BULLETIN OF THE POLISH
ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 34, No. 9-10, 1986

COMPUTER AND INFORMATION SCIENCE

On Decision Tables

by

Zdzisław PAWLAK

Presented by Z. PAWLAK on November 12, 1985

Summary. This paper is an extended and modified version of previous papers in which the rough set approach as a basis for decision tables theory is proposed.

1. Introduction. We show in this article (cf. [3–4]) that the concept of the rough set [2] can be used as a basis for the decision tables theory [5]. The ideas introduced in this paper have been applied to the implementation of cement kiln control algorithm [1] and showed considerable practical advantages as compared to other methods.

2. Decision tables

2.1. *Concept of a decision table.* In this section we recall after [3] and [4] a formal definition of a decision table which will be used throughout this paper.

A decision table is a system

$$S = (\mathrm{Univ},\ \mathrm{Att},\ \mathrm{Val},\ f)$$

where:

Univ — is a finite set of states, called the universe,

$\mathrm{Att} = \mathrm{Con} \cup \mathrm{Dec}$ — is the set of attributes; Con — is the set of conditions attributes and Dec — is the set of decisions attributes,

$\mathrm{Val} = \bigcup_{a \in \mathrm{Att}} \mathrm{Val}_a$, where Val_a is the set of values of an attribute $a \in \mathrm{Att}$ (domain of a),

$f\colon \mathrm{Univ} \times \mathrm{Att} \to \mathrm{Val}$ — is a total function, called the decision function, such that $f(x, a) \in \mathrm{Val}_a$ for every $x \in \mathrm{Univ}$ and $a \in \mathrm{Att}$.

A decision rule in S is a function f_x: Att $\to$ Val, such that $f_x(a) = f(x, a)$ for every $x \in$ Univ and $a \in$ Att.

If f_x is a decision rule in S then f_x/Con and f_x/Dec are called conditions and decisions of the decision rule f_x, respectively.

A decision rule f_x in S is deterministic (consistent) if for every $y \in$ Univ, $y \neq x$, f_x/Con $= f_y$/Con implies f_x/Dec $= f_y$/Dec; otherwise the decision rule f_x is nondeterministic (inconsistent).

A decision table S is deterministic (consistent) if all its decision rules are deterministic, otherwise the decision table S is nondeterministic (inconsistent).

A decision table $S' = (X, \text{Att}, \text{Val}', f')$ is said to be an X-restriction of the decision table $S = (\text{Univ}, \text{Att}, \text{Val}, f)$, if $X \subseteq$ Univ, $f' = f/X \times$ Att and $\text{Val}' = \{v \in \text{Val}: \bigvee_{x \in X} f_x(a) = v\}$.

2.2. *Rough sets.* Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be a decision table and let $a \in$ Att, $y \in$ Univ.

With every subset of attributes $A \subseteq$ Att we associate the equivalence relation $\tilde{A}$ defined thus

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

If $(x, y) \in \tilde{A}$ we say that x and y are indiscernible with respect to A in S (A-indiscernible) and $\tilde{A}$ is called an indiscernibility relation in S. Equivalence classes of the indiscernibility relation $\tilde{A}$ are called A-elementary sets in S and the family of all equivalence classes of $\tilde{A}$ is denoted by A^*.

Let $A \subseteq$ Att and $X \subseteq$ Univ in a decision table $S = (\text{Univ}, \text{Att}, \text{Val}, f)$.

By A-lower (A-upper) approximation of X in S we mean the sets

$$\underline{A}X = \{x \in \text{Univ}: [x]_{\tilde{A}} \subseteq X\}$$

$$\bar{A}X = \{x \in \text{Univ}: [x]_{\tilde{A}} \cap X \neq \emptyset\}.$$

Set $Bn_A(X) = \bar{A}X - \underline{A}X$ will be called A-boundary of X in S.

We shall use also the following definitions: A-positive region of set X is the set $\underline{A}X$; A-doubtful region of set X is the set $Bn_A(X)$; A-negative region of set X is the set $\text{Neg}_A X = \text{Univ} - \bar{A}X$.

If $\underline{A}X = \bar{A}X$ we say that set X is A-definable in S; otherwise set X is A-nondefinable in S.

Nondefinable sets will be called also rough sets in S.

The number

$$\alpha_A(X) = \frac{\text{card } \underline{A}X}{\text{card } \bar{A}X}$$

will be called the accuracy of the X with respect to A in S, and the number

$$\varrho_A(X) = 1 - \alpha_A(X)$$

will be called the roughness of the set X with respect to A in S.

