Rough Sets

Zdzisław Pawlak

Institute of Theoretical and Applied Informatics,
Polish Academy of Sciences,
ul. Bałtycka 5, 44 100 Gliwice, Poland

University of Information Technology and Management
ul. Newelska 6, 01-447 Warsaw, Poland
zpw@ii.pw.edu.pl

Contents

Introduction

Rough set theory is a new mathematical approach to imperfect knowledge.

The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently it became also a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imperfect knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Zadeh [2].

Rough set theory proposed by the author in [1] presents still another attempt to this problem. The theory has attracted attention of many researchers and practitioners all over the world, who contributed essentially to its development and applications.

Rough set theory has an overlap with many other theories. However we will refrain to discuss these connections here. Despite of the above mentioned connections rough set theory may be considered as the independent discipline in its own rights.

Rough set theory has found many interesting applications. The rough set approach seems to be of fundamental importance to AI and cognitive sciences, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition.

The main advantage of rough set theory in data analysis is that it does not need any preliminary or additional information about data – like probability in statistics, or basic probability assignment in Dempster-Shafer theory, grade of membership or the value of possibility in fuzzy set theory.

The proposed approach

- provides efficient algorithms for finding hidden patterns in data,
- finds minimal sets of data (data reduction),
- evaluates significance of data,
- generates sets of decision rules from data,
- it is easy to understand,
- offers straightforward interpretation of obtained results,
- most algorithms based on the rough set theory are particularly suited for parallel processing.

Basic ideas of rough set theory and its extensions, as well as many interesting applications can be found on the internet, e.g., http://www.roughsets.org

The booklet is organized as follows:

Chapter 1 (Basic Concepts) contains general formulation of basic ideas of rough set theory together with brief discussion of its place in classical set theory.

Chapter 2 (Rough Sets and Reasoning from Data) presents the application of rough set concept to reason from data (data mining).

Chapter 3 (Rough Sets and Bayes' Theorem) gives a new look on Bayes' theorem and shows that Bayes' rule can be used differently to that offered by classical Bayesian reasoning methodology.

In Chapter 4 (Data Analysis and Flow Graphs) we show that many problems in data analysis can be boiled down to flow analysis in a flow network.

Chapter 5 (Rough Sets and Conflict Analysis) discuses the application of rough set concept to study conflict.

This booklet is a modified version of lectures delivered at the Tarragona University seminar on Formal Languages and Rough Sets in August 2003.

References

[1] Z. Pawlak: Rough sets, International Journal of Computer and Information Sciences, 11, 341-356, 1982

[2] L. Zadeh: Fuzzy sets, Information and Control, 8, 338-353, 1965

CHAPTER 1

Rough Sets – Basic Concepts

1. Introduction

In this chapter we give some general remarks on a concept of a set and the place of rough sets in set theory.

The concept of a set is fundamental for the whole mathematics. Modern set theory was formulated by George Cantor [1].

Bertrand Russell has discovered that the intuitive notion of a set proposed by Cantor leads to antinomies [8]. Two kinds of remedy for this discontent have been proposed: axiomatization of Cantorian set theory and alternative set theories.

Another issue discussed in connection with the notion of a set is vagueness. Mathematics requires that all mathematical notions (including set) must be exact (Gottlob Frege[2]). However philosophers and recently computer scientists got interested in vague concepts.

The notion of a fuzzy set proposed by Lotfi Zadeh [10] is the first very successful approach to vagueness. In this approach sets are defined by partial membership, in contrast to crisp membership used in classical definition of a set.

Rough set theory, introduced by the author, [4] expresses vagueness, not by means of membership, but employing a boundary region of a set. If the boundary region of a set is empty it means that the set is crisp, otherwise the set is rough (inexact). Nonempty boundary region of a set means that our knowledge about the set is not sufficient to define the set precisely.

In this paper the relationship between sets, fuzzy sets and rough sets will be outlined and briefly discussed.

2. Sets

The notion of a set is not only basic for the whole mathematics but it also plays an important role in natural language. We often speak about sets (collections) of various objects of interest, e.g., collection of books, paintings, people etc.

Intuitive meaning of a set according to some dictionaries is the following:

> "A number of things of the same kind that belong or are used together."
>
> *Webster's Dictionary*

> "Number of things of the same kind, that belong together because they are similar or complementary to each other."
>
> *The Oxford English Dictionary*

Thus a set is a collection of things which are somehow related to each other but the nature of this relationship is not specified in these definitions.

In fact these definitions are due to the original definition given by the creator of set theory, George Cantor [1], which reads as follows:

> "Unter einer Mannigfaltigkeit oder Menge verstehe ich nämlich allgenein jedes Viele, welches sich als Eines denken lässt, d.h. jeden Inbegriff bestimmter Elemente, welcher durch ein Gesetz zu einem Ganzen verbunden werden kann."

Thus according to Cantor a set is a collection of any objects, which according to some law can be considered as a whole.

All mathematical objects, e.g., relations, functions, numbers, etc., are some kind of sets. In fact set theory is needed in mathematics to provide rigor.

Bertrand Russell discovered that the intuitive notion of a set given by Cantor leads to *antinomies* (contradictions) [8]. One of the best known antinomies called the powerset antinomy goes as follows: consider (infinite) set X of all sets. Thus X is the greatest set. Let Y denote the set of all subsets of X. Obviously Y is greater then X, because the number of subsets of a set is always greater the number of its elements. For example, if $X = \{1, 2, 3\}$ then
$Y = \{\emptyset,\{1\},\{2\},\{3\},\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$, where $\emptyset$ denotes the empty set. Hence X is not the greatest set as assumed and we arrived at contradiction.

Thus the basic concept of mathematics, the concept of a set, is contradictory. That means that a set cannot be a collection of arbitrary elements as was stipulated by Cantor.

As a remedy for this defect several improvements of set theory have been proposed. For example,

- Axiomatic set theory (Zermello and Fraenkel, 1904)
- Theory of types (Whitehead and Russell, 1910)
- Theory of classes (v. Neumann, 1920)

All these improvements consist on restrictions, put on objects which can form a set. The restrictions are expressed by properly chosen axioms, which say how the set can be build. They are called, in contrast to Cantors' intuitive set theory, axiomatic set theories.

Instead of improvements of Cantors' set theory by its axiomatization, some mathematicians proposed escape from classical set theory by creating completely new idea of a set, which would free the theory from antinomies. Some of them are listed below.

- Mereology (Leśniewski, 1915)
- Alternative set theory (Vopenka, 1970)
- "Penumbral" set theory (Apostoli and Kanada, 1999)

No doubt the most interesting proposal was given by Stanisław Leśniewski [3], who proposed instead of membership relation between elements and sets, employed in classical set theory, the relation of "being a part". In his set theory, called *mereology*, this relation is a fundamental one.

None of the three mentioned above "new" set theories were accepted by mathematicians, however Leśniewski's mereology attracted some attention of philosophers and recently also computer scientists, (e.g., Lech Polkowski and Andrzej Skowron [7]).

In classical set theory a set is uniquely determined by its elements. In other words, it means that every element must be uniquely classified as belonging to the set or not. That is to say the notion of a set is a *crisp* (precise) one. For example, the set of odd numbers is crisp because every number is either odd or even. In mathematics we have to use crisp notions, otherwise precise reasoning would be impossible. However philosophers for many years were interested also in *vague* (imprecise) notions.

For example, in contrast to odd numbers, the notion of a beautiful painting is vague, because we are unable to classify uniquely all paintings into two classes: beautiful and not beautiful. Some paintings cannot be decided whether they are beautiful or not and thus they remain in the doubtful area. Thus *beauty* is not a precise but a vague concept.

Almost all concepts we are using in natural language are vague. Therefore common sense reasoning based on natural language must be based on vague concepts and not on classical logic. This is why vagueness is important for philosophers and recently also for computer scientists.

Vagueness is usually associated with the boundary region approach (i.e., existing of objects which cannot be uniquely classified to the set or its complement) which was first formulated in 1893 by the father of modern logic Gottlob Frege [2]. He wrote:

> "Der Begrieff muss scharf begrenzt sein. Einem unscharf begrenzten Begriff würde ein Bezirk ensprechen, der nicht überall ein scharfe Grentzlinie hätte, sondern stellenweise gantz verschwimmend in die Umgebung übergine. Das wäre eigentlich gar kein Bezirk; und so wird ein unscharf definirter Begrieff mit Unrecht Begrieff gennant. Solche begriffsartige Bildungen kann die Logik nicht als Begriffe anerkennen; es is unmäglich, von ihnen genaue Gesetze auszustellen. Das Gesetz des ausgeschlossenen Drititten ist ja eigentlich nur in anderer From die Forderung, dass der Begriff scharf begrentz sei. Ein beliebiger Gegenstand *x* fält entwerder unter der Begriff *y*, oder er fällt nich unter ihn: *tertium non datur*."

Thus according to Frege

> "The concept must have a sharp boundary. To the concept without a sharp boundary there would correspond an area that had not a sharp boundary-line all around."

I.e., mathematics must use crisp, not vague concepts, otherwise it would be impossible to reason precisely.

Summing up, vagueness is

- Not allowed in mathematics
- Interesting for philosophy
- Necessary for computer science

3. Fuzzy Sets

Lotfi Zadeh proposed completely new, elegant approach to vagueness called *fuzzy set theory* [10]. In his approach an element can belong to a set to a degree *k* ($0 \le k \le 1$), in contrast to classical set theory where an element must definitely belong or not to a set. E.g., in classical set theory one can be definitely ill or healthy, whereas in fuzzy set theory we can say that someone is ill (or healthy) in 60 percent (i.e. in the degree 0.6). Of course, at once the question arises where we get the value of degree from. This issue raised a lot of discussion, but we will refrain from considering this problem here.

Thus fuzzy membership function can be presented as

$$\mu_X(x) \in <0,1>$$

where, *X* is a set and *x* is an element.

Let us observe that the definition of fuzzy set involves more advanced mathematical concepts, real numbers and functions, whereas in classical set theory the notion of a set is

used as a fundamental notion of whole mathematics and is used to derive any other mathematical concepts, e.g., numbers and functions. Consequently fuzzy set theory cannot replace classical set theory, because, in fact, the theory is needed to define fuzzy sets.

Fuzzy membership function has the following properties.

a) $\mu_{U-X}(x) = 1 - \mu_X(x)$ for any $x \in U$

b) $\mu_{X \cup Y}(x) = max(\mu_X(x), \mu_Y(x))$ for any $x \in U$

c) $\mu_{X \cap Y}(x) = min(\mu_X(x), \mu_Y(x))$ for any $x \in U$

That means that the membership of an element to the union and intersection of sets is uniquely determined by its membership to constituent sets. This is a very nice property and allows very simple operations on fuzzy sets, which is a very important feature both theoretically and practically.

Fuzzy set theory and its applications developed very extensively over last years and attracted attention of practitioners, logicians and philosophers worldwide.

4. Rough Sets

Rough set theory [4] is still another approach to vagueness. Similarly to fuzzy set theory it is not an alternative to classical set theory but it is embedded in it. Rough set theory can be viewed as a specific implementation of Frege's idea of vagueness, i.e., imprecision in this approach is expressed by a boundary region of a set, and not by a partial membership, like in fuzzy set theory.

Rough set concept can be defined quite generally by means of topological operations, *interior* and *closure*, called *approximations.*

Let us describe this problem more precisely. Suppose we are given a set of objects U called the *universe* and an indiscernibility relation $R \subseteq U \times U$, representing our lack of knowledge about elements of U. For the sake of simplicity we assume that R is an equivalence relation. Let X be a subset of U. We want to characterize the set X with respect to R. To this end we will need the basic concepts of rough set theory given below.

- The *lower approximation* of a set X with respect to R is the set of all objects, which can be for *certain* classified as X with respect to R (are *certainly* X with respect to R).
- The *upper approximation* of a set X with respect to R is the set of all objects which can be *possibly* classified as X with respect to R (are *possibly* X in view of R).
- The *boundary region* of a set X with respect to R is the set of all objects, which can be classified neither as X nor as not-X with respect to R.

Now we are ready to give the definition of rough sets.

- Set X is *crisp* (exact with respect to R), if the boundary region of X is empty.
- Set X is *rough* (inexact with respect to R), if the boundary region of X is nonempty.

Thus a set is *rough* (imprecise) if it has nonempty boundary region; otherwise the set is *crisp* (precise). This is exactly the idea of vagueness proposed by Frege.

The approximations and the boundary region can be defined more precisely. To this end we need some additional notation.

The equivalence class of R determined by element x will be denoted by $R(x)$. The indiscernibility relation in certain sense describes our lack of knowledge about the universe.

Equivalence classes of the indiscernibility relation, called *granules* generated by R, represent elementary portion of knowledge we are able to perceive due to R. Thus in view of the indiscernibility relation, in general, we are an able to observe individual objects but we are forced to reason only about the accessible granules of knowledge.

Formal definitions of approximations and the boundary region are as follows:

- *R-lower approximation* of X

$$R_*(x) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$$

- *R-upper approximation* of X

$$R^*(x) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \varnothing\}$$

- *R-boundary region* of X

$$RN_R(X) = R^*(X) - R_*(X)$$

As we can see from the definition approximations are expressed in terms of granules of knowledge. The lower approximation of a set is union of all granules which are entirely included in the set; the upper approximation – is union of all granules which have non-empty intersection with the set; the boundary region of set is the difference between the upper and the lower approximation.

This definition is clearly depicted in Figure 1.

