

Motto: “Let the data speak for themselves”

R.A. Fisher

A new Approach to Drawing Conclusions from Data A Rough Set Perspective

Zdzisław Pawlak

Institute for Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Bałtycka 5, 44 00 Gliwice, Poland
zpw@ii.pw.edu.pl

Abstract. Rough set theory is a new mathematical approach to vague and uncertain data analysis. The paper explains basic concepts of the theory and outlines briefly applications of the method to drawing conclusions from factual data. The presented approach has found many real-life applications and it seems that can be used in some kind of legal reasoning.

1. Introduction

This paper is an extension of the paper [3]. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently it became also a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate the imperfect knowledge. The most successful one is, no doubt, fuzzy set theory proposed by Zadeh.

Rough set theory is still another approach to this problem. From philosophical point of view rough set theory is a new approach to vagueness and uncertainty - practically it is a new method of data analysis.

The proposed method has some important advantages:

- provides efficient algorithms for finding hidden patterns in data
- finds reduced sets of data (data reduction)
- evaluates significance of data
- generates minimal sets of decision rules from data
- it is easy to understand

- offers straightforward interpretation of results
- can be used to both qualitative and quantitative data analysis
- identifies relationships that would not be found using statistical methods

Rough set theory overlaps with many other theories, e.g., fuzzy sets, evidence theory, statistics and others, nevertheless it can be viewed in its own rights, as an independent, complementary, and not competing discipline.

The rough set methodology has found many real-life applications in various domains. It seems that the rough set approach can be also used in some kind of legal reasoning, in particular for drawing conclusions from factual data.

In the paper basic concepts of rough set theory are given and its applications to drawing conclusions from data are briefly outlined. The theory is based on sound mathematical foundations, but for the sake of simplicity we will try to explain rudiments of the theory only. The real life applications of the theory require advanced extension of the theory but we will not dwell upon this issue here.

Rudiments of rough set theory can be found in [1,4].

More about the theory and its applications can be found in [2,5,6,7,8,9] and the web.

2. An Example

We start our considerations from a very simple tutorial example concerning driving a car in various driving conditions. In Table 1, 6 facts concerning 98 cases of driving a car in various driving conditions are presented. In the table columns labeled *weather*, *road* and *time*, called *condition attributes*, represent driving conditions. The column labeled by *accident*, called *decision attribute*, contains information whether an accident has occurred or not. *N* denotes the number of analogous cases.

<i>Fact no.</i>	<i>Driving conditions</i>			<i>Consequence</i>	<i>N</i>
	<i>weather</i>	<i>road</i>	<i>time</i>	<i>accident</i>	
1	<i>misty</i>	<i>icy</i>	<i>day</i>	<i>yes</i>	8
2	<i>foggy</i>	<i>icy</i>	<i>night</i>	<i>yes</i>	10
3	<i>misty</i>	<i>not icy</i>	<i>night</i>	<i>yes</i>	4
4	<i>sunny</i>	<i>icy</i>	<i>day</i>	<i>no</i>	50
5	<i>foggy</i>	<i>not icy</i>	<i>dusk</i>	<i>yes</i>	6
6	<i>misty</i>	<i>not icy</i>	<i>night</i>	<i>no</i>	20

Table 1

The problem we are interested in is the following one: we want to find the relationship between accidents (or lack of accidents) and the driving conditions, i.e., to describe the set of facts $\{1,2,3,5\}$ (or the set of facts $\{4,6\}$) in terms of attributes *weather*, *road* and *time*.

Note that the data are *inconsistent* because the facts number 3 and 6 are inconsistent, i.e., they have the same conditions but different consequences, therefore the set of all accidents cannot be described in terms of attributes *weather*, *road* and *time*. However we can describe the set of accidents approximately. To this end let us observe what the data are telling us:

- $\{1,2,5\}$ is the *maximal* set of facts that can *certainly* be classified as accidents in terms of the driving conditions,
- $\{1,2,3,5,6\}$ is the set of all facts that can *possible* be classified as accidents in terms of the driving conditions.
- $\{3,6\}$ is the set of facts that can be classified neither as accident nor no accidents in term of the driving conditions

Note that the set $\{3,6\}$ is the difference between sets $\{1,2,3,5,6\}$ and $\{1,2,5\}$.