Let us notice that each subset of attributes $A \subseteq \mathrm{Att}$ in a decision table $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ defines uniquely the topological space $T_S = (\mathrm{Univ}, \mathrm{Def}_A(S))$, where $\mathrm{Def}_A(S)$ is the family of all A-definable sets in S, and the lower and upper approximations are interior and closure in the topological space T_S, thus approximations have the following properties:

1) $\underline{A}X \subseteq X \subseteq \bar{A}X$
2) $\underline{A}\emptyset = \bar{A}\emptyset = \emptyset$; $\underline{A}\,\mathrm{Univ} = \bar{A}\,\mathrm{Univ} = \mathrm{Univ}$,
3) $\underline{A}(X \cup Y) \supseteq \underline{A}X \cup \underline{A}Y$
4) $\bar{A}(X \cup Y) = \bar{A}X \cup \bar{A}Y$
5) $\underline{A}(X \cap Y) = \underline{A}X \cap \underline{A}Y$
6) $\bar{A}(X \cap Y) \subseteq \bar{A}X \cap \bar{A}Y$
7) $\underline{A}(-X) = -\bar{A}(X)$
8) $\bar{A}(-X) = -\underline{A}(X)$.

Moreover in this topological space we have the following two properties:

9) $\underline{A}\,\underline{A}X = \bar{A}\underline{A}X$
10) $\bar{A}\bar{A}X = \underline{A}\bar{A}X$.

From the topological view the rough sets can be classified as follows:

a) Set X is roughly A-definable in S if $\underline{A}X \neq \emptyset$ and $\bar{A}X \neq \mathrm{Univ}$,
b) Set X is internally A-nondefinable in S if $\underline{A}X = \emptyset$ and $\bar{A}X \neq \mathrm{Univ}$,
c) Set X is externally A-nondefinable in S if $\bar{A}X = \mathrm{Univ}$ and $\underline{A}X \neq \emptyset$,
d) Set X is totally A-nondefinable in S if $\underline{A}X = \emptyset$ and $\bar{A}X = \mathrm{Univ}$.

2.3. *Dependence of attributes.* Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be a decision table, $F = \{X_1, X_2, \ldots, X_n\}$, where $X_i \subseteq \mathrm{Univ}$, a family of subsets of Univ and $A \subseteq \mathrm{Att}$.

By A-lower (A-upper) approximation of F in S we mean the families

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

The A-positive region of a family F is the set

$$\mathrm{Pos}_A(F) = \bigcup_{X_i \in F} \underline{A}X_i.$$

The A-doubtful region of a family F is the set

$$Bn_A(F) = \bigcup_{X_i \in F} Bn_A X_i.$$

The A-negative region of a family F is the set

$$\mathrm{Neg}_A(F) = \mathrm{Univ} - \bigcup_{X_i \in F} \bar{A} X_i.$$

The number

$$\gamma_A(F) = \frac{\mathrm{card}\ \mathrm{Pos}_A(F)}{\mathrm{card\ Univ}}$$

will be called the quality at the approximation of F by A in S.

Let $B, C \subseteq \mathrm{Att}$ be two subsets of attributes in $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$, and k — real number such that $0 \leqslant k \leqslant 1$.

We say that C depends in a degree k on B in S, in symbols $B \xrightarrow{k} C$, if $k = \gamma_B(C^*)$.

If $k = 1$ we say that C totally depends on B in S and we write also $B \to C$ instead of $B \xrightarrow{1} C$. If $0 < k < 1$ we say that C roughly depends on B in S. If $k = 0$ we say that C is totally independent on B in S.

The following properties are valid:

Property 2.3.1. A decision table $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ is deterministic iff $\mathrm{Con} \to \mathrm{Dec}$ in S.

A decision table $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ is called roughly deterministic if $\mathrm{Con} \xrightarrow{k} \mathrm{Dec}$ and $0 < k < 1$.

Property 2.3.2. The following properties are true:

1) $\mathrm{Con} \xrightarrow{1} \mathrm{Dec}$ in $S/\mathrm{Pos}_{\mathrm{Con}}(\mathrm{Dec}^*)$

2) $\mathrm{Con} \xrightarrow{0} \mathrm{Dec}$ in $S/Bn_{\mathrm{Con}}(\mathrm{Dec}^*)$.

Note. The above property says that every decision table can be decomposed into two parts (possibly empty) such that one is deterministic and the second totally nondeterministic.