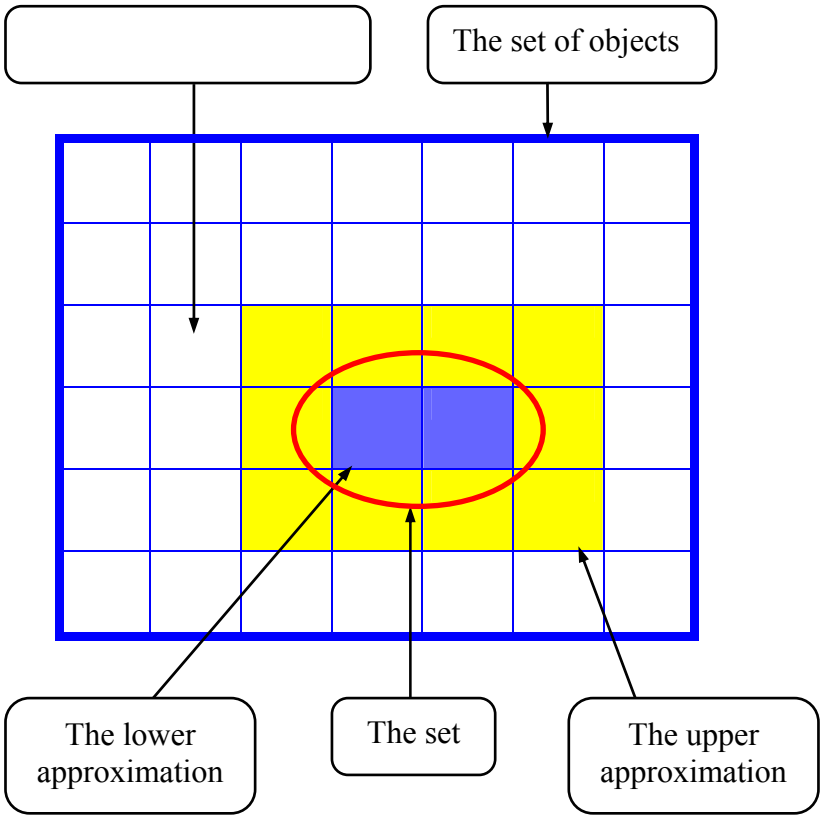

Fig. 1

It is interesting to compare definitions of classical sets, fuzzy sets and rough sets. Classical set is a primitive notion and is defined intuitively or axiomatically. Fuzzy sets are defined by employing the fuzzy membership function, which involves advanced mathematical structures,

numbers and functions. Rough sets are defined by approximations. Thus this definition also requires advanced mathematical concepts.

Approximations have the following properties:

1) $R_*(X) \subseteq X \subseteq R^*(X)$
2) $R_*(\varnothing) = R^*(\varnothing) = \varnothing; R_*(U) = R^*(U) = U$
3) $R^*(X \cup Y = R^*(X) \cup R^*(Y)$
4) $R_*(X \cap Y) = R_*(X) \cap R_*(Y)$
5) $R_*(X \cup Y) \supseteq R_*(X) \cup R_*(Y)$
6) $R^*(X \cap Y) \subseteq R^*(X) \cap R^*(Y)$
7) $X \subseteq Y \rightarrow R_*(X) \subseteq R_*(Y) \,\&\, R^*(X) \subseteq R^*(Y)$
8) $R_*(-X) = -R^*(X)$
9) $R^*(-X) = -R_*(X)$
10) $R_*R_*(X) = R^*R_*(X) = R_*(X)$
11) $R^*R^*(X) = R_*R^*(X) = R^*(X)$

It is easily seen that approximations are in fact interior and closure operations in a topology generated by data. Thus fuzzy set theory and rough set theory require completely different mathematical setting.

Rough sets can be also defined employing, instead of approximation, rough membership function [5]

$$\mu_X^R : U \rightarrow < 0,1 >$$

where

$$\mu_X^R(x) = \frac{|X \cap R(x)|}{|R(x)|}$$

and $|X|$ denotes the cardinality of X.

The rough membership function expresses conditional probability that x belongs to X given R and can be interpreted as a degree that x belongs to X in view of information about x expressed by R.

The meaning of rough membership function can be depicted as shown in Fig.2.

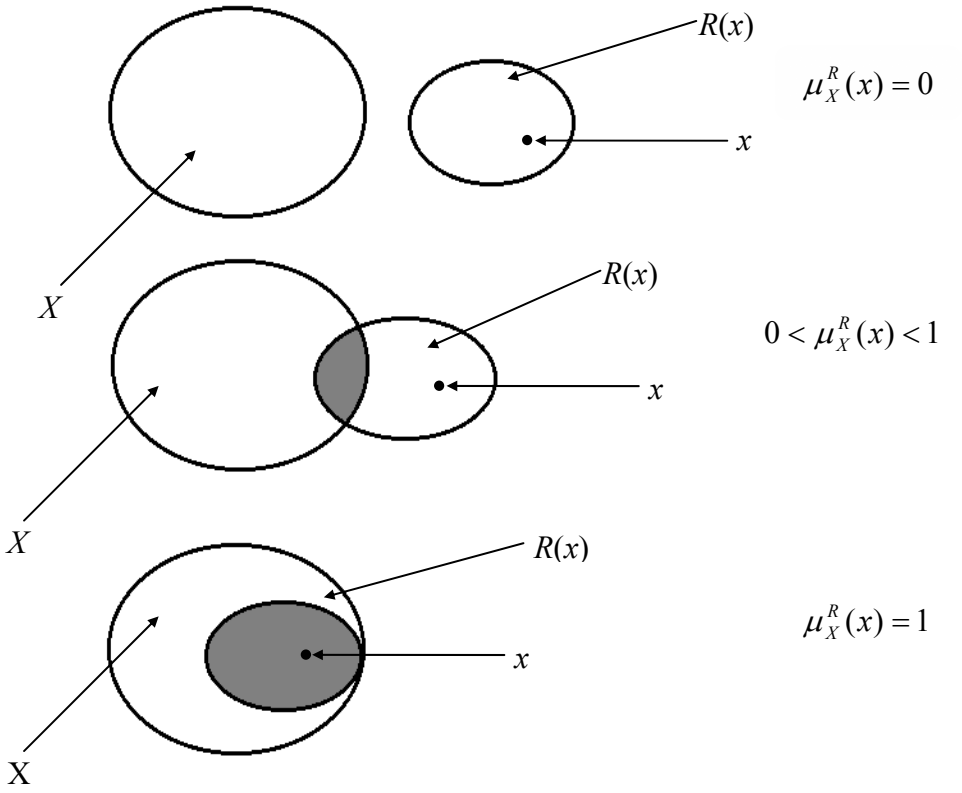

Fig. 2

The rough membership function can be used to define approximations and the boundary region of a set, as shown below:

$$R_*(X) = \{x \in U : \mu_X^R(x) = 1\},$$
$$R^*(X) = \{x \in U : \mu_X^R(x) > 0\},$$
$$RN_R(X) = \{x \in U : 0 < \mu_X^R(x) < 1\}.$$

It can be shown that the membership function has the following properties [5]:

1) $\mu_X^R(x) = 1$ *iff* $x \in R_*(X)$
2) $\mu_X^R(x) = 0$ *iff* $x \in U - R^*(X)$
3) $0 < \mu_X^R(x) < 1$ *iff* $x \in RN_R(X)$
4) $\mu_{U-X}^R(x) = 1 - \mu_X^R(x)$ *for any* $x \in U$
5) $\mu_{X \cup Y}^R(x) \geq max\ (\mu_X^R(x), \mu_Y^R(x))$ *for any* $x \in U$
6) $\mu_{X \cap Y}^R(x) \leq min\ (\mu_X^R(x), \mu_Y^R(x))$ *for any* $x \in U$

From the properties it follows that the rough membership differs essentially from the fuzzy membership, for properties 5) and 6) show that the membership for union and intersection of sets, in general, cannot be computed – as in the case of fuzzy sets – from their constituents membership. Thus formally the rough membership is a generalization of fuzzy membership. Besides, the rough membership function, in contrast to fuzzy membership function, has a probabilistic flavour.

Now we can give two definitions of rough sets.

Definition 1: Set X is *rough* with respect to R if $R_*(X) \neq R^*(X)$.

Definition 2: Set X *rough* with respect to R if for some x, $0 < \mu_X^R(x) < 1$.

It is interesting to observe that definition 1 and definition 2 are not equivalent [5], but we will not discuss this issue here.

Let us also mention that rough set theory clearly distinguishes two very important concepts, vagueness and uncertainty, very often confused in the AI literature. Vagueness is the property of sets and can be described by approximations, whereas uncertainty is the property of elements of a set and can expressed by the rough membership function.

5. Summary

Basic concept of mathematics, the set, leads to antinomies, i.e., it is contradictory.

This deficiency of sets, has rather philosophical than practical meaning, for sets used in mathematics are free from the above discussed faults. Antinomies are associated with very "artificial" sets constructed in logic but not found in sets used in mathematics. That is why we can use mathematics safely.

Fuzzy set and rough set theory are two different approaches to vagueness and are not remedy for classical set theory difficulties.

Both theories represent two different approaches to vagueness. Fuzzy set theory addresses gradualness of knowledge, expressed by the fuzzy membership – whereas rough set theory addresses granularity of knowledge, expressed by the indiscernibility relation.

References

[1] G. Cantor: Grundlagen einer allgemeinen Mannigfaltigkeitslehre, Leipzig, 1883

[2] G. Frege, Grundlagen der Arithmetik, 2, Verlag von Herman Pohle, Jena, 1893

[3] St. Leśniewski: Grungzüge eines neuen Systems der Grundlagen der Mathematik, Fundamenta Matemaicae, XIV, 1929, 1- 81

[4] Z. Pawlak: Rough sets, Int. J. of Information and Computer Sciences, 11, 5, 341-356, 1982

[5] Z. Pawlak, A. Skowron: Rough membership function, in: R. E Yeager, M. Fedrizzi and J. Kacprzyk (eds.), Advaces in the Dempster-Schafer of Evidence, Wiley, New York, 1994, 251-271

[6] L. Polkowski: Rough Sets, Mathematical Foundations, Advances in Soft Computing, Physica – Verlag, A Springer-Verlag Company, 2002

[7] L. Polkowski, A. Skowron: Rough mereological calculi granules: a rough set approach to computation, computational intelligence: An International Journal 17, 2001, 472-479

[8] B. Russell: The Principles of Mathematics, London, George Allen & Unwin Ltd., 1st Ed. 1903 (2nd Edition in 1937)

[9] A. Skowron, J. Komorowski, Z. Pawlak, L. Polkowski: A rough set perspective on data and knowledge, in: W. Kloesgen, J. Zytkow (eds.), Handbook of KDD, Oxford University Press, 2002, 134-149

[10] L. Zadeh: Fuzzy sets, Information and Control, 8, 338-353, 1965

CHAPTER 2

Rough Sets and Reasoning from Data

1. Introduction

In this chapter we define basic concepts of rough set theory in terms of data, in contrast to general formulation presented in Chapter 1. This is necessary if we want to apply rough sets to reason from data.

As mentioned in the previous chapter rough set philosophy is based on the assumption that, in contrast to classical set theory, we have some additional information (knowledge, data) about elements of a *universe of discourse*. Elements that exhibit the same information are indiscernible (similar) and form blocks that can be understood as elementary granules of knowledge about the universe. For example, patients suffering from a certain disease, displaying the same symptoms are indiscernible and may be thought of as representing a granule (disease unit) of medical knowledge. These granules are called *elementary sets* (*concepts*), and can be considered as elementary building blocks of knowledge. Elementary concepts can be combined into compound concepts, i.e., concepts that are uniquely determined in terms of elementary concepts. Any union of elementary sets is called a *crisp set*, and any other sets are referred to as *rough* (*vague*, *imprecise*).

Due to the granularity of knowledge, rough sets cannot be characterized by using available knowledge. Therefore with every rough set we associate two *crisp* sets, called its *lower* and *upper approximation*. Intuitively, the lower approximation of a set consists of all elements that *surely* belong to the set, whereas the upper approximation of the set constitutes of all elements that *possibly* belong to the set. The difference of the upper and the lower approximation is a *boundary region*. It consists of all elements that cannot be classified uniquely to the set or its complement, by employing available knowledge. Thus any rough set, in contrast to a crisp set, has a non-empty boundary region.

In rough set theory sets are defined by approximations. Notice, that sets are usually defined by the membership function. Rough sets can be also defined using, instead of approximations, membership function, however the membership function is not a primitive concept in this approach, and both definitions are not equivalent.

2. An Example

For the sake of simplicity we first explain the proposed approach intuitively, by means of a simple tutorial example.

Data are often presented as a table, columns of which are labeled by *attributes*, rows by *objects* of interest and entries of the table are *attribute values*. For example, in a table containing information about patients suffering from a certain disease objects are *patients* (strictly speaking their ID's), attributes can be, for example, *blood pressure, body temperature* etc., whereas the entry corresponding to object *Smith* and the attribute *blood preasure* can be

normal. Such tables are known as *information systems, attribute-value* tables or *information tables*. We will use here the term *information table*.

Below an example of information table is presented.

Suppose we are given data about 6 patients, as shown in Table 1.

Patient	*Headache*	*Muscle-pain*	*Temperature*	*Flu*
*p*1	*no*	*yes*	*high*	*yes*
*p*2	*yes*	*no*	*high*	*yes*
*p*3	*yes*	*yes*	*very high*	*yes*
*p*4	*no*	*yes*	*normal*	*no*
*p*5	*yes*	*no*	*high*	*no*
*p*6	*no*	*yes*	*very high*	*yes*

Table 1

Columns of the table are labeled by attributes (symptoms) and rows – by objects (patients), whereas entries of the table are attribute values. Thus each row of the table can be seen as information about specific patient. For example, patient *p*2 is characterized in the table by the following attribute-value set

(*Headache*, *yes*), (*Muscle-pain*, *no*), (*Temperature*, *high*), (*Flu*, *yes*),

which form the information about the patient.

In the table patients *p*2, *p*3 and *p*5 are indiscernible with respect to the attribute *Headache*, patients *p*3 and *p*6 are indiscernible with respect to attributes *Muscle-pain* and *Flu*, and patients *p*2 and *p*5 are indiscernible with respect to attributes *Headache*, *Muscle-pain* and *Temperature*. Hence, for example, the attribute *Headache* generates two elementary sets {*p*2, *p*3, *p*5} and {*p*1, *p*4, *p*6}, whereas the attributes *Headache* and *Muscle-pain* form the following elementary sets: {*p*1, *p*4, *p*6}, {*p*2, *p*5} and {*p*3}. Similarly one can define elementary sets generated by any subset of attributes.