3. Approximations

It follows from the considerations above that some decisions cannot be described by means of conditions. However they can be describe with some approximations. Therefore in what follows we will use the following terminology:

- the set $\{1,2,5\}$ is the *lower approximation* of the set $\{1,2,3,5\}$,
- the set $\{1,2,3,5,6\}$ is the *upper approximation* of the set $\{1,2,3,5\}$,
- the set $\{3,6\}$ is the *boundary region* of the set $\{1,2,3,5\}$.

Approximations are basic concepts of rough set theory and are used to draw conclusions from data. Informal definitions of approximations are the following:

- the *lower approximation* of a set X with respect to data D is the set of all facts that can be *for certain* classified as X (are *certainly* X) in view of the data D ,
- the *upper approximation* of a set X with respect to data D is the set of all facts that can be *possibly* classified as X (are *possibly* X) in view of the data D ,
- the *boundary region* of a set X with respect to data D is the set of all facts that can be classified as neither X nor non- X in view of the data D .

Mathematical definitions of approximations are given next.

Let us first define 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 .

Thus, the set of elements is rough (inexact) if it cannot be defined in terms of the data, i.e. it has some elements that can be classified neither as member of the set nor its complement in view of the data.

4. Data dependencies and reduction

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 .

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

$$k = \gamma(C, D) = \frac{\text{card}(\text{POS}_C(D))}{\text{card}(U)},$$

where

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

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

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

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 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*.

Next important issue in data analysis is reduction of data. Very often we want to know whether some data is superfluous, i.e., whether we can remove some data from the data table in such a way that drawing conclusions from the data is intact. It turns out that in order to reduce the data without affecting this property we have to preserve the degree of dependency among the attributes.

A minimal subset of condition attributes $C' \subseteq C$ that preserves the degree of dependency among condition and decision attributes, i.e., $\gamma(C', D) = \gamma(C, D)$ is called a *reduct* of C .

For example, the set of condition attributes $\{weather, road, time\}$ has two reducts $\{weather, road\}$ and $\{weather, time\}$.

That means that Table 1 contains superfluous data and consequently the table can be replaced either by Table 2 or Table 3.

<i>Fact no.</i>	<i>Driving conditions</i>		<i>Consequence</i>	<i>N</i>
	<i>weather</i>	<i>road</i>	<i>accident</i>	
1	<i>misty</i>	<i>icy</i>	<i>yes</i>	8
2	<i>foggy</i>	<i>icy</i>	<i>yes</i>	10
3	<i>misty</i>	<i>not icy</i>	<i>yes</i>	4
4	<i>sunny</i>	<i>icy</i>	<i>no</i>	50
5	<i>foggy</i>	<i>not icy</i>	<i>yes</i>	6
6	<i>misty</i>	<i>not icy</i>	<i>no</i>	20

Table 2

<i>Fact no.</i>	<i>Driving conditions</i>		<i>Consequence</i>	<i>N</i>
	<i>weather</i>	<i>time</i>	<i>accident</i>	
1	<i>misty</i>	<i>day</i>	<i>yes</i>	8
2	<i>foggy</i>	<i>night</i>	<i>yes</i>	10
3	<i>misty</i>	<i>night</i>	<i>yes</i>	4
4	<i>sunny</i>	<i>day</i>	<i>no</i>	50
5	<i>foggy</i>	<i>dusk</i>	<i>yes</i>	6
6	<i>misty</i>	<i>night</i>	<i>no</i>	20

Table 3

In general there are many reducts of condition attributes, thus simplification of data tables yields no unique results. However, in some sense, the reduced data tables are equivalent to the original data table because they ensure drawing identical conclusions from the reduced data.

Intersection of all reducts is called the *core* of condition attributes. For example, in Table 1 the core is the attribute *weather*. The core can not be removed from the table without affecting patterns in the data. Thus the core is the set of most important attributes, which cannot be eliminated in any circumstances.

The algorithms for data reduction are rather sophisticated and for now we refrain from dwelling upon this issue. More about data reduction can be found [5,6,9].

5. Decision rules

In order to reason about data we need a language of *decision rules*, called sometimes also *association rules* or *production rules*. A decision rule is an implication in the form *if Φ then Ψ* , (in symbols $\Phi \rightarrow \Psi$), where Φ is called the *condition* and Ψ – the *decision* of the rule and Φ, Ψ are logical formulas build up from attributes and attribute values combined by means of logical connectives “and”, “or” and “not” in the standard way..

Every fact in the data table determines a decision rule.

