DECISION TABLES - A ROUGH SET APPROACH

ZDZISŁAW PAWLAK

Department of Complex Control Systems
Polish Academy of Sciences, Poland
and
Department of Computer Science
The University of North Carolina at Charlotte
NC 28223, USA

ABSTRACT. We propose in this article the application of the rough set concept as a basis for decision tables theory. The proposed approach yields interesting, new theoretical results and has given rise to a wide class of practical applications.

Keywords and phrases: decision tables and decision tables analysis, rough sets.

CR Categories: 412

1.Introduction. The main objective of this article is to show that the rough set concept [6] can be used as a mathematical basis for the decision tables theory [7], [8]. In particular the decision tables analysis and simplification [10] seems to be the most interesting area of research within the framework of the proposed approach.

Decision tables are commonly meant to be used mainly in programming [1-4],[10], however it turns out that machine learning [13], expert systems [5], vague data analysis [9] and other fields of artificial intelligence may also be chosen as possible fields of applications , especially in the rough set setting.

2.Decision Tables. We assume that the reader is familiar with the basic concepts of decision tables ,and we will start from the formal definition of a decision table employed in our considerations throughout this paper.

A *decision table* is a system

$$S = (U, C, D, V, f)$$

where

U-is a nonempty, finite set called the *universe*

C, D - are finite sets such that $C \cup D \neq 0$ and $C \cap D = 0$, called *condition* and *decision* (*action*) attributes respectively

$V = \bigcup_{a \in C \cup D} V_a$, V_a-is the set called the *domain* of attribute a

$f : U \times (C \cup D) \longrightarrow V$ is a *decision function* (total) such that $f(x, a) \in V_a$, for every $a \in C \cup D$ and $x \in U$.

A function $g : C \cup D \longrightarrow V$ is called a *decision rule* in S if there exists an $x \in U$ such that $g = f_x$.

If g is a decision rule, then the restriction g to C ,denoted $g|C$,and the restriction of g to D , denoted $g|D$ are called *conditions* and *decisions* (*actions*) of g respectively.

The decision rule g is *deterministic* (in S) if for every $x \in U$,$f_x|C = g|C$ implies $f_x|D = g|D$; otherwise g is *nondeterministic*.

Decision table is deterministic (consistent) if all its decision rules are deterministic; otherwise decision table is nondeterministic (inconsistent).

An example of decision table is shown below. In what follows we assume that decision rules are labelled by integers.

U	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	0	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1

Decision Table 1

Attributes a, b, c are conditions whereas d, e are decision attributes. For example the decision rule 1 is nondeterministic and the decision rule 3 is deterministic in the table.

3.Rough Sets. In this section we will present the basic concepts concerning rough sets which will be used throughout this paper.

It is easy to see that every subset of attribute $A \subseteq C \cup D$ generates a binary realtion over U ,denoted $\widetilde{A}$ and defined thus

$$(x, y) \in \widetilde{A} \text{ iff } f_x(a) = f_y(a) \text{ for every } a \in A.$$

The relation will be called an *indiscernibility* relation. Obviously $\widetilde{A}$ is equivalence relation over U. Hence every subset of attributes A generates a partition of U ,denoted A^*. In other words A^* is the family of all equivalence classes of the relation $\widetilde{A}$. The equivalence classes of $\widetilde{A}$ are called *blocks* of A^*.

For example the attribute a in the decision table 1 generates the partition of U consisting of three blocks $X_1 = \{1,4,5\}, X_2 = \{2,8\}$ and $X_3 = \{3,6,7\}$,whereas the set of condition attributes generates the partition consisting of 6 following blocks: $Y_1 = \{1,5\}$,$Y_2 = \{2,8\}$,$Y_3 = \{3\}$,$Y_4 = \{4\}$, $Y_5 = \{6\}$,$Y_6 = \{7\}$.

Our next important concept is that of approximation of a set.

Let $S = (U, C, D, V, f)$ be a decision table, $A \subseteq C \cup D$ and $X \subseteq U$. The *A-lower approximation* of X, denoted $\underline{A}X$ and *A-upper approximation* of X ,denoted $\overline{A}X$, are defined as below

$$\underline{A}X = \{x \in U : [x]_{\widetilde{A}} \subseteq X\}$$

$$\overline{A}X = \{x \in U : [x]_{\widetilde{A}} \cap X \neq \phi\}$$

where $[x]_{\widetilde{A}}$ denotes block of the partition A^* conaining the element $x \in U$.

Set $Bn_A(X) = \overline{A}X - \underline{A}X$ will be called the *A-boundary* of X.