Patient *p*2 has flu, whereas patient *p*5 does not, and they are indiscernible with respect to the attributes *Headache*, *Muscle-pain* and *Temperature*, hence flu cannot be characterized in terms of attributes *Headache*, *Muscle-pain* and *Temperature*. Hence *p*2 and *p*5 are the boundary-line cases, which cannot be properly classified in view of the available knowledge. The remaining patients *p*1, *p*3 and p6 display symptoms which enable us to classify them with certainty as having flu, patients p2 and p5 cannot be excluded as having flu and patient *p*4 for sure does not have flu, in view of the displayed symptoms. Thus the lower approximation of the set of patients having flu is the set {*p*1, *p*3, *p*6} and the upper approximation of this set is the set {*p*1, *p*2, *p*3, *p*5, *p*6}, whereas the boundary-line cases are patients *p*2 and *p*5. Similarly *p*4 does not have flu and *p*2, *p*5 cannot be excluded as having flu, thus the lower approximation of this concept is the set {*p*4} whereas - the upper approximation – is the set {*p*2, *p*4, *p*5} and the boundary region of the concept "not flu" is the set {*p*2, *p*5}, the same as in the previous case.

3. Rough Sets and Approximations

As mentioned in the introduction, the starting point of rough set theory is the indiscernibility relation, generated by information about objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge we are unable to discern some objects employing the available information. That means that, in general, we are unable to

deal with single objects but we have to consider clusters of indiscernible objects, as fundamental concepts of our theory.

Now we present above considerations more precisely.

Suppose we are given two finite, non-empty sets U and A, where U is the *universe*, and A – a set of *attributes*. With every attribute $a \in A$ we associate a set V_a, of its *values*, called the *domain* of a. Any subset B of A determines a binary relation $I(B)$ on U, which will be called an *indiscernibility relation*, and is defined as follows:

$xI(B)y$ if and only if $a(x) = a(y)$ for every $a \in A$,
where $a(x)$ denotes the value of attribute a for element x.

Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., partition determined by B, will be denoted by $U/I(B)$, or simple U/B; an equivalence class of $I(B)$, i.e., block of the partition U/B, containing x will be denoted by $B(x)$.

If (x, y) belongs to $I(B)$ we will say that x and y are *B-indiscernible*. Equivalence classes of the relation $I(B)$ (or blocks of the partition U/B) are referred to as *B-elementary sets*. In the rough set approach the elementary sets are the basic building blocks (concepts) of our knowledge about reality.

The indiscernibility relation will be used next to define approximations, basic concepts of rough set theory.

Now approximations can be defined as follows:

$$B_*(X) = \{x \in U : B(x) \subseteq X\},$$

$$B^*(X) = \{x \in U : B(x) \cap X \neq \emptyset\},$$

assigning to every subset X of the universe U two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X, respectively. The set

$$BN_B(X) = B^*(X) - B_*(X)$$

will be referred to as the *B-boundary region* of X.

If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then the set X is *crisp* (*exact*) with respect to B; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, the set X is to as *rough* (*inexact*) with respect to B.

The properties of approximations can be presented now as:

1) $B_*(X) \subseteq X \subseteq B^*(X)$,
2) $B_*(\emptyset) = B^*(\emptyset) = \emptyset; B_*(U) = B^*(U) = U$,
3) $B^*(X \cup Y = B^*(X) \cup B^*(Y)$,
4) $B_*(X \cap Y) = B_*(X) \cap B_*(Y)$,
5) $X \subseteq Y$ implies $B_*(X) \subseteq B_*(Y)$ and $B_*(X) \subseteq B^*(Y)$,
6) $B_*(X \cup Y) \supseteq B_*(X) \cup B_*(Y)$,
7) $B^*(X \cup Y) \subseteq B^*(X) \cap B^*(Y)$,
8) $B_*(-X) = -B^*(X)$,
9) $B^*(-X) = -B_*(X)$,
10) $B_*(B_*(X)) = B^*(B_*(X)) = B_*(X)$,

11) $B^*(B^*(X)) = B_*(B^*(X)) = B^*(X)$,

where $-X$ denotes $U - X$.

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

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

a) $B_*(X) \neq \varnothing$ $B^*(X) \neq$ and, iff X is *roughly B-definable*,

b) $B_*(X) = \varnothing$ and $B^*(X) \neq U$, iff X is *internally B-indefinable*,

c) $B_*(X) \neq \varnothing$ and $B_*(X) = U$, iff X is *externally B-definable*,

d) $B_*(X) = \varnothing$ and $B^*(X) = U$, iff X is *totally B-indefinable*.

The intuitive meaning of this classification is the following.

If X is roughly *B-definable*, this means that we are able to decide for some elements of U whether they belong to X or $-X$, using B.

If X is internally B-indefinable, this means that we are able to decide whether some elements of U belong to $-X$, but we are unable to decide for any element of U, whether it belongs to X or not, using B.

If X is externally B-indefinable, this means that we are able to decide for some elements of U whether they belong to X, but we are unable to decide, for any element of U whether it belongs to $-X$ or not, using B.

If X is totally B-indefinable, we are unable to decide for any element of U whether it belongs to X or $-X$, using B.

Rough set can be also characterized numerically by the following coefficient

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|}$$

called *accuracy of approximation*, where $|X|$ denotes the cardinality of X. 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).

Let us depict above definitions by examples referring to Table 1. Consider the concept "flu", i.e., the set $X = \{p1, p2, p3, p6\}$ and the set of attributes B = {*Headache*, *Muscle-pain*, *Temperature*}. Concept "flu" is roughly B-definable, because $B_*(X) = \{p1, p3, p6\} \neq \varnothing$ and $B^*(X) = \{p1, p2, p3, p5, p6\} \neq U$. For this case we get α_B("*flu*") = 3/5. It means that the concept "flu" can be characterized partially employing symptoms *Headache*, *Muscle-pain* and *Temperature*. Taking only one symptom B = {*Headache*} we get $B_*(X) = \varnothing$ and $B^*(X) = U$, which means that the concept "flu" is totally indefinable in terms of attribute *Headache*, i.e., this attribute is not characteristic for flu whatsoever. However, taking single attribute

B = {Temperature} we get $B_*(X) = \{p3, p6\}$ and $B^*(X) = \{p1, p2, p3, p5, p6\}$, thus the concept "flu" is again roughly definable, but in this case we obtain $\alpha_B(X)$= 2/5, which means that the single symptom Temperature is less characteristic for flu, than the whole set of symptoms, and patient p1 cannot be now classified as having flu in this case.

4. Rough Sets and Membership Function

As shown Chapter 1 rough sets can be also defined using a *rough membership function* [3], defined as

$$\mu_X^B(x) = \frac{|X \cap B(x)|}{|B(x)|}.$$

Obviously

$$\mu_X^B(x) \in [0,1].$$

Value of the membership function $\mu_X(x)$ is a kind of conditional probability, and can be interpreted as a degree of *certainty* to which x belongs to X (or $1 - \mu_X(x)$, as a degree of *uncertainty*).

The rough membership function can be used to define approximations and the boundary region of a set, as shown below:

$$B_*(X) = \{x \in U : \mu_X^B(x) = 1\},$$

$$B^*(X) = \{x \in U : \mu_X^B(x) > 0\},$$

$$BN_B(X) = \{x \in U : 0 < \mu_X^B(x) < 1\}.$$

The rough membership function has the following properties [3]:

a) $\mu_X^B(x) = 1$ iff $x \in B_*(X)$,

b) $\mu_X^B(x) = 0$ iff $x \in -B^*(X)$,

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

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

e) If $xI(B)y$, then $\mu_X^B(x) = \mu_X^B(y)$ provided $I(B)$,

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

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

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

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

It can be easily seen that there exists a strict connection between vagueness and uncertainty. As we mentioned above vagueness is related to sets (concepts), whereas uncertainty is related to elements of sets. Rough set approach shows clear connection between these two concepts.

5. Decision Tables and Decision Algorithms

Sometimes we distinguish in an information table two classes of attributes, called *condition* and *decision* (*action*) attributes. For example, in Table 1 attributes *Headache*, *Muscle-pain*

and *Temperature* can be considered as condition attributes, whereas the attribute *Flu* – as a decision attribute.

Each row of a decision table determines a *decision rule*, which specifies decisions (*actions*) that should be taken when conditions pointed out by *condition* attributes are satisfied. For example, in Table 1 the condition (*Headache*, *no*), (*Muscle-pain*, *yes*), (*Temperature*, *high*) determines uniquely the decision (*Flu*, *yes*). Objects in a decision table are used as labels of decision rules.

Decision rules 2) and 5) in Table 1 have the same conditions but different decisions. Such rules are called *inconsistent* (*nondeterministic*, *conflicting*); otherwise the rules are referred to as *consistent* (*certain*, *deterministic*, *non-conflicting*). Sometimes consistent decision rules are called *sure* rules, and inconsistent rules are called *possible* rules. Decision tables containing inconsistent decision rules are called *inconsistent* (*nondeterministic*, *conflicting*); otherwise the table is *consistent* (*deterministic*, *non-conflicting*).

The number of consistent rules to all rules in a decision table can be used as *consistency factor* of the decision table, and will be denoted by $\gamma(C, D)$, where C and D are condition and decision attributes respectively. Thus if $\gamma(C, D) = 1$ the decision table is consistent and if $\gamma(C, D) \neq 1$ the decision table is inconsistent. For example, for Table 1, we have $\gamma(C, D) = 4/6$Decision rules are often presented as implications called "*if...then...*" rules. For example, rule 1) in Table 1 can be presented as implication

if (*Headache*, *no*) *and* (*Muscle-pain*, *yes*) *and* (*Temperature*, *high*) *then* (*Flu*, *yes*).

A set of decision rules is called a *decision algorithm*. Thus with each decision table we can associate a decision algorithm consisting of all decision rules occurring in the decision table.

We must however, make distinction between decision tables and decision algorithms. A decision table is a collection of data, whereas a decision algorithm is a collection of implications, e.g., logical expressions. To deal with data we use various mathematical methods, e.g., statistics but to analyze implications we must employ logical tools. Thus these two approaches are not equivalent, however for simplicity we will often present here decision rules in form of implications, without referring deeper to their logical nature, as it is often practiced in AI.

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 words, D depends totally on C, if there exists a functional dependency between values of D and C. For example, in Table 1 there are no total dependencies whatsoever. If in Table 1, the value of the attribute *Temperature* for patient *p*5 were "*no*" instead of "*high*", there would be a total dependency $\{Temperature\} \Rightarrow \{Flu\}$, because to each value of the attribute *Temperature* there would correspond unique value of the attribute *Flu*.

We would need also a more general concept of dependency of attributes, called a *partial dependency* of attributes.

Let us depict the idea by example, referring to Table 1. In this table, for example, the attribute *Temperature* determines uniquely only some values of the attribute *Flu*. That is, (*Temperature*, *very high*) implies (*Flu*, *yes*), similarly (*Temperature*, *normal*) implies (*Flu*, *no*), but (*Temperature*, *high*) does not imply always (*Flu*, *yes*). Thus the partial dependency means that only some values of D are determined by values of 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* ($0 \le k \le 1$), denoted $C \Rightarrow_k D$, if $k = \gamma(C, D)$.

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.

Thus the concept of dependency of attributes is strictly connected with that of consistency of the decision table.

For example, for dependency {*Headache*, *Muscle-pain*, *Temperature*}⇒{*Flu*} we get $k = 4/6 = 2/3$, because four out of six patients can be uniquely classified as having flu or not, employing attributes *Headache*, *Muscle-pain* and *Temperature*.

If we were interested in how exactly patients can be diagnosed using only the attribute Temperature, that is – in the degree of the dependence {*Temperature*}⇒{*Flu*}, we would get $k = 3/6 = 1/2$, since in this case only three patients $p3$, $p4$ and $p6$ out of six can be uniquely classified as having flu. In contrast to the previous case patient $p4$ cannot be classified now as having flu or not. Hence the single attribute *Temperature* offers worse classification than the whole set of attributes *Headache*, *Muscle-pain* and *Temperature*. It is interesting to observe that neither *Headache* nor *Muscle-pain* can be used to recognize flu, because for both dependencies {*Headache*}⇒{*Flu*} and {*Muscle-pain*}⇒{*Flu*} we have $k = 0$.

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

If *D depends in degree k*, $0 \le k \le 1$, on C, then

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

where

$$POS_C(D) = \bigcup_{X \in U/I(D)} C_*(X).$$

The expression $POS_C(D)$, 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.

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

7. Reduction of Attributes

We often face a question whether we can remove some data from a data table preserving its basic properties, that is – whether a table contains some superfluous data.

For example, it is easily seen that if we drop in Table 1 either the attribute Headache or Muscle-pain we get the data set which is equivalent to the original one, in regard to approximations and dependencies. That is we get in this case the same accuracy of approximation and degree of dependencies as in the original table, however using smaller set of attributes.

In order to express the above idea more precisely we need some auxiliary notions. Let B be a subset of A and let a belong to B.

- We say that a is *dispensable* in B if $I(B) = I(B - \{a\})$; otherwise a is *indispensable* in B.
- Set B is *independent* if all its attributes are indispensable.
- Subset B' of B is a *reduct* of B if B' is independent and $I(B') = I(B)$.

Thus a reduct is a set of attributes that preserves partition. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe.

Reducts have several important properties. In what follows we will present two of them.

First, we define a notion of a *core of attributes*.

Let B be a subset of A. The *core* of B is the set off all indispensable attributes of B.

The following is an important property, connecting the notion of the core and reducts

$$Core(B) = \bigcap Red(B),$$

where $Red(B)$ is the set off all reducts of B.

Because the core is the intersection of all reducts, it is included in every reduct, i.e., each element of the core belongs to some reduct. Thus, in a sense, the core is the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

To further simplification of an information table we can eliminate some values of attribute from the table in such a way that we are still able to discern objects in the table as the original one. To this end we can apply similar procedure as to eliminate superfluous attributes, which is defined next.