For example, Table 1 can be represented by the following set of decision rules:

- 1) *if (weather, misty) and (road, icy) and (time, day) then (accident, yes)*
- 2) *if (weather, foggy) and (road, icy) and (time, night) then (accident, yes)*
- 3) *if (weather, misty) and (road, not icy) and (time, night) then (accident, yes)*
- 4) *if (weather, sunny) and (road, icy) and (time, day) then (accident, no)*
- 5) *if (weather, foggy) and (road, not icy) and (time, dusk) then (accident, yes)*
- 6) *if (weather, misty) and (road, not icy) and (time, night) then (accident, no)*

Now we define decision rules more formally.

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:

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

A *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.

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 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

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

called the *certainty factor* and denoted $cer_S(\Phi, \Psi)$. We have

$$cer_s(\Phi, \Psi) = \pi_s(\Psi | \Phi) = \frac{card(\|\Phi \wedge \Psi\|_s)}{card(\|\Phi\|_s)}$$

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

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

Besides, we will also use a *coverage factor* denoted $cov_s(\Phi, \Psi)$ and defined as

$$cov_s(\Phi, \Psi) = \pi_s(\Phi | \Psi) = p_U(\|\Phi\|_s | \|\Psi\|_s).$$

Obviously we have

$$cov_s(\Phi, \Psi) = \pi_s(\Phi | \Psi) = \frac{card(\|\Phi \wedge \Psi\|_s)}{card(\|\Psi\|_s)}.$$

In other words

$$cer_s(\Phi, \Psi) = \frac{\text{number of all cases satisfying } \Phi \text{ and } \Psi}{\text{number of all cases satisfying } \Phi}$$

and

$$cov_s(\Phi, \Psi) = \frac{\text{number of all cases satisfying } \Phi \text{ and } \Psi}{\text{number of all cases satisfying } \Psi}.$$

The certainty factor is the frequency of Ψ s in Φ and the coverage factor is the frequency of Φ s in Ψ .

If a decision rule $\Phi \rightarrow \Psi$ uniquely determines decisions in terms of conditions, i.e., if $\pi(\Phi | \Psi) = 1$, then the rule is called *certain*.

If a decision rule $\Phi \rightarrow \Psi$ does not determine uniquely decisions in terms of conditions, i.e., if $0 < \pi(\Phi | \Psi) < 1$, then the rule is called *uncertain*.

For example, for the decision rule

if (weather, misty) and (road, icy) and (time, day) then (accident, yes)

the certainty factor equals to 1.00, thus the decision rule is certain, whereas, for the decision rule

if (weather, misty) and (road, not icy) and (time, night) then (accident, no)

the certainty factor is equal 0.83 thus the decision rule is uncertain.

The number

$$\sigma_s(\Phi, \Psi) = \frac{\text{supp}_s(\Phi, \Psi)}{\text{card}(U)} = \text{cer}_s(\Phi, \Psi) \cdot \pi_s(\Phi)$$

will be called the *strength* of the decision rule $\Phi \rightarrow \Psi$ in S . The strength of a decision rule plays an important role in our consideration and will be discussed in detail in sec. 8.

6. Decision algorithm

Any set of mutually, exclusive and exhaustive, i.e., covering all facts in S decision rules in S will be called a *decision algorithm* in S and denoted $Dec(S)$. For exact definitions see [2].

An example of decision algorithm in the decision Table 1 is given below

- 1') *if (weather, misty) and (road, icy) then (accident, yes)*
- 2') *if (weather, foggy) then (accident, yes)*
- 3') *if (weather, misty) and (road, not icy) then (accident, yes)*
- 4') *if (weather, sunny) then (accident, no)*
- 5') *if (weather, misty) and (road, not icy) then (accident, no)*

Finding a minimal decision algorithm is not a simple task. Many methods have been proposed to solve this problem, but we will not consider this problem here. Interested reader is advised to consult [5,6,9].

The certainty coverage factors for this algorithm are shown in Table 4.

rule	strength	certainty	coverage
1'	0.08	1.00	0.29
2'	0.17	1.00	0.57
3'	0.04	0.17	0.14
4'	0.51	1.00	0.71
5'	0.20	0.83	0.29

Table 4

Sometimes we may be interested in *explanation* of decisions in terms of conditions. To this end we will need an *inverse* decision algorithm which is obtained by replacing mutually conditions and decisions in every decision rule in the decision algorithm.

