldots, A_k\}$ of the set A of attributes into non-empty sets $A_1, \ldots, A_k$, and a given set $W = \{w_1, \ldots, w_k\}$ of weights, $w_i \in [0, 1]$ for $i = 1, 2, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$ we let

$$\mu_{o,P,W}(x, y) = \sum_{i=1}^{k} w_i \cdot \frac{\|IND_i(x, y)\|}{\|A_i\|}$$

where $IND_i(x, y) = \{a \in A_i : a(x) = a(y)\}$. We call $\mu_{o,P,W}$ a *pre-rough inclusion.*

The function $\mu_{o,P,W}$ is rough-invariant i.e. if $a(x) = a(x')$ and $a(y) = a(y')$ for each $a \in A$ then $\mu_{o,P,W}(x, y) = \mu_{o,P,W}(x', y')$. $\mu_{o,P,W}$ can be extended to a rough inclusion on the set 2^U [339] e.g. via the formula: $\mu(X, Y) = \top\{\perp \{\mu_{o,P,W}(x, y) : y \in Y\} : x \in X\}$ where $\top$ is a t-norm and $\perp$ is a t-conorm.

An advantage of having rough inclusions in this form is that we can optimize weights w_i in the learning stage.

Example 2.1.1 Consider an information system H

The table below shows values of the initial rough inclusion $\mu_{o,P,W}(x, y) = \frac{card(IND(x,y))}{3}$ i.e. we consider the simplest case when $k = 1$, $w_1 = 1$. □

Example 2.1.2 In addition to the information system (agent) H from Example 1 we consider agents B and HB. Together with H they form the string $\mathbf{ag} = (H)(B)(HB)$ i.e. HB takes objects: x sent by H and y sent by B and assembles a complex object xy.

The values of the initial rough inclusion $\mu = \mu_{o,P,W}$ are calculated for (B) and (BH) by the same procedure as in Example 2.1.1. □

2.1.2 Approximate Mereological Connectives

An important ingredient in our scheme is related to the problem of rough mereological connectives: given information systems $\mathbf{A}, \mathbf{B}, \mathbf{C}, ...$ we will say that $\mathbf{A}$ *results from* $\mathbf{B}, \mathbf{C}, ...$ if there exists a (partial) mapping (an *operation*) $o_{\mathbf{A}} : U_{\mathbf{B}} \times U_{\mathbf{C}} \times ... \to U_{\mathbf{A}}$ with $rng\ o_A = U_{\mathbf{A}}$ i.e. any $x \in U_{\mathbf{A}}$ is of the form $o_{\mathbf{A}}(y, z, ...)$ where $y \in U_{\mathbf{B}}$ and $z \in U_{\mathbf{C}}, ...$. In the case when pre - rough inclusions $\mu_A, \mu_B, \mu_C, ...$ are defined in respective universes $U_{\mathbf{A}}, U_{\mathbf{B}}, U_{\mathbf{C}}, ...$ there arises

	pis	cut	kni	cr
y_1	1	1	1	1
y_2	1	0	0	0
y_3	0	0	1	0
y_4	1	1	1	0

Table 2.3: The information system B

	har	lar	off	tar
x_1y_1	1	0	1	0
x_1y_3	0	1	0	0
x_2y_1	1	0	0	0
x_2y_3	0	1	0	1

Table 2.4: The information system HB

the problem of *uncertainty propagation* i.e. we have to decide in what way is the measure μ_A related to measures $\mu_B, \mu_C, ...$. Formally, we have to find an (approximation) to a function f satisfying the following property:

```
for any $\epsilon_1, \epsilon_2, \epsilon_3, ... \in [0,1]$ :
for any $x, x_1 \in U_{\mathbf{B}}, y, y_1 \in U_{\mathbf{C}}, ...$ :
    if $\mu_{\mathbf{B}}(x, x_1) \geq \epsilon_1$ , $\mu_{\mathbf{C}}(y, y_1) \geq \epsilon_2$ and ...
    then $\mu_{\mathbf{A}}(o_A(x, y, ...), o_A(x_1, y_1, ...)) \geq f(\epsilon_1, \epsilon_2, ...)$
```

In practice, it is unrealistic to expect the existence of a function f satisfying the above condition globally; therefore, we localize this notion. To this end, let us select a subset $S_A \subseteq U_A$ of objects which we will call *standard objects*; we will use the symbol $xyz...$ to denote the object $o_A(x, y, ...) \in U_A$. Given a standard $s = xyz...$, we will call a function f an (approximation to) *rough mereological connective relative to* o_A and s in case it satisfies the condition:

```
for any $\epsilon_1, \epsilon_2, \epsilon_3, ... \in [0,1]$ :
for any $x_1 \in U_{\mathbf{B}}, y_1 \in U_{\mathbf{C}}, ....$ :
    if $\mu_{\mathbf{B}}(x_1, x) \geq \epsilon_1$ , $\mu_{\mathbf{C}}(y_1, y) \geq \epsilon_2$ and ...
    then $\mu_{\mathbf{A}}(o_A(x_1, y_1, ...), s) \geq f(\epsilon_1, \epsilon_2, ...)$
```

We outline an algorithm which may be used to extract from information systems approximations to uncertainty functions (rough mereological connectives).

Example 2.1.3 We will determine an approximation to the mereological connective at the standard x_1y_1 in Table 2.4 of Example 2.1.2 i.e. a function f such that (for simplicity of notation we omit subscripts of μ):

$$\begin{array}{rl} \textbf{if} & \mu(x, x_1) \geq \epsilon_1 \text{ and } \mu(y, y_1) \geq \epsilon_2 \\ \textbf{then} & \mu(xy, x_1y_1) \geq f(\epsilon_1, \epsilon_2), \text{ for any pair } \epsilon_1, \epsilon_2. \end{array}$$

The following tables show conditions which f is to fulfill.

□

x	$\mu(x, x_1)$	y	$\mu(y, y_1)$	$\mu(xy, x_1y_1)$
x_1	1	y_1	1	1
x_1	1	y_2	0.25	0.5
x_1	1	y_3	0.25	0.25
x_1	1	y_4	0.75	1
x_2	0.33	y_1	1	0.75
x_2	0.33	y_2	0.25	0.25
x_2	0.33	y_3	0.25	0.00
x_2	0.33	y_4	0.75	0.5

Table 2.5: The conditions for f (first part)

x	$\mu(x, x_1)$	y	$\mu(y, y_1)$	$\mu(xy, x_1y_1)$
x_3	0.33	y_1	1	0.75
x_3	0.33	y_2	0.25	0.25
x_3	0.33	y_3	0.25	0.25
x_3	0.33	y_4	0.75	0.75
x_4	0.66	y_1	1	1
x_4	0.66	y_2	0.25	0.5
x_4	0.66	y_3	0.25	0.25
x_4	0.66	y_4	0.75	1

Table 2.6: The conditions for f (second part)

This full set T_0 of conditions can be reduced: we can find a minimal set T of vectors of the form $(\varepsilon_1^{'}, \varepsilon_2^{'}, \varepsilon)$ such that if f satisfies the condition $f(\varepsilon_1^{'}, \varepsilon_2^{'}) = \varepsilon$ for each $(\varepsilon_1^{'}, \varepsilon_2^{'}, \varepsilon) \in T$ then f extends by the formula (2.1), below.

The following algorithm produces a minimal set T of conditions.

Algorithm

Input : table T_0 of vectors

$$(\mu(x, x_1), \mu(y, y_1), \mu(xy, x_1y_1));$$

Step 1. For each pair $(\mu(x, x_1) = \varepsilon_1, \mu(y, y_1) = \varepsilon_2)$, find $\varepsilon(\varepsilon_1, \varepsilon_2) = \min\{\varepsilon : \varepsilon_1' \geq \varepsilon_1, \varepsilon_2' \geq \varepsilon_2, (\varepsilon_1', \varepsilon_2', \varepsilon) \in T_0\}$. Let T_1 be the table of vectors $(\varepsilon_1, \varepsilon_2, \varepsilon(\varepsilon_1, \varepsilon_2))$.

Step 2. For each ε^* such that $(\varepsilon_1, \varepsilon_2, \varepsilon(\varepsilon_1, \varepsilon_2) = \varepsilon^*) \in T_1$, find: $row(\varepsilon^*) = (\varepsilon_1^*, \varepsilon_2^*, \varepsilon^*)$ where $(\varepsilon_1^*, \varepsilon_2^*, \varepsilon^*) \in T_1$ and if $(\varepsilon_1^{'}, \varepsilon_2^{'}, \varepsilon^*) \in T_1$ then $\varepsilon_1^{'} \geq \varepsilon_1^*, \varepsilon_2^{'} \geq \varepsilon_2^*$.

Output: table T of vectors of the form $row(\varepsilon)$.

One can check that Table 2.7 shows a minimal set T of vectors for the case of Tables 2.5, 2.6.

One can extract from the algorithm the synthesis formula of f from conditions T_0 :

$$f(\varepsilon_1, \varepsilon_2) = \min \quad \{\varepsilon^{'} : (\varepsilon_1^{'}, \varepsilon_2^{'}, \varepsilon^{'}) \in T_0 \wedge (\varepsilon_1^{'} \geq \varepsilon_1) \wedge (\varepsilon_2^{'} \geq \varepsilon_2)\} \tag{2.1}$$

ε_1	ε_2	ε
0.66	0.75	1
0.33	0.75	0.5
0.66	0.25	0.25
0.33	0.25	0.00

Table 2.7: A minimal set T of vectors

2.2 Reasoning in Multi–Agent Systems

We outline here basic ideas of reasoning under uncertainty by intelligent units (agents) in multi - agent systems. Schemes based on these ideas may be - in our opinion - applied in the following areas of application, important for development of automated techniques [5]:

- computer-aided manufacturing or computer-aided design [5], [49],[127] where a complex object=a final artifact (assembly) is produced (designed) from inventory (elementary) parts by a dedicated team of agents.
- logistics [78] where complex structures are organized from existing elementary structures (units) to perform a task according to a given specification.
- adaptive control of complex systems [202], [407] where the task consists in maintaining a given constraint (specification) by adaptive adjustment of behavior of some parts (organs, physiological processes etc.).
- business re-engineering [78], [443] where the task is to adaptively modify a complex object (structure, organization, resources, etc.) according to the current economic situation (specification).
- cooperative/distributed problem solving including planning, dynamic task assignment etc. [78], [86], [443] where the task is to organize a system of agents into a scheme of local teams for solving a problem (specification).
- automated fabrication [32] where the task is to build complex objects (e.g. mechanisms) by layer-after -layer synthesis.
- preliminary stage of design process [356] where the approximate reasoning about objects and processes is crucial as it is carried out in an informal, often natural, language.

The general scheme for approximate reasoning can be represented by the following tuple

$$Appr_Reas = (Ag, Link, U, St, Dec_Sch, O, Inv, Unc_mes, Unc_prop)$$

where

(i) The symbol Ag denotes the set of agent names.

(ii) The symbol $Link$ denotes a set of non-empty strings over the alphabet Ag; for $v(ag) = ag_1 ag_2 ... ag_k ag \in Link$, we say that $v(ag)$ defines an *elementary synthesis scheme* with the *root* ag and the *leaf agents* $ag_1, ag_2, ..., ag_k$. The intended meaning of $v(ag)$ is that the agents ag_1, ag_2,..,ag_k are the children of the agent ag which can send to ag some simpler constructs for assembling a more complex artifact. The relation $\leq$ defined via

$ag \leq ag'$ iff ag is a leaf agent in $v(ag')$ for some $v(ag')$, is usually assumed to be at least an ordering of Ag into a type of an acyclic graph; we assume for simplicity that $(Ag, \leq)$ is a tree with the root $root(Ag)$ and leaf agents in the set $Leaf(Ag)$.

(iii) The symbol U denotes the set $\{U(ag) : ag \in Ag\}$ of *universes* of agents.

(iv) The symbol St denotes the set $\{St(ag) : ag \in Ag\}$ where $St(ag) \subset U(ag)$ is the set of *standard objects* at the agent ag.

(v) The symbol O denotes the set $\{O(ag) : ag \in Ag\}$ of *operations* where $O(ag) = \{o_i(ag)\}$ is the set of *operations at ag*.

(vi) The symbol Dec_Sch denotes the set of *decomposition schemes*; a particular decomposition scheme dec_sch_j is a tuple

$$(\{st(ag)_j : ag \in Ag\}, \{o_j(ag) : ag \in Ag\})$$

which satisfies the property that if $v(ag) = ag_1 ag_2 ... ag_k ag \in Link$ then

$$o_j(ag)(st(ag_1)_j, st(ag_2)_j, .., st(ag_k)_j) = st(ag)_j$$

for each j.

The intended meaning of dec_sch_j is that when any child ag_i of ag submits the standard construct $st(ag_i)_j$ then the agent ag assembles from $st(ag_1)_j$, $st(ag_2)_j$, ..., $st(ag_k)_j$ the standard construct $st(ag)_j$ by means of the operation $o_j(ag)$.

The rule dec_sch_j establishes therefore a decomposition scheme of any standard construct at the agent *root* (Ag) into a set of consecutively simpler standards at all other agents. The standard constructs of leaf agents are *primitive (inventory) standards.* We can regard the set of decomposition schemes as a skeleton about which the approximate reasoning is organized. Any rule dec_sch_j conveys a certain knowledge that standard constructs are synthesized from specified simpler standard constructs by means of specified operations. This ideal knowledge is a reference point for real synthesis processes in which we deal as a rule with constructs which are not standard: in adaptive tasks, for instance, we process new, unseen yet, constructs (objects, signals).

(vii) The symbol Inv denotes the *inventory set* of primitive constructs. We have $Inv = \cup\{U(ag) : ag \in Leaf(Ag)\}$.

(viii) The symbol Unc_mes denotes the set $\{Unc\_mes(ag) : ag \in Ag\}$ of *uncertainty measures of agents*, where $Unc\_mes(ag) = \{\mu_j(ag)\}$ and $\mu_j(ag) \subseteq U(ag) \times U(ag) \times V(ag)$ is a relation (possibly function) which determines a distance between constructs in $U(ag)$ valued in a set $V(ag)$; usually, $V(ag) = [0, 1]$, the unit interval.

(ix) The symbol Unc_prop denotes the set of *uncertainty propagation rules* $\{Unc\_prop(v(ag)) : v(ag) \in Link\}$; for $v(ag) = ag_1 ag_2 ... ag_k ag \in Link$, the set $Unc\_prop(v(ag))$ consists of functions $f_j : V(ag_1) \times V(ag_2) \times ... \times V(ag_k) \Leftrightarrow\!\!\!\rightarrow V(ag)$ such that

$$\textbf{if } \mu_j(ag_i)(x_i, st(ag_i)_j) = \varepsilon_i \text{ for } i = 1, 2, .., k$$
$$\textbf{then } \mu_j(ag)(o_j(x_1, x_2, .., x_k), st(ag)_j) = \varepsilon \geq f_j(\varepsilon_1, \varepsilon_2, .., \varepsilon_k).$$

The functions f_j relate values of uncertainty measures at the children of ag and at ag.

This general scheme may be adapted to the particular cases.

As an example, we will interpret this scheme in the case of a fuzzy controller. In its version due to Mamdani [202] in its simplest form, we have two agents: *input, output*, and standards of agents are expressed in terms of linguistic labels like *positively small, negative, zero* etc. Operations of the agent *output* express the control rules of the controller e.g. the symbol $o(positively\ small, negative) = zero$ is equivalent to the control rule of the form if $st(input)_i$ is *positively small* and $st(input)_j$ is *negative* then $st(output)_k$ is *zero*. Uncertainty measures of agents are introduced as fuzzy membership functions corresponding to fuzzy sets representing standards i.e. linguistic labels. An input construct (signal) $x(input)$ is fuzzyfied i.e. its distances from input standards are calculated and then the fuzzy logic rules are applied. By means of these rules uncertainty propagating functions are defined which allow for calculating the distances of the output construct $x(output)$ from the output standards. On the basis of these distances the construct $x(output)$ is evaluated by the defuzzyfication procedure.

The process of synthesis by a scheme of agents of a complex object x which is an approximate solution to a requirement Φ consists in our approach of the two communication stages viz. the top - down communication/negotiation process and the bottom - up synthesis process. We outline the two stages here.

In the process of top - down communication, a requirement Φ received by the scheme from an external source is decomposed into approximate specifications of the form

$$(\Phi(ag), \varepsilon(ag))$$

for any agent ag of the scheme. The intended meaning of the approximate specification $(\Phi(ag), \varepsilon(ag))$ is that a construct $z \in U(ag)$ satisfies $(\Phi(ag), \varepsilon(ag))$ iff there exists a standard $st(ag)$ with the properties that $st(ag)$ satisfies the predicate $\Phi(ag)$ and

$$\mu(ag)(z, st(ag)) \geq \varepsilon(ag).$$

The uncertainty bounds of the form $\varepsilon(ag)$ are defined by the agents viz. the root agent *root(Ag)* chooses $\varepsilon(root(Ag))$ and $\Phi(root(Ag))$ as such that according to it any construct x satisfying $(\Phi(root(Ag), \varepsilon(root(Ag)))$ should satisfy the external requirement Φ in an acceptable degree. The choice of $(\Phi(root(Ag), \varepsilon(root(Ag)))$ can be based on the previous learning process; the other agents choose their approximate specifications in negotiations within each elementary scheme $v(ag) \in Link$. The result of the negotiations is successful when there exists a decomposition scheme dec_sch_j such that for any $v(ag) \in Link$, where $v(ag) = ag_1 ag_2 ... ag_k ag$, from the conditions $\mu(ag_i)(x_i, st(ag_i)_j) \geq \varepsilon(ag_i)$ and $st(ag_i)_j$ satisfies $\Phi(ag_i)$ for $i = 1, 2, .., k$, it follows that $\mu(ag)(o_j(x_1 x_2, .., x_k), st(ag)_j) \geq \varepsilon(ag)$ and st(ag)$_j$ satisfies $\Phi(ag)$.

The uncertainty bounds $\varepsilon(ag)$ are evaluated on the basis of uncertainty propagating functions whose approximations are extracted from information systems of agents.

The synthesis of a complex object x is initiated at the leaf agents: they select primitive constructs (objects) and calculate their distances from their respective standards; then, the selected constructs are sent to the parent nodes of leaf agents along with vectors of distance values. The parent nodes synthesize complex constructs from the sent primitives and calculate the new vectors of distances from their respective standards. Finally, the root agent $root(Ag)$ receives from its children the constructs from which it assembles the final construct and calculates the distances of this construct from the root standards. On the basis of the found values, the root agent classifies the final construct x with respect to the root standards as eventually satisfying $(\Phi(root(Ag), \varepsilon(root(Ag)))$.

Our approach is analytic: all logical components (uncertainty measures, uncertainty functions etc.) necessary for the synthesis process are extracted from the empirical knowledge of agents represented in their information systems; it is also intensional in the sense that rules for propagating uncertainty are local as they depend on a particular elementary synthesis scheme and on a particular local standard.