- We will say that the value of attribute $a \in B$, is *dispensable* for x, if $[x]_{I(B)} = [x]_{I(B - \{a\})}$; otherwise the value of attribute a is *indispensable* for x.
- If for every attribute $a \in B$ the value of a is indispensable for x, then B will be called *orthogonal* for x.
- Subset $B' \subseteq B$ is a *value reduct* of B for x, iff B' is orthogonal for x and $[x]_{I(B)} = [x]_{I(B')}$.

The set of all indispensable values of attributes in B for x will be called the *value core* of B for x, and will be denoted $CORE^x(B)$.

Also in this case we have

$$CORE^x(B) = \bigcap Red^x(B),$$

where $Red^x(B)$ is the family of all reducts of B for x.

Suppose we are given a dependency $C \Rightarrow D$. It may happen that the set D depends not on the whole set C but on its subset C' and therefore we might be interested to find this subset. In order to solve this problem we need the notion of a *relative reduct*, which will be defined and discussed next.

Let $C, D \subseteq A$. Obviously if $C' \subseteq C$ is a D-reduct of C, then C' is a minimal subset of C such that

$$\gamma(C, D) = \gamma(C', D).$$

- We will say that attribute $a \in C$ is *D-dispensable* in C, if $POS_C(D) = POS_{(C-\{a\})}(D)$; otherwise the attribute a is *D-indispensable* in C.
- If all attributes $a \in C$ are C-indispensable in C, then C will be called *D-independent*.

- Subset $C' \subseteq C$ is a *D-reduct* of C, iff C' is D-independent and $POS_C(D) = POS_{C'}(D)$.

The set of all D-indispensable attributes in C will be called *D-core* of C, and will be denoted by $CORE_D(C)$. In this case we have also the property

$$CORE_D(C) = \bigcap Red_D(C),$$

where $Red_D(C)$ is the family of all D-reducts of C.

If $D = C$ we will get the previous definitions.

For example, in Table 1 there are two relative reducts with respect to *Flu*, {*Headache*, *Temperature*} and {*Muscle-pain*, *Temperature*} of the set of condition attributes {*Headache*, *Muscle-pain*, *Temperature*}. That means that either the attribute *Headache* or *Muscle-pain* can be eliminated from the table and consequently instead of Table 1 we can use either Table 2

Patient	*Headache*	*Temperatur e*	*Flu*
*p*1	*no*	*high*	*yes*
*p*2	*yes*	*high*	*yes*
*p*3	*yes*	*very high*	*yes*
*p*4	*no*	*normal*	*no*
*p*5	*yes*	*high*	*no*
*p*6	*no*	*very high*	*yes*

Table 2

or Table 3

Patient	*Muscle-pain*	*Temperatur e*	*Flu*
*p*1	*yes*	*high*	*yes*
*p*2	*no*	*high*	*yes*
*p*3	*yes*	*very high*	*yes*
*p*4	*yes*	*normal*	*no*
*p*5	*no*	*high*	*no*
*p*6	*yes*	*very high*	*yes*

Table 3

For Table 1 the relative core of with respect to the set {*Headache*, *Muscle-pain*, *Temperature*} is the *Temperature*. This confirms our previous considerations showing that Temperature is the only symptom that enables, at least, partial diagnosis of patients.

We will need also a concept of a *value reduct* and *value core*. Suppose we are given a dependency $C \Rightarrow D$ where C is relative D-reduct of C. To further investigation of the dependency we might be interested to know exactly how values of attributes from D depend on values of attributes from C. To this end we need a procedure eliminating values of attributes form C which does not influence on values of attributes from D.

- We say that value of attribute $a \in C$, is *D-dispensable* for $x \in U$, if

  $$[x]_{I(C)} \subseteq [x]_{I(D)} \text{ implies } [x]_{I(C-\{a\})} \subseteq [x]_{I(D)};$$

  otherwise the value of attribute a is *D-indispensable* for x.
- If for every attribute $a \in C$ value of a is D-indispensable for x, then C will be called *D-independent* (*orthogonal*) for x.

- Subset $C' \subseteq C$ is a *D-reduct* of C for x (a value reduct), iff C' is D-independent for x and

$$[x]_{I(C)} \subseteq [x]_{I(D)} \text{ implies } [x]_{I(C')} \subseteq [x]I(D).$$

The set of all D-indispensable for x values of attributes in C will be called the *D-core* of C for x (the value core), and will be denoted $CORE_D^x(C)$.

We have also the following property

$$CORE_D^x(C) = \bigcap Red_D^x(C),$$

where $Red_D^x(C)$ is the family of all D-reducts of C for x.

Using the concept of a value reduct, Table 2 and Table 3 can be simplified as follows

Patient	*Headache*	*Temperatur e*	*Flu*
*p*1	*no*	*high*	*yes*
*p*2	*yes*	*high*	*yes*
*p*3	–	*very high*	*yes*
*p*4	–	*normal*	*no*
*p*5	*yes*	*high*	*no*
*p*6	–	*very high*	*yes*

Table 4

Patient	*Muscle-pain*	*Temperatur e*	*Flu*
*p*1	*yes*	*high*	*yes*
*p*2	*no*	*high*	*yes*
*p*3	–	*very high*	*yes*
*p*4	–	*normal*	*no*
*p*5	*no*	*high*	*no*
*p*6	–	*very high*	*yes*

Table 5

We can also present the obtained results in a form of a decision algorithm.
For Table 4 we get

if (*Headache*, *no*) *and* (*Temperature*, *high*) *then* (*Flu*, *yes*),
if (*Headache*, *yes*) *and* (*Temperature*, *high*) *then* (*Flu*, *yes*),
if (*Temperature*, *very high*) *then* (*Flu*, *yes*),
if (*Temperature*, *normal*) *then* (*Flu*, *no*),
if (*Headache*, *yes*) *and* (*Temperature*, *high*) *then* (*Flu*, *no*),
if (*Temperature*, *very high*) *then* (*Flu*, *yes*).

and for Table 5 we have

if (*Muscle-pain*, *yes*) *and* (*Temperature*, *high*) *then* (*Flu*, *yes*),
if (*Muscle-pain*, *no*) *and* (*Temperature*, *high*) *then* (*Flu*, *yes*),
if (*Temperature*, *very high*) *then* (*Flu*, *yes*),
if (*Temperature*, *normal*) *then* (*Flu*, *no*),
if (*Muscle-pain*, *no*) *and* (*Temperature*, *high*) *then* (*Flu*, *no*),

if (*Temperature*, *very high*) *then* (*Flu*, *yes*).

The following important property

a) $B' \Rightarrow B - B'$, where B' is a reduct of B,

connects reducts and dependency.

Besides, we have:

b) If $B \Rightarrow C$, then $B \Rightarrow C'$, for every $C' \subseteq C$,

in particular

c) If $B \Rightarrow C$, then $B \Rightarrow \{a\}$, for every $a \in C$.

Moreover, we have:

d) If B' is a reduct of B, then neither $\{a\} \Rightarrow \{b\}$ nor $\{b\} \Rightarrow \{a\}$ holds, for every $a, b \in B'$, i.e., all attributes in a reduct are pairwise independent.

8. Indiscernibility Matrices and Functions

To compute easily reducts and the core we will use discernibility matrix [4], which is defined next.

By an discernibility matrix of $B \subseteq A$ denoted $M(B)$ we will mean $n \times n$ matrix defined as:

$$(c_{ij}) = \{a \in B : (x_i) \neq a(x_j)\} \text{ for } i, j = 1,2,\ldots,n .$$

Thus entry c_{ij} is the set of all attributes which discern objects x_i and x_j.

The discernibility matrix $M(B)$ assigns to each pair of objects x and y a subset of attributes $\delta(x, y) \subseteq B$, with the following properties:

i) $\delta(x, x) = \varnothing$,

ii) $\delta(x, y) = \delta(y, x)$,

iii) $\delta(x, z) \subseteq \delta(x, y) \cup \delta(y, z)$.

These properties resemble properties of semi-distance, and therefore the function δ may be regarded as *qualitative semi-matrix* and $\delta(x, y)$ – *qualitative semi-distance*. Thus the discernibility matrix can be seen as a *semi-distance* (*qualitative*) matrix.

Let us also note that for every $x, y, z \in U$ we have

iv) $|\delta(x, x)| = 0$,

v) $|\delta(x, y)| = |\delta(y, x)|$,

vi) $|\delta(x, z)| \leq |\delta(x, y)| + |\delta(y, z)|$.

It is easily seen that the core is the set of all single element entries of the discernibility matrix $M(B)$, i.e.,

$$CORE)B) = \{a \in B : c_{ij} = \{a\}, \text{ for some } i, j\}$$

Obviously $B' \subseteq B$ is a reduct of B, if B' is the minimal (with respect to inclusion) subset of B such that

$$B' \cap c \neq \varnothing \text{ for any nonempty entry } c(c \neq \varnothing) \text{ in } M(B).$$

In other words reduct is the minimal subset of attributes that discerns all objects discernible by the whole set of attributes.

Every discernibility matrix $M(B)$ defines uniquely a *discernibility* (*boolean*) *function* $f(B)$ defined as follows.

Let us assign to each attribute $a \in B$ a binary Boolean variable $\overline{a}$, and let $\Sigma\delta(x,y)$ denote Boolean sum of all Boolean variables assigned to the set of attributes $\delta(x,y)$. Then the discernibility function can be defined by the formula

$$f(B) = \prod_{(x,y)\in U^2} \{\Sigma\delta(x,y) : (x,y) \in U^2 \text{ and } \delta(x,y) \neq \varnothing\}.$$

The following property establishes the relationship between disjunctive normal form of the function $f(B)$ and the set of all reducts of B.

All constituents in the minimal disjunctive normal form of the function $f(B)$ *are all reducts of* B.

In order to compute the value core and value reducts for x we can also use the discernibility matrix as defined before and the discernibility function, which must be slightly modified:

$$f^x(B) = \prod_{y\in U} \{\Sigma\delta(x,y) : y \in U \text{ and } \delta(x,y) \neq \varnothing\}.$$

Relative reducts and core can be computed also using discernibility matrix, which needs slight modification

$$c_{ij} = \{a \in C : a(x_i) \neq a(x_j) \text{ and } w(x_i, x_j)\},$$

where $w(x_i, x_j \equiv x_i \in POS_C(D)$ and $x_j \notin POS_C(D)$ or

$x_i \notin POS_C(D)$ and $x_j \in POS_C(D)$ or

$x_i, x_j \in POS_C(D)$ and $(x_j, x_j) \notin I(D)$

for $i, j = 1,2,\ldots,n$.

If the partition defined by D is definable by C then the condition $w(x_i, x_j)$ in the above definition can be reduced to $(x_i, x_j) \notin I(D)$.

Thus entry c_{ij} is the set of all attributes which discern objects x_i and x_j that do not belong to the same equivalence class of the relation $I(D)$.

The remaining definitions need little changes.

The D-core is the set of all single element entries of the discernibility matrix $M_D(C)$, i.e.,

$$CORE_D(C) = \{a \in C : c_{ij} = (a), \text{for some } i, j\}.$$

Set $C' \subseteq C$ is the D-reduct of C, if C' is the minimal (with respect to inclusion) subset of C such that

$$C' \cap c \neq \varnothing \text{ for any nonempty entry } c(c \neq \varnothing) \text{ in } M_D(C).$$

Thus D-reduct is the minimal subset of attributes that discerns all equivalence classes of the relation $I(D)$.

Every discernibility matrix $M_D(C)$ defines uniquely a *discernibility* (*Boolean*) *function* $f_D(C)$ which is defined as before. We have also the following property:

All constituents in the disjunctive normal form of the function $f_D(C)$ *are all D-reducts of* C.

For computing value reducts and the value core for relative reducts we use as a starting point the discernibility matrix $M_D(C)$ and discernibility function will have the form:

$$f_D^x(C) = \prod_{y \in U} \{\Sigma\delta(x,y) : y \in U \text{ and } \delta(x,y) \neq \varnothing\}.$$

Let us illustrate the above considerations by computing relative reducts for the set of attributes {*Headache*, *Muscle-pain*, *Temperature*} with respect to *Flu*.

The corresponding discernibility matrix is shown in Table 6.

	1	2	3	4	5	6
1						
2						
3						
4	*T*	*H*, *M*, *T*				
5	*H*, *M*		*M*, *T*			
6				*T*	*H*, *M*, *T*	

Table 6

In this table *H*, *M*, *T* denote *Headache*, *Muscle-pain* and *Temperatur*e, respectively.

The discernibility function for this table is

$$T(H+M)(H+M+T)(M+T),$$

where + denotes the boolean sum and the boolean multiplication is omitted in the formula.

After simplication the discernibility function using laws of Boolean algebra we obtain the following expression

$$TH + TH,$$

which says that there are two reducts *TH* and *TM* in the data table and *T* is the core.

9. Significance of Attributes and Approximate Reducts

As it follows from considerations concerning reduction of attributes, they cannot be equally important, and some of them can be eliminated from an information table without losing information contained in the table. The idea of attribute reduction can be generalized by introducing a concept of *significance of attributes*, which enables us evaluation of attributes not only by two-valued scale, *dispensable – indispensable*, but by assigning to an attribute a real number from the closed interval [0,1], expressing how important is an attribute in an information table.

Significance of an attribute can be evaluated by measuring effect of removing the attribute from an information table on classification defined by the table. Let us first start our consideration with decision tables.

Let *C* and *D* be sets of condition and decision attributes respectively and let *a* be a condition attribute, i.e., $a \in C$. As shown previously the number $\gamma(C,D)$ expresses a degree of consistency of the decision table, or 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 removing the attribute *a*, i.e., what is the difference between $\gamma(C,D)$ and $\gamma(C-\{a\},D)$. We can normalize the difference and define the significance of the attribute *a* as

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

and denoted simple by $\sigma(a)$, when *C* and *D* are understood.