More about approxiamtion can be found in [6]. The intuitive meaning of approximations is as follows: the A-lower approximation of X is the set of all elements of U which can certainly be classified as elements of X, employing the set of attributes A ,and the A-upper approximation of X -which can be possibly classified as elements of X using the set of attributes A. The A-boudary of X is the set of elements which cannot be properly classified either to X or $-X$ using the set of atributes A.

Now we are about to define our main notion, the notion of a rough set.

If $\overline{A}X = \underline{A}X$ we shall say that X is *A-discernible*, otherwise set X is *A-indiscernible*, or *rough* with respect to A.

Thus rough sets are sets with unsharply defined boundaries or, in other words, sets which cannot be clearly defined using the set of attributes A.

The following example will depict the above introduced notions of approximations and of boundary. Consider the decision table 1 ,the set of condition attributes $C = \{a, b, c\}$ and the subset of the universe $X = \{1,2,3,4,5\}$. Employing the definition of approximations and boundary it is easy to compute that $\underline{C}X = \{1,3,4,5\}$,$\overline{C}X = \{1,2,3,4,5,8\}$ and the boundary $Bn_C(X) = \{2,8\}$. Thus the set X is rough with respect to the set of condition attributes. That is to say that we are unable to decide whether elements 2 and 8 are members of the set X or not. For the rest of the universe such decision is possible.

Now we shall extend the concept of approximation from sets to families of sets, which will be needed to define further notions.

Let $S = (U, C, D, V, f)$ be a decision table, $A \subseteq C \cup D$ and $F = \{X_1, X_2, ..., X_n\}$ a family of subsets of U, i.e. $X_i \subseteq U$.

Lower and upper approximations are defined as follows

$$\underline{A}F = \{\underline{A}X_1, \underline{A}X_2, ..., \underline{A}X_n\}$$

$$\overline{A}F = \{\overline{A}X_1, \overline{A}X_2, ..., \overline{A}X_n\}.$$

In this paper we will be interested in families which are partitions generated by some subsets of attributes $A \subseteq C \cup D$, i.e. families of

the form A^*, and we will try to define how closely we approximate a partition B^* by another partition A^*, where $A, B \subseteq C \cup D$. In order to do so we shall introduce a coefficient, called the *accuracy of approximation* of B^* by A^*, and defined in the following way:

$$\gamma_A(B) = \frac{card(Pos_A(B^*))}{card(U)},$$

where

$$Pos_A(B^*) = \bigcup_{X \in B^*} \underline{A}X,$$

is called *positive region* of B^* with respect to A.

Of course $0 \leq \gamma_A(B) \leq 1$, for every A, B.

Intuitively speaking, the positive region of B^* with respect to A is the set of all members of the universe U, which can be properly classified to blocks of the partition B^* employing attributes from A.

For illustration let us compute $\gamma_C(D)$ for the decision table 1, where C and D are sets of condition and decision attributes respectively. It is easy to compute that the partition D^* consists of the following blocks, $X_1 = \{1\}$, $X_2 = \{2,7\}$, $X_3 = \{3,6\}$, $X_4 = \{4\}$, $X_5 = \{5,8\}$ and the partition C^* consists of blocks $Y_1 = \{1,5\}$, $Y_2 = \{2,8\}$, $Y_3 = \{3\}$, $Y_4 = \{4\}$, $Y_5 = \{6\}$, $Y_6 = \{7\}$. Next let us compute the C-lower approximation of D^*, and we get $\underline{C}X_1 = \phi$, $\underline{C}X_2 = Y_6$, $\underline{C}X_3 = Y_3 \cup Y_5$, $\underline{C}X_4 = Y_4$ and $\underline{C}X_5 = \phi$. Thus $Pos_C(D^*) = Y_3 \cup Y_4 \cup Y_5 \cup Y_6 = \{3,4,6,7\}$. That is to say that only these elements can be properly classified to blocks of the partition generated by decision attributes D employing the condition attributes C. Hence the accuracy of the approximation $\gamma_C(D) = 0.5$.

4.Dependency of Attributes. One of the basic problem in the decision tables theory is the question whether the set of conditions uniquely determines the set of decisions to be performed if the conditions are satisfied. In order to consider this problem in detail we shall define the notion of dependency of attributes.

Let $S = (U, C, D, V, f)$ be a decision table, and let $A, B \subseteq C \cup D$.

We say that B depends in a degree k $(0 \leq k \leq 1)$ on A (in S) if $k = \gamma_A(B)$. (symbolically $A \longrightarrow^k B$).