2.4. *Reduction of attributes.* Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be a decision table and let $A \subseteq \mathrm{Att}$.

Set A is independent in S if for every $B \subset A$, $\tilde{B} \supset \tilde{A}$. Set A is dependent in S if there exists $B \subset A$ such that $\tilde{B} = \tilde{A}$.

Set $B \subseteq A$ is a reduct of A in S if B is the maximal independent set in S.

Subset $B \subseteq A$ is a reduct of A with respect to $C \subseteq \mathrm{Att}$ in S if B is an independent subset of A such that $\gamma_B(C^*) = \gamma_A(C^*)$ (or $\mathrm{Pos}_B(C^*) = \mathrm{Pos}_A(C^*)$).

Let us notice that if $A = C$ the reduct of A with respect to C coincide with the reduct of A.

Property 2.4.1. If $A \xrightarrow{k} B$ in S and C is a reduct of A, or reduct of A with respect to B in S, then $C \xrightarrow{k} B$.

In particular, if C is a reduct of conditions attributes Con in a decision table S and Con $\xrightarrow{k}$ Dec, then $C \xrightarrow{k}$ Dec. This is to mean that we can simplify the decision table by reducing the set of conditions attributes.

We can also define the approximate reduct (or approximate reduct with respect to a subset C) in the following way:

Let $0 \leqslant \varepsilon \leqslant 1$ be a real number and let $B \subseteq A \subseteq$ Att in a decision table $S = (\text{Univ, Att, Val}, f)$.

Subset B of A is a ε-reduct of A in S if B is independent in S and $\gamma_B(A^*) = 1 - \varepsilon$.

Subset B of A is a ε-reduct of A in S with respect to $C \subseteq$ Att if B is independent in S and $\gamma_B(C^*) = \gamma_A(C^*) = \varepsilon$.

Directly from these definitions we have

Property 2.4.2. If B is a ε-reduct of A in S then $B \xrightarrow{1-\varepsilon} A$.

Property 2.4.3. If B is a ε-reduct of A in S with respect to $C \subseteq$ Att, and $A \xrightarrow{k} C$, then $B \xrightarrow{k-\varepsilon} C$.

In particular, if Con $\xrightarrow{k}$ Dec in S and $C \subseteq$ Con is a ε-reduct of Con in S, then $C \xrightarrow{k-\varepsilon}$ Dec. That is to mean that we can reduce the set of conditions attributes, in such a way that the degree of dependence between decisions and conditions attributes is decreased by the constant ε.

3. The decision language

3.1. *Syntax of the decision language.* With each decision table $S = (\text{Univ, Att, Val}, f)$ we associate a decision language L_S, which consists of terms, formulas and decision algorithms.

Terms are built up from some constants by means of Boolean operation $+$, $\cdot$, $-$. We assume that 0, 1 are constants and Att, Val are some finite sets of constants called attributes and values of attributes, respectively.

The set of terms is the least set satisfying the conditions:

1) Constants 0 and 1 are terms in L,
2) Any expression of the form $(a := v)$ where $a \in$ Att and $v \in \text{Val}_a$ is a term in L,
3) If t and s are terms in L, so are $-t$, $(t+s)$ and $(t \cdot s)$ (or simple (ts)).

The set of formulas in the information language L is the least set satisfying the conditions:

1) Constants T (for true) and F (for false) are formulas in L,
2) If t and s are terms in L, then $t = s$ and $t \Rightarrow s$ are formulas in L.
3) If Φ and Ψ are formulas in L, then $\sim \Phi$, $(\Phi \vee \Psi)$, $(\Phi \wedge \Psi)$, $(\Phi \rightarrow \Psi)$ and $(\Phi \leftrightarrow \Psi)$ are also formulas in L.

Any formula of the form $t \Rightarrow s$ will be called a decision rule in L; t is referred as a predecessor and s the successor of the decision rule respectively.

Any finite set of decision rules in L is called a decision algorithm in L.

With every decision algorithm $\mathfrak{A} = \{t_i \Rightarrow s_i\}_m$, $1 \leqslant i \leqslant m$ in L we associate the formula $\Psi_{\mathfrak{A}} = \bigwedge_{i=1}^{m} (t_i \Rightarrow s_i)$ called the decision formula of $\mathfrak{A}$ in L.