Obviously $0 \le \sigma(a) \le 1$. The more important is the attribute *a* the greater is the number $\sigma(a)$. For example for condition attributes in Table 1 we have the following results:

σ(*Headache*) = 0,
σ(*Muscle-pain*) = 0,
σ(*Temperature*) = 0.75.

Because the significance of the attribute *Temperature* or *Muscle-pain* is zero, removing either of the attributes from condition attributes does not effect the set of consistent decision rules, whatsoever. Hence the attribute *Temperature* is the most significant one in the table. That means that by removing the attribute *Temperature*, 75% (three out of four) of consistent decision rules will disappear from the table, thus lack of the attribute essentially effects the "decisive power" of the decision table.

For a reduct of condition attributes, e.g., {*Headache*, *Temperature*}, we get

σ(*Headache*) = 0.25,
σ(*Temperature*) = 1.00.

In this case, removing the attribute *Headache* from the reduct, i.e., using only the attribute *Temperature*, 25% (one out of four) of consistent decision rules will be lost, and dropping the attribute *Temperature*, i.e., using only the attribute *Headache* 100% (all) consistent decision rules will be lost. That means that in this case making decisions is impossible at all, whereas by employing only the attribute *Temperature* some decision can be made.

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

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

denoted by ε(*B*), if *C* and *D* are understood, where *B* is a subset of *C*.

If *B* is a reduct of *C*, then ε(*B*) = 1, i.e., removing any reduct from a set of decision rules unables to make sure decisions, whatsoever.

Any subset *B* of *C* will be called an *approximate reduct* of *C*, and the number

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

denoted simple 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*. Obviously $\varepsilon(B) = 1 - \sigma(B)$ and $\varepsilon(B) = 1 - \varepsilon(C - B)$. For any subset *B* of *C* we have $\varepsilon(B) \le \varepsilon(C)$. If *B* is a reduct of *C*, then $\varepsilon(B) = 0$.

For example, either of attributes *Headache* and *Temperature* can be considered as approximate reducts of {*Headache*, *Temperature*}, and

ε(*Headache*) = 1,
ε(*Temperature*) = 0.25.

But for the whole set of condition attributes {*Headache*, *Muscle-pain*, *Temperature*} we have also the following approximate reduct

ε (*Headache*, *Muscle-pain*) = 0.75.

The concept of an approximate reduct is a generalization of the concept of a reduct considered previously. The minimal subset B of condition attributes C, such that $\gamma(C,D) = \gamma(B,D)$, or $\varepsilon_{(C,D)}(B) = 0$ is a reduct in the previous sense. The idea of an approximate reduct can be useful in cases when a smaller number of condition attributes is preferred over accuracy of classification.

10. Summary

Rough set Theory has found many applications in medical data analysis, finance, voice recognition, image processing and others. However the approach presented in this paper is too simple to many real-life applications and was extended in many ways by various authors. The detailed discussion of the above issues can be found in [5], [6] and the internet (e.g., http://www.roughsets.org)

References

[1] Z. Pawlak: Rough sets, International Journal of Computer and Information Sciences, 11, 341-356, 1982

[2] Z. Pawlak: Rough Sets – Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Boston, London, Dordrecht, 1991

[3] Z. Pawlak, A. Skowron: Rough membership functions, in: R. R Yaeger, M. Fedrizzi and J. Kacprzyk (eds.), *Advances in the Dempster Shafer Theory of Evidence*, John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, 1994, 251-271

[4] A. Skowron, C. Rauszer: The discernibility matrices and functions in information systems, in: R. Słowiński (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, Kluwer Academic Publishers, Dordrecht, 1992, 311-362

[5] A. Skowron *et al:* Rough set perspective on data and konwedge, Handbook of Data Mining and Knoledge Discovery (W. Klösgen, J. Żytkow eds.), Oxford University Press, 2002, 134-149

[6] L. Polkowski: Rough Sets – Mathematical Foundations, Advances in Soft Computing, Physica-Verlag, Springer-Verlag Company, 2002, 1-534

[7] L. Zadeh: Fuzzy sets, Information and Control 8, 333-353, 1965

CHAPTER 3

Rough Sets and Bayes' Theorem

1. Introduction

The Bayes' theorem is the essence of statistical inference.

"The result of the Bayesian data analysis process is the posterior distribution that represents a revision of the prior distribution on the light of the evidence provided by the data" [5].

"Opinion as to the values of Bayes' theorem as a basic for statistical inference has swung between acceptance and rejection since its publication on 1763" [4].

Rough set theory offers new insight into Bayes' theorem [7]. The look on Bayes' theorem offered by rough set theory is completely different to that used in the Bayesian data analysis philosophy. It does not refer either to prior or posterior probabilities, inherently associated with Bayesian reasoning, but it reveals some probabilistic structure of the data being analyzed. It states that any data set (decision table) satisfies total probability theorem and Bayes' theorem. This property can be used directly to draw conclusions from data without referring to prior knowledge and its revision if new evidence is available. Thus in the presented approach the only source of knowledge is the data and there is no need to assume that there is any prior knowledge besides the data. We simple look what the data are telling us. Consequently we do not refer to any prior knowledge which is updated after receiving some data.

Moreover, the rough set approach to Bayes' theorem shows close relationship between logic of implications and probability, which was first studied by Łukasiewicz [6] (see also [1]). Bayes' theorem in this context can be used to "invert" implications, i.e., to give reasons for decisions. This is a very important feature of utmost importance to data mining and decision analysis, for it extends the class of problem which can be considered in this domains.

Besides, we propose a new form of Bayes' theorem where basic role plays strength of decision rules (implications) derived from the data. The strength of decision rules is computed from the data or it can be also a subjective assessment. This formulation gives new look on Bayesian method of inference and also simplifies essentially computations.

2. Bayes' Theorem

"In its simplest form, if H denotes an hypothesis and D denotes data, the theorem says that

$$P(H|D) = P(D|H) \times P(H)/P(D).$$

With $P(H)$ regarded as a probabilistic statement of belief about H before obtaining data D, the left-hand side $P(H|D)$ becomes an probabilistic statement of belief about H after obtaining D. Having specified $P(D|H)$ and $P(D)$, the mechanism of the theorem provides a solution to the problem of how to learn from data.

In this expression, $P(H)$, which tells us what is known about H without knowing of the data, is called the *prior* distribution of H, or the distribution of H *priori*. Correspondingly,

$P(H|D)$, which tells us what is known about H given knowledge of the data, is called the *posterior* distribution of H given D, or the distribution of H a *posteriori*" [3].

"A prior distribution, which is supposed to represent what is known about unknown parameters before the data is available, plays an important role in Bayesian analysis. Such a distribution can be used to represent prior knowledge or relative ignorance" [4].

3. Information Systems and Decision Rules

Every decision table describes decisions (actions, results etc.) determined, when some conditions are satisfied. In other words each row of the decision table specifies a decision rule which determines decisions in terms of conditions.

In what follows we will describe decision rules more exactly.

Let $S = (U, C, D)$ be a decision table. Every $x \in U$ determines a sequence $c_1(x),\ldots,c_n(x), d_1(x),\ldots,d_m(x)$ where $\{c_1,\ldots,c_n\} = C$ and $\{d_1,\ldots,d_m\} = D$.

The sequence will be called a *decision rule induced by x* (in S) and denoted by $c_1(x),\ldots,c_n(x) \to d_1(x),\ldots,d_m(x)$ or in short $C \to_x D$.

The number $supp_x(C,D) = |C(x) \cap D(x)|$ will be called a *support* of the decision rule $C \to_x D$ and the number

$$\sigma_x(C,D) = \frac{supp_x(C,D)}{|U|},$$

will be referred to as the *strength* of the decision rule $C \to_x D$, where $|X|$ denotes the cardinality of X. With every decision rule $C \to_x D$ we associate the *certainty factor* of the decision rule, denoted $cer_x(C, D)$ and defined as follows:

$$cer_x(C,D) = \frac{|C(x) \cap D(x)|}{|C(x)|} = \frac{supp_x(C,D)}{|C(x)|} = \frac{\sigma_x(C,D)}{\pi(C(x))},$$

where $\pi(C(x)) = \frac{|C(x)|}{|U|}$.

The certainty factor may be interpreted as a conditional probability that y belongs to $D(x)$ given y belongs to $C(x)$, symbolically $\pi_x(D|C)$.

If $cer_x(C, D) = 1$, then $C \to_x D$ will be called a *certain decision* rule in S; if $0 < cer_x(C, D) < 1$ the decision rule will be referred to as an *uncertain decision rule* in S.

Besides, we will also use a *coverage factor* of the decision rule, denoted $cov_x(C, D)$ defined as

$$cov_x(C,D) = \frac{|C(x) \cap D(x)|}{|D(x)|} = \frac{supp_x(C,D)}{|D(x)|} = \frac{\sigma_x(C,D)}{\pi(D(x))}$$

where $\pi(D(x)) = \frac{|D(x)|}{|U|}$.

Similarly

$$cov_x(C,D) = \pi_x(C|D).$$

If $C \to_x D$ is a decision rule then $C \to_x D$ will be called an *inverse decision rule*. The inverse decision rules can be used to give explanations (*reasons*) for a decision.

Let us observe that

$$cer_x(C,D) = \mu^C_{D(x)}(x) \text{ and } cov_x(C, D).$$

That means that the certainty factor expresses the degree of membership of x to the decision class $D(x)$, given C, whereas the coverage factor expresses the degree of membership of x to condition class $C(x)$, given D.

4. Decision Language

It is often useful to describe decision tables in logical terms. To this end we define a formal language called a *decision language*.

Let $S = (U, A)$ be an information system. With every $B \subseteq A$ we associate a formal language, i.e., a set of formulas $For(B)$. Formulas of $For(B)$ are built up from attribute-value pairs (a, v) where $a \in B$ and $v \in V_a$ by means of logical connectives $\wedge$ (*and*), $\vee$ (*or*), $\sim$ (*not*) in the standard way.

For any $\Phi \in For(B)$ by $\|\Phi\|_S$ we denote the set of all objects $x \in U$ satisfying Φ in S and refer to as the *meaning* of Φ in S.

The meaning $\|\Phi\|_S$ of Φ in S is defined inductively as follows:
$\|(a, v)\|_S = \{x \in U : a(v) = x\}$ for all $a \in B$ and $v \in V_a$, $\|\Phi \vee \Psi\|_S = \|\Phi\|_S \cup \|\Psi\|_S$, $\|\Phi \wedge \Psi\|_S = \|\Phi\|_S \cap \|\Psi\|_S$, $\|\sim \Phi\|_S = U - \|\Phi\|_S$.

If $S = (U, C, D)$ is a decision table then with every row of the decision table we associate a decision rule, which is defined next.

A *decision rule* in S is an expression $\Phi \rightarrow_S \Psi$ or simply $\Phi \rightarrow \Psi$ if S is understood, read *if* Φ *then* Ψ, where $\Phi \in For(C)$, $\Psi \in For(D)$ and C, D are condition and decision attributes, respectively; Φ and Ψ are referred to as *conditions* part and *decisions* part of the rule, respectively.

The number $supp_S(\Phi,\Psi) = |(\|\Phi \wedge \Psi\|_S)|$ will be called the *support* of the rule $\Phi \rightarrow \Psi$ in S. We consider a probability distribution $p_U(x) = 1/|U|$ for $x \in U$ where U is the (non-empty) universe of objects of S; we have $p_U(X) = |X|/|U|$ for $X \subseteq U$. For any formula Φ we associate its probability in S defined by

$$\pi_S(\Phi) = p_U(\| \Phi \|_S).$$

With every decision rule $\Phi \rightarrow \Psi$ we associate a conditional probability

$$\pi_S(\Psi \mid \Phi) = p_U(\| \Psi \|_S \mid \| \Phi \|_S)$$

called the *certainty factor* of the decision rule, denoted $cer_S(\Phi,\Psi)$. This idea was used first by Łukasiewicz [6] (see also [1]) to estimate the probability of implications. We have

$$cer_S(\Phi,\Psi) = \pi_S(\Psi \mid \Phi) = \frac{|(\| \Phi \wedge \Psi \|_S)|}{|(\| \Phi \|_S)|}$$

where $\| \Phi \|_S \neq \varnothing$.

This coefficient is now widely used in data mining and is called *confidence coefficient*.

If $\pi_S(\Psi \mid \Phi) = 1$, then $\Phi \rightarrow \Psi$ will be called a *certain decision* rule; if $0 < \pi_S(\Psi \mid \Phi < 1$ the decision rule will be referred to as a *uncertain decision* rule.

There is an interesting relationship between decision rules and their approximations: certain decision rules correspond to the lower approximation, whereas the uncertain decision rules correspond to the boundary region.

Besides, we will also use a *coverage factor* of the decision rule, denoted $cov_S(\Phi, \Psi)$ defined by

$$\pi_S(\Phi \mid \Psi) = p_U(\| \Phi \|_S \mid \| \Psi \|_S).$$

Obviously we have

$$cov_S(\Phi, \Psi) = \pi_S(\Phi \mid \Psi) = \frac{|(\| \Phi \wedge \Psi \|_S)|}{|(\| \Psi \|_S)|}.$$

There are three possibilities to interpret the certainty and the coverage factors: statistical (frequency), logical (degree of truth) and mereological (degree of inclusion).

We will use here mainly the statistical interpretation, i.e., the certainty factors will be interpreted as the frequency of objects having the property Ψ in the set of objects having the property Φ and the coverage factor – as the frequency of objects having the property Φ in the set of objects having the property Ψ.

Let us observe that the factors are not assumed arbitrarily but are computed from the data.

The number

$$\sigma_S(\Phi, \Psi) = \frac{supp_S(\Phi, \Psi)}{U} = \pi_S(\Psi \mid \Phi) \cdot \pi_S(\Phi)$$

will be called the *strength* of the decision rule $\Phi \to \Psi$ in *S*, and will play an important role in our approach, which will be discussed in section 6.

We will need also the notion of an equivalence of formulas.

Let Φ, Ψ be formulas in *For*(*A*) where *A* is the set of attributes in $S = (U, A)$.