If $k = 1$ we say that B *depends totally* on A (or in short -depends); if $0 < k < 1$, we say that B *depends partially* on A, and if $k = 0$, we say that B is totally *independent* of A. If $A \longrightarrow^1 B$ we shall also write $A \longrightarrow B$.

Let us notice that $A \longrightarrow B$, iff $\widetilde{A} \subseteq \widetilde{C}$.

The following property connects two improtant notions, that of determinism and dependency of attributes.

PROPERTY 4.1. *A decision table $S = (U, C, D, V, f)$ is deterministic (consistent) iff $C \longrightarrow D$.*

The next property may be regarded as the decomposition theorem.

PROPERTY 4.2. *Each decision table*

$$S = (U, C, D, V, f)$$

can be uniquely decomposed into two decision tables

$$S_1 = (U_1, C, D, V_1, f_1) \text{ and } S_2 = (U_2, C, D, V_2, f_2)$$

such that $C \longrightarrow^1 D$ in S_1 and $C \longrightarrow^0 D$ in S_2, where

$U_1 = Pos_C(D^*)$,

f_1 is the restriction of f to U_1,

$U_2 = \bigcup_{X \in D^*} Bn_C(X)$,

f_2 is the restriction of f to U_2,

and V_1, V_2 are ranges of function f_1 and f_2 respectively.

Employing the above property we can decompose the decision table 1 as shown below:

U_1	a	b	c	d	e
3	2	0	0	1	1
4	1	1	0	2	2
6	2	2	0	1	1
7	2	1	1	1	2

Decision Table 2

U_2	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
5	1	0	2	0	1
8	0	1	1	0	1

Decision Table 3

Decision table 2 is deterministic and decision table 3 is totally non-deterministic, which is to say that all decision rules in table 2 are deterministic, and in the decision table 3- all decision rules are non-deterministic.

5.Reduction of Attributes. Our second most important notion is that of reduction of attributes. We might for example be interested to know whether all condition attributes are necessary to undertake decisions specified in a decision table, and if not - to find out the minimal subset of condition attributes having this property. In order to solve this problem we need some notions which will be defined in what follows.

Let $S = (U, C, D, V, f)$ be a decision table and let $A \subseteq C \cup D$.

We say that the set A is *independent* in S if for every $B \subset A$, $\widetilde{B} \neq \widetilde{A}$; otherwise set A is *dependent*.

A set $B \subseteq A \subseteq C \cup D$ is a *reduct* of A in S, if B is a maximal (in the sense of inclusion) independent subset of A.

For example in the decision table 1, the set of condition attributes has one reduct $\{a, b\}$. A subset of attributes can have more then one reduct, for example in the decision table below the set of condition attributes has three reducts $\{a, b\}$, $\{b, c\}$, $\{a, c\}$.

U	a	b	c	d
1	1	0	2	2
2	0	1	1	0
3	2	0	0	0
4	1	1	0	1
5	1	0	2	2
6	2	0	0	1
7	0	1	1	2
8	1	1	1	0
9	1	0	2	2
10	0	1	1	0

Decision Table 4

The following property can be employed to simplify decision tables.

PROPERTY 5.1. *Let $S = (U, C, D, V, f)$ be a decision table, and B a reduct of C. If $C \longrightarrow^k D$, then $B \longrightarrow^k D$.*

Thus reducing the set of condition attributes, we preserve the dependency between condition and decision attributes in the resulting table.

For example, in the decision table 1 the only reduct of condition attributes is $\{a, b\}$ thus the table can be simplified as shown below:

U	a	b	d	e
1	1	0	2	0
2	0	1	1	2
3	2	0	1	1
4	1	1	2	2
5	1	0	0	1
6	2	2	1	1
7	2	1	1	2
8	0	1	0	1

Decision Table 5

That means that in order to undertake decisions specified by decision attributes d and e, condition attribute c is superfluous and attributes a and b are sufficient.

The idea of a reduct of condition attributes consists in preserving a partition induced by the set of condition attributes. The idea of a reduct can be generalised, so that it would preserve now the positive region induced by the decision attributes - which leads to further simplification of a decision table.

Let $S = (U, C, D, V, f)$ be a decision table and let $A, B \subseteq C \cup D$. We say that A is independent with respect to B if for every $A' \subset A$ $Pos_A(B^*) \neq Pos_{A'}(B^*)$. Let us notice that in particular, if $A = B$ we get the previously introduced concept of dependency of a set of attributes.

$A' \subseteq A$ will be called a *relative reduct* of A with respect to B, or in short -B-reduct of A, if A' is a maximal subset of A with respect to B.

Again, if $A = B$ the relative reduct of A with respect to B coincides with the reduct of A.

