

# Rough Sets and Decision Algorithms

Zdzisław Pawlak

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,  
ul. Bałtycka 5, 44 000 Gliwice, Poland

**Abstract.** Rough set based data analysis starts from a data table, called an *information system*. The information system contains data about objects of interest characterized in terms of some attributes. Often we distinguish in the information system condition and decision attributes. Such information system is called a *decision table*. The decision table describes decisions in terms of conditions that must be satisfied in order to carry out the decision specified in the decision table. With every decision table a set of decision rules, called a *decision algorithm* can be associated. It is shown that every decision algorithm reveals some well known probabilistic properties, in particular it satisfies the Total Probability Theorem and the Bayes' Theorem. These properties give a new method of drawing conclusions from data, without referring to prior and posterior probabilities, inherently associated with Bayesian reasoning.

## 1 Introduction

Rough set based data analysis starts from a data table, called an *information system*. The information system contains data about objects of interest characterized in terms of some attributes. Often we distinguish in the information system condition and decision attributes. Such an information system is called a *decision table*. The decision table describes decisions in terms of conditions that must be satisfied in order to carry out the decision specified in the decision table. With every decision table we can associate a decision algorithm which is a set of *if... then...* decision rules. The decision rules can be also seen as a logical description of approximation of decisions, and consequently a decision algorithm can be viewed as a logical description of basic properties of the data. The decision algorithm can be simplified, what results in optimal description of the data, but this issue will not be discussed in this paper.

In the paper first basic notions of rough set theory will be introduced. Next the notion of the decision algorithm will be defined and some its basic properties will be shown. It is revealed that every decision algorithm has some well known probabilistic features, in particular it satisfies the Total Probability Theorem and the Bayes' Theorem [5]. These properties give a new method of drawing conclusions from data, without referring to prior and posterior probabilities, inherently associated with Bayesian reasoning. Three simple tutorial examples will be given to illustrate the above discussed ideas. The real-life examples are much more sophisticated and will not be presented here.

## 2 Approximation of Sets

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  $a \in B$ , 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.

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  $B$ .

### 3 Decision Rules

In this section we will introduce a formal language to describe approximations in logical terms.

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  $\Phi$  in  $S$  and refer to as the *meaning* of  $\Phi$  in  $S$ .

The meaning  $\|\Phi\|_S$  of  $\Phi$  in  $S$  is defined inductively as follows:

$$\|(a, v)\|_S = \{x \in U : a(v) = x\} \text{ for all } a \in B \text{ and } v \in V_a, \|\Phi \vee \Psi\|_S = \|\Phi\|_S \cup \|\Psi\|_S, \|\Phi \wedge \Psi\|_S = \|\Phi\|_S \cap \|\Psi\|_S, \|\sim \Phi\|_S = U - \|\Phi\|_S.$$

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

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

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

The number  $supp_S(\Phi, \Psi) = card(\|\Phi \wedge \Psi\|_S)$  will be called the *support* of the rule  $\Phi \rightarrow \Psi$  in  $S$ . We consider a probability distribution  $p_U(x) = 1/card(U)$  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  $\Phi$  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  $\Psi$  is true in  $S$  given  $\Phi$  is true in  $S$  called the *certainty factor*, used first by Lukasiewicz [3] 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  $\Phi \rightarrow \Psi$  is true in  $S$ .

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

Besides, we will also use a *coverage factor* (used e.g. by Tsumoto [14] for estimation of the quality of decision rules) defined by

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

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

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

The certainty factors in  $S$  can be also interpreted as the frequency of objects having the property  $\Psi$  in the set of objects having the property  $\Phi$  and the coverage factor – as the frequency of objects having the property  $\Phi$  in the set of objects having the property  $\Psi$ .

