Chapter 1
Elementary Rough Set Granules: Toward a Rough Set Processor

Zdzisław Pawlak

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland
and
University of Information Technology and Management,
Newelska 6, 01-477 Warsaw, Poland
zpw@ii.pw.edu.pl

Summary. In this chapter, the basics of the rough set approach are presented, and an outline of an exemplary processor structure is given. The organization of a simple processor is based on elementary rough set granules and dependencies between them. The rough set processor (RSP) is meant to be used as an additional fast classification unit in ordinary computers or as an autonomous learning machine. In the latter case, the RSP can be regarded as an alternative to neural networks.

1 Introduction

Rough set theory [4] has proved its effectiveness in drawing conclusions from data [6]. However, to take full advantage of the theory in data analysis, adequate processor organization is necessary. The architecture of such processors was proposed first in [4]. In this chapter, another proposal for rough set processor organization is presented.

Rough-set-based data analysis starts from a decision table, which is a data table. The columns of a decision table are labeled with attributes; the rows are labeled with objects of interest; and attribute values are entered in the data cells of the table. Attributes of the decision table are divided into two disjoint groups called condition and decision attributes, respectively. Each row of a decision table induces a decision rule, which specifies the decision (action, results, outcome, etc.) if some conditions are satisfied. If a decision rule uniquely determines a decision in terms of conditions, the decision rule is certain. Otherwise the decision rule is uncertain. Decision rules are closely connected with approximations, which are basic concepts of rough set theory. Roughly speaking, certain decision rules describe the lower approximation of decisions in terms of conditions, whereas uncertain decision rules refer to the upper approximation of decisions.

Two conditional probabilities, called the certainty and the coverage coefficient, are associated with every decision rule. The certainty coefficient expresses the conditional probability that an object belongs to the decision class specified by the decision rule, given that it satisfies the conditions of the rule. The coverage coefficient gives the conditional probability of the reasons for a given decision.

It turns out that the certainty and coverage coefficients satisfy Bayes' theorem. This gives new insight into the interpretation of Bayes' theorem, showing that Bayes' theorem can be used differently for drawing conclusions from data than the use offered by classical Bayesian inference philosophy [5].

This idea is at the foundation of rough set processor organization. In this chapter, the basics of rough set theory are presented, and an outline of an exemplary processor structure is given. The rough set processor is meant to be used as a "rough" classifier, or as a learning machine, and can be regarded as an alternative to neural networks.

2 Information Systems and Decision Tables

In this section we define the basic concept of rough set theory: information systems. The rudiments of rough set theory can be found in [4, 6]. An information system is a data table whose columns are labeled with attributes, rows are labeled with objects of interest, and attribute values are entered in the data cells of the table.

Formally, the *information system* is a pair $S = (U,A)$, where U and A are nonempty finite sets called the *universe* of objects and the set of *attributes*, respectively, such that $a : U \to V_a$, where V_a is the set of all *values* of a, called the *domain* of a, for each $a \in A$. 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:

$$(x,y) \in I(B) \quad \text{if and only if} \quad a(x) = a(y) \quad \text{for every} \quad a \in A,$$

where $a(x)$ denotes the value of the attribute a for the 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., the 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 objects* (*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-elementary granules*.

If we distinguish in the information system two disjoint classes of attributes, called *condition* and *attribute decision*, 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, and $C \cup D = A$. $C(x)$ and $D(x)$ will be referred to as the condition class and the decision class induced by x, respectively.

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

An example of a simple decision table is shown in Table 1. In the table, *age*, *sex*, and *profession* are condition attributes, whereas *disease* is the decision attribute.

Table 1. Decision table

Decision rule	*Age*	*Sex*	*Profession*	*Disease*
1	*Old*	*Male*	*Yes*	*No*
2	*Med.*	*Female*	*No*	*Yes*
3	*Med.*	*Male*	*Yes*	*No*
4	*Old*	*Male*	*Yes*	*Yes*
5	*Young*	*Male*	*No*	*No*
6	*Med.*	*Female*	*No*	*No*

The table contains data on the relationship among age, sex, and profession and a certain vocational disease. Decision tables can be simplified by removing superfluous attributes and attribute values, but we will not consider this issue in this chapter.

3 Decision Rules

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, $C \underset{x}{\longrightarrow} D$. The number $supp_x(C,D) = |A(x)| = |C(x) \cap D(x)|$ will be called the *support* of the decision rule $C \underset{x}{\longrightarrow} D$, and the number,

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

will be referred to as the *strength* of the decision rule $C \underset{x}{\longrightarrow} D$, where $|X|$ denotes the cardinality of X. Another decision table is shown in Table 2. This decision table can be understood as an abbreviation of a bigger decision table containing 1100 rows. Support of the decision rule means the number of identical decision rules in the original decision table.