3.2. *The meaning* (*the semantics*) *of terms and formulas in L.* Now we shall define formally the meaning of terms and formulas in a decision table $S = (\text{Univ, Att, Val}, f)$. Terms are intended to mean subsets of the universe Univ and the meaning of formulas is truth or falsity. Of course the meaning of a certain term or formula can be different in various information systems.

In order to define the meaning of terms and formulas we shall use the meaning function $g_S\colon \text{Ter} \cup \text{For} \rightarrow \gamma(\text{Univ}) \cup \{T, F\}$, where Ter and For denote the set of all terms and formulas, respectively.

The meaning function g_s for terms is defined as follows (we omit the subscript S if S is understood):

1) $g(0) = \emptyset$; $g(1) = \text{Univ}$
2) $g(q := v) = \{x \in \text{Univ}: f(x, q) = v\}$
3) $g(-t) = \text{Univ} - g(t)$

   $g(t+s) = g(t) \cup g(s)$

   $g(ts) = g(t) \cap g(s)$.

The meaning of formulas is defined thus:

1) $g(T) = T$; $\quad g(F) = F$,
2) $g(t = s) = \begin{cases} T, & \text{if} \quad g(t) = g(s) \\ F, & \text{otherwise} \end{cases}$
3) $g(t \Rightarrow s) = \begin{cases} T, & \text{if} \quad g(t) \subseteq g(s) \\ F, & \text{otherwise} \end{cases}$
4) $g(\sim \Phi) = \begin{cases} T, & \text{if} \quad g(\Phi) = F \\ F, & \text{if} \quad g(\Phi) = T \end{cases}$
5) $g(\Phi \vee \Psi) = g(\Phi) \vee g(\Psi)$

$g(\Phi \wedge \Psi) = g(\Phi) \wedge g(\Psi)$

$g(\Phi \rightarrow \Psi) = g(\sim \Phi) \vee g(\Psi)$

$g(\Phi \leftrightarrow \Psi) = g(\Phi \rightarrow \Psi) \wedge g(\Phi \leftarrow \Psi)$.

If $g_s(\Phi) = T$ we say that Φ is true in S; if $g_s(\Phi) = F$ then Φ is said to be false in S. If Φ is true in S we shall write $\models_S \Phi$ or simply $\models \Phi$ when S is known.

If $\models_S (t = s)$ we say that terms t and s are equivalent in S; if $\models_S (t \Rightarrow s)$ we say that term t implies term s in S. If $\models_S (\Phi \leftrightarrow \Psi)$ we say that formulas Φ and Ψ are equivalent in S and if $\models_S (\Phi \rightarrow \Psi)$ we say that formula Φ implies formula Ψ in S.

For the transformation of terms we shall use the axioms of Boolean algebra and the following specific axiom

$$(a := v) = -\sum_{u \neq v,\, u \in \mathrm{Val}_a} (a := u).$$

For the transformation of formulas we shall employ the axioms of propositional calculus.

A term t in L is A-elementary ($A \subseteq \mathrm{Att}$) if $t = \prod_{a \in A} (a := v_a)$.

A term s in L is an A-normal form if $t = \sum s$, where all s are A-elementary.

Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be a decision table $A \subseteq \mathrm{Att}$ subset of attributes, and L_A — an information language with the set of attributes A.

Property 3.2.1. For every term t in L_P there exists the term s in L_A in A-normal form, such that $\models_S t = s$; s is referred as the A-normal form of t in L_A.

Subset $X \subseteq \mathrm{Univ}$ is said to be A-definable in L ($A \subseteq \mathrm{Att}$) if there exists a term t in L_A such that $g_S(t) = X$; the term t is called the A-description of X in L.

If set $X \subseteq \mathrm{Univ}$ is not A-definable in L, then the terms t and s such that $g_S(t) = \underline{A}X$ and $g_S(s) = \bar{A}X$ are called the A-lower and A-upper descriptions of X in L, respectively.

3.3. *Decision rules.* Our basic concept is that of a decision rule. We shall discuss this concept in some details in this section.

Let $t \Rightarrow s$ be a decision rule in L, and let A, B be two subsets of attributes which occurs in t and s, respectively. We shall call then $t \Rightarrow s$ an (A, B)-decision rule. If A and B are single element sets, for the sake of simplicity, we shall use the expression (a, b)-decision rule.

Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be a decision table and $t \Rightarrow s$ an (A, B)-decision rule in L.

We say that an (A, B)-decision rule is B-deterministic in S if $g_S(s) \in \mathscr{B}^*$, i.e. $g_S(s)$ is a description of some equivalence class of the equivalence relation B; otherwise the decision rule is B-nondeterministic.