We will now give a more detailed account of the process of synthesis.

2.3 Synthesis Schemes

Synthesis agents.
We start with the set *Ag* of synthesis agents and the set *Inv* of inventory objects. Any synthesis agent *ag* has assigned a *label* $lab(ag) = \{U(ag), \mathcal{A}(ag), St(ag), L(ag), \mu_o(ag), F(ag)\}$ where: $U(ag)$ is the universe of objects at *ag*, $\mathcal{A}(ag) = \{U(ag), A(ag), V(ag)\}$ is the information system of *ag*, $St(ag) \subseteq U(ag)$ is the set of *standard objects* (*standards*) at *ag*, $L(ag)$ is a set of *unary predicates* at *ag* (specifying properties of objects in $U(ag)$). Predicates of $L(ag)$ are constructed as formulas in $C(A(ag), V)$ (i.e. Boolean combinations of descriptors over $A(ag)$ and V); $\mu_o(ag) \subseteq U(ag) \times U(ag) \times [0,1]$ is a pre-rough inclusion at *ag* generated from $\mathcal{A}(ag)$; $F(ag)$ is a set of functions at *ag* called *mereological connectives (*cf. below, the notion of a *(C,*Φ, ε)⇔scheme). Synthesis agents reason about objects by means of the approximate logic of synthesis.

Approximate logic of synthesis.
Consider a synthesis agent *ag*. The symbol b_{ag} will denote the variable which runs over objects in U_{ag}. A *valuation* v_X where X is a set of synthesis agents is a function which assigns to any b_{ag} for $ag \in X$ an element $v_X(b_{ag}) \in U_{ag}$. The symbol v^x_{ag} denotes $v_{\{ag\}}$ with $v_{\{ag\}}(b_{ag}) = x$.

We now define *approximate specifications* at *ag* as formulas of the form $\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ where $st(ag) \in St(ag), \Phi(ag) \in L(ag)$ and $\varepsilon(ag) \in [0,1]$. We say that $v = v_{\{ag\}}$ *satisfies a formula* $\alpha = \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$, symbolically $v \models \alpha$, in case $\mu(ag)(v(b_{ag}), st(ag)) \geq \varepsilon$ and $st(ag) \models \Phi(ag)$. We write $x \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ iff $v^x_{ag} \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$. The meaning of $\alpha = \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ is thus the set $[\alpha]_{ag}$ of objects x satisfactorily (as determined by $\varepsilon(ag)$) close to a standard (viz. $st(ag)$) satisfying $\Phi(ag)$. How the agents cooperate is determined by a chosen scheme; selection of a scheme is itself an adaptive process of design [407].

The synthesis language
Link. The synthesis agents are organized into a hierarchy (which may be an empty relation in case of autonomous agents system). We describe this hierarchy in a language *Link* over the alphabet *Ag*. The agents $ag_1, ag_2, .., ag_k, ag_0$ in *Ag* form the string $\mathbf{ag} = ag_1 ag_2 ... ag_k ag_0$ $\in$ *Link* if and only if there exist a mapping $\rho(\mathbf{ag})$:$U(ag_1) \times ... \times U(ag_k) \Leftrightarrow\rightarrow U(ag_0)$ (meaning: the agent ag$_0$ can assemble by means of $\rho(\mathbf{ag})$ the object $\rho(\mathbf{ag})$ $(x_1, ..., x_k) \in U(ag_0)$ from any tuple ($x_1 \in U(ag_1), ..., x_k \in U(ag_k)$).

Elementary constructions.
If $\mathbf{ag} = ag_1 ag_2 ... ag_k ag_0 \in$ *Link*, then the pair

$$c = (\mathbf{ag}, \{\langle st(ag_i), \Phi(ag_i), \varepsilon(ag_i)\rangle : i = 0, 1, .., k\})$$

will be called an *elementary construction.* We write: $Ag(c) = \{ag_0, ag_1, ..., ag_k\}$, $Root(c) = ag_0$, $Leaf(c) = Ag(c) \Leftrightarrow \{ag_0\}$.

Constructions.
For elementary constructions c, c' with $Ag(c) \cap Ag(c') = \{ag\}$ where $ag = Root(c) \in Leaf(c')$, we define *the ag-composition* $c \star_{ag} c'$ of c and c' with $Root(c \star_{ag} c') = Root(c')$, $Leaf(c \star_{ag} c') = (Leaf(c) \Leftrightarrow \{ag\}) \cup (Leaf(c')$, $Ag(c \star_{ag} c') = Ag(c) \cup Ag(c')$. A *construction* is any expression C obtained from a set of elementary constructions by applying the composition operation a finite number of times.

(C, Φ, ε)**-schemes.**
For an elementary construction $c = c(\mathbf{ag})$ as above, we define a $(c, \Phi, \varepsilon) \Leftrightarrow$*scheme* as

$$(\mathbf{ag}, \{\langle st(ag_i), \Phi(ag_i), \varepsilon(ag_i)\rangle : i = 0, 1, .., k\}$$

where $f(ag_0) \in F(ag_0)$ satisfies the condition:

---

if $\mu(ag_i)(x_i, st(ag_i)) \geq \varepsilon(ag_i)$ for $i = 1, 2, .., k$

then

$$\begin{aligned} \mu_o(ag)(x, st(ag_0)) &\geq (\varepsilon(ag_1), \varepsilon(ag_2), .., \varepsilon(ag_k)) \\ &\geq \varepsilon(ag_0). \end{aligned}$$

A construction C composed of elementary constructions $c_1, .., c_m$, c_o with $Root(C) = Root(c_o) = ag_o$ is the *support of a* $(C, \Phi, \varepsilon) \Leftrightarrow$*scheme* when each c_i is the support of a $(c_i, \Phi_i, \varepsilon_i)$-*scheme*, where $\Phi_i = \Phi(Root(c_i))$, $\varepsilon_i = \varepsilon(Root(c_i))$, $\Phi = \Phi(ag_0)$ and $\varepsilon = \varepsilon(ag_0)$. The (C, Φ, ε)-scheme $\mathbf{c}$ defines a function $F_{\mathbf{c}}$ called *the output function* of $\mathbf{c}$ given by $F_{\mathbf{c}}(v_{Leaf(C)}) = x$ where $x \in U(ag_0)$ is the unique object produced by C from $v_{Leaf(C)}$. The following statement expresses the sufficiency criterion of synthesis by a scheme of an object satisfying the approximate requirement $\langle st(ag_o), \Phi(ag_o), \varepsilon(ag_o)\rangle$.

Theorem 2.3.1 (*the sufficiency criterion of synthesis*). For any valuation v_X on the set X of leaf agents $ag(1), .. ., ag(m)$ of the $(C, \Phi, \varepsilon) \Leftrightarrow$scheme with $ag_o = Root(C)$ such that

$$v(b_{ag(i)}) \models \langle st(ag(i)), \Phi(ag(i)), \varepsilon(ag(i))\rangle$$

for $i = 1, 2, ..., m$, we have

$$F_{\mathbf{c}}(v_X) \models \langle st(ag_o), \Phi(ag_o), \varepsilon(ag_o)\rangle$$

■

Let us emphasize the fact that the functions $f(ag)$, called *mereological connectives* above, are expected to be extracted from experiments with samples of objects (see Example 3, above). The above property allows for an easy to justify correctness criterion of a given (C, Φ, ε)-scheme provided that all parameters in this scheme have been chosen properly. The searching process for these parameters and synthesis of an uncertainty propagation scheme satisfying the formulated conditions constitutes the main and not easy part of synthesis (and design as well).

2.4 Mereological Controllers

The approximate specification (Φ, ε) can be regarded as an invariant to be kept over the universe of global states (complex objects) of the distributed system. A mereological controller generalizes the notion of a fuzzy controller. The control problems can be divided into several classes depending on the model of controlled object. In this work we deal with the simplest case. In this case, the model of a controlled object is the (C, Φ, ε)-scheme $\mathbf{c}$ which can be treated as a model of the unperturbed by noise controlled object whose states are satisfying the approximate specification (Φ, ε).

We assume the leaf agents of the (C, Φ, ε)-scheme $\mathbf{c}$ are partitioned into two disjoint sets, namely the set $Un\_control(\mathbf{c})$ of *uncontrollable (noise) agents* and the set $Control(\mathbf{c})$ of controllable *agents.*

We present now two examples of a control problem for a given (C, Φ, ε) -scheme.

(OCP) OPTIMAL CONTROL PROBLEM:

Input: (C, Φ, ε)-scheme $\mathbf{c}$; information about actual valuation v of leaf agents i.e. the values $v(b_{ag})$ for any $ag \in Control(\mathbf{c})$ and a value ε' such that $F_{\mathbf{c}}(v) \models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon' \rangle$.

Output: A new valuation v' such that $v'(b_{ag}) = v(b_{ag})$ for $ag \in Un\_control(\mathbf{c})$ and $F_{\mathbf{c}}(v')$ $\models (st(ag_{\mathbf{c}}), \Phi, \varepsilon_0)$ where $\varepsilon_0 = \sup\{\delta : F_{\mathbf{c}}(w) \models \langle st(ag_{\mathbf{c}}), \Phi, \delta \rangle$ for some w such that $w(b_{ag}) = v(b_{ag})$ for $ag \in Un\_control(\mathbf{c})\}$.

These requirements can hardly be satisfied directly. A relaxation of (OCP) is

---

(CP) ∇-CONTROL PROBLEM

Input: (C, Φ, ε)-scheme $\mathbf{c}$; information about actual valuation v of leaf agents (i.e. the values $v(b_{ag})$ for any $ag \in Control(\mathbf{c})$) and a value ε' such that $F_{\mathbf{c}}(v) \models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon' \rangle$.

Output: A new valuation v' such that $v'(b_{ag}) = v(b_{ag})$ for $ag \in Un\_control(\mathbf{c})$ and $F_{\mathbf{c}}(v')$ $\models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon_0 \rangle$ where $\varepsilon_0 \rangle \varepsilon' + \nabla$ for some given threshold ∇.

We will now describe the basic idea on which our controllers of complex dynamic objects represented by distributed systems of intelligent agents are built. The main component of the controller are Δ - *incremental rules.*

Δ-*rules* have the form:

$$(\Delta(\mathbf{ag})) \qquad \begin{aligned} &(\Delta\varepsilon(ag_{i_1}), ..., \Delta\varepsilon(ag_{i_r})) \\ &= h(\varepsilon(ag), \Leftrightarrow \Delta\varepsilon(ag), \varepsilon(ag_1), ..., \varepsilon(ag_k)) \end{aligned}$$

where $ag_{i_1}, ..., ag_{i_r}$ are all controllable children of ag (i.e. children of ag having descendents in $Control(\mathbf{c})$), $h : R^{k+2} \to R^r$ and R is the set of reals.

Approximations to the function h are extracted from experimental data.

The meaning of $\Delta(\mathbf{ag})$ is : if $x' \models \langle st(ag), \Phi(ag), \varepsilon'(ag) \rangle$ for $x^{'} \in U(ag)$ where $\varepsilon'(ag) = \varepsilon(ag) + \Delta\varepsilon(ag)$ then if the controllable children $ag_{i_1}, ..., ag_{i_r}$ of ag will issue objects $y_{i_1}, ..., y_{i_r}$ with $y_{i_j} \models \langle st(ag_{i_j}), \Phi(ag_{i_j}), \varepsilon(ag_{i_j}) + \Delta\varepsilon(ag_{i_j}) \rangle$ for $j = 1, ..., r$ where $(\Delta\varepsilon(ag_{i_1}), ..., \Delta\varepsilon(ag_{i_r})) =$

$h(\varepsilon(ag), \Leftrightarrow\Delta\varepsilon(ag), \varepsilon(ag_1), ..., \varepsilon(ag_k))$ then the agent ag will construct an object y such that $y \models \langle st(ag), \Phi(ag), \varepsilon\rangle$ where $\varepsilon \geq \varepsilon(ag)$.

In the above formula, we assume $\Delta\varepsilon(ag) \leq 0$ and $\Delta\varepsilon(ag_{i_1}) \geq 0, ..., \Delta\varepsilon(ag_{i_r}) \geq 0$. The above semantics covers the case when Δ - rules allow to compensate in one step the influence of noise.

$\Delta\Leftrightarrow$ rules can be composed in an obvious sense. $\Delta(\mathbf{ag}\star\mathbf{ag}')$ denotes the composition of $\Delta(\mathbf{ag})$ and $\Delta(\mathbf{ag}')$ over $\mathbf{ag}\star\mathbf{ag}'$. The variable $\Delta(\mathbf{c})$ will run over compositions of $\Delta\Leftrightarrow$rules over $\mathbf{c}$. We can sum up the above discussion in a counterpart of Theorem 2.3.1 which formulates a goodness - of - controller criterion.

Theorem 2.4.1 (*the sufficiency criterion of correctness of the controller*). Let $F_{\mathbf{c}}(v) \models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon\rangle$ where v is the valuation of leaf agents of the (C, Φ, ε) -scheme $\mathbf{c}$ and let $F_{\mathbf{c}}(v') \models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon'\rangle$ where v' is a valuation of leaf agents of $\mathbf{c}$ such that $v'(b_{ag}) = v(b_{ag})$ for $ag \in Control(\mathbf{c})$, $\varepsilon' < \varepsilon$. If $\{\varepsilon_{new}(ag)\}$ is a new assignment to agents defined by a composition $\Delta(\mathbf{c})$ of some Δ-rules such that $\varepsilon_{new}(ag) = \varepsilon(ag)$ for $ag \in Un\_control(\mathbf{c})$, $\varepsilon_{new}(ag_{\mathbf{c}}) = \varepsilon$ and $\{x_{ag} : ag \in Control(\mathbf{c})\}$ is the set of control parameters (inventory objects) satisfying $x_{ag} \models \langle st(ag), \Phi(ag), \varepsilon_{new}(ag)\rangle$ for $ag \in Control(\mathbf{c})$ then for the object $x_{new} = F_{\mathbf{c}}(v_1)$ constructed over the valuation v_1 of leaf agents in $\mathbf{c}$ such that $v_1(b_{ag}) = v'(b_{ag})$ for $ag \in Un\_control(\mathbf{c})$ and $v_1(b_{ag}) = x_{ag}$ for $ag \in Control(\mathbf{c})$ it holds that $x_{new} \models \langle st(ag_{\mathbf{c}}), \Phi, \varepsilon\rangle$. ■

The meaning of Theorem 2.4.1 is that the controllable agents are able to compensate of noise which perturbs the state $\langle st(ag_{\mathbf{c}}), \Phi, \varepsilon\rangle$ to $\langle st(ag_{\mathbf{c}}), \Phi, \varepsilon'\rangle$ in case the search in the space of Δ - rules results in finding a composition of them which defines new, better uncertainty coefficients $\varepsilon_{new}(ag)$ for $ag \in Control(\mathbf{c})$; the new valuation v_1 defined by $\{\varepsilon_{new}(ag) : ag \in Control(\mathbf{c})\}$ satisfies the state $\langle st(ag_{\mathbf{c}}), \Phi, \varepsilon\rangle$.

The above approach can be treated as a first step towards modeling complex distributed dynamical systems. We expect that it can be extended to solve control problem for complex dynamical systems i.e. dynamical systems which are distributed, highly nonlinear, with vague concepts involved in their description. One can hardly expect that classical methods of control theory can be successfully applied to such complex systems.

2.5 Adaptive Calculus of Granules

Within paradigm of rough mereology one may formalize adaptive calculus of granules [342]. The metaphor of a granule, already present in rough set theory, has been recently advocated as a central notion of soft computing [544], [543].

2.5.1 Information Granules

We would like to present a general view on the problem of information granule construction and information granule calculus. Our main claims are :

- granules can be identified with finite relational structures (finite models);
- composition operations of granules (knowledge fusion) can be represented by operations on finite models;

- granules are fused, transformed and converted into decision by intelligent computing units (agents) or their schemes;
- schemes of agents are seen as decision classifiers and may be regarded as terms over granules and operations on them whose values are decisions;
- structure of granules and composition operations as well as the agent scheme and conversion operations on granules should be adaptively learned from data (the accessible information about objects).

We propose to realize this program on the basis of rough mereology. Let us mention that fuzzy set approach can be treated as a particular case of this methodology. One can consider also this approach as a basis for feature extraction in pattern recognition and machine learning. In the feature extraction process for instance we would like to develop learning strategies extracting from initial accessible information $Inf_A(x_i)$ about object x_i and decision $d(x_i)$ on x_i (for $i = 1, ..., n$ where n is the number of objects in data table) appropriate granules (finite structures) in the form of e.g. a finite model M and valuations v_i as well as formulas α_i for $i = 1, ..., k$ expressible in a language of the signature the same as M (and such that total size of these formulas is as small as possible) for which the following conditions hold

$d(x_i) \neq d(x_j)$ implies $\{\alpha_p : M, v_i \models \alpha_p\} \neq \{\alpha_p : M, v_j \models \alpha_p\}$

for any $i \neq j; i, j = 1.,,,, k$.

We start our presentation assuming the simple form of information granules defined by information systems. We have a variety of indiscernibility spaces $\{U/IND_B : B \subseteq A\}$; the Boolean algebra generated over the set of atoms U/IND_B by means of set - theoretical operations of union, intersection and complement is said to be the *B-algebra $CG(B)$ of B - pre-granules.* Any member of $CG(B)$ is called a *B - pre-granule.*

We have an alternative logical language in which we can formalize the notion of an *information pre-granule*; for a set of attributes $B \subseteq A$, we recall the definition of the B-logic L_B: elementary formulas of L_B are of the form (a, v) where $a \in B$ and $v \in V_a$. Formulas of L_B are built from elementary formulas by means of logical connectives $\vee, \wedge$; thus, each formula in DNF is represented as $\vee_{j\in J} \wedge_{i\in I_j} (a_i, v_i)$. The formulas of L_B, called *information pre-granules*, are interpreted in the set of objects U: the denotation $[(a, v)]$ of an elementary formula (a, v) is the set of objects satisfying the equation $a(x) = v$ i.e. $[(a, v)] = \{u \in U : a(u) = v\}$ and this is extended by structural induction viz. $[\alpha \vee \beta] = [\alpha] \cup [\beta]$, $[\alpha \wedge \beta] = [\alpha] \cap [\beta]$ for $\alpha, \beta \in L_B$.

Clearly, given a B -pre-granule $G \in CG(B)$, there exists an information pre-granule α_G of L_B such that $[\alpha_G] = G$.

An atom of the Boolean algebra $CG(B)$ will be called an *elementary B-pre-granule;* clearly, for any atom G of $CG(B)$ there exists an *elementary information pre-granule α_G* of the form $\wedge_{a\in B}(a, v_a)$ such that $[\alpha_G] = G$.