We say that Φ and Ψ are equivalent in *S*, or simply, equivalent if *S* is understood, in symbols $\Phi \equiv \Psi$, if and only if $\Phi \to \Psi$ and $\Psi \to \Phi$. It means that $\Phi \equiv \Psi$ if and only if $\| \Phi \| = \| \Psi \|$.

We need also approximate equivalence of formulas which is defined as follows:

$$\Phi \equiv_k \Psi \text{ and only if } cer(\Phi, \Psi) = cov(\Phi, \Psi) = k.$$

Besides, we define also approximate equivalence of formulas with the accuracy $\varepsilon (0 \le \varepsilon \le 1)$, which is defined as follows:

$$\Phi \equiv_{k,\varepsilon} \Psi \text{ if and only if } k = min\{(cer(\Phi, \Psi), cov(\Phi, \Psi)\} \text{ and } |cer(\Phi, \Psi) - cov(\Phi, \Psi)| \le \varepsilon.$$

5. Decision Algorithms

In this section we define the notion of a decision algorithm, which is a logical counterpart of a decision table.

Let $Dec(S) = \{\Phi_i \to \Psi_i\}_{i=1}^m$, $m \ge 2$, be a set of decision rules in a decision table $S = (U, C, D)$.

1) If for every $\Phi \to \Psi$, $\Phi' \to \Psi' \in Dec(S)$ we have $\Phi = \Phi'$ or $\| \Phi \wedge \Phi' \|_S = \varnothing$, and $\Psi = \Psi'$ or $\| \Psi \wedge \Psi' \|_S = \varnothing$, then we will say that $Dec(S)$ is the set of pairwise *mutually exclusive* (*independent*) decision rules in *S*.

2) If $\| \bigvee_{i=1}^{m} \Phi_i \|_S = U$ and $\| \bigvee_{i=1}^{m} \Psi_i \|_S = U$ we will say that the set of decision rules $Dec(S)$ *covers* U.

3) If $\Phi \to \Psi \in Dec(S)$ and $supp_S(\Phi, \Psi) \neq 0$ we will say that the decision rule $\Phi \to \Psi$ is *admissible* in S.

4) If $\bigcup_{X \in U/D} C_*(X) = \bigvee_{\Phi \to \Psi \in Dec^+(S)} \Phi \|_S$, where $Dec^+(S)$ is the set of all certain decision rules from $Dec(S)$, we will say that the set of decision rules $Dec(S)$ preserves the *consistency* part of the decision table $S = (U, C, D)$.

The set of decision rules $Dec(S)$ that satisfies 1), 2) 3) and 4), i.e., is independent, covers U, preserves the consistency of S and all decision rules $\Phi \to \Psi \in Dec(S)$ are admissible in S – will be called a *decision algorithm* in S.

Hence, if $Dec(S)$ is a decision algorithm in S then the conditions of rules from $Dec(S)$ define in S a partition of U. Moreover, the *positive region of D with respect to C*, i.e., the set

$$\bigcup_{X \in U/D} C_*(X)$$

is partitioned by the conditions of some of these rules, which are certain in S.

If $\Phi \to \Psi$ is a decision rule then the decision rule $\Psi \to \Phi$ will be called an *inverse* decision rule of $\Phi \to \Psi$.

Let $Dec^*(S)$ denote the set of all inverse decision rules of $Dec(S)$.

It can be shown that $Dec^*(S)$ satisfies 1), 2), 3) and 4), i.e., it is a decision algorithm in S.

If $Dec(S)$ is a decision algorithm then $Dec^*(S)$ will be called an *inverse* decision algorithm of $Dec(S)$.

The inverse decision algorithm gives *reasons* (explanations) for decisions pointed out by the decision algorithms.

A decision algorithm is a description of a decision table in the decision language.

Generation of decision algorithms from decision tables is a complex task and we will not discuss this issue here, for it does not lie in the scope of this paper. The interested reader is advised to consult the references.

6. Probabilistic Properties of Decision Tables

Decision tables have important probabilistic properties which are discussed next.

Let $C \to_x D$ be a decision rule in S and let $\Gamma = C(x)$ and $\Delta = D(x)$. Then the following properties are valid:

1) $$\sum_{y \in \Gamma} cer_y(C, D) = 1$$

2) $$\sum_{y \in \Gamma} cov_y(C, D) = 1$$

3) $$\pi(D(x)) = \sum_{y \in \Gamma} cer_y(C, D) \cdot \pi(C(y)) = \sum_{y \in \Gamma} \sigma_y(C, D)$$

4) $$\pi(C(x)) = \sum_{y \in \Delta} cov_y(C, D) \cdot \pi(D(y)) = \sum_{y \in \Delta} \sigma_y(C, D)$$

5) $$cer_x(C,D)=\frac{cov_x(C,D)\cdot\pi(D(x))}{\sum_{y\in\Gamma}cov_y(C,D)\cdot\pi(D(y))}=\frac{\sigma_x(C,D)}{\sum_{y\in\Delta}\sigma_y(C,D)}=\frac{\sigma_x(C,D)}{\pi(C(x))}$$

6) $$cov_x(C,D)=\frac{cer_x(C,D)\cdot\pi(C(x))}{\sum_{y\in\Gamma}cer_y(C,D)\cdot\pi(C(y))}=\frac{\sigma_x(C,D)}{\sum_{y\in\Gamma}\sigma_x(C,D)}=\frac{\sigma_x(C,D)}{\pi(D(x))}$$

That is, any decision table, satisfies 1),...,6). Observe that 3) and 4) refer to the well known *total probability theorem*, whereas 5) and 6) refer to *Bayes' theorem*.

Thus in order to compute the certainty and coverage factors of decision rules according to formula 5) and 6) it is enough to know the strength (support) of all decision rules only. The strength of decision rules can be computed from data or can be a subjective assessment.

7. Illustrative Example

Let us now consider an example, shown in Table 1.

Fact	*Disease*	*Age*	*Sex*	*Test*	*Support*
1	*yes*	*old*	*man*	+	400
2	*yes*	*middle*	*woman*	+	80
3	*no*	*old*	*man*	–	100
4	*yes*	*old*	*man*	–	40
5	*no*	*young*	*woman*	–	220
6	*yes*	*middle*	*woman*	–	60

Table 1

Attributes *disease, age* and *sex* are condition attributes, whereas *test* is the decision attribute.

We want to explain the test result in terms of patients state, i.e., to describe attribute *test* in terms of attributes *disease*, *age* and *sex.*

The strength, certainty and coverage factors for decision table are shown in Table 2.

Fact	*Strength*	*Certainty*	*Coverage*
1	0.44	0.92	0.83
2	0.09	0.56	0.17
3	0.11	1.00	0.24
4	0.04	0.08	0.10
5	0.24	1.00	0.52
6	0.07	0.44	0.14

Table 2

Below a decision algorithm associated with Table 1 is presented.

1) *if* (*disease*, *yes*) *and* (*age*, *old*) *then* (*test*, +)
2) *if* (*disease*, *yes*) *and* (*age*, *middle*) *then* (*test*, +)
3) *if* (*disease*, *no*) *then* (*test*, –)

4) *if* (*disease*, *yes*) *and* (*age*, *old*) *then* (*test*, −)

5) *if* (*disease*, *yes*) *and* (*age*, *middle*) *then* (*test*, −)

The certainty and coverage factors for the above algorithm are given in Table 3.

Rule	*Strengt*	*Certainty*	*Coverage*
1	0.44	0.92	0.83
2	0.09	0.56	0.17
3	0.36	1.00	0.76
4	0.04	0.08	0.10
5	0.24	0.44	0.14

Table 3

The certainty factors of the decision rules lead the following conclusions:

– 92% ill and old patients have positive test result

– 56% ill and middle age patients more positive test result

– all healthy patients have negative test result

– 8% ill and old patients have negative test result

– 44% ill and old patients have negative test result

In other words:

– ill and old patients most probably have positive test result (probability = 0.92)

– ill and middle age patients most probably have positive test result (probability = 0.56)

– healthy patients have certainly negative test result (probability = 1.00)

Now let us examine the inverse decision algorithm, which is given below:

1') *if* (*test*, +) *then* (*disease*, *yes*) *and* (*age*, *old*)

2') *if* (*test*, +) *then* (*disease*, *yes*) *and* (*age*, *middle*)

3') *if* (*test*, −) *then* (*disease*, *no*)

4') *if* (*test*, −) *then* (*disease*, *yes*) *and* (*age*, *old*)

5') *if* (*test*, −) *then* (*disease*, *yes*) *and* (*age*, *middle*)

Employing the inverse decision algorithm and the coverage factor we get the following explanation of test results:

– reason for positive test results are most probably patients disease and old age (probability = 0.83)

– reason for negative test result is most probably lack of the disease (probability = 0.76)

It follows from Table 2 that there are two interesting approximate equivalences of test results and the disease.

According to rule 1) the disease and old age are approximately equivalent to positive test result (k = 0.83, ε = 0.11), and lack of the disease according to rule 3) is approximately equivalent to negative test result (k = 0.76, ε = 0.24).

8. Summary

From the example it is easily seen the difference between employing Bayes' theorem in statistical reasoning and the role of Bayes' theorem in rough set based data analysis.

Bayesian inference consists in update prior probabilities by means of data to posterior probabilities.

In the rough set approach Bayes' theorem reveals data patterns, which are used next to draw conclusions from data, in form of decision rules.

References

[1] E. W. Adams: The Logic of Conditionals, an Application of Probability to Deductive Logic. D. Reidel Publishing Company, Dordrecht, Boston, 1975

[2] T. Bayes: An essay toward solving a problem in the doctrine of chances, Phil. Trans. Roy. Soc., 53, 370-418; (1763); Reprint Biometrika 45, 296-315, 1958

[3] J. M. Bernardo, A. F. M. Smith: Bayesian Theory, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1994

[4] G. E. P. Box, G. C. Tiao: Bayesian Inference in: Statistical Analysis, John Wiley and Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, 1992

[5] M. Berthold, D. J. Hand: Intelligent Data Analysis, an Introduction, Springer-Verlag, Berlin, Heidelberg, New York, 1999

[6] J. Łukasiewicz: Die logischen Grundlagen der Wahrscheinilchkeitsrechnung. Kraków, 1913, in: L. Borkowski (ed.), Jan Łukasiewicz – Selected Works, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw, 1970

[7] Z. Pawlak: Rough Sets and Decision Algorithms, in: W. Ziarko, Y. Y. Yao (eds.), Second International Conference, Rough Sets and Current Trends in Computing, RSCTC 2000, Banff, Canada, October 2000, LNAI 2000, 30-45

CHAPTER 4

Data Analysis and Flow Graphs

1. Introduction

In [2] Jan Łukasiewicz proposed to use logic as mathematical foundation of probability. He claims that probability is "purely logical concept" and that his approach frees probability from its obscure philosophical connotation. He recommends to replace the concept of *probability* by *truth values* of *indefinite propositions*, which are in fact propositional functions.

Let us explain this idea more closely. Let U be a non empty finite set, and let $\Phi(x)$ be a propositional function. The meaning of $\Phi(x)$ in U, denoted by $|\Phi(x)|$, is the set of all elements of U, that satisfies $\Phi(x)$ in U. The truth value of $\Phi(x)$ is defined as *card* $|\Phi(x)|$ / *card* U. For example, if $U = \{1, 2, 3, 4, 5, 6\}$ and $\Phi(x)$ is the propositional function $x > 4$, then the truth value of $\Phi(x) = 2/6 = 1/3$. If the truth value of $\Phi(x)$ is 1, then the propositional function is *true*, and if it is 0, then the function is *false*. Thus the truth value of any propositional function is a number between 0 and 1. Further, it is shown that the truth values can be treated as probability and that all laws of probability can be obtained by means of logical calculus.

In this paper we show that the idea of Łukasiewicz can be also expressed differently. Instead of using truth values in place of probability, stipulated by Łukasiewicz, we propose, in this paper, using of deterministic flow analysis in flow networks (graphs). In the proposed setting, flow is governed by some probabilistic rules (e.g., Bayes' rule), or by the corresponding logical calculus proposed by Łukasiewicz, though, the formulas have entirely deterministic meaning, and need neither probabilistic nor logical interpretation. They simply describe flow distribution in flow graphs. However, flow graphs introduced here are different from those proposed by Ford and Fulkerson [1] for optimal flow analysis, because they model rather, e.g., flow distribution in a plumbing network, than the optimal flow.

The flow graphs considered in this paper are basically meant not to physical media (e.g., water) flow analysis, but to information flow examination in decision algorithms. To this end branches of a flow graph are interpreted as decision rules. With every decision rule (i.e. branch) three coefficients are associated, the *strength*, *certainty* and *coverage factors*. In classical decision algorithms language they have probabilistic interpretation. Using Łukasiewicz's approach we can understand them as truth values. However, in the proposed setting they can be interpreted simply as flow distribution ratios between branches of the flow graph, without referring to their probabilistic or logical nature.

This interpretation, in particular, leads to a new look on Bayes' theorem, which in this setting, has entirely deterministic explanation.

The presented idea can be used, among others, as a new tool for data analysis, and knowledge representation.

We start our considerations giving fundamental definitions of a flow graph and related notions. Next, basic properties of flow graphs are defined and investigated. Further, the

relationship between flow graphs and decision algorithms is discussed. Finally, a simple tutorial example is used to illustrate the consideration.

2. Flow Graphs

A flow graph is a *directed, acyclic*, *finite* graph $G = (N, \mathtt{B}, \varphi)$, where N is a set of *nodes*, $\mathtt{B} \subseteq N \times N$ is a set of *directed branches,* $\varphi : \mathtt{B} \to R^+$ is a *flow function* and R^+ is the set of non-negative reals.

If $(x, y) \in \mathtt{B}$ then x is an *input* of y and y is an *output* of x.

If $x \in N$ then $I(x)$ is the set of all inputs of x and $O(x)$ is the set of all outputs of x.

Input and *output* of a graph G *are* defined $I(G) = \{x \in N : I(x) = \varnothing\}$, $O(G) = \{x \in N : O(x) = \varnothing\}$.

