Rough Sets - a New Way of Data Analysis

Zdzisław Pawlak
Institute of Computer Science
Warsaw University of Technology
00-665 Warsaw, ul Nowowiejska 15/19
e-mali: zpw@ii.pw.edu.pl

Abstract

Rough set theory is a new mathematical approach to data analysis. In this paper basic concepts of rough set theory will be defined and the methodology of applications briefly discussed. A tutorial illustrative example will be used to make the introduced notions more intuitive.

1 Introduction

Rough set theory is a new mathematical approach to data analysis.

Rough set philosophy is founded on the assumption that with every object of the *universe of discourse* some information is associated Objects characterized by the same information are *indiscernible*. The indiscernibility relation 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 a *crisp* (*precise*) set - otherwise the set is *rough* (*imprecise, vague*). Each rough set has *boundary-line* cases, i.e., objects which cannot be with certainty classified, by employing the available knowledge, as members of the set or its complement. Obviously rough sets, in contrast to precise sets, cannot be characterized in terms of information about their elements.

In the proposed approach with any rough set a pair of precise sets - called the *lower* and the *upper approximation* of the rough set is associated. The lower approximation consists of all objects which *surely* belong to the set and the upper approximation contains all objects which *possibly* belong to the set. The difference between the upper and the lower approximation constitutes the *boundary region* of the rough set. Approximations are two basic operations in the rough set theory and are employed to define *dependencies* (*total or partial*) between attributes, *reduction of attributes*, *decision rule generation* and other.

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. It seems of particular importance to decision support systems and data mining.

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

set approach seems also important for various engineering applications, like machine diagnosis, process control, material sciences etc..

Rough set approach to data analysis has many important advantages. 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.

This paper gives rudiments of rough set theory. The basic concepts of the theory are illustrated by a simple tutorial example. In order to tackle many sophisticated real life problems the theory has been generalized in various ways but we will not discuss these generalizations here. More about rough set theory and its applications can be found in the references.

2 Information systems

Starting point of rough set based data analysis is a data set, called an information system.

An information system is a data table, whose columns are labeled by attributes, rows are labeled by objects of interest and entries of the table are attribute values.

Formally, by an *information system* we will understand a pair $S = (U, A)$, where U and A, are finite, nonempty sets called the *universe*, and the set of *attributes,* respectively. 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 defined as follows: $(x, y) \in I(B)$ 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., a partition determined by B, will be denoted by $U/I(B)$, or simply by 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* (*indiscernible with respect to B*). Equivalence classes of the relation $I(B)$ (or blocks of the partition U/B) are referred to as *B-elementary sets* or *B-granules*.

If we distinguish in an information system two disjoint classes of attributes, called *condition* and *decision attributes*, respectively, then the system will be called a *decision table* and will be denoted by $S = (U, C, D)$, where C and D are disjoint sets of condition and decision attributes, respectively.

3 Approximation

Suppose we are given an information system $S = (U, A)$, $X \subseteq U$, and $B \subseteq A$. Our task is to describe the set X in terms of attribute values from B. To this end we define two operations assigning to every $X \subseteq U$ two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X, respectively, and defined as follows:

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

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

Hence, the B-lower approximation of a set is the union of all B-granules that are included in the set, whereas the *B-upper* approximation of a set is the union of all B-granules that have a nonempty intersection with the set. 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 X is *crisp* (*exact*) with respect to B; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, X is referred to as *rough* (*inexact*) with respect to B.

4 Dependency of attributes and reducts

Next important concept in rough set theory is the dependency 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 C and D.

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

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

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

where

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

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

Obviously

$$\gamma(C, D) = \sum_{X \in U/D} \frac{card(C_*(X))}{card(U)}.$$

If $k = 1$ we say that D *depends totally* on C, and if $k < 1$, we say that D *depends partially* (in a *degree* k) on C. The degree of dependency $\gamma(C, D)$ can be also understood as a *consistency measure* of the decision table $S = (U, C, D)$.

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

Observe 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.

Next important task in rough set based data analysis is data reduction.

To this end we remove from the data table superfluous attributes.

Let $C, D \subseteq A$, be sets of condition and decision attributes respectively. We will say that $C' \subseteq C$ is a *D-reduct* (reduct with *respect* to D) of C, if C' is a minimal subset of C such that

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

There are also other concepts of reducts but we will not discuss them hare.

The intersection of all D-reducts is called a *D-core* (core with *respect* to D).

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, the non-empty core consists of the most important subset of attributes, for none of its elements can be removed without affecting the classification power of attributes.