For given non-empty sets $B, C \subseteq A$, a pair (G_B, G_C) where $G_B \in CG(B)$ and $G_C \in CG(C)$ is called a (B,C) - *granule of knowledge.* There exists therefore *an information granule* $(\alpha_{G_B}, \alpha_{G_C})$ such that $\alpha_{G_B} \in L_B, \alpha_{G_C} \in L_C$, $[\alpha_{G_B}]$=G_B and $[\alpha_{G_C}]$=G_C. If G_B, G_C are atoms then the pair (G_B, G_C) is called an *elementary (B,C) - granule.*

One can associate with any granule (G', G) where $G' \in CG(B'), G \in CG(B)$ a rule $\alpha_G \Longrightarrow \alpha_{G'}$[342].

The notion of a granule corresponds to the logical content (*logic of knowledge*) of the information system; however, there is more to the notion of a granule of knowledge: we have to take into account the restrictions which on the choice of *good granules* are imposed by the structure and demands of two interfaces: *input_interface* which controls the input objects (signals) and *output_interface* which controls the output objects (actions, signals).

Consider a granule (G, G'); let $G = [\alpha_G]$, $G' = [\alpha_{G'}]$. There are two characteristics of the granule (G, G') important in applications to adaptive synthesis of complex objects viz. the characteristic whose values measure what part of $[G']$ is in $[G]$ (the *strength of the covering* of the rule $\alpha_G \Longrightarrow \alpha_{G'}$ and the characteristic whose values measure what part of $[G]$ is in $[G']$ (the *strength of the support* for the rule $\alpha_G \Longrightarrow \alpha_{G'}$).

To select sufficiently strong rules, we would set thresholds : $tr, tr' \in [0,1]$. We define then, by analogy with machine learning techniques, two characteristics:

(ρ) $\rho(\alpha, \alpha') = \mu([\alpha], [\alpha']);$

(η) $\eta(\alpha, \alpha') = \mu([(\alpha')], [\alpha])$ for any (B, C)-information granule (α, α')

and we call a $(\mu, B, C, tr, tr')\Leftrightarrow$ *information granule of knowledge* any (B, C)-information granule (α, α') such that

(i) $\rho(\alpha, \alpha') \geq tr$ and

(ii) $\eta(\alpha, \alpha') \geq tr'$.

The set of all $(\mu, B, C, tr, tr')\Leftrightarrow$ granules corresponding to $(\mu, B, C, tr, tr')\Leftrightarrow$ information granules generates a Boolean algebra of (μ, B, C, tr, tr')-*granules of knowledge.* Let us observe that given sets $B, C \subseteq A, \alpha \in L_B$ and $\alpha' \in L_C$, we may define the value $Gr(B, C, \mu, tr, tr', \alpha, \alpha')$ to be $TRUE$ in the case when the pair (α, α') is an $(\mu, B, C, tr, tr')\Leftrightarrow$granule of knowledge. In this way we define the relation Gr which we call the *granulation relation* induced by the triple *(input_interface, logic of knowledge, output_interface)* related to the information system **A.**

The functions η, ρ and thresholds tr, tr' introduced above have been used to present an example of the interface between two sources of information e.g. between input information source and inner agent knowledge. This is necessary because very often the exact interpretation of information from one source into another one is impossible. Different methods for constructing and tuning these interfaces up, crucial for granular computing, can be developed using rough set and fuzzy set approaches.

Rough inclusion μ_0 may enter our discussion of a granule and of the relation Gr in each of the following ways:

- concerning the definitions $(\eta), (\rho)$ of functions η and ρ , we may replace in them the rough membership function μ with μ_0 possibly better fitted to a context.

- the function μ_0 can be extended to a rough inclusion μ^* by means of a formula above, relative to a t-conorm $\perp$ and a t-norm $\top$ suitably chosen. This rough inclusion μ^* can now be used to measure the information granule closeness.

2.5.2 Information Granules of the Form (Φ, ε)

In applications it is often convenient to use another language for information granules description allowing for more compressed description of information granules. Let us assume that a pre-rough inclusion is generated over a set B of attributes. Let μ^* be the rough inclusion generated from μ_0. Now, a new information granule language $\mathbf{L}_B$ for $B \subseteq A$ consists of pairs (Φ, ε) where Φ is an elementary B-information pre-granule and $\varepsilon \in [0,1]$. By $[\Phi, \varepsilon]_B$ we denote the B-granule $[\alpha_{\Phi,\varepsilon}]$ where $\alpha_{\Phi,\varepsilon}$ is a B-information granule being a disjunction of all $\wedge u$ where u is of the form $\{(a, v) : a \in B, v \in V_a\}$ (where V_a is the value set of a) and $\mu^*([\wedge u], [\Phi]) \geq \varepsilon$.

We say that

$$x \models_{ag} (\Phi, \varepsilon) \text{ iff } \mu^*([\wedge u(x)], [\Phi]) \geq \varepsilon$$

where $u(x)$ is $\{(a, a(x)) : a \in B\}$.

One can also consider more general information granules taking instead of formulas (Φ, ε) sets $(\Phi, \varepsilon_1), ..., (\Phi, \varepsilon_n)$ where $\varepsilon_1 \leq ... \leq \varepsilon_n$ are interpreted as closeness degrees to Φ.

2.5.3 Synthesis in Terms of Granules

We adopt here our scheme for approximate synthesis and we refer to the notation therein. We will show how the synthesis process over this scheme can be driven by granule exchange among agents. We include the following changes in comparison to this scheme.

1. We introduce an additional agent Cs (the *customer*), where $Cs \notin Ag$, whose actions consist in issuing approximate specifications Ψ describing the desired object (signal, action etc.) to be synthesized by the scheme (Inv, Ag) where Inv is the set of inventory objects of agents from Ag.

2. We assume that the $customer \Leftrightarrow root\ of\ Ag\ interface$ produces approximate formulas of the form
$$(\Phi(root(Ag)), \varepsilon(root(Ag))$$
for the root agent $root(ag)$ of the agent scheme corresponding to approximate specifications Ψ in the sense that any object satisfying the formula
$$(\Phi(root(Ag)), \varepsilon(root(Ag))$$
is regarded as satisfying the approximate specification Ψ to a satisfactory degree.

3. Let $[\Phi(ag), \varepsilon(ag)]_{ag} =$
$$\{x \in U(ag) : x \models_{ag} (\Phi(ag), \varepsilon(ag))\}.$$

4. For a given specification $(\Phi(ag), \varepsilon(ag))$ at ag, k-ary operation $o(ag)$ at ag and a mereological connective $f(ag)$ (see above) the decomposition process returns a sequence
$$(\Phi_1, \varepsilon_1), ..., (\Phi_k, \varepsilon_k)$$
of specifications (in the language of ag) for agents ag_1,...,ag_k satisfying
$$\emptyset \neq o(ag)([\Phi_1, \varepsilon_1]_{ag} \times ... \times [\Phi_k, \varepsilon_k]_{ag})$$
$$\subseteq [\Phi(ag), f(ag)(\varepsilon_1, ..., \varepsilon_k)]_{ag}]$$
where $f(ag)(\varepsilon_1, ..., \varepsilon_k) \geq \varepsilon(ag)$.

5. Next, information granules α_1, ..., α_k at ag_1, ..., ag_k, respectively, are chosen in such a way that granules
$$[\alpha_1]_{ag_1}, ..., [\alpha_k]_{ag_k}$$
and respectively
$$[\Phi_1, \varepsilon_1]_{ag}, ..., [\Phi_k, \varepsilon_k]_{ag}$$
are sufficiently close (in the tuning process parameters ρ, η, tr, tr' are fixed). The closeness of the granules should guarantee that the following inclusion is true
$o(ag)([\alpha_1]_{ag_1} \times ... \times [\alpha_k]_{ag_k})$
$$\subseteq [\Phi(ag), f(ag)(\varepsilon_1, ..., \varepsilon_k)]_{ag}.$$

Formulae α_1,...,α_k at ag_1,...,ag_k are described in the form

$$(\Phi(ag_1), \varepsilon(ag_1)), ..., (\Phi(ag_k), \varepsilon(ag_k))$$

for agents ag_1,...,ag_k, respectively.

The meaning of an expression $x \models_S (\Phi, \varepsilon)$ is that an agent scheme S is yielding at the $root(S)$ the object x. A goodness - of - granule synthesis scheme criterion can be formulated in the following:

Theorem 2.5.1 *([342])*. Let S be an agent scheme satisfying the following conditions:

(i) the root of S denoted by $root(S)$ has attached a specification

$$(\Phi(root(S), \varepsilon(root(S)));$$

(ii) any non-leaf and non-root agent ag of S satisfies conditions stated in (4)-(5);

(iii) for any leaf agent ag of S the attached specification $(\Phi_{ag}, \varepsilon_{ag})$ is satisfied by some object from the inventory object set INV.

Then the scheme S is yielding at $root(S)$ an object x satisfying

$$x \models_S (\Phi(root(S)), \varepsilon(root(S))).$$

■

2.5.4 Adaptivity of calculus of granules

The adaptivity of our scheme is due to the several factors. Among them are

- The possibility of changing the parameters

  $$\mu, tr, tr', B, C$$

  in the granulation predicate

  $$Gr(ag)(B, C, \mu, tr, tr', \alpha, \alpha')$$

  for any agent $ag \in Ag$.

- The possibility of new granule formation at any agent $ag \in Ag$ in the dynamic process of synthesis.

- The possibility of forming new rough inclusion at any agent $ag \in Ag$ in the dynamic process of synthesis e.g. by choosing $\top, \perp$ in the definition of μ.

In conclusions we discuss some other potential applications of rough mereological approach.

2.6 Conclusions

The rough set approach to data analysis has many important advantages. Some of them are listed below.

- Synthesis of efficient algorithms for finding hidden patterns in data.
- Identification of relationships that would not be found using statistical methods.
- Representation and processing of both qualitative and quantitative parameters and mixing of user-defined and measured data.
- Reduction of data to a minimal representation.
- Evaluation of the significance of data.
- Synthesis of classification or decision rules from data.
- Legibility and straightforward interpretation of synthesized models.

Most algorithms based on the rough set theory are particularly suited for parallel processing, but in order to exploit this feature fully, a new computer organization based on rough set theory is necessary.

Although rough set theory has many achievements to its credit, nevertheless several theoretical and practical problems require further attention.

Especially important is widely accessible efficient software development for rough set based data analysis, particularly for large collections of data.

Despite of many valuable methods of efficient, optimal decision rule generation methods from data, developed in recent years based on rough set theory - more research here is needed, particularly, when quantitative and quantitative attributes are involved. Also an extensive study of a new approach to missing data is very important. Comparison to other similar methods still requires due attention, although important results have been obtained in this area. Particularly interesting seems to be a study of the relationship between neural network and rough set approach to feature extraction from data.

Last but not least, rough set computer is badly needed for more advanced applications. Some research in this area is already in progress.

We would like to stress some areas of research related to the rough mereological approach. They are in particular important for further development of rough set theory and soft computing. They can be characterized as new algorithmic methods for inducing structures of information granules and information granule calculus from data (also in distributed and multi–agent environments). Among them are adaptive algorithmic methods for:

- extracting logical (algebraic) structures of information granules from data: this belongs to the process of searching for a model couched in logical (algebraic) terms;
- constructing interfaces among various knowledge structures: this group of problems is relevant in granular computing; as observed above, granules of knowledge are the result, among other factors, of uncertainty immanent to interfaces among various sources of knowledge;
- extracting distance functions for similarity measures from data: here we would like to have clustering methods based on closeness measures to construct aggregation - based models;

- inducing exact dependencies: this group of problems belong to the second step i.e. searching for elements of the model and dependencies among them; exact dependencies constitute the skeleton along which we organize schemes for approximate reasoning;
- inducing approximate dependencies: here we search for approximate i.e. close to exact dependencies and possible ways of expressing them like described in this collection and literature default rules, templates, rough classifiers, rough mereological connectives etc.;
- inducing networks of dependencies: emulation of schemes of approximate reasoning including also algorithmic methods for inducing concurrent data models from data.

We propose rough mereology as a general framework for investigations in these directions. Taking this point of view the research should be concentrated around two main groups of problems, namely methods of adaptive learning from data of components of schemes of approximate reasoning (like standards, rough inclusions, mereological connectives, decomposition schemes etc. (see Sect. 2.4)) and adaptive learning of schemes of approximate reasoning (synthesis schemes of agents).

The research results in the above mentioned areas will also have important impact on development of new methods for KDD, in particular for development of algorithmic methods for pattern extraction from data (also in multi–agent environment) and extracting from data calculus for approximate reasoning on sets of extracted patterns (e.g. algorithmic methods for large data table decomposition, synthesis of global laws from local findings).

The advantage of employing various sources of knowledge and various structures of knowledge in data mining and knowledge discovery implies that new algorithmic methods are desirable for hybrid systems in which rough set methods will be applied along with methods based on (one or more of) the following: fuzzy sets; neural nets; evolutionary strategies; statistical reasoning; belief nets; evidence theory.

Problems of knowledge discovery should be studied also from the point of view of complexity. The following topics seem to be important:

- analysis of complexity of knowledge discovery processes: complexity of extracting classes of problems solvers;
- analysis of complexity of problems of approximate reasoning: complexity of feature and model extraction/selection, complexity of data mining processes, complexity of knowledge discovery processes;
- quality of heuristics for hard problems of approximate reasoning: quality - complexity trade - off;
- refinements of complexity theory for problems of approximate reasoning: classification issues, relations to minimal length description.

Let us observe that this analysis would require in many cases transgressing the classical complexity theory.

In addition to areas of application discussed in the paper we would like to point to some areas where perspectives of applications of rough sets are promising as borne out by the current research experience. Among them are applications in: data mining and knowledge discovery; process control; case based reasoning; conflict analysis and negotiations; natural language processing; software engineering.

Finally, the progress in the discussed above rough merology, being an extension of rough set theory, should bring forth:

- new computation model based on information granulation and granule calculus;
- new software systems for mentioned above important applications;
- hardware developments: a rough mereological processor and a rough mereological computer.

We are convinced that progress in the above areas is of the utmost importance for creating new methods, algorithms, software as well as hardware systems which prove the applicability of rough set techniques to challenging problems of Data Mining, Knowledge Discovery and other important areas of applications.

Acknowledgments

The results presented in this tutorial are due to a long term collaborative effort of several international groups. The authors wish to thank their colleagues who contributed to this tutorial both directly and indirectly. Several results from joint work were re-used here and citations provided as detailed as it was judged feasible.

This research has been supported in part by the European Union 4th Framework Telematics project CardiAssist, by the ESPRIT-CRIT 2 project #20288, and by the 8T11C 02417 grant from the National Committee for Scientific Research.

Bibliography

[1] A. Aamodt, J. Komorowski (Eds.) (1995), *Proc. Fifth Scandinavian Conference on Artificial Intelligence.* Trondheim, Norway, May 29–31, Frontiers. In: Artifical Intelligence and Applications **28**, IOS Press.

[2] R.L. Ackoff, M.W. Sasieni (1968), *Fundamentals of operational research.* Wiley, New York.

[3] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I. (1996), *Fast discovery of assocation rules.* In: V.M. Fayad, G.Piatetsky Shapiro, P. Smyth, R. Uthurusamy (Eds.): Advanced in Knowledge Discovery and Data Mining, AAAI/MIT Press, pp. 307-328.

[4] H. Almuallim, T. G. Dietterich (1994), *Learning Boolean concepts in the presence of many irrelevant features.* Artificial Intelligence **69(1-2)**, pp. 279–305.

[5] S. Amarel et al. (1991), *PANEL on AI and design.* In: Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), Morgan Kaufmann, San Mateo, pp. 563–565.

[6] A. An, C. Chan, N. Shan, N. Cercone, W. Ziarko (1997), *Applying knowledge discovery to predict water–supply consumption.* IEEE Expert **12/4**, pp. 72–78.

[7] A. An, N. Shan, C. Chan, N. Cercone, W. Ziarko (1995), *Discovering rules from data for water demand prediction.* In: Proceedings of the Workshop on Machine Learning in Engineering (IJCAI'95), Montreal, pp. 187–202; see also, Journal of Intelligent Real-Time Automation, Engineering Applications of Artificial Intelligence **9/6**, pp. 645–654.

[8] T. Arciszewski, W. Ziarko (1987), *Adaptive expert system for preliminary design of wind–bracings in steel skeleton structures.* In: Second Century of Skyscraper, Van Norstrand, pp. 847–855.

[9] P. Balbiani (1998), *Axiomatization of logics based on Kripke models with relative accessibility relations.* In: Orłowska [276], pp. 553-578.

[10] C.A. Bana e Costa (1990), *Readings in multiple–criteria decision aid.* Springer–Verlag, Berlin.

[11] M. Banerjee (1997), *Rough sets and 3-valued Lukasiewicz logic.* Fundamenta Informaticae **32/1**, pp. 213–220.

[12] M. Banerjee, M.K. Chakraborty (1998), *Rough logics: A survey with further directions.pedrycz@ee.ualberta.ca* In: Orłowska [276] pp. 579-600.

[13] M. Banerjee, S.K. Pal (1996), *Roughness of a fuzzy set.* Information Sciences Informatics and Comp. Sc. **93/3-4**. pp. 235–246.

[14] J. Bazan (1998), *A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables.* In: Polkowski and Skowron [344], pp. 321–365.

[15] J. Bazan, H.S. Nguyen, T.T. Nguyen, A. Skowron, J. Stepaniuk (1994), *Some logic and rough set applications for classifying objects.* Institute of Computer Science, Warsaw University of Technology, ICS Research Report **38/94**.

[16] J. Bazan, H.S. Nguyen, T.T. Nguyen, A. Skowron, J. Stepaniuk (1995), *Application of modal logics and rough sets for classifying objects.* In: M. De Glas, Z. Pawlak (Eds.), Proceedings of the Second World Conference on Fundamentals of Artificial Intelligence (WOCFAI'95), Paris, July 3-7, Angkor, Paris, pp. 15–26.

[17] J.G. Bazan, H.S. Nguyen, T.T. Nguyen, A. Skowron, J. Stepaniuk (1998), *Synthesis of decision rules for object classification.* In: Orłowska [276], pp. 23–57.

[18] J. Bazan, A. Skowron, P. Synak (1994), *Market data analysis.* Institute of Computer Science, Warsaw University of Technology, ICS Research Report **6/94**.

[19] J. Bazan, A. Skowron P. Synak (1994), *Dynamic reducts as a tool for extracting laws from decision tables.* Proc. Symp. on Methodologies for Intelligent Systems, Charlotte, NC, USA, Oct. 16-19, LNAI, **869**, Springer-Verlag, pp. 346–355.

[20] D. Brindle (1995), *Speaker-independent speech recognition by rough sets analysis.* In: Lin, Wildberger [196], pp. 101–106.

[21] T. Beaubouef, F.E. Petry (1993), *A rough set model for relational databases.* In: Ziarko [550], pp. 100–107.