The number

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

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

## 4 Decision Algorithms

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

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

- 1) If for every  $\Phi \rightarrow \Psi, \Phi' \rightarrow \Psi' \in \text{Dec}(S)$  we have  $\Phi = \Phi'$  or  $\|\Phi \wedge \Phi'\|_S = \emptyset$ , and  $\Psi = \Psi'$  or  $\|\Psi \wedge \Psi'\|_S = \emptyset$ , then we will say that  $\text{Dec}(S)$  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  $\text{Dec}(S)$  *covers*  $U$ .
- 3) If  $\Phi \rightarrow \Psi \in \text{Dec}(S)$  and  $\text{supp}_S(\Phi, \Psi) \neq 0$  we will say that the decision rule  $\Phi \rightarrow \Psi$  is *admissible* in  $S$ .
- 4) If  $\bigcup_{X \in U/D} C_*(X) = \|\bigvee_{\Phi \rightarrow \Psi \in \text{Dec}^+(S)} \Phi\|_S$  where  $\text{Dec}^+(S)$  is the set of all certain decision rules from  $\text{Dec}(S)$ , we will say that the set of decision rules  $\text{Dec}(S)$  preserves the *consistency* of the decision table  $S = (U, C, D)$ .

The set of decision rules  $\text{Dec}(S)$  that satisfies 1), 2) 3) and 4), i.e., is independent, covers  $U$ , preserves the consistency of  $S$  and all decision rules

$\Phi \rightarrow \Psi \in Dec(S)$  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 a decision algorithm then  $Dec^*(S)$  will be called an *inverse decision algorithm* of  $Dec(S)$ .

The number

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

where  $D(\Phi) = \{\Psi : \Phi \rightarrow \Psi \in Dec(S)\}$  will be referred to as the *efficiency* of the decision algorithm  $Dec(S)$  in  $S$ , and the sum is stretching 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 decision rules  $\Phi \rightarrow \Psi$  with maximal strength  $\sigma_S(\Phi, \Psi)$  among rules  $\Phi \rightarrow \Psi \in Dec(S)$  with satisfied  $\Phi$  on these objects. In other words, the efficiency says how well the decision algorithm classifies objects when the decision rules with maximal strength are used only.

## 5 Decision algorithms and approximations

Decision algorithms can be used as a formal language for describing approximations (see [5]).

Let  $Dec(S)$  be a decision algorithm in  $S$  and let  $\Phi \rightarrow \Psi \in Dec(S)$ . By  $C(\Psi)$  we denote the set of all conditions of  $\Psi$  in  $Dec(S)$  and by  $D(\Phi)$  - the set of all decisions of  $\Phi$  in  $Dec(S)$ .

Then we have the following relationships:

$$\begin{aligned} \text{a) } C_*(\|\Psi\|_S) &= \bigvee_{\Phi' \in C(\Psi), \pi(\Psi|\Phi')=1} \Phi' \|_S, \\ \text{b) } C^*(\|\Psi\|_S) &= \bigvee_{\Phi' \in C(\Psi), 0 < \pi(\Psi|\Phi') \leq 1} \Phi' \|_S, \end{aligned}$$

$$c) BN_C(\|\Psi\|_S) = \bigvee_{\Phi' \in C(\Psi), 0 < \pi(\Psi|\Phi') < 1} \Phi' \|_S.$$

From the above properties we can get the following definitions:

- i) If  $\|\Phi\|_S = C_*(\|\Psi\|_S)$ , then formula  $\Phi$  will be called the *C-lower approximation* of the formula  $\Psi$  and will be denoted by  $C_*(\Psi)$ ;
- ii) If  $\|\Phi\|_S = C^*(\|\Psi\|_S)$ , then the formula  $\Phi$  will be called the *C-upper approximation* of the formula  $\Psi$  and will be denoted by  $C^*(\Psi)$ ;
- iii) If  $\|\Phi\|_S = BN_C(\|\Psi\|_S)$ , then  $\Phi$  will be called the *C-boundary* of the formula  $\Psi$  and will be denoted by  $BN_C(\Psi)$ .

The above properties say that any decision  $\Psi \in Dec(S)$  can be uniquely described by the following certain and uncertain decision rules respectively:

$$C_*(\Psi) \rightarrow \Psi,$$

$$BN_C(\Psi) \rightarrow \Psi.$$

This property is an extension of some ideas given by Ziarko [16]. The approximations can also be defined more generally, as proposed in [15] by Ziarko, and consequently we obtain more general probabilistic decision rules.

## 6 Some properties of decision algorithms

Decision algorithms have interesting probabilistic properties which are discussed in this section.