For example, the following inverse decision algorithm can be understood as explanation of car accidents in terms of driving conditions:

- 1'') *if (accident, yes) then (road, icy) and (weather, misty)*

- 2") *if (accident, yes) then (weather, foggy)*
- 3") *if (accident, yes) then (road, not icy) and (weather, misty)*
- 4") *if (accident, no) then (weather, sunny)*
- 5") *if (accident, no) then (road, not icy) and (weather, misty)*

Note that for inverse decision rules the certainty and coverage factors are mutually exchanged.

7. Decision algorithm and approximations

There is an interesting relationship between decision algorithm and approximations.

Certain decision rules describe the *lower approximation* of the set of facts pointed out by the decision of the rule, whereas *uncertain* decision rules describe the *boundary region* of the set of facts pointed out by the decision of the rule.

We have the following relationships between approximations and decision rules in the illustrative example:

- Certain decision rules describing accidents (the lower approximation of the set of facts {1,2,3,5})
 - 1') *if (weather, misty) and (road, icy) then (accident, yes)*
 - 2') *if (weather, foggy) then (accident, yes)*
- Uncertain decision rule describing accidents (the boundary region {3,6} of the set of facts {1,2,3,5})
 - 3') *if (weather, misty) and (road, not icy) then (accident, yes)*
- Certain decision rule describing lack of accidents (the lower approximation of the set of facts {4,6})
 - 4') *if (weather, sunny) then (accident, no)*
- Uncertain decision rule describing lack of accidents (the boundary region {3,6} of the set of facts {4,6})
 - 5') *if (weather, misty) and (road, not icy) then (accident, no)*

These decision rules constitute a logical counterpart of approximations. Consequently instead of speaking of approximations we can employ the language of decision rules, and conversely. The language of approximations suits better to prove some properties of data, whereas the language of decision rules is better when practical applications are considered.

8. Decision algorithm and the Bayes' theorem

It turned out that the decision algorithms have interesting probabilistic properties: every decision algorithm (decision table) satisfies the total probability theorem and the Bayes' theorem, i.e., every data set has this two very important properties. This leads, in particular, to a new interpretation of Bayes' theorem, which does not require concepts of *prior* or *posterior* probabilities, inherently associated with Bayesian reasoning philosophy.

Besides, these relationships reveal connection between rough set theory and the Bayes' theorem.

Let $\Phi \rightarrow \Psi$ be a decision rule in $Dec(S)$. By $C(\Psi)$ we denote the set of all conditions of Ψ , such that if $\Phi' \in C(\Psi)$ and $\Phi' \rightarrow \Psi \in Dec(S)$, and by $D(\Phi)$ we mean the set of all decisions of Φ such that if $\Psi' \in D(\Phi)$ and $\Phi \rightarrow \Psi' \in Dec(S)$. Moreover we assume that all conditions in $C(\Psi)$ and all decisions in $D(\Phi)$ are pairwise mutually exclusive, i.e., if $\Phi', \Phi \in C(\Psi)$ then $\|\Phi' \wedge \Phi\|_S = \emptyset$ and if $\Psi', \Psi \in D(\Phi)$ then $\|\Psi' \wedge \Psi\|_S = \emptyset$. Then the following properties hold:

$$\sum_{\Phi' \in C(\Psi)} cer_S(\Phi', \Psi) = 1 \quad (1)$$

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

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

$$\pi_S(\Phi) = \sum_{\Psi' \in D(\Phi)} cov_S(\Phi, \Psi') \cdot \pi_S(\Psi') = \sum_{\Psi' \in D(\Phi)} \sigma_S(\Phi, \Psi') \quad (4)$$

$$cer_S(\Phi, \Psi) = \frac{cov_S(\Phi, \Psi) \cdot \pi_S(\Psi)}{\sum_{\Psi' \in D(\Phi)} cov_S(\Phi, \Psi') \cdot \pi_S(\Psi')} = \frac{\sigma_S(\Phi, \Psi)}{\sum_{\Psi' \in D(\Phi)} \sigma_S(\Phi, \Psi')} = \frac{\sigma_S(\Psi, \Phi)}{\pi_S(\Phi)} \quad (5)$$

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

Formulas 3) and 4) refer to the *total probability theorem*, whereas formulas 5) and 6) refer to the *Bayes' theorem*. The meaning of Bayes' theorem in this case differs from that postulated in statistical inference, where we assume that *prior* probability about some parameters without knowledge about the data is given. The *posterior* probability is computed next, which tells us what can be said about *prior* probability in view of the data.