[22] T. Beaubouef, F.E. Petry (1994), *Rough querying of crisp data in relational databases.* In: Ziarko [550], pp. 85–88.

[23] S. Bodjanova (1997), *Approximation of fuzzy concepts in decision making.* Fuzzy Sets and Systems **85**, pp. 23–29.

[24] P. Borne, G. Dauphin–Tanguy, C. Sueur, S. El Khattabi (Eds.), *Proceedings of IMACS Multiconference, Computational Engineering in Systems Applications (CESA'96).* July 9–12, Lille, France, Gerf EC Lille – Cite Scientifique **3/4**.

[25] B. Bouchon–Meunier, M. Delgado, J.L. Verdegay, M.A. Vila, R.R. Yager (1996), *Proceedings of the Sixth International Conference. Information Processing Management of Uncertainty in Knowledge–Based Systems (IPMU'96).* July 1-5, Granada, Spain **1–3**, pp. 1–1546.

[26] B. Bouchon–Meunier, R.R. Yager (1998), *Proc. Seventh Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU'98).* July 6–10, Université de La Sorbonne, Paris, France, Edisions, E.D.K., Paris, pp. 1–1930.

[27] Z. Bonikowski, E. Bryniarski, and U. Wybraniec–Skardowska (1998), *Extensions and intensions in the rough set theory.* Information Sciences **107**, 149–167.

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

[29] C. Browne, I. Düntsch, G. Gediga (1998), *IRIS revisited, A comparison of discriminant enhanced rough set data analysis.* In: Polkowski and Skowron [345], pp. 347–370.

[30] E. Bryniarski E., U. Wybraniec-Skardowska (1997), *Generalized rough sets in contextual spaces.* In: Lin and Cercone [191] pp. 339–354.

[31] A. Budihardjo, J.W. Grzymała–Busse, L. Woolery (1991), *Program LERS–LB 2.5 as a tool for knowledge acquisition in nursing.* In: Proceedings of the Fourth International Conference on Industrial Engineering Applications of Artificial Intelligence Expert Systems, Koloa, Kauai, Hawaii, June 2–5, pp. 735–740.

[32] M. Burns (1993), *Resources: Automated Fabrication. Improving Productivity in Manufacturing.* Prentice Hall, Englewood Cliffs, NJ.

[33] W. Buszkowski, E. Orłowska (1998), *Indiscernibility–based formalisation of dependencies in information systems.* In: Orłowska [276], pp. 293–315.

[34] W. Buszkowski (1998), *Approximation spaces and definability for incomplete information systems.* In: Polkowski and Skowron [346], pp. 115–122.

[35] U. Carlin, J. Komorowski, A. Øhrn (1998), *Rough set analysis of patients with suspected acute appendicitis.* In: Bouchon–Meunier and Yager [26], pp. 1528–1533.

[36] J. Catlett (1991), *On changing continuos attributes into ordered discrete attributes.* In: Y. Kodratoff, (Ed.), Machine Learning-EWSL-91, Proc. of the European Working Session on Learning, Porto, Portugal, March 1991, LNAI, pp. 164–178.

[37] G. Cattaneo (1996), *A unified algebraic approach to fuzzy algebras and rough approximations.* In: R. Trappl (Ed.), Proceedings of the 13th European Meeting on Cybernetics and Systems Research (CSR'96), April 9–12, The University of Vienna **1**, pp. 352–357.

[38] G. Cattaneo (1996), *Abstract rough approximation spaces (Bridging the gap between fuzziness and roughness).* In: Petry and Kraft [321], pp. 1129–1134.

[39] G. Cattaneo (1996), *Mathematical foundations of roughness and fuzziness*, In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484]. pp. 241–247.

[40] G. Cattaneo (1997), *Generalized rough sets. Preclusivity fuzzy-intuitionistic (BZ) lattices. Studia Logica* **58**, pp. 47–77.

[41] G. Cattaneo (1998), *Abstract approximation spaces for rough theories.* In: Polkowski and Skowron [344], pp. 59–98.

[42] G. Cattaneo, R. Giuntini, R. Pilla (1997), *MVBZ algebras and their substructures. The abstract approach to roughness and fuzziness.* In: Mares, Meisar, Novak, and Ramik [206], pp. 155–161.

[43] G. Cattaneo, R. Giuntini, R. Pilla (1997), *$MVBZ^{dM}$ and Stonian algebras (Applications to fuzzy sets and rough approximations).* In: Fuzzy Sets and Systems.

[44] M.K. Chakraborty, M. Banerjee (1997), *In search of a common foundation for rough sets and fuzzy sets.* In: Zimmermann [558], pp. 218–220.

[45] Y.-Y. Chen, K. Hirota, J.-Y. Yen (Eds.) (1996), *Proceedings of 1996 ASIAN FUZZY SYSTEMS SYMPOSIUM – Soft Computing in Intelligent Systems Information Processing.* December 11–14, Kenting, Taiwan, ROC.

[46] M.R. Chmielewski, J.W. Grzymala-Busse (1994), In: Lin [188] pp. 294–301.

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

[48] S. Comer (1991), *An algebraic approach to the approximation of information.* Fundamenta Informaticae **14**, pp. 492–502.

[49] J.H. Connolly J.H., E.A. Edmunds E.A. (1994), *CSCW and Artificial Intelligence.* Springer - Verlag, Berlin.

[50] A. Czajewski (1998), *Rough sets in optical character recognition.* In: Polkowski and Skowron [346], pp. 601–604.

[51] A. Czyżewski (1995), *Some methods for detection interpolation of impulsive distortions in old audio recordings.* In: IEEE ASSP Workshop on Applications of Signal Processing to Audio Acoustics Proceedings, October 15-18, New York, USA.

[52] A. Czyżewski (1996), *Speaker–independent recognition of digits – Experiments with neural networks, fuzzy logic rough sets.* Journal of the Intelligent Automation and Soft Computing **2/2**, pp. 133–146.

[53] A. Czyżewski (1996), *Mining knowledge in noisy audio data.* In: Simoudis, Han, and Fayyad [402] pp. 220–225.

[54] A. Czyżewski (1997), *Learning algorithms for audio signal enhancement – Part I, Neural network implementation for the removal of impulse distortions.* Journal of the Audio Engineering Society **45/10**, pp. 815–831.

[55] A. Czyżewski (1997), *Learning algorithms for audio signal enhancement - Part II, Rough set method implementation for the removal of hiss.* Journal of the Audio Engineering Society **45/11**, pp. 931–943.

[56] A. Czyżewski (1998), *Soft processing of audio signals.* In: Polkowski and Skowron [345], pp. 147–165.

[57] A. Czyżewski, A. Kaczmarek (1995), *Speaker-independent recognition of isolated words using rough sets.* In: Wang [513], pp. 397–400.

[58] A. Czyżewski, B. Kostek (1996), *Rough set-based filtration of sound applicable to hearing prostheses.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 168–175.

[59] A. Czyżewski, B. Kostek (1996), Restoration of old records employing artificial intelligence methods. Proceedings of IASTED International Conference – Artificial Intelligence, Expert Systems Neural Networks, August 19–21, Honolulu, Hawaii, USA, pp. 372–375.

[60] A. Czyżewski, R. Królikowski (1997), *New methods of intelligent filtration coding of audio.* In: 102nd Convention of the Audio Engineering Society, March 22–25, preprint **4482**, Munich, Germany .

[61] A. Czyżewski, R. Królikowski (1998), Applications of fuzzy logic rough sets to audio signal enhancement. In this book.

[62] A. Czyżewski, R. Królikowski, P. Skórka (1996), *Automatic detection of speech disorders.* In: Zimmermann [557], pp. 183–187.

[63] B.V. Dasarathy (Ed.) (1991), *Nearest Neighbor Pattern Classification Techniques.* IEEE Computer Society Press.

[64] R. Deja (1996), *Conflict analysis.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 118–124.

[65] S. Demri (1996), *A class of information logics with a decidable validity problem.* In: Lecture Notes in Computer Science **1113**, Springer-Verlag pp. 291–302.

[66] S. Demri, E. Orłowska (1998), *Logical analysis of indiscernibility.* In: Orłowska [276], pp. 347–380.

[67] J.S. Deogun, V.V. Raghavan, A. Sarkar, H. Sever (1997), *Data mining: Trends in research and development.* In: Lin and Cercone [191]. pp. 9–45.

[68] J. Dougherty, R. Kohavi, M. Sahami (1995), *Supervised and unsupervised discretization of continuous features.* Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, pp. 194–202.

[69] D. Dubois, H. Prade (1980), *Fuzzy Sets and Systems: Theory and Applications.* Academic Press.

[70] D. Dubois, H. Prade (1987), *Twofold fuzzy sets and rough sets - Some issues in knowledge representation.* Fuzzy Sets and Systems **23**, pp. 3–18.

[71] D. Dubois, H. Prade (1990), *Rough fuzzy sets and fuzzy rough sets.* International J. General Systems **17**, pp. 191–209.

[72] D. Dubois, H. Prade (1992), *Putting rough sets and fuzzy sets together.* In: Słowiński [428], pp. 203–232.

[73] D. Dubois, H. Prade (1992), *Comparison of two fuzzy set–based logics: similarity logic and possibilistic logic.* In: Proceedings FUZZ-IEEE'95, Yokohama, Japan, pp. 1219–1236.

[74] D. Dubois, H. Prade, R.R. Yager(Eds.) (1993), *Readings in Fuzzy Sets for Intelligent Systems*, Morgan Kaufmann, San Mateo.

[75] D. Dubois, F. Esteva, P. Garcia, L. Godo, and H. Prade (1997), *Logical approach to interpolation based on similarity.* Journal of Approximate Reasoning (to appear) (available as IIIA Research Report 96-07).

[76] D. Dubois, H. Prade (1998), *Similarity-based approximate reasoning.* In: J.M. Zurada, R.J. Marks II, and X.C.J. Robinson (Eds.), *Proceedings of the IEEE Symposium*, Orlando, FL, June 17–July 1st, IEEE Press, pp. 69–80.

[77] D. Dubois, H. Prade (1998), *Similarity versus preference in fuzzy set-based logics.* In: Orłowska [276] pp. 440–460.

[78] E.H. Durfee (1988), *Coordination of Distributed Problem Solvers.* Kluwer Academic Publishers, Boston.

[79] I. Düntsch (1997), *A logic for rough sets.* Theoretical Computer Science **179/1-2** pp. 427–436.

[80] I. Düntsch (1998), *Rough sets and algebras of relations.* In: Orłowska [276], pp. 95–108.

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

[82] I. Düntsch, G. Gediga (1997), *Non–invasive data analysis.* Proceedings of the Eighth Ireland Conference on Artificial Intelligence, Derry, pp. 24–31.

[83] I. Düntsch, G. Gediga (1997), *Algebraic aspects of attribute dependencies in information systems.* Fundamenta Informaticae **29**, pp. 119–133.

[84] I. Düntsch, G. Gediga (1998), *IRIS revisited: A comparison of discriminant and enhanced rough set data analysis.* In: Polkowski and Skowron [344], pp. 347–370.

[85] A.E. Eiben, T.J. Euverman, W. Kowalczyk, F. Slisser (1998), *Modelling customer retention with statistical techniques, rough data models, genetic programming.* In this book.

[86] E. Ephrati, J.S. Rosenschein (1994), *Divide and conquer in multi-agent planning.* In: Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94) AAAI Press/MIT Press, Menlo-Park, CA, pp. 375–380.

[87] F. Esteva, P. Garcia, L. Godo (1993), *On the relatonsship between preference and similarity–based approaches to possibilistic reasoning.* In: Proceedings of the 2nd IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93), San Francisco, CA, March 28–April 1st, pp. 918–923.

[88] F. Esteva, P. Garcia, L. Godo (1997), *Similarity reasoning: Logical grounds and applications.* In: Zimmermann [558], pp. 113–117.

[89] L. Farinas del Cerro, E. Orłowska (1985), *DAL–A logic for data analysis.* Theoretical Computer Science **36**, pp. 251–264; see also, Corrigendum, ibidem **47** (1986), pp. 345.

[90] L. Farinas del Cerro, H. Prade (1986), *Rough sets, twofold fuzzy sets and modal logic – Fuzziness in indiscernibility and partial information.* In: A. Di Nola, A.G.S. Ventre (Eds.), The Mathematics of Fuzzy Systems, Verlag TUV Rheinland, Köln, pp. 103–120.

[91] U.M. Fayyad, K.B. Irani (1992), *The attribute election problem in decision tree generation.* Proc. of AAAI-92, July 1992, San Jose, CA., MIT Press, pp. 104-110.

[92] U. Fayyad, G. Piatetsky-Shapiro, G. (Eds.) (1996), *Advances in Knowledge Discovery and Data Mining.* MIT/AAAI Press.

[93] U.M. Fayyad, R. Uthurusamy (Eds.) (1995), *Proceedings of the First International Conference on Knowledge Discovery Data Mining (KDD'95).* Montreal, August, AAAI Press, Menlo Park, CA, pp. 1–354.

[94] J. Fibak, Z. Pawlak, K. Słowiński, R. Słowiński (1986), *Rough sets based decision algorithm for treatment of duodenal ulcer by HSV.* Bull. Polish Acad. Sci. Biological Sci. **34/10-12**, pp. 227–246.

[95] J. Fibak, K. Słowiński, R. Słowiński (1986), *The application of rough sets theory to the verification of treatment of duodenal ulcer by HSV.* In: Proceedings of the Sixth International Workshop on Expert Systems their Applications, Agence de l'Informatique, Paris, pp. 587–599.

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

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

[98] M.C. Fernandes–Baizan, E. Menasalvas Ruiz, M. Castano (1996), *Integrating RDMS data mining capabilities using rough sets.* In: Bouchon–Meunier, Delgado, Verdegay, Vila, and Yager [25] **2**, pp. 1439–1445.

[99] K. Funakoshi, T.B. Ho (1996), *Information retrieval by rough tolerance relation.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 31–35.

[100] H. Furuta, M. Hirokane, Y. Mikumo (1998), *Extraction method based on rough set theory of rule-type knowledge from diagnostic cases of slope-failure danger levels.* In: Polkowski and Skowron [345], pp. 178–192.

[101] B. Ganter, R. Wille (1996), *Formale Begriffsanalyse: Mathematische Grundlagen.* Springer–Verlag, Berlin, pp. 1–286.

[102] M. L. Geleijnse, A. Elhendy, R. van Domburg et al. (1996), *Prognostic Value of Dobutamine-Atropine Stress Technetium-99m Sestamibi Perfusion Scintigraphy in Patients with Chest Pain.* Journal of the American College of Cardiologists, **28/2**, August, pp. 447–454.

[103] R. Golan, W. Ziarko (1995), *A methodology for stock market analysis utilizing rough set theory.* In: Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, New York City, pp. 32–40.

[104] S. Greco, B. Matarazzo, R. Słowiński (1997), *Rough set approach to multi-attribute choice ranking problems.* Institute of Computer Science, Warsaw University of Technology, ICS Research Report **38/95** (1995); see also, G. Fandel, T. Gal (Eds.), Multiple Criteria Decision Making: Proceedings of 12th International Conference in Hagen, Springer–Verlag, Berlin, pp. 318–329.

[105] S. Greco, B. Matarazzo, R. Słowiński (1998), *A new rough set approach to evaluation of bankruptcy risk.* In: C. Zopounidis (Ed.), New Operational Tools in the Management of Financial Risks, Kluwer Academic Publishers, Dordrecht pp. 121–136.

[106] S. Greco, B. Matarazzo, R. Słowinski (1998), *Rough approximation of a preference relation in a pairwise comparison table.* In: Polkowski and Skowron [345] pp. 13–36.

[107] J.W. Grzymała-Busse (1986), *Algebraic properties of knowledge representation systems.* In: Z.W. Ras, M. Zemankova (Eds.), Proceedings of the ACM SIGART International Symposium on Methodologies for Intelligent Systems (ISMIS'86), Knoxville, Tennessee, October 22–24, ACM Special Interest Group on Artificial Intelligence, pp. 432–440.

[108] J.W. Grzymała–Busse (1993), *ESEP, an expert system for enviromental protection.* In: Ziarko [550], pp. 499–508.

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

[110] J.W. Grzymała–Busse (1998), *Applications of the rule induction system LERS.* In: Polkowski and Skowron [344], pp. 366–375.

[111] J.W. Grzymała-Busse L.K. Goodwin (1997), *Predicting preterm birth risk using machine learning from data with missing values.* In: S. Tsumoto (Ed.), Bulletin of International Rough Set Society **1/2**, pp. 17–21.

[112] J.W. Grzymała–Busse, L.K. Goodwin (1996), *A comparison of less specific versus more specific rules for preterm birth prediction.* In: Proceedings of the First Online Workshop on Soft Computing WSC1 on the Internet, served by Nagoya University, Japan, August 19–30, pp. 129–133.

[113] J.W. Grzymała–Busse, J.D. Gunn (1995), *Global temperature analysis based on the rule induction system LERS.* In: Proceedings of the Fourth International Workshop on Intelligent Information Systems, Augustów, Poland, June 5–9, 1995, Institute od Computer Science, Polish Academy of Sciences, Warsaw, pp. 148–158.

[114] J.W. Grzymała–Busse, L.J. Old (1997), *A machine learning experiment to determine part of speach from word-endings.* In: Ras and Skowron [363], pp. 497–506.

[115] J.W. Grzymała–Busse, S. Than (1993), *Data compression in machine learning applied to natural language.* Behavior Research Methods, Instruments & Computers **25**, pp. 318–321.

[116] J.W. Grzymała–Busse, C.P.B. Wang (1996), *Classification rule induction based on rough sets.* In: [321], pp. 744–747.

[117] J.W. Grzymała–Busse, L. Woolerly (1994), *Improving prediction of preterm birth using a new classification scheme rule induction.* In: Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care (SCAMC), November 5-9, Washington D. C. pp. 730–734.

[118] J.D. Gunn, J.W. Grzymała-Busse (1994), *Global temperature stability by rule induction. An interdisciplinary bridge.* Human Ecology **22**, pp. 59–81.

[119] M. Hadjimichael, A. Wasilewska (1993), *Rough sets-based study of voter preference in 1988 USA presidential election.* In: Słowiński [428], pp. 137–152.

[120] D. Heath, S. Kasif, S. Salzberg (1993), *Induction of oblique decision trees.* Proc. 13th International Joint Conf. on AI, Chambery, France, pp. 1002-1007.

[121] R.C. Holt (1993), *Very simple classification rules perform well on most commonly used datasets.* Machine Learning **11**, pp. 63–90.

[122] Y. Hou, W. Ziarko (1996), *A rough sets approach to handwriting classification.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp.372–382.

[123] U. Höhle (1988), *Quotients with respect to similarity relations. Fuzzy Sets and Systems* **27**, pp. 31–44.