Let  $Dec(S)$  be a decision algorithm and let  $\Phi \rightarrow \Psi \in Dec(S)$ . Then the following properties are valid:

$$\sum_{\Phi' \in C(\Psi)} \pi_S(\Phi'|\Psi) = 1 \quad (1)$$

$$\sum_{\Psi' \in D(\Phi)} \pi_S(\Psi'|\Phi) = 1 \quad (2)$$

$$\pi_S(\Psi) = \sum_{\Phi' \in C(\Psi)} \pi_S(\Psi|\Phi') \cdot \pi_S(\Phi') \quad (3)$$

$$\pi_S(\Phi|\Psi) = \frac{\pi_S(\Psi|\Phi) \cdot \pi_S(\Phi)}{\sum_{\Phi' \in C(\Psi)} \pi_S(\Psi|\Phi') \cdot \pi_S(\Phi')} \quad (4)$$

That is, any decision algorithm, and consequently any decision table, satisfies (1), (2), (3) and (4). Observe that (3) is the well known *Total Probability Theorem* and (4) is the *Bayes' Theorem*. Note that we are not referring to prior and posterior probabilities – fundamental in Bayesian data analysis philosophy. The Bayes' Theorem in our case says that: if an implication  $\Phi \rightarrow \Psi$

is true in the degree  $\pi_S(\Psi|\Phi)$  then the inverse implication  $\Psi \rightarrow \Phi$  is true in the degree  $\pi_S(\Phi|\Psi)$ .

Let us observe that the Total Probability Theorem can be presented in the form

$$\pi_S(\Psi) = \sum_{\Phi' \in C(\Psi)} \sigma_S(\Phi', \Psi) \quad (5)$$

and the Bayes' Theorem will assume the form

$$\pi_S(\Phi|\Psi) = \frac{\sigma_S(\Phi, \Psi)}{\sum_{\Phi' \in C(\Psi)} \sigma_S(\Phi', \Psi)} = \frac{\sigma_S(\Phi, \Psi)}{\pi_S(\Psi)} \quad (6)$$

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

In other words, if we know the ratio of  $\Phi_S$  in  $\Psi$ , thanks to the Bayes' Theorem, we can compute the ratio of  $\Psi_S$  in  $\Phi$ .

## 7 Illustrative examples

In this section we will illustrate the concepts introduced previously by means of simple tutorial examples.

### Example 1

Let us consider Table 1 in which data on the relationships between color of eyes and color of hair is given.

**Table 1.** Simple data table

<i>Eyes</i>	<i>Hair</i>	
	<i>blond</i>	<i>dark</i>
<i>blue</i>	16	0
<i>brown</i>	8	56

The above data can be presented as a decision table shown in Table 2.

**Table 2.** Decision table

<i>Rule number</i>	<i>Eyes</i>	<i>Hair</i>	<i>Support</i>
1	<i>blue</i>	<i>blond</i>	16
2	<i>blue</i>	<i>dark</i>	0
3	<i>hazel</i>	<i>blond</i>	8
4	<i>hazel</i>	<i>dark</i>	56

Assume that *Hair* is a decision attribute and *Eyes* is a condition attribute. The corresponding decision algorithm is given below:

- 1) *if (Eyes, blue) then (Hair, blond),*
- 2) *if (Eyes, blue) then (Hair, dark),*
- 3) *if (Eyes, hazel) then (Hair, blond),*
- 4) *if (Eyes, hazel) then (Hair, dark).*

The certainty and coverage factors for the decision rules are given in Table 3.

**Table 3.** Certainty and coverage factors

<i>Rule number</i>	<i>Cert.</i>	<i>Cov.</i>	<i>Support</i>	<i>Strength</i>
1	1.000	0.67	16	0.2
2	0.000	0.00	0	0.0
3	0.125	0.33	8	0.1
4	0.875	1.00	56	0.7

From the certainty factors of the decision rules we can conclude that:

- every person in the data table having *blue eyes* is for certain a *blond*,
- for certain there are no people in the data table having *blue eyes* who are *dark-haired*,
- the probability that a person having *hazel eyes* is a *blond* is 0.125,
- the probability that the person having *hazel eyes* is *dark-haired* equals to 0.875.