We say that an (A, B)-decision rule $t \Rightarrow s$ is in $A \cup B$-normal form if t and s are in $A \cup B$-normal form.

Property 3.3.1. An (A, B)-decision rule $t \Rightarrow s$ is true in S iff all non-empty $A \cup B$-elementary terms occurring in $A \cup B$-normal form of t occur also in the $A \cup B$-normal form of s.

This property enables us to prove the validity of any decision rule in a simple syntactical way.

3.4. *Decision algorithms.* Now we shall discuss the most important concept of our approach — the decision algorithm.

A decision algorithm $\mathfrak{A}$ in L is said to be correct in S if $\models_S \Psi_{\mathfrak{A}}$.

A decision algorithm $\mathfrak{A}$ in L is A-deterministic in S $(A \subseteq \mathrm{Att})$ if all its decision rules are A-deterministic in S; otherwise the algorithm is A-nondeterministic.

If A and B are the sets of all attributes occurring in the predecessors and successors of the decision rules in an decision algorithm $\mathfrak{A}$, then $\mathfrak{A}$ will be called the (A, B)-decision algorithm.

An (A, B)-decision algorithm is total in S if for every equivalence class X of the equivalence relation B, there exists a decision rule $t_i \Rightarrow s_i$ in $\mathfrak{A}$ such that $g_S(s_i) \supseteq X_j$; otherwise the decision algorithm is partial in S.

The following properties are used as transformation rules for decision algorithms:

Property 3.4.1.

1) $$\models_S \bigwedge_{i=1}^{m} (t_i \Rightarrow s) \to \Big(\sum_{i=1}^{m} t_i \Rightarrow s\Big),$$

2) $$\models_S \big((t \Rightarrow s) \wedge ((t \Rightarrow s) \to (p \Rightarrow r))\big) \to (p \Rightarrow r).$$

Property 3.4.1. 2) can be regarded as a "modus ponens" for decision rules.

The following important property establishes a relationship between dependency of attributes and the decision algorithm.

Property 3.4.2. Let $\mathfrak{A}$ be an (A, B)-decision algorithm in L.

$$\models_S \Psi_{\mathfrak{A}} \quad \text{iff} \quad A \underset{S}{\to} B.$$

4. Example. Let us consider the following decision table taken from Mrózek [1], and describing cement kiln control.

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

where a, b, c and d are conditions attributes and e and f are decision attributes.

It is easy to check that the decision table is deterministic, i.e. $\{a, b, c, d\} \to \to \{e, f\}$, and the set of control attributes as well the set of decision attributes are independent.

The corresponding decision algorithm is the following:

$$(a: = 3)\ (d: \neq 3) \Rightarrow (e: = 2)\ (f: = 4)$$

$$(a: = 2) \Rightarrow (e: = 1)\ (f: = 4)$$

$$(c: = 2)\ (d: = 3) \Rightarrow (e: = 2)\ (f: = 3)$$

$$(c: = 3) \Rightarrow (e: = 2)\ (f: = 2).$$

where $d: \neq 3$ is an abbreviation of $(d: = 0) + (d: = 1) + (d: = 2)$.

DEPARTMENT OF COMPLEX CONTROL SYSTEMS, POLISH ACADEMY OF SCIENCES, BAŁTYCKA 5, 44-100 GLIWICE
(ZAKŁAD SYSTEMÓW AUTOMATYKI KOMPLEKSOWEJ PAN) *
DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF NORTH CAROLINA, NC 28223, USA

REFERENCES

[1] A. Mrózek, *Information systems and control algorithms*, Bull. Pol. Ac.: Tech., 33 (1985), 195–204.

[2] Z. Pawlak, *Rough Sets*, Int. J. Inform. Comp. Sci., **11** (1982), 341–356.

[3] Z. Pawlak, *Decision tables and decision algorithms*, Bull. Pol. Ac.: Tech., **33** (1985), 487–494.

[4] Z. Pawlak, *Rough sets and decision tables*, Lecture Notes, Springer-Verlag (in press).

[5] S. Pollack, H. Hicks, W. Harrison, *Decision tables: theory and practice*, Wiley and Sons Inc., New York, 1971.

З. Павляк, **О таблицах принятия решений**

Настоящая статья представляет собой расширенный и модифицированный вариант работ автора, в которых предлагалось использование приближенных множеств в качестве основы для теории таблиц принятия решений.