[124] X. Hu, N. Cercone (1995), *Rough set similarity based learning from databases.* In: Fayyad and Uthurusamy [93], pp. 162–167.

[125] X. Hu, N. Cercone (1997), *Learning maximal generalized decision rules via discretization, generalization, and rough set feature selection.* In: Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, Newport Beach, CA, pp. 548–557.

[126] X. Hu, N. Cercone, W. Ziarko (1997), *Generation of multiple knowledge from databases based on rough set theory.* In: Lin and Cercone [191], pp. 109–121.

[127] T. Ishida (1994), *Parallel, Distributed and Multiagent Production Systems.* Lecture Notes in Computer Science **878**, Springer–Verlag, Berlin.

[128] T. Iwiński (1987), *Algebraic approach to rough sets.* Bull. Polish Acad. Sci. Math. **35**, pp. 673–683.

[129] T. Iwiński (1988), *Rough order and rough concepts.* Bull. Polish Acad. Sci. Math. pp. 187–192.

[130] J. Järvinen (1997), *A Representation of dependence spaces and some basic algorithms.* Fundamenta Informaticae **29/4**, pp. 369–382.

[131] A.G. Jackson, S.R. Leclair, M.C. Ohmer, W. Ziarko, H. Al-Kamhawi (1996), *Rough sets applied to material data.* Acta Metallurgica et Materialia, pp.44–75.

[132] J. Jelonek, K. Krawiec, R. Słowiński, *Rough set reduction of attributes and their domains for neural networks.* In: [CI], pp.339–347

[133] J. Jelonek, K. Krawiec, R. Słowiński, J. Stefanowski, J. Szymas, (1994), *Neural networks rough sets – Comparison combination for classification of histological pictures.* In: Ziarko [550], pp. 426–433.

[134] J. Jelonek, K. Krawiec, R. Słowinski, J. Stefanowski, J. Szymas (1994), *Rough sets as an intelligent front–end for the neural network.* In: Proceedings of the First National Conference on Neural Networks their Applications **2**, Częstochowa, Poland, pp. 116–122.

[135] J. Jelonek, K. Krawiec, R. Słowiński, J. Szymas (1995), *Rough set reduction of features for picture–based reasoning.* In: Lin and Wildberger [196], pp.89–92.

[136] T.-K. Jensen, J. Komorowski A. Øhrn (1998), *Improving Mollestad's algorithm for default knowledge discovery.* In: Polkowski and Skowron [346], pp. 373–380.

[137] G. John, R. Kohavi, K. Pfleger (1994), *Irrelevant features subset selection problem.* Proc. of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, pp. 121–129.

[138] J. Johnson (1998), *Rough mereology for industrial design.* In: Polkowski and Skowron [346], pp. 553–556.

[139] M. Kandulski, J. Marciniec, K. Tukałło (1992), *Surgical wound infection – Conductive factors and their mutual dependencies.* In: Słowiński [428], pp. 95–110.

[140] J.D. Katzberg, W. Ziarko W. (1996), *Variable precision extension of rough sets.* Fundamenta Informaticae **27**, pp. 155–168.

[141] W.H. Kautz (1968), *Fault testing and diagnosis in combinational digital circuits*, IEEE Transactions on Computers, April, pp. 352–366.

[142] K. Keiser, A. Szladow, W. Ziarko (1992), *Rough sets theory applied to a large multi-species toxicity database.* In: Proceedings of the Fifth International Workshop on QSAR in Environmental Toxicology, Duluth, Minnesota.

[143] R.E. Kent (1996), *Rough concept analysis: A synthesis of rough sets and formal concept analysis.* Fundamenta Informaticae **27/2–3**, pp. 169–181.

[144] R. Kerber (1992), *Chimerge. Discretization of numeric attributes.* Proc. of the Tenth National Conference on Artificial Intelligence, MIT Press, pp. 123–128.

[145] S. Kłopotek, S. Wierzchoń (1997), *Qualitative versus quantitative interpretation of the mathematical theory of evidence.* In: Ras and Skowron [363], pp. 391–400.

[146] S. Kobayashi, T, Yokomori (1997), *Approximately learning regular languages with respect to reversible languages: A rough set based analysis.* In: Wang [514], pp. 91–94.

[147] Y. Kodratoff, R. Michalski (1990), *Machine learning, An Artificial Intelligence approach* **3**, Morgan Kaufmann, 1990.

[148] J. Komorowski, Z.W. Ras (Eds.) (1993), *Proceedings of the Seventh International Symposium on Methodologies for Intelligent Systems (ISMIS'93)*, Trondheim, Norway, June 15–18, Lecture Notes in Computer Science **689**, Springer–Verlag, Berlin.

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

[150] B. Konikowska (1997), *A logic for reasoning about relative similarity.* Studia Logica **58/1**, pp. 185–226.

[151] B. Konikowska B. (1998), *A logic for reasoning about similarity.* In: Orłowska [276] pp. 461–490.

[152] B. Kostek (1995) *Computer based recognition of musical phrases using the rough set approach.* In: Wang [513], pp. 401–404.

[153] B. Kostek (1997) *Soft set approach to the subjective assessment of sound quality.* In: Proceedings of the Symposium on Computer Science and Engineering Cybernetics, August 18–23, Baden–Baden, Germany, The International Institute for Advanced Studies in Systems Research and Cybernetics (to appear).

[154] B. Kostek (1997), *Sound quality assessment based on the rough set classifier.* In: Zimmermann [558], pp. 193–195.

[155] B. Kostek (1998), *Soft computing–based recognition of musical sounds.* In: Polkowski and Skowron [345], pp. 193–213.

[156] B. Kostek (1998), *Assesment of concert hall acoustics using rough set fuzzy approach.* In this book.

[157] B. Kostek, A. Czyżewski (1996), *Automatic classification of musical timbres based on learning algorithms applicable to Cochlear implants.* In: Proceedings of IASTED International Conference – Artificial Intelligence, Expert Systems and Neural Networks, August 19–21, Honolulu, Hawaii, USA, pp. 98–101.

[158] B. Kostek, M. Szczerba (1996), *Parametric representation of musical phrases.* In: 101st Convention of the Audio Engineering Society, November 8–11, Los Angeles, California, USA, preprint **4337**.

[159] B. Kostek, M. Szczerba, A. Czyżewski (1997), *Rough set based analysis of computer musical storage.* In: Proceedings of the International Conference on Computing Intelligent Multimedia Applications (ICCIMA'97), February 10–13, Brisbane, Australia.

[160] W. Kowalczyk (1996), *Analyzing temporal patterns with rough sets.* In: Zimmermann [557], pp. 139–143.

[161] W. Kowalczyk (1996), *Analyzing signals with AI-techniques, two case studies.* A contribution to the International Competition for Signal Analysis Processing by Intelligent Techniques held during the Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT'96), September 2–5, 1996, Aachen, Germany; available by anonymous ftp from `ftp.cs.vu.nl`, the directory `/pub/wojtek`.

[162] W. Kowalczyk (1998), *Rough data modelling, A new technique for analyzing data.* In: Polkowski and Skowron [344], pp. 400–421.

[163] W. Kowalczyk (1998), *An empirical evaluation of the accuracy of rough data models.* In: Bouchon–Meunier and Yager [26], pp. 1534–1538.

[164] W. Kowalczyk, Z. Piasta (1998), *Rough sets–inspired approach to knowledge discovery in business databases.* In: The Second Pacific–Asian Conference on Knowledge Discovery Data Mining, (PAKDD'98), Melbourne, Australia, April 15–17.

[165] W. Kowalczyk, F. Slisser (1997), *Analyzing customer retention with rough data models.* In: Komorowski and Żytkow [149], pp. 4–13.

[166] K. Krawiec, K., R. Słowiński, and D. Vanderpooten (1996), *Construction of rough classifiers based on application of a similarity relation.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484] pp. 23–30.

[167] K. Krawiec, R. Słowiński, and D. Vanderpooten (1998), *Learning decision rules from similarity based rough approximations.* In: Polkowski and Skowron [345], pp. 37–54.

[168] M. Kretowski, J. Stepaniuk J. (1996), *Selection of objects and attributes, a tolerance rough set approach.* In: Proceedings of the Poster Session of Ninth International Symposium on Methodologies for Intelligent Systems, Zakopane Poland, June 10-13, pp. 169–180.

[169] E. Krusińska, R. Słowiński, J. Stefanowski (1992), *Discriminant versus rough set approach to vague data analysis*, Applied Stochastic Models and Data Analysis **8**, pp. 43–56.

[170] M. Krynicki, L. Szczerba (1998), On the logic with rough quantifier. In: Orłowska [276], pp. 601–613.

[171] J. Krysiński (1990), *Rough set approach to the analysis of structure–activity relationship of quaternary imidazolium compounds.* Arzneimittel Forschung / Drug Research **40/11**, pp. 795–799.

[172] J. Krysiński (1992), *Grob Mengen Theorie in der Analyse der Struktur Wirkungs Beziehungen von quartaren Pyridiniumverbindungen.* Pharmazie **46/12**, pp.878–881.

[173] J. Krysiński (1992), *Analysis of structure–activity relationships of quaternary ammonium compounds.* In: Słowiński [428], pp. 119–136.

[174] J. Krysiński (1995), *Application of the rough sets theory to the analysis of structure–activity relationships of antimicrobial pyridinium compounds.* Die Pharmazie **50**, pp. 593–597.

[175] J. Krysiński (1995), *Rough sets in the analysis of the structure–activity relationships of antifungal imidazolium compounds.* Journal of Pharmaceutical Sciences **84/2**, pp. 243–247.

[176] M. Kryszkiewicz (1995), *Rough set approach to incomplete information systems.* In: Wang [513], pp. 194–197.

[177] M. Kryszkiewicz (1997), *Generation of rules from incomplete information systems.* In: Komorowski and Żytkow [149], pp. 156–166.

[178] G. Lambert–Torres, R. Rossi, J. A. Jardini, A. P. Alves da Silva, V. H. Quintana (1998), *Power system security analysis based on rough classification.* In this book.

[179] A. Lenarcik, Z. Piasta (1992), *Discretization of condition attributes space* In: Słowiński [428], pp. 373–389.

[180] A. Lenarcik, Z. Piasta (1993), *Probabilistic approach to decision algorithm generation in the case of continuous condition attributes.* Foundations of Computing and Decision Sciences **18/3–4**, pp. 213–223.

[181] A. Lenarcik, Z. Piasta (1994), *Deterministic rough classifiers.* In: Lin [188]. pp. 434–441.

[182] A. Lenarcik, Z. Piasta (1997), *Probalilistic rough classifiers with mixture of discrete and continuous attributes*, In: Lin and Cercone [191] pp. 373–383.

[183] S. Leśniewski (1992), *Foundations of the general theory of sets*, In: Surma, Srzednicki, Barnett, Rickey (Eds.), Stanislaw Leśniewski Collected Works, Kluwer Academic Publishers, Dordrecht, pp. 128–173.

[184] T.Y. Lin (1989), *Neighborhood systems and approximation in database and knowledge base systems.* In: Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems.

[185] T.Y. Lin (1989), *Granular computing on binary relations I.* In: Polkowski and Skowron [344], pp. 107–121.

[186] T.Y. Lin (1989), *Granular computing on binary relations II.* In: Polkowski and Skowron [344], pp. 122–140.

[187] T.Y. Lin (1994), *Fuzzy reasoning and rough sets.* In: Ziarko [550], pp. 343–348.

[188] T.Y. Lin (Ed.) (1994), *Proceedings of the Third International Workshop on Rough Sets Soft Computing (RSSC'94).* San Jose State University, San Jose, California, USA, November 10–12.

[189] T.Y. Lin (1996), *Neighborhood systems – A qualitative theory for fuzzy and rough sets.* In: P.P. Wang (Ed.), Advances in Machine Intelligence and Soft Computing **4**, pp. 132–155.

[190] T.Y. Lin (1996), *An overview of rough set theory from the point of view of relational databases.* In: S. Tsumoto (Ed.), Bulletin of International Rough Set Society **1/1**, pp. 23–30.

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

[192] T.Y. Lin, R. Chen (1997), *Finding reducts in very large databases.* In: Wang [514], pp. 350–352.

[193] T.Y. Lin, J.C. Liau (1997), *Probabilistics multivalued random variables – Belief functions and granular computing.* In: Zimmermann [558], pp. 221–225.

[194] T.Y. Lin, J.C. Liau (1997), *Belief functions based on probabilistic multivalued random variables.* In: Wang [514], pp. 269–272.

[195] T.Y. Lin, Q. Liu (1996), *First–order rough logic I, Approximate reasoning via rough sets.* Fundamenta Informaticae **27**, pp. 137–153.

[196] T.Y. Lin, A.M. Wildberger (Eds.) (1995), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery.* Simulation Councils, Inc., San Diego, CA.

[197] P. Lingras (1996), *Rough neural networks.* In: Bouchon–Meunier, Delgado, Verdegay, Vila, and Yager [25] **2**, pp. 1445–1450.

[198] P. Lingras (1996), *Learning using rough Kohonen neural networks classifiers.* In: Borne, Dauphin–Tanguy, Sueur, and El Khattabi [24], pp. 753–757.

[199] P. Lingras (1997), *Comparison of neofuzzy rough neural networks.* In: Wang [514], pp. 259–262.

[200] W. Lipski (1981), *On databases with incomplete information.* Journal of the ACM **28**, pp. 41–70.

[201] T. Løken (1998), *Vibration analysis using Rosetta, A practical application of rough sets.* (Project in Technical Report), Knowledge Systems Group, The Norwegian University of Science Technology, Trondheim, Norway.

[202] E.H. Mamdani, S. Assilian (1975), *An experiment in linguistic synthesis with a fuzzy logic controller.* International Journal of Man - Machine Studies **7**,pp. 1–13.

[203] S. Marcus (1994), *Tolerance rough sets, Cech topologies, learning processes.* Bull. Polish Acad. Sci. Tech. Sci. **42/3**, pp. 471–487.

[204] W. Marek, Z. Pawlak (1976), *Information storage and retrieval: Mathematical foundations.* Theoretical Computer Science **1**, pp. 331-354.

[205] W. Marek, Z. Pawlak (1984), *Rough sets and information systems.* Fundamenta Informaticae **17**, pp. 105-115.

[206] M. Mares, R. Meisar, V. Novak, and J. Ramik (Eds.) (1997), *Proceedings of the Seventh International Fuzzy Systems Assotiation World Congress (IFSA'97).* June 25–29, Academia, Prague **1**, pp. 1–529.

[207] P. Maritz (1996), *Pawlak and topological rough sets in terms of multifunctions.* Glasnik Matematicki **31/51**, pp. 159–178.

[208] Z. Michalewicz (1992), *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag, Berlin.

[209] S. Miyamoto (1996), *Fuzzy multisets and application to rough approximation of fuzzy sets.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 255–260.

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

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

[212] M. Millan, F. Machuca (1997), *Using the rough set theory to exploit the data mining potential in relational databases systems.* In: Wang [514], pp. 344–347.

[213] S. Mitra, M. Banerjee (1996), *Knowledge–based neural net with rough sets.* In: T. Yamakawa et al. (Eds.), Methodologies for the Conception, Design, Application of Intelligent Systems, Proceedings of the Fourth International Conference on Soft Computing (IIZUKA'96), Iizuka, Japan 1996, World Scientific, pp. 213–216.

[214] M. Modrzejewski (1993), *Feature selection using rough sets theory.* In: P.B. Brazdil (Ed.), Proceedings of the European Conference on Machine Learning, pp. 213–226.

[215] T. Łuba, J. Rybnik (1992), *Rough sets some aspects of logical synthesis.* In: R. Słowiński [428], pp. 181–202.

[216] T. Mollestad (1997), *A Rough set approach to data mining, extracting a logic of default rules from data.* Ph.D. thesis, Norwegian University of Science and Technology.

[217] T. Mollestad J. Komorowski (1998), *A rough set framework for propositional default rules data mining.* In this book.

[218] T. Mollestad A. Skowron (1996), *A rough set framework for data mining of propositional default rules.* In: Pawlak and Ras [362], pp. 448–457. Full version available at `http://www.idi.ntnu.no`.

[219] H. Moradi, J.W. Grzymała–Busse, J. Roberts (1995), *Entropy of English text, Experiments with humans machine learning system based on rough sets.* In: Wang [513], pp. 87–88.

[220] A. Mrózek (1985), *Information systems and control algorithms.* Bull. Polish Acad. Sci. Tech. Sci. **33**, pp. 195–212.

[221] A. Mrózek (1992), *Rough sets in computer implementation of rule–based control of industrial processes.* In: Słowiński [428], pp. 19–31.

[222] A. Mrózek, L. Płonka (1993), *Rough sets in image analysis.* Foundations of Computing Decision Sciences **18/3–4**, pp. 259–273.

[223] A. Mrózek, L. Płonka (1994), *Knowledge representation in fuzzy rough controllers.* In: M. Dąbrowski, M. Michalewicz, Z.W. Ras (Eds.), Proceedings of the Third International Workshop on Intelligent Information Systems, Wigry, Poland, June 6–10. Institute of Computer Science, Polish Academy of Sciences, Warsaw, pp. 324–337.

[224] A. Mrózek, L. Płonka (1998), *Rough sets in industrial applications.* In: Polkowski and Skowron [345], pp. 214–237.

[225] A. Mrózek, K. Skabek (1998), *Rough sets in economic applications.* In: Polkowski and Skowron [345], pp. 238–271.

[226] A. Mrózek, L. Płonka, R. Winiarczyk, J. Majtan (1994), *Rough sets for controller synthesis.* In: Lin [188], pp. 498–505.

[227] A. Mrózek, L. Płonka, J. Kędziera (1996), *The methodology of rough controller synthesis.* In: In: Petry and Kraft [321], pp. 1135–1139.

[228] T. Munakata (1995), *Commercial industrial AI a future perspective on rough sets.* In: Lin and Wildberger [196], pp. 219–222.

[229] T. Munakata (1995), *Rough control, Basic ideas applications.* In: Wang [513], pp. 340–343.

[230] T. Munakata (1997), *Rough control, A perspective.* In: Lin and Cercone [191], pp. 77–88.

[231] Murthy, S., Aha, D. (1996), *UCI repository of machine learning data tables.* `http.//www/ics.uci.edu/~mlearn.`

[232] S. Murthy, S. Kasif, S. Saltzberg, R. Beigel,(1993), *OC1. Randomized induction of oblique decision trees.* Proc. of the Eleventh National Conference on AI, pp. 322–327.

[233] S. Murthy, S. Kasif, S. Saltzberg (1994). A system for induction of oblique decision trees. In. Proc. of Sixth International Machine Learning Workshop, Ithaca N.Y., Morgan Kaufmann.

[234] A. Nakamura (1993), *On a multi–modal logic based on the graded classifications.* In: [FCDS'93] pp. 275–292.