In other words the decision algorithm says that:



- 12,5% persons with *hazel eyes* are *blond*,
- 87,5% persons with *hazel eyes* are *dark-haired*,
- 100% persons with *blue eyes* are *blond*.

From the above we can conclude that:

- people with *hazel eyes* are most probably *dark-haired*,
- people with *blue eyes* are for certain *blond*.

The efficiency of the decision algorithm is 0.9.

The inverse decision algorithm is given below:

- 1') if (*Hair, blond*) then (*Eyes, blue*),
- 2') if (*Hair, dark*) then (*Eyes, blue*),
- 3') if (*Hair, blond*) then (*Eyes, hazel*),
- 4') if (*Hair, dark*) then (*Eyes, hazel*).

The coverage factors says that:

- the probability that a *blond* has *blue eyes* is 0.67,
- for certain there are no *dark-haired* people in the data table having *blue eyes*,
- the probability that a *blond* has *brown eyes* is 0.33,
- for certain every *dark-haired* person in the data table has *hazel eyes*.

In other words:

- 33% *blond* have *hazel eyes*,
- 67% *blond* have *blue eyes*,
- 100% *dark-haired* persons have *hazel eyes*.

Thus we can conclude that:

- *blond* have most probably *blue eyes*,
- *dark-haired* people have for certain *hazel eyes*.

The efficiency of the inverse decision algorithm is 0.9.

### **Example 2**

In Table 4 information about nine hundred people is represented. The population is characterized by the following attributes: *Height, Hair, Eyes* and *Nationality*.

**Table 4.** Characterization of nationalities

<i>U</i>	<i>Height</i>	<i>Hair</i>	<i>Eyes</i>	<i>Nationality</i>	<i>Support</i>
1	<i>tall</i>	<i>blond</i>	<i>blue</i>	<i>Swede</i>	270
2	<i>medium</i>	<i>dark</i>	<i>hazel</i>	<i>German</i>	90
3	<i>medium</i>	<i>blond</i>	<i>blue</i>	<i>Swede</i>	90
4	<i>tall</i>	<i>blond</i>	<i>blue</i>	<i>German</i>	360
5	<i>short</i>	<i>red</i>	<i>blue</i>	<i>German</i>	45
6	<i>medium</i>	<i>dark</i>	<i>hazel</i>	<i>Swede</i>	45

Suppose that *Height*, *Hair* and *Eyes* are condition attributes and *Nationality* is the decision attribute, i.e., we want to find description of each nationality in terms of condition attributes.

Below a decision algorithm associated with Table 4 is given:

- 1) *if (Height, tall) then (Nationality, Swede),*
- 2) *if (Height, medium) and (Hair, dark) then (Nationality, German),*
- 3) *if (Height, medium) and (Hair, blond) then (Nationality, Swede),*
- 4) *if (Height, tall) then (Nationality, German),*
- 5) *if (Height, short) then (Nationality, German),*
- 6) *if (Height, medium) and (Hair, dark) then (Nationality, Swede).*

The certainty and coverage factors for the decision rules are shown in Table 5.

**Table 5.** Certainty and coverage factors

<i>Rule number</i>	<i>Cert.</i>	<i>Cov.</i>	<i>Support</i>	<i>Strength</i>
1	0.43	0.67	270	0.3
2	0.67	0.18	90	0.1
3	1.00	0.22	90	0.1
4	0.57	0.73	360	0.4
5	1.00	0.09	45	0.05
6	0.33	0.11	45	0.05

From the certainty factors of the decision rules we can conclude that:

- 43% *tall* people are *Swede*,
- 57% *tall* people are *German*,
- 33% *medium and dark-haired* people are *Swede*,
- 67% *medium and dark-haired* people are *German*,
- 100% *medium and blond* people are *Swede*,
- 100% *short* people are *German*.

Summing up:

- *tall* people are most probably *German*,
- *medium and dark-haired* people are most probably *German*,
- *medium and blond* people are for certain *Swede*,
- *short* people are for certain *German*.

The efficiency of the above decision algorithm is 0.65.