The notion of a relative reduct enables us, when applied to condition and decision attributes, to reduce the set of condition attributes preserving the positive region of partition induced by the decision attributes, and consequently, the dependency between the condition and decision attributes.

An example will exhibit in more detail the introduced concept.

Let us consider the decision table 6, which represents cement klin control [5]. Attributes a, b, c, d are condition attributes and e, f are decision attributes.

U	a	b	c	d	e	f
1	3	2	2	2	2	4
2	3	2	2	1	2	4
3	2	2	2	1	1	4
4	2	2	2	2	1	4
5	3	2	2	3	2	3
6	3	3	2	3	2	3
7	4	3	2	3	2	3
8	4	3	3	3	2	2
9	4	4	3	3	2	2
10	4	4	3	2	2	2
11	4	3	3	2	2	2
12	4	2	3	2	2	2
13	3	3	2	2	2	4

Decision Table 6

In the decision table the set of condition attributes is independent, but it is dependent with respect to the decision attributes. The only D-reduct of condition attributes is the set $\{a, b, c\}$, thus the condition attribute b is superfluous when making decisions specified by the decision attributes. Hence the table can be simplyfied as below (after removing duplicate decision rules, which are obviously redundant in the table).

U	a	c	d	e	f
1	3	2	2	2	4
2	3	2	1	2	4
3	2	2	1	1	4
4	2	2	2	1	4
6	3	2	3	2	3
7	4	2	3	2	3
8	4	3	3	2	2
11	4	3	2	2	2

Decision Table 7

The relative reduct can be exploited also in another way leading to further simplification of a decision table. Let us remark that any partition consisting of n blocks can be represented as a crossproduct of n partitions consisting of two suitable blocks. In the case of decision tables this means that, for example, two decision attributes e and f can be replaced by four two valued decision attributes as shown in the next decision table.

U	a	c	d	g_1	g_2	g_3	g_4
1	3	2	2	1	0	0	0
2	3	2	1	1	0	0	0
4	2	2	1	0	1	0	0
5	2	2	2	0	1	0	0
7	3	2	3	0	0	1	0
8	4	2	3	0	0	1	0
10	4	3	3	0	0	0	1
11	4	3	2	0	0	0	1

Decision Table 8

If the value of attribute g_1 equals 1 in this decision table, this means: perform actions specified by value 2 of attribute e and value 4 of attribute f, etc. Computing reducts of condition attributes with respect to attributes g_1, g_2, g_3 and g_4 we get $\{a, d\}$, $\{a\}$, $\{c, d\}$ and $\{c\}$

respectively. This means that in order to perform decision g_i we have to know values of all attributes belonging to the g_i-reduct of condition attributes. Consequently the decision table 7 can be represented as below

U	a	c	d	e	f
1	3	x	1	2	4
2	3	x	2	2	4
4	2	x	x	1	4
7	x	2	3	2	3
10	x	3	x	2	2

Decision Table 9

in which all duplicate decision rules are removed, and crosses denote "don't care" values of attributes.

Because in general a subset of attributes can have more than one reduct (relative reduct), the simplification of decision tables does not yield unique results, and can be optimised according to preassumed criteria.

In the end let us notice that the decision table 9 is an abbreviation of a "fully expanded decision table" without "don't care" entries.

6.Decision Algorithms. With each decision table one can associate a decision algorithm. For example with the decision table 9 we can associate the following decision algorithm:

$$(a := 3).(b := 1) \Rightarrow (e := 2).(f := 4)$$
$$(a := 3).(b := 2) \Rightarrow (e := 1).(f := 4)$$
$$(a := 2) \Rightarrow (e := 1).(f := 4)$$
$$(c := 2).(d := 3) \Rightarrow (e := 2).(f := 3)$$
$$(c := 3) \Rightarrow (e := 2).(f := 2)$$

The *decision algorithm* is a finite set of decision rules, expressed in a certain formal language. Formally, decision rules take the form $t \Rightarrow s$, where t and s are *terms* called condition and decison respectively. The terms are built up from expressions of the form $(a := v)$, combined by means of boolean operators ".", "+" and "-".

In order to connect the concept of a decision table with that of decision algorihm we shall employ the *semantics function*, denoted $\| \|_S$, which associates to each term a subset of the universe U of the decision table S, and to each formula (decision rule) - the truth (T) or falsity (F).