[235] A. Nakamura (1994), *Fuzzy quantifiers and rough quantifiers.* In: P.P. Wang (Ed.), Advances in Fuzzy Theory and Technology **II** pp. 111–131.

[236] A. Nakamura, *A rough logic based on incomplete information and its application.* International Journal of Approximate Reasoning **15/4**, pp. 367–378.

[237] A. Nakamura, *Conflict logic with degrees.* In this book.

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

[239] H.S. Nguyen(1998), *From optimal hyperplanes to optimal decision trees.* Fundamenta Informaticae **34**, 1998, pp. 145–174.

[240] S.H. Nguyen (1999), *Data regularity analysis and its applications in data mining.* Ph.D. Thesis, Warsaw University.

[241] S.H. Nguyen, H.S. Nguyen (1996), *Some efficient algorithms for rough set methods.* In: Bouchon–Meunier, Delgado, Verdegay, Vila, and Yager [25], pp. 1451–1456.

[242] H.S. Nguyen, S.H. Nguyen (1998), *Discretization methods in data mining.* In: Polkowski and Skowron [344], pp. 451–482 .

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

[244] H.S. Nguyen, S.H. Nguyen (1998), *Pattern extraction from data.* In: Bouchon–Meunier and Yager [26], pp. 1346–1353.

[245] H.S. Nguyen, A. Skowron (1995), *Quantization of real values attributes: Rough set and Boolean reasoning approaches.* In: Wang [513], pp. 34–37.

[246] H.S. Nguyen, S.H. Nguyen, A. Skowron (1996), *Searching for Features defined by Hyperplanes.* In: Raś and Michalewicz [362], pp. 366–375.

[247] H.S. Nguyen, A. Skowron (1996), *Quantization of real value attributes: Rough set and Boolean reasoning approach.* In: S. Tsumoto (Ed.), Bulletin of International Rough Set Society **1/1** (1996), pp. 5–16 also in Information Sciences (1998) (in print).

[248] H.S. Nguyen, A. Skowron (1997), *Boolean reasoning for feature extraction problems.* In: Ras and Skowron [363], pp. 117–126.

[249] H.S. Nguyen, A. Skowron (1999), *Boolean reasoning scheme with some applications in data mining*, In: Proceedings of the 3-rd European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'99), September 15–18, Prague, Lecture Notes in Artificial Intelligence (in print).

[250] S.H. Nguyen, T.T. Nguyen, A. Skowron, P. Synak (1996), *Knowledge discovery by rough set methods.* In: Nagib C. Callaos (Ed.), Proceedings of the International Conference on Information Systems Analysis and Synthesis (ISAS'96), July 22–26, Orlando, USA, pp. 26–33.

[251] S.H. Nguyen, L. Polkowski, A. Skowron, P. Synak, J. Wróblewski (1996), *Searching for approximate description of decision classes.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 153–161.

[252] S.H. Nguyen, A. Skowron (1997), *Searching for relational patterns in data.* In: Komorowski and Żytkow [149], pp. 265–276.

[253] S.H. Nguyen, A. Skowron, P. Synak (1998), *Discovery of data patterns with applications to decomposition and classification problems.* In: Polkowski and Skowron [345], pp. 55–97.

[254] H.S. Nguyen, M. Szczuka, D. Ślęzak (1997), *Neural network design, Rough set approach to real–valued data.* In: Komorowski and Żytkow [149], pp. 359–366.

[255] S.H. Nguyen, A. Skowron, P. Synak, J. Wróblewski (1997), *Knowledge discovery in data bases, Rough set approach.* In: Mares, Meisar, Novak, and Ramik, [206], pp. 204–209.

[256] T. Nguyen, R. Swiniarski, A. Skowron, J. Bazan, K. Thagarajan (1995), *Applications of rough sets, neural networks and maximum likelihood for texture classification based on singular value decomposition.* In: Lin and Wildberger [196], pp. 157–160.

[257] J. Nieminen (1988), *Rough tolerance equality.* Fundamenta Informaticae **11/3**, pp. 289–296.

[258] R. Nowicki, R. Słowiński, J. Stefanowski (1990), *Possibilities of applying the rough sets theory to technical diagnostics.* In: Proceedings of the Ninth National Symposium on Vibration Techniques Vibroacoustics, Kraków, December 12–14, AGH University Press, Kraków, pp. 149–152.

[259] R. Nowicki, R. Słowiński, J. Stefanowski (1992), *Rough sets analysis of diagnostic capacity of vibroacoustic symptoms.* Journal of Computers Mathematics with Applications **24**, pp.109–123.

[260] R. Nowicki, R. Słowiński, J. Stefanowski (1992), *Evaluation of vibroacoustic diagnostic symptoms by means of the rough sets theory.* Journal of Computers in Industry **20**, pp. 141–152.

[261] R. Nowicki, R. Słowiński, J. Stefanowski (1992), *Analysis of diagnostic symptoms in vibroacoustic diagnostics by means of the rough set theory.* In: Słowiński [428], pp. 33–48.

[262] M. Novotny (1998), *Dependence spaces of information systems.* In: Orłowska [276], pp. 193–246.

[263] M. Novotny (1998), *Applications of dependence spaces.* In: Orłowska [276], pp. 247–289.

[264] M. Novotny, Z. Pawlak (1991), *Algebraic theory of independence in information systems.* Fundamenta Informaticae **14**, pp. 454–476.

[265] M. Novotny, Z. Pawlak (1992), *On problem concerning dependence space.* Fundamenta Informaticae **16/3–4**, pp. 275–287.

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

[267] A. Obtułowicz (1988), *Rough sets and Heyting algebra valued sets.* Bull. Polish Acad. Sci. Math. **35/9–10**, pp. 667–671.

[268] E. Orłowska (1984), *Modal logics in the theory of information systems.* Zeitschrift für Mathematische Logic und Grundlagen der Mathematik, **10/3**, pp.213–222.

[269] E. Orłowska (1985), *A logic of indiscernibility relations.* In: A. Skowron (Ed.), Computation Theory, Lecture Notes in Computer Science **208**, pp. 177–186.

[270] E. Orłowska (1985), *Logic of nondeterministic information.* Studia Logica **44**, pp. 93–102.

[271] E. Orłowska (1988), *Kripke models with relative accessibility and their application to inferences from incomplete information.* In: G. Mirkowska, H. Rasiowa (Eds.), Mathematical Problems in Computation Theory, Banach Center Publications **21** pp. 329–339.

[272] E. Orłowska (1990), *Kripke semantics for knowledge representation logics.* Studia Logica **49** pp. 255–272.

[273] E. Orłowska (1990), *Verisimilitude based on concept analysis.* Studia Logica **49**, pp. 307–320.

[274] E. Orłowska (1995), *Information algebras.* Lecture Notes in Computer Science **936**, Springer–Verlag, Berlin, pp. 55–65.

[275] E. Orłowska (1998), *Introduction: What you always wanted to know about rough sets.* In: Orłowska [276] pp. 10–29.

[276] E. Orłowska (Ed.) (1998), *Incomplete Information, Rough Set Analysis.* Physica–Verlag, Heidelberg.

[277] E. Orłowska, M.W. Orłowski, *Maintenance of knowledge in dynamic information systems.* In: Słowiński [428], pp. 315–330.

[278] E. Orłowska, Z. Pawlak (1984), *Representation of nondeterministic information.* Theoretical Computer Science **29** pp. 27–39.

[279] A. Øhrn, S. Vinterbo, P. Szymański J. Komorowski (1997), *Modelling cardiac patient set residuals using rough sets.* Proc. AMIA Annual Fall Symposium (formerly SCAMC), Nashville, TN, USA, Oct. 25–29, pp. 203–207.

[280] A. Øhrn J. Komorowski (1997), ROSETTA – *A Rough Set Toolkit for Analysis of Data.* Proc. Third International Joint Conference on Information Sciences, Durham, NC, USA, March 1–5, **3**, pp. 403–407.

[281] A. Øhrn J. Komorowski (1998), *Analyzing The Prognostic Power of Cardiac Tests Using Rough Sets.* Proc. of the Invited Session on Intelligent Prognostic Methods in Medical Diagnosis and Treatment Planning at the Computational Engineering in Systems Applications conference – CESA'98, April Nabeul-Hammamet, Tunisia, 6 pages.

[282] A. Øhrn, J. Komorowski, A. Skowron P. Synak (1998), *The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets - The* ROSETTA *system.* In: Polkowski and Skowron [344], pp. 376–399.

[283] P. Pagliani (1993), *From concept lattices to approximation spaces, Algebraic structures of some spaces of partial objects.* Fundamenta Informaticae **18/1**, pp. 1–25.

[284] P. Pagliani (1998), *Rough set theory and logic-algebraic structures.* In: Orłowska [276] pp. 109–190.

[285] P. Pagliani (1998), *Modalizing relations by means of relations: A general framework for two basic approaches to knowledge discovery in databases.* In: Bouchon–Meunier and Yager [26], pp. 1175–1182.

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

[287] Y.-H. Pao, I. Bozma (1986), *Quantization of numerical sensor data for inductive learning.* In: J. S. Kowalik (Ed.), Coupling Symbolic and Numerical Computing in Expert System. Elsevier Science Publ., Amsterdam, pp. 69–81.

[288] P. Paszek, A. Wakulicz–Deja (1996), *Optimalization diagnose in progressive encephalopathy applying the rough set theory.* In: Zimmermann [557] **1**, pp. 192–196.

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

[290] G. Paun, L. Polkowski, A. Skowron (1997), *Rough set approximations of languages.* Fundamenta Informaticae **32/2**, pp. 149–162.

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

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

[293] Z. Pawlak (1984), *Rough probability.* Bull. Polish Acad. Sci. Math. **132/9–10**, pp. 607–612.

[294] Z. Pawlak (1984), *On conflicts.* Int. J. of Man-Machine Studies **21**, pp. 127-134.

[295] Z. Pawlak (1985), *Rough sets and fuzzy sets.* J. of Fuzzy Sets and Systems **17**, pp. 99–102.

[296] Z. Pawlak (1989), *Knowledge, reasoning and classification – A rough set perspective.* Bulletin of the European Association for Theoretical Computer Science (EATCS) **38**, pp. 199–210.

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

[298] Z. Pawlak (1993), *Anatomy of conficts.* Bulletin of the European Association for Theoretical Computer Science **50**, pp. 234-247.

[299] Z. Pawlak (1994), *Hard and soft sets.* In: Ziarko [550], pp. 130–135.

[300] Z. Pawlak (1994), *Decision analysis using rough sets.* International Trans. Opr. Res. **1/1**, pp. 107–114.

[301] Z. Pawlak (1995), *Rough sets and fuzzy sets.* In: C. Jinshong (Ed.), Proceedings of ACM, Computer Science Conference, February 28 – March 2, Nashville, Tennessee, pp. 262–264.

[302] Z. Pawlak (1997), *Conflict analysis.* In: Zimmermann [558], pp. 1589–1591.

[303] Z. Pawlak (1997), *Rough set approach to knowledge–based decision support.* European Journal of Operational Research **2933**, pp. 1–10.

[304] Z. Pawlak (1998), *Reasoning about data – A rough set perspective.* In: Polkowski and Skowron [346], pp. 25–34.

[305] Z. Pawlak, J.W. Grzymała–Busse, R. Słowiński, and W. Ziarko (1995), *Rough sets.* Communications of the ACM **38(11)**, pp. 89–95.

[306] Z. Pawlak, T. Munakata (1996), *Rough control, Application of rough set theory to control.* In: Zimmermann [557], pp. 209–218.

[307] Z. Pawlak, K. Słowiński, R. Słowiński (1986), *Rough classification of patients after highly selected vagotomy for duodenal ulcer.* Journal of Man–Machine Studies **24**, pp. 413–433.

[308] Z. Pawlak, R. Słowiński (1994), *Decision analysis using rough sets. International Transactions in Operational Research.* **1/1**, pp. 107–114.

[309] Z. Pawlak, R. Słowiński (1994), *Rough set approach to multi–attribute decision analysis.* European Journal of Operational Research **72**, pp. 443–459 (Invited Review).

[310] Z. Pawlak A. Skowron (1994), *Rough membership functions.* In: R. Yager, M. Fedrizzi J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, Wiley, New York, pp. 251–271.

[311] Z. Pawlak, A. Skowron,(Ed.)(1996), *Logic, algebra and computer science, Helena Rasiowa and Cecylia Rauszer in Memoriam.* Bulletin of the Section of Logic **25/3–4**, pp. 174–184.

[312] Z. Pawlak, S.K.M. Wong, W. Ziarko (1988), *Rough sets: Probabilistic versus deterministic approach.* International Journal of Man–Machine Studies **29**, pp. 81–95.

[313] W. Pedrycz (1998), *Computational Intelligence: An Introduction.* CRC Press, Boca Ratou.

[314] W. Pedrycz (1998), *Shadowed sets: Bridging fuzzy and rough sets.* In this book.

[315] J. F. Peters III, S. Ramanna (1998), *Software deployability decision system framework, A rough set approach.* In: Bouchon–Meunier and Yager [26], pp. 1539–1545.

[316] J. F. Peters, A. Skowron, Z. Suraj, S. Ramanna, A. Paryzek (1998), *Modelling real – time decision – making systems with rough fuzzy Petri nets.* In Proceedings of EUFIT-98, Aachen (to appear).

[317] J. F. Peters, K. Ziaei, S. Ramanna (1998), *Approximate time rough control, Concepts applications to satelite attitude control.* In: Polkowski and Skowron [346], pp. 491–498.

[318] J. F. Peters III, S. Ramanna (1998), *A rough set approach to assessing software quality: Concepts and rough Petri net models.* In this book.

[319] J. F. Peters III (1998), *Time clock information systems: Concepts and roughly fuzzy Petri net models.* In: Polkowski and Skowron [345], pp. 387–419.

[320] G. I. Peterson (1994), *Rough classification of pneumonia patients using a clinical data–base.* In: Ziarko [550], pp. 412–419.

[321] F.E. Petry, D.H. Kraft (1996), Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96). September 8–11, New Orleans, Louisiana, pp. 1–2214.

[322] Z. Piasta (1993), *Statistical and logical classifiers: A comparative study.* In: W. Ziarko (Ed.): Proceedings of the Second International Workshop on Rough Sets and Knowledge Discovery (RSKD'93). Banff, Alberta, Canada, October 12–15.

[323] Z. Piasta (1996), *Rough classifiers in intelligent support of business decisions.* In: Proceedings of the First Polish Conference on Theory Applications of Artificial Intelligence (CAI'96), Łódź, Poland, pp. 103–111.

[324] Z. Piasta (1998), *Transforming data into engineering knowledge with rough classifiers.* In: A.M. Brandt (Ed.), Optimization Methods for Material Design of Cement–based Composites, Thomson Science & Professional, London (to appear).

[325] Z. Piasta (1998), *Data mining knowledge discovery in marketing financial databases with rough classifiers.* Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław (to appear, in Polish).

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

[327] Z. Piasta, A. Lenarcik (1998), *Learning rough classifiers from large databases with missing values.* In: Polkowski and Skowron [345], pp. 483–499.

[328] Z. Piasta, A. Lenarcik, S. Tsumoto (1997), *Machine discovery in databases with probabilistic rough classifiers.* In: S. Tsumoto (Ed.): Bulletin of International Rough Set Society **1/2**, pp. 51–57.

[329] L. Płonka, A. Mrózek (1995), *Rule–based stabilization of the inverted pendulum.* Computational Intelligence: An International Journal **11/2**, pp.348–356.

[330] L. Polkowski (1993), *Mathematical morphology of rough sets.* Bull. Polish Acad. Sci. Math. **41/3**, pp. 241–273.

[331] L. Polkowski (1994), *Concerning mathematical morphology of almost rough sets.* Bull. Polish Acad. Sci. Tech. **42/1**, pp. 141–152.

[332] L. Polkowski (1994), *Concerning mathematical morphology of rough sets.* Bull. Polish Acad. Sci. Tech. **42/1**, pp. 125–140.

[333] L. Polkowski (1998), *Rough set approach to mathematical morphology, approximate compression of data.* In: Bouchon–Meunier and Yager [26], pp. 1183–1189.

[334] L. Polkowski (1998), *Approximate mathematical morphology, rough set approach.* In this book.

[335] L. Polkowski, A. Skowron (1994), *Rough mereology.* In: Proceedings of the Symposium on Methodologies for Intelligent Systems, Charlotte, NC, October 16–19, Lecture Notes in Artificial Intelligence **869**, Springer–Verlag, Berlin (1994) pp. 85–94.

[336] L. Polkowski, A. Skowron (1995), *Rough mereology analytical mereology, New developments in rough set theory.* In: M. De Glas, Z. Pawlak (Eds.), Proceedings of the Second World Conference on Fundamentals of Artificial Intelligence (WOCFAI'95), July 13–17, Angkor, Paris, pp. 343–354.

[337] L. Polkowski, A. Skowron (1996), *Rough mereological approach to knowledge – based distributed AI.* In: J.K. Lee, J. Liebowitz, and J.M. Chae (Eds.), Critical Technology, Proceedings of the Third World Congress on Expert Systems, February 5–9, Seoul, Korea, Cognizant Communication Corporation, New York, pp. 774–781.

[338] L. Polkowski, A. Skowron (1996), *Implementing fuzzy containment via rough inclusions: Rough mereological approach to distributed problem solving.* In: Petry and Kraft [321], pp. 1147–1153.

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

[340] L. Polkowski, A. Skowron (1996), *Adaptive decision-making by systems of cooperative intelligent agents organized on rough mereological principles.* Journal of the Intelligent Automation and Soft Computing **2/2**, pp. 121-132.

[341] L. Polkowski, A. Skowron (1998), *Rough mereology analytical morphology.* In: Orłowska [276], pp. 399–437.

[342] L. Polkowski, A. Skowron (1998), *Towards adaptive calculus of granules.* In: Proceedings of the FUZZ-IEEE'98 International Conference, Anchorage, Alaska, USA, May 5–9, pp. 111–116.

[343] L. Polkowski, A. Skowron (1998), *Rough sets: A perspective.* In: Polkowski and Skowron [344], pp. 31–58.

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

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

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

[347] L. Polkowski, A. Skowron, J. Komorowski (1997), *Towards a rough mereology-based logic for approximate solution synthesis, Part 1. Studia Logica* **58/1**, pp. 143-184.

[348] L. Polkowski, A. Skowron, and J. Żytkow (1995), *Tolerance based rough sets.* In: Lin and Wildberger [196], pp. 55–58.

[349] J.A. Pomykała (1987), *Approximation operations in approximation space.* Bull. Polish Acad.Sci.Ser. Sci. Math. **35**, pp. 653–662.

[350] J.A. Pomykała (1988), *On definability in the nondeterministic information system*, Bull. Polish Acad. Sci.Ser. Sci. Math., **36**, pp. 193–210.

[351] M. Quafafou (1996), *Towards a transition from the crisp rough set theory to a fuzzy one.* In: Ras and Michalewicz [362], pp. 67–80.