Table 2. Support and strength

Decision rule	*Age*	*Sex*	*Profession*	*Disease*	*Support*	*Strength*
1	*Old*	*Male*	*Yes*	*No*	200	0.18
2	*Med.*	*Female*	*No*	*Yes*	70	0.06
3	*Med.*	*Male*	*Yes*	*No*	250	0.23
4	*Old*	*Male*	*Yes*	*Yes*	450	0.41
5	*Young*	*Male*	*No*	*No*	30	0.03
6	*Med.*	*Female*	*No*	*No*	100	0.09

With every decision rule $C \underset{x}{\longrightarrow} D$, we associate the *certainty factor* of the decision rule, denoted $cer_x(C,D)$ and defined as follows:

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

where $\pi\left[C(x)\right] = \frac{|C(x)|}{|U|}$.

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 $C \underset{x}{\longrightarrow} D$ will be called a *certain decision rule*; if $0 < cer_x(C,D) < 1$, the decision rule will be referred to as an *uncertain decision rule*. We will also use a *coverage factor* of the decision rule, denoted $cov_x(C,D)$ [7] defined as

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

where $D(x) \neq \emptyset$ and $\pi\left[D(x)\right] = \frac{|D(x)|}{|U|}$. Similarly,

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

If $C \underset{x}{\longrightarrow} D$ is a decision rule, then $D \underset{x}{\longrightarrow} C$ will be called an *inverse decision rule*. Inverse decision rules can be used to give *explanations* (*reasons*) for a decision.

4 Approximation of Sets

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

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

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

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.

There is an interesting relationship between approximations and decision rules. Let $C \underset{x}{\longrightarrow} D$ be a decision rule. The set,

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \subseteq D(x)\},$$

is equal to the lower approximation of the decision class $D(x)$, by condition classes $C(y)$, whereas the set,

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \cap D(x) \neq \emptyset\},$$

is equal to the upper approximation of the decision class by condition classes $C(y)$.

That means that approximations and decision rules are two different methods for expressing imprecision. Approximations are better suited to expressing topological properties of decision tables, whereas rules describe hidden patterns in data in a simple way.

5 Probabilistic Properties of Decision Tables

Decision tables have important probabilistic properties that are discussed next.

Let $C \underset{x}{\longrightarrow} D$ be a decision rule, and let $\Gamma = C(x)$ and $\Delta = D(x)$. Then the following properties are valid:

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

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

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

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

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

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

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

These properties will be used as a basis for the rough set processor organization. The certainty and coverage factors for the decision table presented in Table 2 are shown in Table 3.

Table 3. Certainty and coverage factors

Decision rule	*Strength*	*Certainty*	*Coverage*
1	0.18	0.31	0.34
2	0.06	0.40	0.13
3	0.23	1.00	0.43
4	0.41	0.69	0.87
5	0.03	1.00	0.06
6	0.09	0.60	0.17

Let us observe that, according to formulas (5) and (6), the certainty and coverage factors can be computed employing only the strength of decision rules. In Table 2, decision rules 3 and 5 are certain, whereas the remaining decision rules are uncertain. This means that middle-aged males having a profession and young males not having a profession are certainly healthy. Old males having a profession are most probably ill (probability = .69), and middle-aged females not having a profession are most probably healthy (probability = .60).

The inverse decision rules say that healthy persons are most probably middle-aged males having a profession (probability = .43) and ill persons are most probably old males having a profession (probability = .87).

6 Decision Tables and Flow Graphs

With every decision table, we associate a *flow graph*, i.e., a directed acyclic graph defined as follows: to every decision rule $C \underset{x}{\longrightarrow} D$, we assign a *directed branch* x connecting the *input node* $C(x)$ and the *output node* $D(x)$. The strength of the decision rule represents a *throughflow* of the corresponding branch. The throughflow of the graph is governed by formulas (1) – (6).

Formulas (1) and (2) say that the outflow of an input node or an output node is equal to their respective inflows. Formula (3) states that the outflow of the output node amounts to the sum of its inflows; whereas formula (4) says that the sum of the outflows of the input node equals its inflow. Finally, formulas (5) and (6) reveal how throughflow in the flow graph is distributed between its inputs and outputs.
The flow graph associated with the decision table presented in Table 2 is shown in Fig. 1.