The inverse algorithm is as follows:

- 1') *if (Nationality, Swede) then (Height, tall)*,
- 2') *if (Nationality, German) then (Height, medium) and (Hair, dark)*,
- 3') *if (Nationality, Swede) then (Height, medium) and (Hair, blond)*,
- 4') *if (Nationality, German) then (Height, tall)*,
- 5') *if (Nationality, German) then (Height, short)*,
- 6') *if (Nationality, Swede) then (Height, medium) and (Hair, dark)*.

From the coverage factors we get the following characterization of nationalities:

- 11% *Swede* are *medium and dark-haired*,
- 22% *Swede* are *medium and blond*,
- 67% *Swede* are *tall*,
- 9% *German* are *short*,
- 18% *German* are *medium and dark-haired*,
- 73% *German* are *tall*.

Hence we conclude that:

- *Swede* are most probably *tall*,
- *German* are most probably *tall*.

The efficiency of the inverse decision algorithm is 0.7.

Observe that there are no certain decision rules in the inverse decision algorithm nevertheless it can properly classify 70% objects.

Of course it is possible to find another decision algorithm from Table 4.

Observe that there are three methods of computation of the certainty and coverage factors: either directly from definition employing the data, or using formula (4) or (6).

Similarly,  $\pi_S(\Psi)$  can be computed in three ways: using the definition and the data, or formula (3) or (5).

The obtained results are valid for the data only. In the case of another bigger data set the results may not be valid anymore .

Whether they are valid or not it depends if Table 4 is a representative sample of a bigger population or not.

**Example 3**

Now we will consider an example taken from [12], which will show clearly the difference between the Bayesian and rough set approach to data analysis.

We will start from the data table presented below:

**Table 6.** Voting Intentions

Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>1</sub>			
		1	2	3	4
1	1	28	8	7	0
	2	153	114	53	14
	3	20	31	17	1
2	1	1	1	0	1
	2	165	86	54	6
	3	30	57	18	4

where Y<sub>1</sub> represents Voting Intentions (1 = Conservatives, 2 = Labour, 3 = Liberal Democrat, 4 = Others), Y<sub>2</sub> represents Sex (1 = male, 2 = female) and Y<sub>3</sub> represents Social Class (1 = high, 2 = middle, 3 = low).

**Remark.** In the paper [12] wrongly 1 = low and 3 = high instead of 1 = high and 3 = low.

We have to classify voters according to their Voting Intentions on the basis of Sex and Social Class.

First we create from Table 6 a decision table shown in Table 7:

**Table 7.** Voting Intentions

<i>U</i>	<i>Y</i> <sub>2</sub>	<i>Y</i> <sub>3</sub>	<i>Y</i> <sub>1</sub>	<i>Support</i>	<i>Strength</i>
1	1	1	1	28	0.03
2	1	1	2	8	0.01
3	1	1	3	7	0.01
4	1	2	1	153	0.18
5	1	2	2	114	0.13
6	1	2	3	53	0.06
7	1	2	4	14	0.02
8	1	3	1	20	0.02
9	1	3	2	31	0.04
10	1	3	3	17	0.02
11	1	3	4	1	0.00
12	2	1	1	1	0.00
13	2	1	2	1	0.00
14	2	1	4	1	0.00
15	2	2	1	165	0.19
16	2	2	2	86	0.10
17	2	2	3	54	0.06
18	2	2	4	6	0.01
19	2	3	1	30	0.03
20	2	3	2	57	0.07
21	2	3	3	18	0.02
22	2	3	4	4	0.00

Next we simplify the decision table by employing only the decision rules with maximal strength, and we get the decision table presented in Table 8.

**Table 8.** Simplified Decision Table

<i>U</i>	<i>Y<sub>2</sub></i>	<i>Y<sub>3</sub></i>	<i>Y<sub>1</sub></i>	<i>Support</i>	<i>Strength</i>
1	1	1	1	28	0.07
2	1	2	1	153	0.35
3	1	3	2	31	0.07
4	2	2	1	165	0.38
5	2	3	2	57	0.13

It can be easily seen that the set of condition attributes can be reduced (see[4]) and the only reduced is the attribute  $Y_3$  (Social Class).