[352] M. Quafafou (1997), *α-RST: A generalization of rough set theory.* In: Wang [514], pp. 173–176.

[353] E. Martienne, M. Quafafou (1998), *Vahueness and data reduction in concept learning.* In: E. Prade (Ed.), Proc. of the 13th European Conference on Arificial Intelligence (ECAI'98), August 23–28, Brighton, UK, Wiley, Chichester, pp. 351–355.

[354] J.R. Quinlan (1986), *Induction of decision trees.* In: Machine Learning **1**, pp. 81–106.

[355] J.R. Quinlan (1993), *C4.5. Programs for Machine Learning.* Morgan Kaufmann, San Mateo CA.

[356] R.B. Rao, S.C.-Y. Lu (1993), *Building models to support synthesis in early stage product design.* In: Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93), MIT Press, Cambridge, MA, pp. 277–282.

[357] Z.W. Ras (1996), *Cooperative knowledge–based systems.* Journal of the Intelligent Automation and Soft Computing **2/2**, pp. 193–202.

[358] Z.W. Ras (1997), *Collaboration control in distributed knowledge–based systems.* Information Sciences **96/3–4**, pp. 193–205.

[359] Z.W. Ras (1998), *Answering non-standard queries in distributed knowledge–based systems.* In: Polkowski and Skowron [345], pp. 98–108.

[360] Z.W. Ras, S. Joshi (1997), *Query answering system for an incomplete DKBS.* Fundamenta Informaticae **30/3–4**, pp. 313–324.

[361] Z.W. Ras, A. Kudelska, N. Chilumula (1995), *Can we simplify international physical performance test profile using rough set approach?* In: Wang [513], pp. 393–396.

[362] Z.W. Ras, M. Michalewicz, (Eds.), (1996), *Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems, Foundations of Intelligent Systems (ISMIS'96)*, October 15-18, Charlotte, NC, USA, Lecture Notes in Artificial Intelligence **1079**, Springer-Verlag, Berlin, pp. 1–664.

[363] Z.W. Ras, A. Skowron (Eds.) (1997), *Proceedings of the Tenth International Symposium on Methodologies for Intelligent Systems, Foundations of Intelligent Systems (ISMIS'97)*, October 15-18, Charlotte, NC, USA, Lecture Notes in Artificial Intelligence **1325**, Springer-Verlag, Berlin, pp. 1–630.

[364] H. Rasiowa (1986), *Rough concepts and multiple–valued logic.* In: Proceedings of the 16th ISMVL'86, Blacksburg, VA, IEEE Computer Society Press, pp. 282–288.

[365] H. Rasiowa (1987), *Logic approximating sequences of sets.* In: Proceedings of Advanced Intern. School and Symp. on Math. Logic and its Applications, honorably dedicated to 80th anniversary of Kurt Gödel, Plenum Press, New York, pp. 167–186.

[366] H. Rasiowa (1987), *An algebraic approach to some approximate reasonings.* In: Proceedings of the 17th ISMVL'87, Boston, MA, May 24–26, IEEE Computer Society Press, pp. 342–347.

[367] H. Rasiowa (1990), *On approximation logics, A survey.* Jahrbuch 1990 Kurt Gödel Gesellschaft, Vienna, pp. 63–87.

[368] H. Rasiowa (1991), *Mechanical proof systems for logic of reaching consensus by groups of intelligent agents.* Intern. Journ. of Approximate Reasoning **5/4** pp. 415–432.

[369] H. Rasiowa, G. Epstein (1987), *Approximation reasoning and Scott's information systems.* In: Z. Ras, M. Zemankova (Eds.), Proceedings of the Second International Symposium on Methodologies for Intelligent Systems, Charlotte, N.C., October 14–17, North Holland, Amsterdam pp. 33–42.

[370] H. Rasiowa, W. Marek (1987), *Gradual approximating sets by means of equivalence relations.* Bull. Polish Acad. Sci. Math. **35/3–4**, pp. 233–238.

[371] H. Rasiowa, W. Marek (1987), *Approximating sets with equivalence relations.* Theoretical Computer Science **48**, pp. 145–152.

[372] H. Rasiowa, W. Marek (1989), *On reaching consensus by groups of intelligent agents.* In: Z.W. Ras (Ed.), Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems (ISMIS'89), North Holland, Amsterdam, pp. 234–243.

[373] H. Rasiowa, W. Marek (1992), *Mechanical proof systems for logic II: Consensus programs and their processing.* Journal of Intelligent Information Systems **2/2**, pp. 149–164.

[374] H. Rasiowa (1985), A. Skowron, *Approximation logic.* In: Proceedings of Mathematical Methods of Specification and Synthesis of Software Systems Conference, Akademie Verlag **31**, Berlin pp. 123–139.

[375] C. Rauszer (1985), *Remarks on logic for dependencies.* Bull. Polish Acad. Sci. Math. **33**, pp. 249–252.

[376] C. Rauszer (1985), *Algebraic properties of functional dependencies.* Bull. Polish Acad. Sci. Math. **33**, pp. 561–569.

[377] C. Rauszer (1993), *Dependencies in relational databases. Algebraic and logical approach.* Fundamenta Informaticae **19**, pp. 235–274.

[378] C. Rauszer (1993), *An equivalence between theory of functional dependencies and a fragment of intuitionistic logic.* Bull. Polish Acad. Sci. Math. **33**, pp. 571–579.

[379] C. Rauszer (1993), *Communication systems in distributed information systems.* In: Proceedings of the International Workshop on Intelligent Information Systems, June 7–11, 1993, Augustów, Poland, Institute of Computer Science, Polish Academy of Sciences, Warsaw, pp. 15–29.

[380] C. Rauszer, (1993) *Approximation methods for knowledge representation systems.* In: J. Komorowski, Z.W. Ras (Eds.), Proceedings of the Seventh International Symposium on Methodologies for Intelligent Systems (ISMIS'93), Trondheim, Norway, June 15–18, 1993, Lecture Notes in Computer Science **689** (1993) pp. 326–337.

[381] C. Rauszer (1994), *Knowledge representation systems for groups of agents.* In: J. Woleński (Ed.), Philosophical Logic in Poland, Kluwer Academic Publishers, Dordrecht, pp. 217–238.

[382] C. Rauszer, W. Marek (1988), *Query optimization in the database distributed by means of product of equivalence relations.* Fundamenta Informaticae **11**, pp. 241–286.

[383] C. Rauszer, H. de Swart(1995), *Different approaches to knowledge, common knowledge and Aumann's theorem.* In: A. Laux, H. Wansing (Eds.), Knowledge and Belief in Philosophy and Artificial Intelligence, Akademie Verlag, Berlin, pp. 87–12.

[384] A. Reinhard, B. Stawski, T. Weber, U. Wybraniec–Skardowska (1992), *An application of rough set theory in the control conditions on a polder.* In: Słowiński [428], pp. 331–362.

[385] R.O. Rogriguez, P. Garcia, L. Godo (1995), *Similarity based models: Counterfactual and belif changes.* Research Report IIIA-95/5, IIIA-CSIS, University of Barcelona, Ballaterra, Spain.

[386] B. Roy (1985), *Méthodologie Multicritère d'Aide à la Décision.* Economica, Paris.

[387] B. Roy (1989), *Main sources of inaccurate determination, uncertainty and imprecision in decision models.* Mathematical and Computer Modelling **12**, pp. 1245–1254.

[388] B. Roy, D. Bouyssou (1993), *Aide Multicritère à la Décision: Méthods et Cas.* Economica, Paris.

[389] B. Roy (1993), *Decision sicence or decision aid science.* European Journal of Operational Research **86/2**, pp. 184–203.

[390] B. Roy, R. Słowiński, W. Treichel (1992), *Multicriteria programming of water supply systems for rural areas.* Water Resources Bulletin **28/1**, pp. 13–31.

[391] S. Rubin, S. Vanderpooten, W. Michałowski, R. Słowiński (1996), *Developing an emergency room for diagnostic check list using rough sets – A case study of appendicitis.* In: J. Anderson, M. Katzper (Eds.), Simulation in the Medical Sciences, Proceedings of the Western Conference of the Society for Computer Simulation, Simulation Councils. Inc. San Diego, CA, pp. 19–24.

[392] G. Ruhe (1996), *Qualitative analysis of software engineering data using rough sets.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 292–299.

[393] G. Ruhe (1996), *Knowledge discovery from software engineering data: Rough set analysis its interaction with goal–oriented measurement.* In: Komorowski and Żytkow [149], pp. 167–177.

[394] E.H. Ruspini (1991), *On truth, utility, and similarity.* Proceedings of the International Fuzzy Engineering Sympsoim (IFES'91), Yokohama, Japan, November 13–15, Fuzzy Engineering Towards Human Friendly Systems **1**, pp. 42–50.

[395] E.H. Ruspini (1990), *Similarity–based interpretations of fuzzy logic concepts.* Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks (IIZUKA'90) Iizuka, Japan, July 20–24, **2**, pp.735–780.

[396] M. Sarkar, B. Yegnanarayana (1998), *Fuzzy–rough sets and fuzzy integrals in modular neural networks.* In this book.

[397] J.A. Schreider (1975), *Equality, Resemblance and Order.* Mir Publishers, Moscow.

[398] B. Selman, H. Kautz and D. McAllester (1997), *Ten challenges in propositional reasoning and search*, Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97), Nagoya, Achi, Japan.

[399] M. Semeniuk–Polkowska (1996), *Rough sets in librarian science (in Polish).* Chair of Formal Linguistics, Warsaw University, pp. 1–109.

[400] G. Shafer (1976), *A Mathematical Theory of Evidence.* Princeton University Press, Princeton.

[401] N. Shan, W. Ziarko, H. Hamilton, N. Cercone (1995), *Using rough sets as tools for knowledge discovery.* In: Fayyad and Uthurusamy [93], pp. 263–268.

[402] E. Simoudis, J. Han, U. Fayyad (Eds.) (1996), *Proceedings of the Second International Conference on Knowledge Discovery Data Mining (KDD'96).* August 2–4, Portland, Oregon, USA, AAAI Press, Menlo Park, pp. 1–391.

[403] A. Skowron (1995), *Synthesis of adaptive decision systems from experimental data.* In: Aamodt and Komorowski [1], pp. 220–238.

[404] A. Skowron, J. Grzymała-Busse) (1994), *From rough set theory to evidence theory.* In: R.R. Yager, M. Fedrizzi, and J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, John Wiley and Sons, New York, pp. 193-236.

[405] A. Skowron, L. Polkowski (1996), *Analytical morphology: mathematical morphology of decision tables.* Fundamenta Informaticae **27/2–3**, pp.255–271.

[406] A. Skowron, L. Polkowski (1997), *Synthesis of Decision Systems from Data Tables.* In: Lin and Cercone [191], pp. 259–300.

[407] A. Skowron, L. Polkowski (1998), *Rough mereological foundations for design, analysis, synthesis, and control in distributive systems.* Information Sciences **104/1-2**, pp. 129–156.

[408] A. Skowron, L. Polkowski, J. Komorowski (1996), *Learning Tolerance Relations by Boolean Descriptors, Automatic Feature Extraction from Data Tables.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 11–17.

[409] A. Skowron, C. Rauszer (1992), *The Discernibility Matrices and Functions in Information Systems.* In: Słowiński [428], pp. 331–362.

[410] A. Skowron, J. Stepaniuk (1991), *Towards an approximation theory of discrete problems.* Fundamenta Informaticae **15/2** pp. 187–208.

[411] A. Skowron, J. Stepaniuk (1995), *Generalized approximation spaces.* In: Lin and Wildberger [196], pp. 18–21.

[412] A. Skowron, J. Stepaniuk (1996), *Tolerance approximation spaces.* Fundamenta Informaticae **27**, pp. 245–253.

[413] A. Skowron, J. Stepaniuk (1998), *Information granules and approximation spaces.* In: Bouchon–Meunier and Yager [26], pp. 1354–1361.

[414] A. Skowron, Z. Suraj (1993), *Rough sets and concurrency.* Bull. Polish Acad. Sci. Tech. **41/3**, pp. 237–254

[415] A. Skowron, Z. Suraj (1995), *Discovery of concurrent data models from experimental data tables, A rough set approach.* In: Fayyad and Uthurusamy [93], pp. 288–293.

[416] A. Skowron, Z. Suraj (1996), *A parallel algorithm for real–time decision making, A rough set approach.* Journal of Intelligent Information Systems **7**, pp. 5–28.

[417] D. Ślęzak (1996), *Approximate reducts in decision tables.* In: Bouchon–Meunier, Delgado, Verdegay, Vila, and Yager [25] **3**, pp. 1159–1164.

[418] D. Ślęzak (1997), *Attribute set decomposition of decision tables.* In: Zimmermann [558] **1**, pp. 236–240.

[419] D. Ślęzak (1998), *Decomposition and synthesis of decision tables with respect to generalized decision functions.* In this book.

[420] D. Ślęzak (1998), *Searching for dynamic reducts in inconsistent ecision tables.* In: Bouchon–Meunier and Yager [26], pp. 1362–1369.

[421] D. Ślęzak, M. Szczuka (1997), *Hyperplane–based neural networks for real–valued decision tables.* In: Wang [514], pp. 265–268.

[422] K. Słowiński, E.S. Sharif (1993), *Rough sets approach to analysis of data of diagnostic peritoneal lavage applied for multiple injuries patients.* In: Ziarko [550], pp. 420–425.

[423] K. Słowiński, R. Słowiński, J. Stefanowski (1988), *Rough sets approach to analysis of data from peritoneal lavage in acute pancreatitis.* Medical Informatics **13/3**, pp. 143–159.

[424] K. Słowiński, J. Stefanowski (1996), *On limitations of using rough set approach to analyze non-trivial medical information systems.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 176–184.

[425] K. Słowiński, J. Stefanowski (1998), *Multistage rough set analysis of therapeutic experience with acute pancreatitis.* In this book.

[426] K. Słowiński, J. Stefanowski, A. Antczak, Z. Kwas (1995), *Rough sets approach to the verification of indications for treatment of urinary stones by extracorporeal shock wave lithotripsy (ESWL).* In: Lin and Wildberger [196], pp. 93–96.

[427] K. Słowiński, J. Stefanowski, W. Twardosz (1997), *Rough set theory rule induction techniques for discovery of attribute dependencies in experience with multiple injured patients.* Institute of Computer Science, Warsaw University of Technology, ICS Research Report **6/97**.

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

[429] K. Słowiński (1992), *Rough classification of HSV patients.* In: Słowiński [428], pp. 77–93.

[430] R. Słowiński (1993), *Rough set learning of preferential attitude in multi–criteria decision making.* In: Komorowski and Ras [148], pp. 642–651.

[431] R. Słowiński (1994), *Handling various types of uncertainty in the rough set approach.* In: Ziarko [550], pp. 366–376.

[432] R. Słowiński (1994), *Rough set analysis of multi-attribute decision problems.* In: Ziarko [550], pp. 136–143.

[433] R. Słowiński (1995), *Rough set approach to decision analysis.* AI Expert **10**, pp. 18–25.

[434] R. Słowiński (1995), *Rough set theory and its applications to decision aid.* Belgian Journal of Operation Research, Francoro **35/3-4**, pp. 81–90.

[435] R. Słowiński, K. Słowiński (1989), *An expert system for treatment of duodenal ulcer by highly selective vagotomy (in Polish).* Pamiętnik 54. Jubil. Zjazdu Towarzystwa Chirurgów Polskich, Kraków **I**, pp. 223–228.

[436] R. Słowiński, D. Vanderpooten (1995), *Similarity relation as a basis for rough approximations.* In: P. Wang (Ed.): Advances in Machine Intelligence & Soft Computing, Bookwrights, Raleigh NC (1997) pp. 17–33.

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

[438] R. Słowiński, C. Zopounidis (1995), *Applications of the rough set approach to evaluation of bankruptcy risk.* International J. Intelligent Systems in Accounting, Finance & Management, **4/1**, pp. 27–41.

[439] R. Słowiński, C. Zopounidis (1994), *Rough set sorting of firms according to bankruptcy risk.* In: M. Paruccini (Ed.), Applying Multiple Criteria Aid for Decision to Environmental Management, Kluwer Academic Publishers, Dordrecht, pp. 339–357.

[440] P. Srinivasan (1989), *Approximations for information retrieval.* In: Z. W. Ras, (Ed.), Methodologies for Intelligent Systems **4**, Elsevier, pp. 128–136.

[441] P. Srinivasan (1989), *Intelligent information retrieval using rough set approximations.* Information Processing and Management **25**, pp. 347–361.

[442] P. Srinivasan (1991), *The importance of rough approximations for information retrieval.* Journal of Man–Machine Studies **34**, pp. 657-671.

[443] D. Sriram, R. Logcher, S. Fukuda (1991), *Computer - aided cooperative product development.* Lecture Notes in Computer Sience **492**, Springer–Verlag, Berlin.

[444] J. Stefanowski (1998), *On rough set based approaches to induction of decision rules.* In: Polkowski and Skowron [344], pp. 500–529.

[445] J. Stefanowski, K. Słowiński, R. Słowiński (1991), *Supporting of therapeutic decisions based on the rough sets theory.* In: Proceedings of the 16th Meeting of the EURO Working Group, Operational Research Applied to Health Services, the First Polish National Conference on Operational Research Applied to Health Systems, Książ 30.07–4.08, Wydawnictwo Politechniki Wrocławskiej, Wrocław, pp. 249–255.

[446] J. Stefanowski, R. Słowiński, R. Nowicki (1992), *The rough sets approach to knowledge analysis for classification support in technical diagnostics of mechanical objects.* In: F. Belli, F.J. Radermacher (Eds.), Industrial & Engineering Applications of Artificial Intelligence Expert Systems, Lecture Notes in Economics Mathematical Systems **604**, Springer-Verlag, Berlin, pp. 324–334.

[447] J. Stepaniuk (1996), *Similarity based rough sets and learning.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 18–22.

[448] J. Stepaniuk (1998), *Approximation spaces, reducts and representatives.* In: Polkowski and Skowron [345], pp. 109–126.

[449] J. Stepaniuk, J. Tyszkiewicz (1991), *Probabilistic properties of approximation problems.* Bull. Polish Acad. Sci. Tech. **39/3**, pp. 535–555.

[450] Z. Suraj (1996), *An application of rough set methods to cooperative information systems re–engineering.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 364–371.

[451] Z. Suraj (1996), *Discovery of concurrent data models from experimental tables, A rough set approach.* Fundamenta Informaticae **28/3–4**, pp. 353–376.

[452] Z. Suraj (1997), *Reconstruction of cooperative information systems under cost constraints, A rough set approach.* In: Wang [514], pp. 399–402.

[453] Z. Suraj (1998), *The synthesis problem of concurrent systems specified by dynamic information systems.* In: Polkowski and Skowron [345], pp. 420–450.