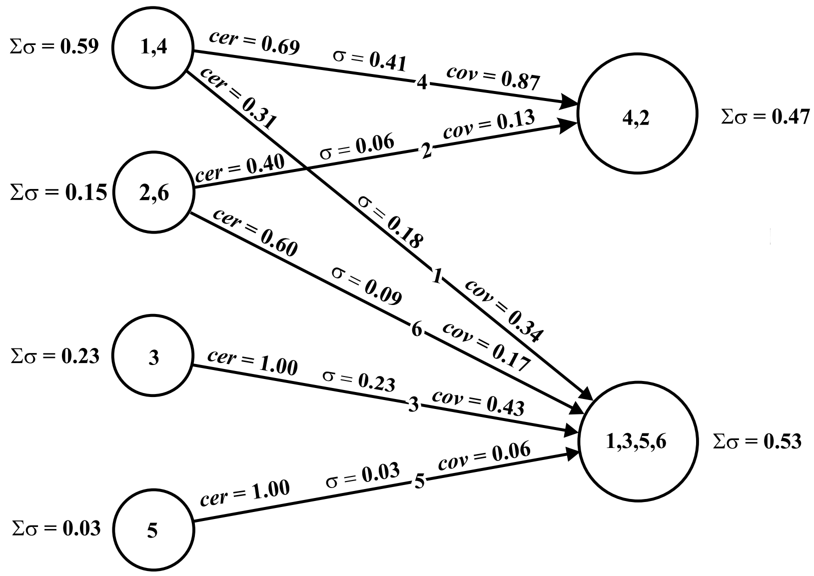

Fig. 1. Flow graph

The application of flow graphs to represent decision tables gives very clear insight into the decision process. The classification of objects in this representation boils down to finding the maximal output flow in the flow graph; whereas, the explanation of the decisions is connected to the maximal input flow associated with the given decision (see also [1] and [6]).

7 Rough Set Processor

To make the most of rough set theory in data analysis, a special microprocessor, the RSP, is necessary, to speed up the classification process. The RSP should perform operations pointed out by the flow graph of a decision table, that is, first it should compute its strengths from the supports of decision rules and afterward compute the certainty and coverage factors of all decision rules. Finally, the maximal certainty (coverage) factor should be computed, pointing out the most probable decision class (reason) for the classified object.

Many hardware implementations of this idea are possible. An example of a simplified RSP structure is depicted in Fig. 2.

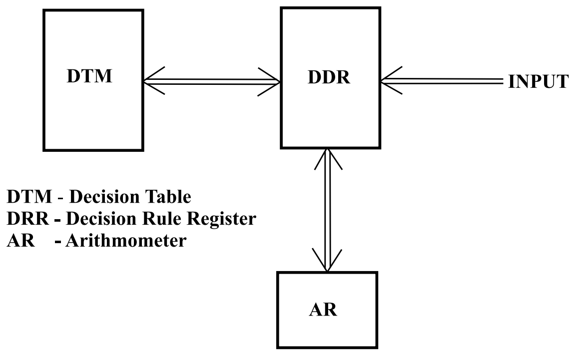

Fig. 2. RSP structure

The RSP consists of decision table memory (DTM), a decision rule register (DRR) and an arithmometer (AR). Decision rules are stored in the DTM. The word structure of the decision table memory is shown in Fig. 3.

Condition	*Decision*	*Support*	*Strength*	*Certainty*	*Coverage*

Fig. 3. Word structure

At the initial state, only conditions, decisions, and support of each decision rule are given. Next, the strength of each decision rule is computed. Afterward, certainty and coverage factors are computed. Finally, the maximal certainty (coverage) factors are ascertained. They will be used to define the most probable decision rules (inverse decision rules) induced by data. Let us also observe that the flow graph can easily be implemented as an analogue electrical circuit.

8 Conclusion

Rough-set-based data analysis consists of discovering hidden patterns in decision tables. It is shown that decision tables display basic probabilistic features; particularly, they satisfy the total probability theorem and Bayes' theorem. Moreover, rough set theory allows us to represent the above theorems in a very simple way using only the strengths of the decision rules. This property allows us to represent decision tables in the form of a flow graph and interpret the decision structure of the decision table as throughflow in the graph. The flow graph interpretation of decision tables can be employed as a basis for rough set processor organization.

References

1. M.Berthold, D.J. Hand. *Intelligent Data Analysis: An Introduction*. Springer, Heidelberg, 1999.
2. J. Łukasiewicz. Die Logischen Grundlagen der Wahrscheinlichkeitsrechnung. Kraków, 1913. In L. Borkowski, editor, *Jan Łukasiewicz: Selected Works*, North–Holland, Amsterdam and Polish Scientific Publishers, Warsaw, 16–63, 1970.
3. A. Kanasugi. A design of architecture for rough set processor. In *Proceedings of the International Workshop on Rough Set Theory and Granular Computing (RSTGC 2001)*, volume 5(1/2) of the *Bulletin of International Rough Set Society*, 201–204, 2001.
4. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
5. Z. Pawlak. Rough sets and decision algorithms. In *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, LNAI 2005, 30–45, Springer, Berlin, 2001.
6. A. Skowron. Rough sets in KDD (plenary talk at IFIP 2000). In *Proceedings of the Conference on Intelligent Information Processing (IIP 2000)*, 1–17, Publishing House of Electronic Industry, Beijing, 2001.
7. D. Ślęzak. Data model based on approximate bayesian network. In B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, R.R. Yager, editors, *Technologies for Constructing Intelligent Systems 2: Tools*, 313–326, Physica, Heidelberg, 2002.
8. S. Tsumoto, H. Tanaka. Discovery of functional components of proteins based on PRIMEROSE and domain kowledge hierarchy. In *Proceedings of the 3rd Workshop on Rough Sets and Soft Computing (RSSC'94)*, 280–285, San Jose, CA, 1995.