5 Decision rules

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 $v \in V_a$, $||\Phi \vee \Psi||_S = ||\Phi||_S \cup ||\Psi||_S$, $||\Phi \wedge \Psi||_S = ||\Phi||_S \cap ||\Psi||_S$, $|| \sim \Phi||_S = U - ||\Phi||_S$.

A formula Φ is *true* in S if $||\Phi||_S = U$.

A *decision rule* in S is an expression $\Phi \rightarrow \Psi$, 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* and *decisions* of the rule, respectively.

A decision rule $\Phi \rightarrow \Psi$ is *true* in S if $||\Phi||_S \subseteq ||\Psi||_S$.

We consider a probability distribution $p_U(x) = 1/card(U)$ for $x \in U$ where U is the (non-empty) universe of objects of S; we have $p_U(X) = card(X)/card(U)$ 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

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

that Ψ is true in S given Φ is true in S called the *certainty factor* to estimate the probability of implications. We have

$$\pi_S(\Psi|\Phi) = \frac{card(||\Phi \wedge \Psi||_S)}{card(||\Phi||_S)}$$

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

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

Obviously, $\pi_S(\Psi|\Phi) = 1$ if and only if $\Phi \to \Psi$ is true in S.

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

Besides, we will also use a *coverage factor* defined by

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

which is the conditional probability that Φ is true in S, given Ψ is true in S with the probability $\pi_S(\Psi)$. Obviously we have

$$\pi_S(\Phi|\Psi) = \frac{card(||\Phi \wedge \Psi||_S)}{card(||\Psi||_S)}.$$

The certainty factors in S can be also 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 certain decision rules are associated with the lower approximation, whereas the possible decision rules correspond to the boundary region of a set. Thus instead of approximations we can use decision rules, which form convenient formal language for descriptions of properties of sets.

6 Decision algorithm

Let $Dec(S) = \{\Phi_i \to \Psi_i\}_{i=1}^m$, $m \geq 2$, 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 $||\Phi \wedge \Phi'||_S = \emptyset$, and $\Psi = \Psi'$ or $||\Psi \wedge \Psi'||_S = \emptyset$, then we will say that $Dec(S)$ is the set of *mutually disjoint* (*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 $||\Phi \wedge \Psi||_S \neq \emptyset$ we will say that the decision rule $\Phi \to \Psi$ is *admissible* in S. $card(||\Phi \wedge \Psi||_S)$ will be called the *support* of the rule in S and will be denoted by $supp_S(\Phi, \Psi)$ or in short $supp(\Phi, \Psi)$.

4) If $\bigcup_{X \in U/D} C_*(X) = ||\bigvee_{\Phi \in Dec^+(S)} \Phi||_S$ 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* 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 $\Phi \rightarrow \Psi$ is a decision rule then the decision rule $\Psi \rightarrow \Phi$ will be called an *inverse* decision rule of $\Phi \rightarrow \Psi$.

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 an decision algorithm in S.

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

Let $Dec(S)$ be a decision algorithm in S. The number

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

will be called the *strength* of the decision rule $\Phi \rightarrow \Psi$ in S and

$$\eta(Dec(S)) = \sum_{\Phi \rightarrow \Psi \in Dec(S)} max\{\sigma_S(\Phi, \Psi)\}_{\Psi \in D(\Phi)}$$

where $D(\Phi) = \{\Psi : \Phi \rightarrow \Psi \in Dec(S)\}$ will be referred to as the *efficiency* of the decision algorithm $Dec(S)$ in S, where the sum is stretched over all decision rules in the algorithm.

The efficiency of a decision algorithm is the probability (ratio) of all objects of the universe, that are classified to decision classes, by means of the decision rule $\Phi \rightarrow \Psi$ with maximal strength $\sigma_S(\Phi, \Psi)$ among rules $\Phi \rightarrow \Psi \in Dec(S)$ with satisfied Φ on these objects.

7 Illustrative example

Now we will illustrate the concepts introduced previously by means of a simple example shown in Table 1.

Table 1: An example of an information system

F	*driving conditions*			*consequence*	*Support*
	weather	*road*	*time*	*accident*	
1	*misty*	*icy*	*day*	*yes*	6
2	*foggy*	*icy*	*night*	*yes*	8
3	*misty*	*not icy*	*night*	*yes*	5
4	*sunny*	*icy*	*day*	*no*	55
5	*foggy*	*not icy*	*dusk*	*yes*	11
6	*misty*	*not icy*	*night*	*no*	15

In Table 1 six facts concerning a hundred cases of driving a car in various weather conditions are presented. In the table *W (weather)*, *R (road)* and *T (time)*, are *condition attributes*, represent driving conditions, *A (accident)*, is a *decision attribute*, giving information whether an accident has occurred or not. N is the number of similar cases.