Inputs and outputs of G are *external nodes* of G; other nodes are *internal nodes* of G.

If $(x, y) \in \mathtt{B}$ then $\varphi(x, y)$ is a *troughflow* from x to y. We will assume in what follows that $\varphi(x, y) \neq 0$ for every $(x, y) \in \mathtt{B}$.

With every node x of a flow graph G we associate its *inflow*

$$\varphi_+(x) = \sum_{y \in I(x)} \varphi(y, x),$$

and *outflow*

$$\varphi_-(x) = \sum_{y \in O(x)} \varphi(x, y).$$

Similarly, we define an inflow and an outflow for the whole flow graph G, which are defined as

$$\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x)$$

$$\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x)$$

We assume that for any internal node x, $\varphi_+(x) = \varphi_-(x) = \varphi(x)$, where $\varphi(x)$ is a *troughflow* of node x.

Obviously, $\varphi_+(G) = \varphi_-(G) = \varphi(G)$, where $\varphi(G)$ is a *troughflow* of graph G.

The above formulas can be considered as *flow conservation equations* [1].

We will define now a *normalized flow graph.*

A normalized flow graph is a *directed, acyclic*, *finite* graph $G = (N, \mathtt{B}, \sigma)$, where N is a set of *nodes*, $\mathtt{B} \subseteq N \times N$ is a set of *directed branches* and $\sigma : \mathtt{B} \to \langle 0,1 \rangle$ is *a normalized flow* of (x, y) and

$$\sigma(x, y) = \frac{\varphi(x, y)}{\varphi(G)}$$

is *strength* of (x, y). Obviously, $0 \leq \sigma(x, y) \leq 1$. The strength of the branch expresses simply the percentage of a total flow through the branch.

In what follows we will use normalized flow graphs only, therefore by a flow graphs we will understand normalized flow graphs, unless stated otherwise.

With every node x of a flow graph G we associate its normalized *inflow* and *outflow* defined as

$$\sigma_+(x) = \frac{\varphi_+(x)}{\varphi(G)} = \sum_{y \in I(x)} \sigma(y,x),$$

$$\sigma_-(x) = \frac{\varphi_-(x)}{\varphi(G)} = \sum_{y \in O(x)} \sigma(x,y).$$

Obviously for any internal node x, we have $\sigma_+(x) = \sigma_-(x) = \sigma(x)$, where $\sigma(x)$ is a *normalized troughflow* of x.

Moreover, let

$$\sigma_+(G) = \frac{\varphi_+(G)}{\varphi(G)} = \sum_{x \in I(G)} \sigma_-(x),$$

$$\sigma_-(G) = \frac{\varphi_-(G)}{\varphi(G)} = \sum_{x \in O(G)} \sigma_+(x).$$

Obviously, $\sigma_+(G) = \sigma_-(G) = \sigma(G) = 1$.

3. Certainty and coverage factors

With every branch (x,y) of a flow graph G we associate the *certainty* and the *coverage factors* .

The *certainty* and the *coverage* of (x,y) are defined as

$$cer(x,y) = \frac{\sigma(x,y)}{\sigma(x)},$$

and

$$cov(x,y) = \frac{\sigma(x,y)}{\sigma(y)}.$$

respectively, where $\sigma(x) \neq 0$ and $\sigma(y) \neq 0$.

Below some properties, which are immediate consequences of definitions given above are presented:

1) $$\sum_{y \in O(x)} cer(x,y) = 1,$$

2) $$\sum_{x \in I(y)} cov(x,y) = 1,$$

3) $$\sigma(x) = \sum_{y \in O(x)} cer(x,y)\sigma(x) = \sum_{y \in O(x)} \sigma(x,y),$$

4) $$\sigma(y) = \sum_{x \in I(y)} cov(x,y)\sigma(y) = \sum_{x \in I(y)} \sigma(x,y),$$

5) $$cer(x,y) = \frac{cov(x,y)\sigma(y)}{\sigma(x)},$$

6) $$cov(x, y) = \frac{cer(x, y)\sigma(x)}{\sigma(y)}.$$

Obviously the above properties have a probabilistic flavor, e.g., equations (3) and (4) have a form of total probability theorem, whereas formulas (5) and (6) are Bayes' rules. However, these properties in our approach are interpreted in a deterministic way and they describe flow distribution among branches in the network.

A (*directed*) *path* from x to y, $x \neq y$ in G is a sequence of nodes $x_1,\ldots,x_n$ such that $x_1 = x$, $x_n = y$ and $(x_i, x_{i+1}) \in$B for every i, $1 \leq i \leq n$-1. A path from x to y is denoted by $[x...y]$.

The *certainty*, the *coverage* and the *strength* of the path $[x_1 \ldots x_n]$ are defined as

$$cer[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cer(x_i, x_{i+1}),$$

$$cov[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cov(x_i, x_{i+1}),$$

$$\sigma[x...y] = \sigma(x)\ cer[x...y] = \sigma(y)\ cov[x...y],$$

respectively.

The set of all paths from x to y ($x \neq y$) in G denoted $< x, y >$, will be called a *connection* from x to y in G. In other words, connection $< x, y >$ is a sub-graph of G determined by nodes x and y.

For every connection $< x, y >$ we define its certainty, coverage and strength as shown below:

$$cer < x, y > = \sum_{[x \ldots y] \in < x, y >} cer[x \ldots y],$$

the *coverage* of the connection $< x, y >$ is

$$cov < x, y > = \sum_{[x \ldots y] \in < x, y >} cov[x \ldots y],$$

and the *strength* of the connection $< x, y >$ is

$$\sigma < x, y > = \sum_{[x \ldots y] \in < x, y >} \sigma[x \ldots y] = \sigma(x) cer < x, y > = \sigma(y) cov < x, y >.$$

Let [x…y] be a path such that x and y are input and output of the graph G, respectively. Such a *path* will be referred to as *complete*.

The set of all complete paths from x to y will be called a *complete connection* from x to y in G. In what follows we will consider complete paths and connections only, unless stated otherwise.

Let x and y be an input and output of a graph G respectively. If we substitute for every complete connection $< x, y >$ in G a single branch (x, y) such $\sigma(x, y) = \sigma < x, y >$, $cer(x, y) = cer < x, y >$, $cov(x, y) = cov < x, y >$ then we obtain a new flow graph G' such that $\sigma(G) = \sigma(G')$. The new flow graph will be called a *combined* flow graph. The combined flow graph for a given flow graph represents a relationship between its inputs and outputs.

4. Dependencies in flow graphs

Let $(x, y) \in B$. Nodes x and y are independent on each other if

$$\sigma(x, y) = \sigma(x)\,\sigma(y).$$

Consequently

$$\frac{\sigma(x, y)}{\sigma(x)} = cer(x, y) = \sigma(y),$$

and

$$\frac{\sigma(x, y)}{\sigma(y)} = cov(x, y) = \sigma(x).$$

If

$$cer(x, y) > \sigma(y),$$

or

$$cov(x, y) > \sigma(x),$$

then x and y *depend positively* on each other. Similarly, if

$$cer(x, y) < \sigma(y),$$

or

$$cov(x, y) < \sigma(x),$$

then x and y *depend negatively* on each other.

Let us observe that relations of independency and dependencies are symmetric ones, and are analogous to that used in statistics.

For every $(x, y) \in B$ we define a *dependency factor* $\eta(x, y)$ defined as

$$\eta(x, y) = \frac{cer(x, y) - \sigma(y)}{cer(x, y) + \sigma(y)} = \frac{cov(x, y) - \sigma(x)}{cov(x, y) + \sigma(x)}.$$

It is easy to check that if $\eta(x, y) = 0$, then x and y are independent on each other, if $-1 < \eta(x, y) < 0$, then x and y are negatively dependent and if $0 < \eta(x, y) < 1$ then x and y are positively dependent on each other.

5. An Example

Now we will illustrate ideas introduced in the previous sections by means of a simple example concerning votes distribution of various age groups and social classes of voters between political parties.

Consider three disjoint age groups of voters y_1 (*old*), y_2 (*middle aged*) and y_3 (*young*) – belonging to three social classes x_1 (*high*), x_2 (*middle*) and x_3 (*low*). The voters voted for four political parties z_1 (*Conservatives*), z_2 (*Labor*), z_3 (*Liberal Democrats*) and z_4 (*others*)

Social class and age group votes distribution is shown in Fig. 1.

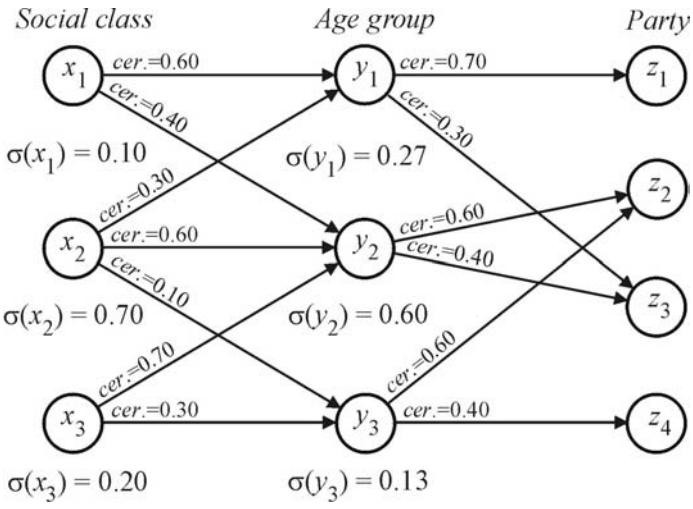

Fig. 1

First we want to find votes distribution with respect to age group. The result is shown in Fig.2.

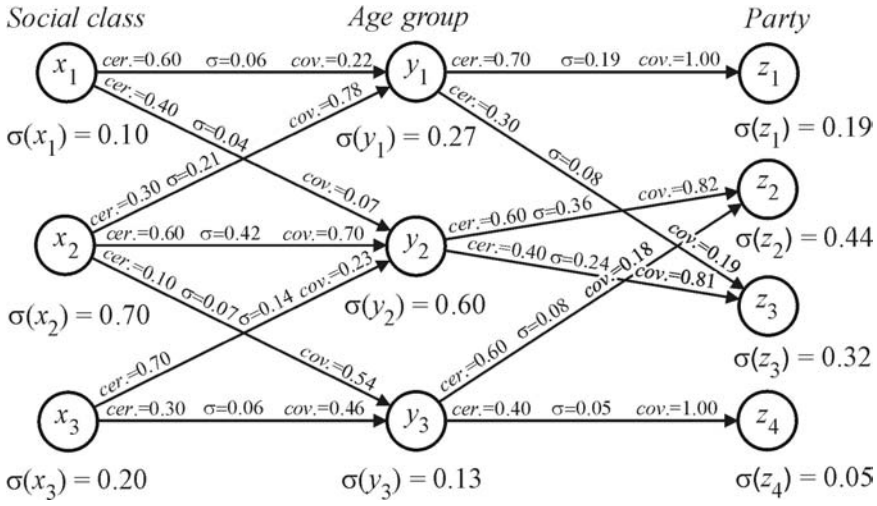

Fig. 2

From the flow graph presented in Fig. 2 we can see that, e.g., party z_1 obtained 19% of total votes, all of them from age group y_1; party z_2 – 44% votes, which 82% are from age group y_2 and 18% – from age group y_3, etc.

If we want to know how votes are distributed between parties with respects to social classes we have to eliminate age groups from the flow graph. Employing the algorithm presented in section 5 we get results shown in Fig. 3.

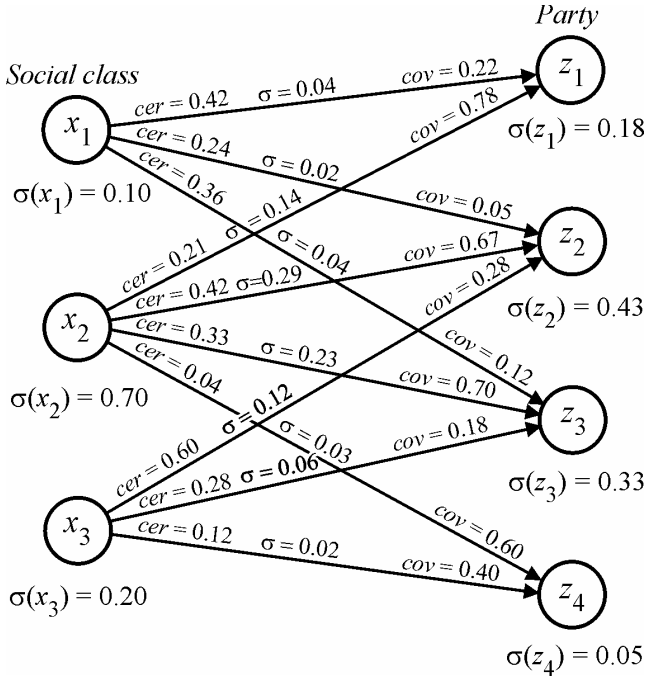

Fig. 3

From the flow graph presented in Fig. 3 we can see that party z_1 obtained 22% votes from social class x_1 and 78% – from social class x_2, etc.

We can also present the obtained results employing decision algorithms. For simplicity we present only some decision rules of the decision algorithm. For example, from Fig.2 we obtain decision rules:

If Party (z_1) *then Age group* (y_1) (0.19)

If Party (z_2) *then Age group* (y_2) (0.36)

If Party (z_2) *then Age group* (y_3) (0.08), etc.

The number at the end of each decision rule denotes strength of the rule.

Similarly, from Fig.3 we get:

If Party (z_1) *then Soc. class* (x_1) (0.04)

If Party (z_1) *then Soc. class* (x_2) (0.14), etc.

We can also invert decision rules and, e.g., from Fig. 3 we have:

If Soc. class (x_1) *then Party* (z_1) (0.04)

If Soc. class (x_1) *then Party* (z_2) (0.02)

If Soc. class (x_1) *then Party* (z_3) (0.04), *etc*

From the examples given above one can easily see the relationship between the role of *modus ponens* and *modus tollens* in logical reasoning and using flow graphs in reasoning about data.