Below, the inductive definition of semantics function is given.

$$1)\ \|(a := v)\|_S = \{x \in U : f_s(a) = v\}$$

$$2)\ \|0\|_S = \phi; \|1\|_S = U$$

$$3)\ \|t + s\|_S = \|t\|_S \cup \|s\|_S$$

$$\|ts\|_S = \|t\|_S \cap \|s\|_S$$

$$\| - t\|_S = U - \|t\|_S$$

$$4)\ \|t \Rightarrow s\|_S = \begin{cases} T, & \text{if } \|t\|_S \subseteq \|s\|_S \\ F, & \text{otherwise.} \end{cases}$$

A decision rule $t \Rightarrow s$ is *consistent* (*correct*) in S if $\|t \Rightarrow s\|_S = T$, otherwise the decision rule is *inconsistent (incorrect)* in S.

A decision algorithm is consistent (correct) in S if all its decision rules are consistent (correct) in S; otherwise the decision algorithm is inconsistent (incorrect) in S.

A decision rule $t \Rightarrow s$ is *deterministic* in S, if $\|s\|_S \in D^*$; otherwise the decision rule is *nondeterministic* in S.

A decision algorithm is deterministic in S if all its decision rules are deterministic in S; otherwise the decision algorithm is nondeterministic.

From the above definitions we see that the notion of decision table is different to that of decision algorithm, for in the latter one we can distinguish between inconsistency and nondeterminism, which is impossible in the case of decision tables.We can have for example a deterministic and incorrect decision algorithm or a nondeterministic correct decision algorithm. The relationship between these two concepts will be studied in more detail in a separate paper.

Finally let us discuss briefly another problem. In some applications we might be rather interested in presenting our decision-making procedure not in the form of a decision table but as a decision algorithm (for example in expert systems, see [9]).

In order to solve this problem we have two options. The first one (semantic) consists of simplifying the original decision table, as shown in the previous sections, and replace the simplified table by corresponding decision algorithm. The second one (syntactic), requires that we first replace the original decision table by proper decision algorithm, which is then simplified by employing logical methods. We recommend the semantical approach, for it is much simpler and faster then the syntactic one.

When deriving decision algorithms from decision tables, special attention must be paid to the nondeterministic (inconsistent) part of the decision table. Suppose we are given a decision table $S = (U, C, D, V, f)$ such that $C \longrightarrow^k D$ and k is high, for example $k = 0.9$. Then we can replace the original decision table by deterministic part, removing in this way all nondeterministic (inconsistent) decision rules,

and use only deterministic rules when making decisions. If k is low, say $k = 0.5$ this procedure will yield rather poor results, since in a substantial number of situations we will be unable to make decisions. When some probabilistic information about nondeterministic rules is available a very elegant method has been employed by Wong and Ziarko (see [11],[12] and [13]), which enables us to derive efficient decision algorithms from highly nondeterministic decision tables.

REFERENCES

1. A. Lew, *Decision Tables for General-Purpose Scientific Programming*, Software Practice and Experience 13 (1981), 181-188.
2. D. E. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading, Mass. (1968).
3. R. Maes, *An Algorithmic Approach to the Conversion of Decision Grid Charts into Compressed Decision Tables*, Comm.ACM 23(5) (1980), 286-293.
4. M.Montalbano, *Decision Tables*, Science Research Associates, Chicago (1974).
5. A. Mrózek, *Rough Sets and Some Aspects of Expert Systems Realization*, 7th International Workshop on Expert Systems and Their Applications, Avignon, France (submitted) (1987).
6. Z. Pawlak, *Rough Sets*, International Journal on Information and Computer Sciences 11(5) (1982), 341-356.
7. Z. Pawlak, *Rough Sets and Decision Tables*, Lecture Notes .Springer Verlag 208 (1986), 186-196.
8. Z. Pawlak, *On Decision Tables*, Bull.Polish Acad.Sci.Tech. 34(9-10) (1986), 563-572.
9. Z. Pawlak, K. Slowiński and R. Slowiński, *Rough Classification of Patients after Highly Selective Vagotomy for Duodenal Ulcer*, Int. Journal of Man-Machine Study 24 (1986), 423-433.
10. S. L. Pollack, H. Hick and W. J. Harrison, *Decision Tables: Theory and Practice*, *Wiley, New York (1971).*
11. S. K. M. Wong and W. Ziarko, *On Optimal Decision Rules in Decision Tables*, Bull. Polish Acad. Sci. 33(11-12) (1986), 693-696.
12. S. K. M. Wong and W. Ziarko, *On Learning and Evaluation of Decision Rules in the Contex of Rough Sets*, Proc. International Symposium on Methodologies for Intelligent Systems (Knoxville) (1986), 308-324.
13. S. K. M. Wong and W. Ziarko, *Algebraic Versus Probabilistic Independence in Decision Theory*, Ibid., 207-212.