Each row of the decision table determines a decision obeyed when specified conditions are satisfied.

It is easy to compute that there are two reducts {*weather, road* } and { *weather, time* } of condition attributes. Thus the core is the attribute *weather*.

An example of decision algorithm associated with Table 1 is given below:

1) $(W,misty) \wedge (R,icy) \rightarrow (A,yes)$

2) $(W,foggy) \rightarrow (A,yes)$

3) $(W,misty) \wedge (R,not\ icy) \rightarrow (A,yes)$

4) $(W,sunny) \rightarrow (A,no)$

5) $(W,misty) \wedge (R,not\ icy) \rightarrow (A,no)$

Below the relationship between approximations and decision rules is given.

- Certain rules describing accidents (the lower approximation of the set of facts {1,2,3,5})

  1) $(W,misty) \wedge (R,icy) \rightarrow (A,yes)$
  2) $(W,foggy) \rightarrow (A,yes)$

- Uncertain rule describing accidents (the boundary region {3,6} of the set of facts {1,2,3,5})

  3) $(W,misty) \wedge (R,not\ icy) \rightarrow (A,yes)$

- Certain rule describing lack of accidents (the lower approximation of the set of facts {4,6})

  4) $(W,sunny) \rightarrow (A,no)$

- Uncertain rule describing lack of accidents (the boundary region {3,6} of the set of facts {4,6})

  5) $(W,misty) \wedge (R,not\ icy) \rightarrow (A,no)$

The certainty and coverage factors for decision rules 1),...,5) are presented in Table 2.

Table 2: Certainty and coverage factors

rule no.	*certainty*	*coverage*	*support*	*strength*
1	1.00	0.20	6	0.06
2	1.00	0.63	19	0.19
3	0.25	0.17	5	0.05
4	1.00	0.79	55	0.55
5	0.75	0.21	15	0.15

The following inverse decision rules can be understood as explanation of car accidents in terms of weather conditions:

1') $(A,yes) \rightarrow (R,icy) \wedge (W,misty)$

2') $(A,yes) \rightarrow (W,foggy)$

3') $(A,yes) \rightarrow (R,not\ icy) \wedge (W,misty)$

4') $(A,no) \rightarrow (W,sunny)$

5') $(A,no) \rightarrow (R,not\ icy) \wedge (W,misty)$

Summing up, the decision algorithm leads to the following conclusions:

- *misty weather and icy road or foggy weather surely cause accident*,
- *misty weather and road not icy most probably does not cause accident*,
- *sunny weather surely does not causes accident*.

The inverse decision algorithm gives the following explanations:

- *foggy weather most probably causes accident*,
- *sunny weather most probably does not cause accident*.

Observe that the results are valid for the data given in Table 1 only. Weather the results are valid for new cases it depends if Table 1 is representative sample of a bigger number of cases or not.

8 Conclusions

In the paper basic concepts of rough set theory have been presented and briefly discussed. The theory attracted attension both theoretitions and practioners world wide and proved to be a valuable new method of data analysis.

References

[1] Grzymała-Busse J. Managing Uncertainty in Expert Systems. Kluwer Academic Publishers, Boston, Dordrecht, 1991.

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

[3] L. Polkowski, A. Skowron (eds.): Proceedings of the First International Conference Rough Sets and Current Trends in Computing (RSCTC'98), Warsaw, Poland, June, Lecture Notes in Artificial Intelligence 1424, Springer–Verlag, Berlin, 1998.

[4] L. Polkowski, A. Skowron (eds.): Rough Sets in Knowledge Discovery Vol. 1–2, Physica-Verlag, Heidelberg, 1998.

[5] L. Polkowski, S. Tsumoto, T. Y. Lin (eds) Rough Set Method and Applications. New Development in Knowledge Discovery in Information Systems. Physica-Verlag, Heidelberg, (to appear)

[6] N. Zhong, A. Skowron, S. Ohsuga (eds.): Proceedings of 7th International Workshop: New Directions in Rough Sets, Data Mining, and Granular –Soft Computing (RSFDGSC'99), Yamaguchi, Japan, November 1999, Lecture Notes in Artificial Intelligence 1711 Springer–Verlag, Berlin, 1999, 1–9.

MORE INFO ABOUT ROUGH SETS CAN BE FOUND IN:

http://www.cs.uregina.ca/ roughset
http://www.infj.ulst.ac.uk /staff/I.Duentsch
http://www-idss.cs.put.poznan.pl/staff/slowinski/
http://alfa/logic/