Rough Set – Basic Concepts

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

1 Introduction

The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently it became also a crucial issue in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imperfect knowledge. The most popular one is, no doubt, fuzzy set theory [7].

Rough set theory [1] is 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 overlaps to a certain degree many other mathematical theories. Despite of the relationships rough set theory can be viewed in its own rights, as the independent discipline.

The theory 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 and data mining.

Rough set theory has been successfully applied in many real-life problems in medicine, pharmacology, engineering, banking, financial and market analysis and others.

The theory has many important advantages for data mining. Some of them are listed below.

- 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.

The aim of this paper is to give rudiments of rough set theory.

More about basics of the theory can be found in the referencies and the internet (http://www.roughsets.org).

2 Basic Philosophy

The rough set concept is a new mathematical approach to vagueness and uncertainty. The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (data, knowledge). E.g., if objects are patients suffering from a certain disease, symptoms of the disease form information about patients. Objects characterized by the same information are *indiscernible* (*similar*) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory.

Any set of all indiscernible (similar) objects is called an *elementary set*, and forms a basic *granule* (*atom*) of knowledge about the universe. Any union of some elementary sets is referred to as *crisp* (*precise*) set – otherwise the set is *rough* (*imprecise, vague*).

Consequently each rough set has *boundary-line cases*, i.e., objects which cannot be with certainty classified as members of the set or of its complement. Obviously crisp sets have no boundary-line elements at all. That means that boundary-line cases cannot be properly classified by employing the available knowledge.

Thus, the assumption that objects can be "seen" only through the information available about them leads to the view that knowledge has granular structure. Due to the granularity of knowledge some objects of interest cannot be discerned and appear as the same (or similar). As, a consequence vague concepts, in contrast to precise concepts, cannot be characterized in terms of information about their elements. Therefore in the proposed approach we assume that any vague concept is replaced by a pair of precise concepts – called the *lower* and the *upper approximation* of the vague concept. The lower approximation consists of all objects which *surely* belong to the concept and the upper approximation contains all objects which *possibly* belong to the concept. Obviously, the difference between the upper and the lower approximation constitute the *boundary region* of the vague concept. Approximations are two basic operations in the rough set theory.

3 An Example

For the sake of simplicity we first explain the above ideas intuitively, by means of a simple 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
p1	no	yes	high	yes
p2	yes	no	high	yes
p3	yes	yes	very high	yes
p4	no	yes	normal	no
p5	yes	no	high	no
p6	no	yes	very high	yes

Table 1

Columns of the table are labelled 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 p2 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 p2, p3 and p5 are indiscernible with respect to the attribute Headache, patients p3 and p6 are indiscernible with respect to attributes Muscle-pain and Flu, and patients p2 and p5 are indiscernible with respect to attributes Headache, Muscle-pain and Temperature. Hence, for example, the attribute Headache generates two elementary sets {p2, p3, p5} and {p1, p4, p6}, whereas the attributes Headache and Muscle-pain form the following elementary sets: {p1, p4, p6}, {p2, p5} and {p3}. Similarly one can define elementary sets generated by any subset of attributes.

Patient p2 has flu, whereas patient p5 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 p2 and p5 are the boundary-line cases, which cannot be properly classified in view of the available knowledge. The remaining patients p1, p3 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 p4 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 {p1, p3, p6} and the upper approximation of this set is the set {p1, p2, p3, p5, p6}, whereas the boundary-line cases are patients p2 and p5. Similarly p4 does not have flu and p2, p5 cannot be exclude as having flu, thus the lower approximation of this concept is the set {p4} whereas – the upper approximation – is the set {p2, p4, p5} and the boundary region of the concept "not flu" is the set {p2, p5}, the same as in the previous case.

4 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 refereed 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.

Let us define now the following two operations on sets

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

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

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.

One can show the following properties of approximations:

(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) $B^*(X \cap Y) \subseteq B^*(X) \cap B^*(Y)$,

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

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

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

d) $B_*(X) = \emptyset$ 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 $B_*(X) = \{p1, p3, p6\} \neq \emptyset$ and $B^*(X) = \{p1, p2, p3, p5, p6\} \neq U$. For this case we get $\alpha_B(\text{"}flu\text{"}) = 3/5$. 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 $B_*(X) = \emptyset$ 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.

5 Rough Sets and Membership Function

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

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

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

The above properties show clearly the difference between fuzzy and rough 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.

6 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 deteremines 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, nonconflicting*). 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, nonconflicting*).

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 tables.

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.

7 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 not total dependencies whatsoever. If in Table 1, the value of the attribute Temperature for patient $p5$ 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 $(0 \leq k \leq 1)$, 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} $\Rightarrow$ {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} $\Rightarrow$ {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} $\Rightarrow$ {Flu} and {Muscle-pain} $\Rightarrow$ {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* $k, 0 \leq k \leq 1$, 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.

8 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 a 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 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 3

Patient	Headache	Temperature	Flu
p1	no	high	yes
p2	yes	high	yes
p3	yes	very high	yes
p4	no	normal	no
p5	yes	high	no
p6	no	very high	yes

Table 3

or Table 4

Patient	Muscle-pain	Temperature	Flu
p1	yes	high	yes
p2	no	high	yes
p3	yes	very high	yes
p4	yes	normal	no
p5	no	high	no
p6	yes	very high	yes

Table 4

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 vlaue 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 3 and Table 4 can be simplified as follow

Patient	Headache	Temperature	Flu
p1	no	high	yes
p2	yes	high	yes
p3	–	very high	yes
p4	–	normal	no
p5	yes	high	no
p6	–	very high	yes

Table 5

Patient	Muscle-pain	Temperature	Flu
p1	yes	high	yes
p2	no	high	yes
p3	–	very high	yes
p4	–	normal	no
p5	no	high	no
p6	–	very high	yes

Table 6

We can also present the obtained results in a form of a decision algorithm. For Table 5 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 6 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.

9 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:

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

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) $\delta(x, x) = \emptyset$,

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

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

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

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

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:

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

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 calss 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

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

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

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:

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

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 7.

	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 7

In this table H, M, T denote Headache, Muscule-pain and Temperature, respectively.

The discernibility dunction for this table is

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

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

After simplication the discernibility function using laws of boolean algebra we abtain the following expression

$$TH + TH,$$

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

10 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 loosing 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 $0 \leq \sigma(a) \leq 1$. 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 attribute 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) consistent decision rule 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 $\varepsilon(B)$, if C and D are understood, where B is a subset of C.

If B is a reduct of C, then $\varepsilon(B) = 1$, 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 $\varepsilon(B) \leq \varepsilon(C)$. 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. A minimal subset B of condition attributes C, such that $\gamma(C, D) = \gamma(B, D)$, or $\varepsilon_{(C,D)}(B) = 0$ is a reduct 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.

11 Conclusions

As mentioned in the introduction the rough set methodology 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 the references and the internet.

References

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

[2] Pawlak Z., (1991), *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Boston, London, Dordrecht.

[3] Pawlak Z., and Skowron A., (1994), "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, 251–271.

[4] Skowron A.,and Rauszer, C., (1992), "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, 311–362.

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

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

[7] Zadeh L., (1965), "Fuzzy sets", Information and Control 8, 333–353.