[454] R. Swiniarski (1993), *Zernike moments: Their application for image recognition.* Internal report of San Diego State University, Department of Mathematical Computer Sciences, USA.

[455] R. Swiniarski (1996), *Rough set expert system for on–line prediction of volleyball game progress for US olympic team.* In: B.D. Czejdo, I.I. Est, B. Shirazi, B. Trousse (Eds.), Proceedings of the Third Biennial European Joint Conference on Engineering Systems Design Analysis, July 1–4, Montpellier, France, pp. 15–20.

[456] R. Swiniarski (1996), *Rough sets expert system for robust texture classification based on 2D fast Fourier transormation spectral features.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 419–425.

[457] R. Swiniarski (1998), *Rough sets and principal component analysis and their applications in data model building and classification.* In this book.

[458] R. Swiniarski (1997), *Design of nonlinear texture data model using localized principal components and rough sets. Application to texture classification.* In: Proceedings of International Symposium on Nonlinear Theory its Applications, Hawaii, USA, November 29 – December 3 (accepted).

[459] R. Swiniarski (1998), *Intelligent feature extraction, Rough sets Zernike moments for data preprocessing feature extraction in handwritten digits recognition.* In: Proceedings of International Symposium on Engineering of Intelligent Systems, EIS98. University of Laguna, Tenerife, Spain, February 11–13.

[460] R. Swiniarski (1998), *Texture recognition based on rough sets 2D FFT feature extraction.* In: Proceedings of World Automation Congress. Anchorage, Alaska, USA, May 9–14.

[461] R. Swiniarski. (1998), *Rough sets Bayesian methods applied to cancer detection.* In: Polkowski and Skowron [346], pp. 609–616.

[462] R. Swiniarski (1998), *Rough sets and neural networks application to handwritten character recognition by complex Zernike moments.* In: Polkowski and Skowron [346], pp. 617-624.

[463] R. Swiniarski (1998), *Texture feature extraction, reduction and recognition based on rough sets.* In: Symposium on Object Recognition Scene Classification from Multispectral Multisensor Pixels. Columbus, Ohio, USA, July 6–10.

[464] R. Swiniarski, A. Berzins (1996), *Rough sets for intelligent data mining, knowledge discovering and designing of an expert systems for on–line prediction of volleyball game progress.* In: Tsumoto, Kobayashi, Yokomori, Tanaka, and Nakamura [484], pp. 413–418.

[465] R. Swiniarski, F. Hunt, D. Chalvet, D. Pearson (1997), *Feature selection using rough sets and hidden layer expansion for rupture prediction in a highly automated production system.* In: Proceedings of the 12th International Conference on Systems Science, September 12–15, Wroclaw, Poland; see also: Systems Science **23/1**.

[466] R. Swiniarski, F. Hunt, D. Chalvet, D. Pearson (1995), *Intelligent data processing and dynamic process discovery using rough sets, statistical reasoning and neural networks in a highly automated production systems.* In: Proceedings of the First European Conference on Application of Neural Networks in Industry, August, Helsinki, Finland.

[467] R. Swiniarski, J. Nguyen (1996), *Rough sets expert system for texture classification based on 2D spectral features.* In: B.D. Czejdo, I.I. Est, B. Shirazi, B. Trousse (Eds.), Proceedings of the Third Biennial European Joint Conference on Engineering Systems Design Analysis, July 1-4, Montpellier, France, pp. 3–8.

[468] R. Swiniarski, H. Zeng (1998), *A new halftoning method based on error diffusion with rough set filtering.* In: Polkowski and Skowron [345], pp. 336–346.

[469] M. Szczuka (1996), *Rough set methods for constructing neural network.* In: Proceedings of the Third Biennal Joint Conference On Engineering Systems Design Analysis, Session on Expert Systems, Montpellier, France, pp. 9–14.

[470] M. Szczuka (1998), *Refining decision classes with neural networks.* In: Bouchon–Meunier and Yager [26], pp. 370–1375.

[471] M. Szczuka (1998), *Rough sets and artificial neural networks.* In: Polkowski and Skowron [345], pp. 451–471.

[472] A.J. Szladow, W. Ziarko (1992), *Knowledge–based process control using rough sets.* In: Słowiński [428], pp. 49–60.

[473] H. Tanaka, Y. Maeda (1998), *Reduction methods for medical data.* In: Polkowski and Skowron [345], pp. 295–306.

[474] J. Teghem, J.-M. Charlet (1992), *Use of 'rough sets' method to draw premonitory factors for earthquakes by emphasing gas geochemistry: The case of a low seismic activity context in Belgium.* In: Słowiński [428], pp. 165–180.

[475] I. Tentush (1995), *On minimal absorbent sets for some types of tolerance relations.* Bull. Polish Acad. Sci. Tech. **43/1** pp. 79–88.

[476] S. Tsumoto (1997), *Domain experts' interpretation of rules induced from clinical databases.* In: In: Zimmermann [558] **1**, pp. 1639–1642.

[477] S. Tsumoto (1997), *Extraction of expert's decision process from clinical databases using rough set model.* In: Komorowski and Żytkow [149], pp. 58–67.

[478] S. Tsumoto (1997), *Induction of positive negative deterministic rules based on rough set model.* In: Ras and Skowron [363], pp. 298–307.

[479] S. Tsumoto (1997), *Empirical induction on medical expert system rules based on rough set model.* Ph.D. dissertation (in Japanese).

[480] S. Tsumoto (1998), *Modelling diagnostic rules based on rough sets.* In: Polkowski and Skowron [346], pp. 475–482.

[481] S. Tsumoto (1998), *Formalization induction of medical expert system rules based on rough set theory.* In: Polkowski and Skowron [345], pp. 307–323.

[482] S. Tsumoto (Ed.) (1998), *Bulletin of International Rough Set Society* **2/1**.

[483] S. Tsumoto (1998), *Induction of expert decision rules using rough sets and set-inclusion.* In this book.

[484] S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka, and A. Nakamura (Eds.) (1996), *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96).* The University of Tokyo, November 6–8.

[485] S. Tsumoto, H. Tanaka (1998), *PRIMEROSE, Probabilistic rule induction method based on rough set theory.* In: Ziarko [550], pp. 274–281 .

[486] S. Tsumoto, H. Tanaka (1994), *Induction of medical expert system rules based on rough sets resampling methods.* In: Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care, Journal of the AMIA **1** (supplement), pp. 1066–1070.

[487] S. Tsumoto, H. Tanaka (1995), *PRIMEROSE, Probabilistic rule induction method based on rough set resampling methods.* In: Computational Intelligence: An International Journal **11/2**, pp. 389–405.

[488] S. Tsumoto, H. Tanaka (1995), *Automated selection of rule induction methods based on recursive iteration of resampling.* In: Fayyad and Uthurusamy [93], pp. 312–317.

[489] S. Tsumoto, H. Tanaka (1995), *Automated discovery of functional components of proteins from amino-acid sequences based on rough sets and change of representation.* In: Fayyad and Uthurusamy [93], pp. 318–324.

[490] S. Tsumoto, H. Tanaka (1996), *Automated discovery of medical expert system rules from clinical databases based on rough sets.* In: Simoudis, Han, and Fayyad [402], pp. 63–69.

[491] S. Tsumoto, H. Tanaka (1996), *Extraction of medical diagnostic knowledge based on rough set based model selection rule induction.* In: [RSFD'96] 426–436; see also: Chen, Hirota, and Yen [45], pp. 145–151.

[492] S. Tsumoto, H. Tanaka (1996), *Automated induction of medical expert system rules from clinical databases based on rough sets.* In: Zimmermann [557], pp. 154–158.

[493] S. Tsumoto, H. Tanaka (1996), *Domain knowledge from clinical databases based on rough set model.* In: In: Borne, Dauphin–Tanguy, Sueur, and El Khattabi [24], pp. 742–747.

[494] S. Tsumoto, H. Tanaka (1996), *Incremental learning of probabilistic rules from clinical databases.* In: Bouchon–Meunier, Delgado, Verdegay, Vila, and Yager [25] **2**, pp. 1457–1462.

[495] S. Tsumoto, H. Tanaka (1996), *Classification rule induction based on rough sets.* In: Petry and Kraft [321], pp. 748–754.

[496] S. Tsumoto, H. Tanaka (1996), *Induction of expert system rules from databases based on rough set theory resampling methods.* In: Ras and Michalewicz [362], pp. 128–138.

[497] S. Tsumoto, H. Tanaka (1996), *Machine discovery of functional components of proteins from amino–acid sequences based on rough sets change of representation.* In: Journal of the Intelligent Automation and Soft Computing **2/2**, pp. 169–180.

[498] S. Tsumoto, H. Tanaka (1996), *PRIMEROSE3. Induction estimation of probabilistic rules from medical databases based on rough sets resampling methods.* In: M. Witten (Ed.), Computational Medicine, Public Health, Biotechnology, Building a Man in the Machine **III**, World Scientific, Singapore, pp. 1173–1189.

[499] S. Tsumoto, W. Ziarko (1996), *The application of rough sets–based data mining technique to differential diagnosis of meningoencephalitis.* In: Ras and Michalewicz [362], pp. 438–447.

[500] S. Tsumoto, W. Ziarko, N. Shan, H. Tanaka (1995), *Knowledge discovery in clinical databases based on variable precision rough sets model.* In: Proceedings of the 19th Annual Symposium on Computer Applications in Medical Care, New Orleans, Journal of American Medical Informatics Association Supplement, pp. 270–274.

[501] A. Tversky (1997), *Features of similarity.* Psychological Review **84/4**, pp. 327–352.

[502] D. Vakarelov (1991), *A modal logic for similarity relations in Pawlak knowledge representation systems.* Fundamenta Informaticae **15**, pp. 61–79.

[503] D. Vakarelov (1991), *Logical analysis of positive and negative similarity relations in property systems.* In: M. De Glas, D. Gabbay (Eds.), *First World Conference on the Fundamentals of AI (WOCFAI'91)*, July 1–5, Paris, France, pp. 491–499.

[504] D. Vakarelov (1995), *A duality between Pawlak's information systems and bi–consequence systems.* Studia Logica **55/1** pp. 205–228.

[505] D. Vakarelov (1998), *Information systems, similarity relations and modal logic.* In: Orłowska [276] pp. 492–550.

[506] V.N. Vapnik (1995), *The Nature of Statistical Learning Theory.* Springer–Verlag, New York.

[507] D. Van den Poel, Z. Piasta (1998), *Purchase prediction in database marketing with the ProbRough system.* In: Polkowski and Skowron [346], pp. 593–600.

[508] D. Van den Poel (1998), *Rough sets for database marketing.* In: Polkowski and Skowron [345], pp. 324–335.

[509] I. Velasco, D. Teo, T.T. Lin (1997), *Design optimization of rough–fuzzy controllers using a genetic algorithm.* In: Wang [514], pp. 313–317.

[510] S. Vinterbo, L. Ohno-Machado H. Fraser (1998), *Prediction of acute myocardial infarction using rough sets.* Submitted for publication.

[511] A. Wakulicz–Deja, M. Boryczka, P. Paszek (1998), *Discretization of continuous attributes of decision system in mitochondrial encephalomyopathies.* In: Polkowski and Skowron [346], pp. 483–490.

[512] A. Wakulicz–Deja, P. Paszek (1997), *Diagnose progressive encephalopathy applying the rough set theory.* International Journal of Medical Informatics **46**, pp. 119–127.

[513] P.P. Wang (Ed.) (1995), *Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS'95).* Wrightsville Beach, North Carolina, 28 September - 1 October, pp. 1–679.

[514] P.P. Wang (Ed.) (1997), *Proceedings of the Fifth International Workshop on Rough Sets Soft Computing (RSSC'97) at Third Annual Joint Conference on Information Sciences (JCIS'97).* Duke University, Durham, NC, USA, Rough Set & Computer Science **3**, March 1–5, pp. 1–449.

[515] A. Wasilewska (1989), *Linguistically definable concepts and dependencies.* Journal of Symbolic Logic **54/2**, pp. 671–672.

[516] A. Wasilewska (1996), *On rough and LT–fuzzy sets.* In: Chen, Hirota, and Yen [45], pp. 13–18.

[517] A. Wasilewska (1997), *Topological rough algebras.* In: Lin and Cercone [191], pp. 411–425.

[518] A. Wasilewska, L. Vigneron (1998), *Rough algebras and automated deduction.* In: Polkowski and Skowron [344], pp. 261–275.

[519] A. Wasilewska, L. Vigneron (1995), *Rough equality algebras.* In: Wang [513], pp. 26–30.

[520] I. Wegener (1987), *The Complexity of Boolean Functions.* Wiley, Stuttgart.

[521] S.M. Weiss, C.A. Kulikowski(1991), *Computer Systems that Learn* Morgan Kaufmann, San Mateo, CA.

[522] R. Wilting (1997), *Predicting card credit. A research on predictable behaviour of clients applying for card credit by means of Rough Data Models.* (in Dutch). Master Thesis, Vrije Universiteit Amsterdam.

[523] *http://www.idi.ntnu.no/˜aleks/rosetta/* – the Rosetta WWW homepage.

[524] D.A. Wilson, T.R. Martinez (1997), *Improved heterogeneous distance functions.* Journal of Artificial Intelligence Research **6**, pp. 1–34.

[525] P. Wojdyłło (1998), *Wavelets, rough sets arificial neural networks in EEG analysis.* In: Polkowski and Skowron [346], pp. 444–449.

[526] L. Woolery, J.W. Grzymała–Busse, S. Summers, A. Budihardjo (1991), *The use of machine learning program $LERS_LB$ 2.5 in knowledge acquisition for expert system development in nursing.* Computers nd Nursing **9**, pp. 227–234.

[527] L. Woolery, M. Van Dyne, J.W. Grzymała–Busse, C. Tsatsoulis (1994), *Machine learning for development of an expert system to support nurses' assessment of preterm birth risk.* In: Nursing Informatics, An International Overview for Nursing in a Technological Era, Proceedings of the Fifth International Conference on Nursing Use of Computers Information Sci. June 17–22, San Antonio, TX, Elsevier, pp. 357–361.

[528] L. Woolery, J.W. Grzymała–Busse (1994), *Machine learning for an expert system to predict preterm birth risk.* Journal of the American Medical Informatics Association 1, pp. 439–446.

[529] S.K.M. Wong (1996), *Interval structure – A qualitative measure of uncertainty.* In: Lin and Wildberger [196], pp. 22–27.

[530] S.K.M. Wong, P. Lingras (1989), *The compatibility view of Shafer–Dempster theory using the concept of rough set.* In: Z.W. Ras (Ed.), Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems (ISMIS'89), Charlotte, North Carolina, October 12–14, North Holland, pp. 33–42.

[531] S.K.M. Wong, W. Ziarko (1986), *Comparison of the probabilistic approximate classification and the fuzzy set model.* Fuzzy Sets and Systems **21**, pp. 357–362.

[532] S.K.M. Wong, W. Ziarko, L.W. Ye (1986), *Comparision of rough set and statistical methods in inductive learning.* Journal of Man–Machine Studies **24**, pp. 53–72.

[533] J. Wróblewski (1995), *Finding minimal reducts using genetic algorithms.* In: Wang [513], pp. 186–189.

[534] J. Wróblewski (1998), *Genetic algorithms in decomposition and classification problems.* In: Polkowski and Skowron [345], pp. 472–492.

[535] U. Wybraniec-Skardowska (1989), *On a generalization of approximation space.* Bull. Polish Acad. Sci. Math. **37**, pp. 51–61.

[536] Y.Y. Yao (1997), *Combination of rough and fuzzy sets based on alpha–level sets.* In: Lin and Cercone [191], pp. 301–321.

[537] Y.Y. Yao (1998), *Generalized rough set models.* In: Polkowski and Skowron [344], pp. 286–318.

[538] Y.Y. Yao, T.Y. Lin (1996), *Generalization of rough sets using modal logic.* Journal of the Intelligent Automation and Soft Computing **2**, pp. 103–120.

[539] Y.Y. Yao, S.K.M. Wong (1995), *Generalization of rough sets using relationships between attribute values.* In: Wang [513], pp. 30–33.

[540] Y.Y. Yao, S.K.M. Wong (1996), *Generalized probabilistic rough set models.* In: Chen, Hirota, and Yen [45], pp. 158–163.

[541] Y.Y. Yao, S.K.M. Wong, T.Y. Lin (1997), *A review of rough set models.* In: T.Y. Lin, N. Cercone [191] pp. 47–75.

[542] L.A. Zadeh (1971), *Similarity relations and fuzzy orderings.* Information Sciences **3**, pp. 177–200.

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

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

[545] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li (1997), *New parallel algorithms for fast discovery of association rules.* In: Data Mining and Knowledge Discovery : An International Journal **1/4**, pp. 343–373.

[546] E.C. Zeeman (1965), *T*he topology of brain and visual perception. In: Fort, K.M. (Ed.): *Topology of 3-manifolds and related topics.* Prentice Hall, Englewood Cliffs N.J., pp. 240–256.

[547] W. Ziarko (1991), *The discovery, analysis and representation of data dependencies in databases.* In: G. Piatetsky–Shapiro, W.J. Frawley (Eds.), Knowledge Discovery in Databases, AAAI Press/MIT Press, pp. 177–195.

[548] W. Ziarko (1992), *Generation of control algorithms for computerized controllers by operator supervised training.* In: Proceedings of the 11th IASTED International Conference on Modelling, Identification Control. Innsbruck, Austria, pp. 510–513.

[549] W. Ziarko (1993), *Variable precision rough set model* . J. of Computer and System Sciences, 46, pp. 39–59.

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

[551] W. Ziarko (1994), *Rough sets and knowledge discovery: An overview.* In: Ziarko [550], pp. 11–15.

[552] W. Ziarko (1998), *Rough sets as a methodology for data mining.* In: Polkowski and Skowron [344], pp. 554–576.

[553] W. Ziarko, R. Golan, D. Edwards (1993), *An application of DATALOGIC/R knowledge discovery tool to identify strong predictive rules in stock market data.* In: Proceedings of AAAI Workshop on Knowledge Discovery in Databases, Washington, DC, pp. 89–101.

[554] W. Ziarko, J. Katzberg (1989), *Control algorithms acquisition, analysis reduction, Machine learning approach.* In: Knowledge–Based Systems Diagnosis, Supervision Control, Plenum Press, Oxford, pp. 167–178.

[555] W. Żakowski (1982), *On a concept of rough sets.* Demonstratio Mathematica **XV**, pp. 1129–1133.

[556] W. Żakowski (1983), *Approximations in the space* (U, Π). Demonstratio Mathematica **XVI**, pp. 761–769.

[557] H.-J. Zimmermann (1997), *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT'96).* September 2–5, Aachen, Germany, Verlag Mainz **1**, pp. 1–702.

[558] H.-J. Zimmermann (1996), *Proceedings of the Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97).* Aachen, Germany, Verlag Mainz **1**, pp. 1–868.