In the rough set approach the meaning of Bayes' theorem is unlike. It reveals some relationships between the data, without referring to prior and posterior probabilities. Instead, the proposed approach connects the total probability theorem and the Bayes'

theorem with the strength of decision rules, giving a very simple way of computing the certainty and the coverage factors.

Thus, the proposed approach can be seen as a new model for Bayes' theorem, which offers a new approach to data analysis, in particular, to inverse decision rules and to compute their certainty factors, which can be used to explain decisions in terms of conditions.

9. What the data are telling us

The rough set approach allows to draw conclusions from the data without additional assumption about the data, like *prior* or *posterior* probabilities. Instead, rough set based reasoning is founded on the patterns hidden in the data, which are discovered by means of approximations induced by the data and revealed in the form of decision algorithms. This leads to drawing *conclusions* from the data and to giving *explanation* of obtained results.

For the tutorial example considered in this paper we get the following results.

From the decision algorithm and the certainty factors we can draw the following conclusions:

- 1') misty weather and icy road certainly caused accidents
- 2') foggy weather certainly caused accidents
- 3') misty weather and not icy road caused accidents in 17% cases
- 4') sunny weather and icy road certainly caused safe driving
- 5') misty weather and not icy road caused safe driving in 83% cases

From the inverse decision algorithm and the coverage factors we get the following explanations:

- 1'') 29% of accidents occurred when the weather was misty and the road icy
- 2'') 57% accidents occurred when the weather was foggy
- 3'') 14% accidents occurred when the weather was misty and the road not icy
- 4'') 71% safe driving took place when the weather was sunny
- 5'') 29% safe driving took place when the weather was misty and the road not icy

Summing up, the decision algorithm and the certainty factors lead to the following conclusions:

- misty weather and icy road or foggy weather *certainly* caused accidents
- sunny weather and icy road *certainly* caused no accidents

- misty weather and not icy road *most probably* caused no accidents

The inverse decision algorithm and the coverage factors give the following explanations of driving accidents:

- the *most probable* reason of accidents is the foggy weather
- the *most probable* reason of lack of accidents is the sunny weather

Other conclusions and explanations can be obtained employing decision algorithms resulting from other reduced decision tables. Thus, as mentioned before, we don't have unique association of accidents with the driving conditions.

Note also that the data table represents closed world, i.e., it is only a sample of a larger world. Therefore the conclusions drawn are not universal but are valid only for the data. Whether they can be generalized depends on whether the data is a representative sample of a bigger data set. This is the central issue of inductive reasoning but we will not discuss the problem here.

10. Conclusions

Rough set theory is a new method of drawing conclusions from data. The approach has found many nontrivial, real life applications. It seems that rough set theory can be also used in some kinds of legal reasoning, but to this end an extensive research is necessary.

References

1. Z. Pawlak, „Rough Sets: Theoretical Aspects and Reasoning about Data”, Kluwer Academic Publishers, 1991.
2. Z. Pawlak, „Decision Rules, Bayes' Rule and Rough Sets” in Rough Sets, Data Mining, and Granular-soft Computing, Springer,1999, pp. 1-9.
3. Z. Pawlak, „A Primer on Rough Sets - a new Approach to Drawing Conclusions from Data”, Cardozo Law Review. (to appear).
4. R. Słowiński, „Rough Set Approach to Decision Analysis”, AI Expert, 10, 18-25, 1995.
5. L. Polkowski, A. Skowron (eds.), „Rough Set and Current Trend in Computing”, Springer Verlag, 1998.
6. L. Polkowski, A. Skowron (eds.), „Rough Sets in Knowledge Discovery”, Physica Verlag, 1998, Vol. 1, 2.
7. T. Munakata, „Fundamentals of the new Artificial Intelligence: Beyond Traditional Paradigms”, Springer, 1998.

8. S. K. Pal, A. Skowron (eds.), „Rough Fuzzy Hybridization: A new Trend in Decision Making”, Springer, 1999.
9. N. Zhong, A. Skowron (eds.), „New Directions in Rough Sets, Data Mining, and Granular-soft Computing”, Springer, 1999.

MORE INFO ABOUT ROUGH SETS CAN BE FOUND AT:

<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.mimuw.edu.pl>

<http://www.idi.ntnu.no/~aleks/rosetta/>

<http://www.infj.ulst.ac.uk/~ccc23/grobian/grobian.html>