Dependencies between Social class and Parties are shown in Fig. 4.

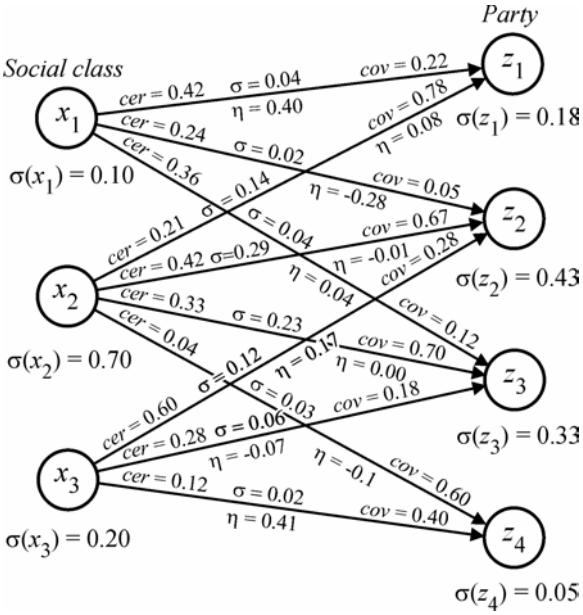

Fig.4

9. Summary

In this paper we have shown a new mathematical model of flow networks, which can be used to decision algorithm analysis. In particular it has been revealed a new interpretation of Bayes' theorem, where the theorem has entirely deterministic meaning, and can be used to decision algorithm study.

Besides, a new look of dependencies in databases, based on Łukasiewicz's ideas of independencies of logical formulas, is presented.

References

[1] L. R. Ford, D. R. Fulkerson: Flows in Networks, Princeton University Press, Princeton, New Jersey

[2] J. Łukasiewicz: Die logishen Grundlagen der Wahrscheinilchkeitsrechnung. Kraków (1913), in: L. Borkowski (ed.), Jan Łukasiewicz – Selected Works, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw, 1970

[3] Z. Pawlak: Flow graphs and decision algorithms, in: G. Wang, Q. Liu, Y. Y. Yao, A. Skowron (eds.), Proceedings of the Ninth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing RSFDGrC'2003), Chongqing, China, May 26-29, 2003, LNAI 2639, Springer-Verlag, Berlin, Heidelberg, New York, 1-11

Chapter 5

Rough Sets and Conflict Analysis

1. Introduction

Conflict analysis and resolution play an important role in business, governmental, political and lawsuits disputes, labor-management negotiations, military operations and others. To this end many mathematical formal models of conflict situations have been proposed and studied, e.g., [1-6].

Various mathematical tools, e.g., graph theory, topology, differential equations and others, have been used to that purpose.

Needless to say that game theory can be also considered as a mathematical model of conflict situations.

In fact there is no, as yet, "universal" theory of conflicts and mathematical models of conflict situations are strongly domain dependent.

We are going to present in this paper still another approach to conflict analysis, based on some ideas of rough set theory – along the lines proposed in [5]. We will illustrate the proposed approach by means of a simple tutorial example of voting analysis in conflict situations.

The considered model is simple enough for easy computer implementation and seems adequate for many real life applications but to this end more research is needed.

2. Basic concepts of conflict theory

In this section we give after [5] definitions of basic concepts of the proposed approach.

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

The pair $S = (U, v)$ will be called a *conflict* situation.

In order to express relations between agents we define three basic binary relations on the universe: *conflict*, *neutrality* and *alliance*. To this end we first define the following auxiliary function:

$$\phi_v(x,y) = \begin{cases} 1, & \text{if } v(x)v(y) = 1 \text{ or } x = y, \\ 0, & \text{if } v(x)v(y) = 0 \text{ and } x \neq y, \\ -1, & \text{if } v(x)v(y) = -1. \end{cases}$$

This means that, if $\phi_v(x,y) = 1$, agents x and y have the same opinion about issue v (are *allied* on v); if $\phi_v(x,y) = 0$ means that at least one agent x or y has neutral approach to issue a (is

neutral on *a*), and if $\phi_v(x,y) = -1$, means that both agents have different opinions about issue *v* (are in *conflict* on *v*).

In what follows we will define three basic relations R_v^+, R_v^0 and R_v^- on U^2 called *alliance*, *neutrality* and *conflict* relations respectively, and defined as follows:

$$R_v^+(x,y) \text{ iff } \phi_v(x,y) = 1,$$

$$R_v^0(x,y) \text{ iff } \phi_v(x,y) = 0,$$

$$R_v^-(x,y) \text{ iff } \phi_v(x,y) = -1.$$

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

(i) $R_v^+(x,x)$,

(ii) $R_v^+(x,y)$ implies $R_v^+(y,x)$,

(iii) $R_v^+(x,y)$ and $R_v^+(y,z)$ implies $R_v^+(x,z)$,

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

For the conflict relation we have the following properties:

(iv) not $R_v^-(x,x)$,

(v) $R_v^-(x,y)$ implies $R_v^-(y,x)$,

(vi) $R_v^-(x,y)$ and $R_v^-(y,z)$ implies $R_v^+(x,z)$,

(vii) $R_v^-(x,y)$ and $R_v^+(y,z)$ implies $R_v^-(x,z)$.

Conditions (vi) and (vii) refer to well known sayings "an enemy of my enemy is my friend" and "a friend of my enemy is my enemy".

For the neutrality relation we have:

(viii) not $R_v^0(x,x)$,

(ix) $R_v^0(x,y) = R_v^0(y,x)$.

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

The following property holds: $R_v^+ \cup R_v^0 \cup R_v^- = U^2$ because if $(x,y) \in U^2$ then $\Phi_v(x,y) = 1$ or $\Phi_v(x,y) = 0$ or $\Phi_v(x,y) = -1$ so $(x,y) \in R_v^+$ or $(x,y) \in R_v^0$ or $(x,y) \in R_v^-$. All the three relations R_v^+, R_v^0, R_v^- are pairwise disjoint, i.e., every pair of objects (*x*, *y*) belongs to exactly one of the above defined relations (is in conflict, is allied or is neutral).

With every conflict situation we will associate a *conflict graph* $G_S = (R_v^+, R_v^0, R_v^-)$.

An example of a conflict graph is shown in Fig. 1.

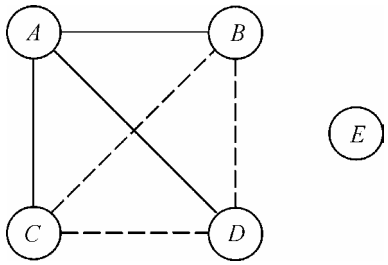

Fig. 1

Solid lines are denoting conflicts, doted line – alliance, and neutrality, for simplicity, is not shown explicitly in the graph. Of course, *B*, *C*, and D form a coalition.

3. An example

In this section we will illustrate the above presented ideas by means of a very simple tutorial example using concepts presented in the previous.

Table 1 presents a decision table in which the only condition attribute is *Party*, whereas the decision attribute is *Voting*. The table describes voting results in a parliament containing 500 members grouped in four political parties denoted *A*, *B*, *C* and *D*. Suppose the parliament discussed certain issue (e.g., membership of the country in European Union) and the voting result is presented in column *Voting*, where +, 0 and – denoted *yes*, *abstention* and *no* respectively. The column *support* contains the number of voters for each option.

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

Table 1

The strength, certainty and the coverage factors for Table 1 are given in Table 2.

Fact	*Strengt*	*Certaint*	*Coverage*
1	0.40	0.833	0.833
2	0.06	0.125	0.353
3	0.02	0.042	0.057
4	0.03	0.375	0.063
5	0.05	0.625	0.143
6	0.04	0.333	0.235
7	0.08	0.667	0.229
8	0.05	0.156	0.104
9	0.07	0.219	0.412
10	0.20	0.625	0.571

Table 2

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

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

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

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

The flow graph associated with Table 2 is shown in Fig. 2.

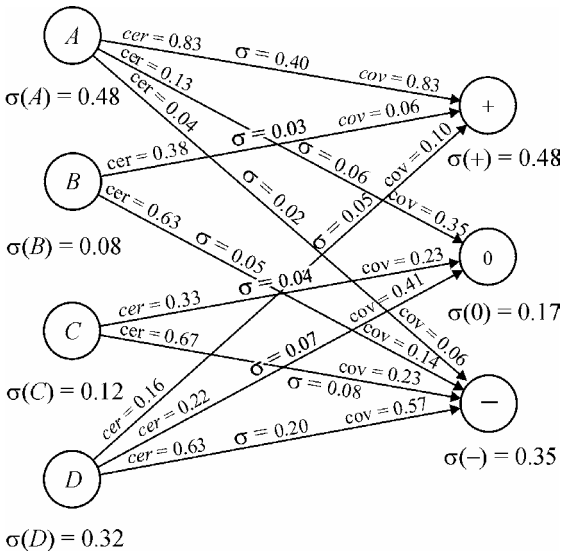

Fig. 2

Branches of the flow graph represent decision rules together with their certainty and coverage factors. For example, the decision rule $A \rightarrow 0$ has the certainty and coverage factors 0.125 and 0.353, respectively.

The flow graph gives a clear insight into the voting structure of all parties.

For many applications exact values of certainty of coverage factors of decision rules are not necessary. To this end we introduce “approximate” decision rules, denoted $C \Rightarrow D$ and read “*C mostly implies D*”. $C \Rightarrow D$ if and only if $cer(C, D) > 0.5$.

Thus we can replace flow graph shown in Fig. 2 by "*approximate*” flow graph presented in Fig. 3.

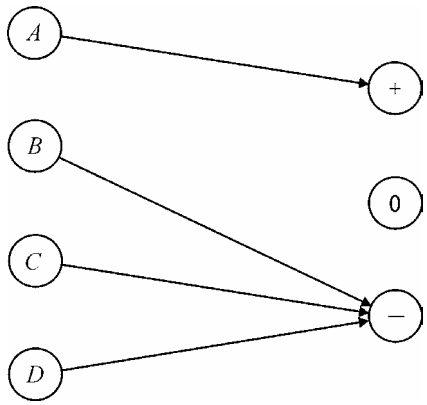

Fig. 3

From this graph we can see that parties *B*, *C* and *D* form a coalition, which is in conflict with party *A*, i.e., every member of the coalition is in conflict with party *A*. The corresponding conflict graph is shown in Fig. 4.

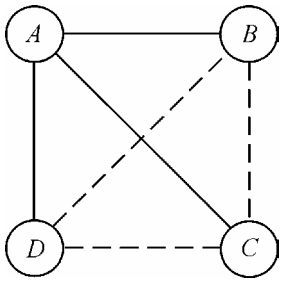

Fig. 4

Moreover from the flow graph shown in Figure 2 we can obtain an “inverse” approximate flow graph which is shown in Fig. 5.

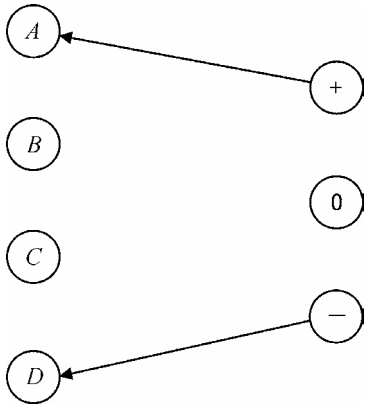

Fig. 5

This flow graph contains all inverse decision rules with certainty factor greater than 0.5. From this graph we can see that *yes* votes were obtained *mostly* from party *A* and *no* votes – *mostly* from party *D*.

We can also compute dependencies between parties and voting results the results are shown in Fig. 6.

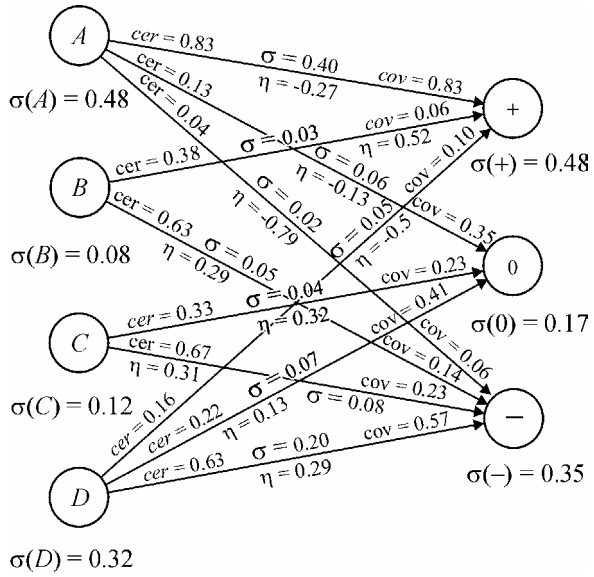

Fig. 6

4. Summary

It is shown that with any conflict situation a flow graph can be associated. Flow distribution in the graph can be used to study the relationship between agents involved in the conflict.

References

[1] J. L. Casti: Alternative Realities – Mathematical Models of Nature and Man, Wiley, New York, 1989

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

[3] R. Deja: Conflict analysis, rough set methods and applications, in: L. Polkowski, S Tsumoto, T.Y. Lin (eds.), Studies in Fuzzyness and Soft Computing, Physica-Verlag, A Springer-Verlag Company, 2000, 491-520

[4] Y. Maeda, K. Senoo, H. Tanaka: Interval density function in conflict analysis, in: N. Zhong, A. Skowron, S. Ohsuga, (eds.), New Directions in Rough Sets, Data Mining and Granular-Soft Computing, Springer, 1999, 382-389

[5] A. Nakamura: Conflict Logic with Degrees, Rough Fuzzy Hybridization – A New Trend in Decison-Making, (S. K. Pal, A. Skowron, eds.), Springer, 1999, 136-150

[6] Z. Pawlak: An inquiry into anatomy of conflicts, Journal of Information Sciences 109, 1998, 65-68

[7] F. Roberts: Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems, Englewood Clifs, 1976, Prince Hall.