Thus Table 8 can be replaced by Table 9

**Table 9.** Reduced Decision Table

<i>U</i>	<i>Y<sub>3</sub></i>	<i>Y<sub>1</sub></i>	<i>Strength</i>	<i>Certainty</i>	<i>Coverage</i>
1	1	1	0.07(0.03)	1.00(0.60)	0.10(0.07)
2	2	1	0.73(0.37)	1.00(0.49)	0.90(0.82)
3	3	2	0.20(0.11)	1.00(0.55)	1.00(0.31)

The numbers in parenthesis refer to Table 7.

From this decision table we get the following decision algorithm:

- 1. high class → Conservative party *cer.* 0.60
- 2. middle class → Conservative party 0.49
- 3. lower class → Labour party 0.55

The efficiency of the decision algorithm is 0.51.

The inverse decision algorithm is given below:

- 1'. Conservative party → high class *cer.* 0.07
- 2'. Conservative party → middle class 0.82
- 3'. Labour party → lower class 0.31

The efficiency of the inverse decision algorithm is 0.48.

From the decision algorithm and the inverse decision algorithm we can conclude the following:

- *60% high class and 49% middle class intend to vote for the Conservative party*
- *55% lower class intend to vote for the Labour party*
- *7% intend to vote for the Conservative party belong to the high class*
- *82% intend to vote for the Conservative party belong to the middle class*
- *31% intend to vote for the Labour party belong to the lower class*

We advise the reader to examine the approach and results presented in [12] and compare them with that shown here.

Clearly, the rough set approach is much simpler and given better results than that discussed in [12].

## 8 Conclusions

The notion of a decision algorithm has been defined and its connection with decision table and other basic concepts of rough set theory discussed. Some probabilistic properties of decision algorithms have been revealed, in particular the relationship with the Total Probability Theorem and the Bayes' Theorem. These relationships give a new efficient method to draw conclusions from data, without referring to prior and posterior probabilities intrinsically associated with Bayesian reasoning.

## Acknowledgments

Thanks are due to Prof. Andrzej Skowron and Prof. Wojciech Ziarko for their critical remarks.

## References

1. Adams, E. W. (1975) *The logic of conditionals, an application of probability to deductive logic*. D. Reidel Publishing Company, Dordrecht, Boston
2. Grzymala-Busse J.(1991) *Managing Uncertainty in Expert Systems*. Kluwer Academic Publishers, Boston, Dordrecht
3. Łukasiewicz, J. *Die logischen Grundlagen der Wahrscheinlichkeitsrechnung*. Krakow, 1913. In: L. Borkowski (ed.), *Jan Łukasiewicz – Selected Works*, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw, 1970
4. Pawlak, Z. (1991) *Rough Sets – Theoretical Aspects of Reasoning about Data*; Kluwer Academic Publishers: Boston, Dordrecht
5. Pawlak, Z.(1999) *Decision rules, Bayes' rule and rough sets*. In: 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, 1–9
6. Pawlak, Z. *Drawing conclusions from data – the rough set way* (to appear)

7. Pawlak, Z. Combining rough sets and Bayes' rule (to appear)
8. Pawlak, Z.; 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, 251–271
9. Polkowski, L., Skowron, A. (1998) *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
10. Polkowski, L., Skowron, A. (1998) *Rough Sets in Knowledge Discovery Vol. 1–2*, Physica-Verlag, Heidelberg
11. Polkowski, L., Tsumoto, S., Lin, T.Y. *Rough Set Methods and Applications. New Development in Knowledge Discovery in Information Systems*, Springer-Verlag (to appear)
12. Ramoni, M., Sebastiani, P. (1999) *Bayesian Methods*. In: M. Berthold, D. Hand (eds.), *Intelligent Data Analysis, An Introduction*, Springer, Berlin
13. Tsumoto, S.; Kobayashi, S. et al. (1996) (eds.), *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96)*. The University of Tokyo, November 6–8
14. Tsumoto, S. (1998) Modelling medical diagnostic rules based on rough sets. In: 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, 475–482
15. Ziarko, W. (1998) Variable precision rough set model. *Journal of Computer and Systems Sciences* 46, 39–59
16. Ziarko, W. (1998) Approximation region-based decision tables. In: L. Polkowski, A. Skowron (eds.), *Rough Sets in Knowledge Discovery Vol.1-2*, Physica-Verlag, Heidelberg, 178–185