Bankruptcy Prediction with Rough Sets

Jan C. Bioch and Viara Popova

ERIM Report Series *Research in Management*	
ERIM Report Series reference number	ERS-2001-11-LIS
Publication	February 2001
Number of pages	17
Email address first author	Bioch@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Erasmus Universiteit Rotterdam PoBox 1738 3000 DR Rotterdam, The Netherlands Phone: # 31-(0) 10-408 1182 Fax: # 31-(0) 10-408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS		
Abstract	The bankruptcy prediction problem can be considered an or dinal classification problem. The classical theory of Rough Sets describes objects by discrete attributes, and does not take into account the ordering of the attributes values. This paper proposes a modification of the Rough Set approach applicable to monotone datasets. We introduce respectively the concepts of monotone discernibility matrix and monotone (object) reduct. Furthermore, we use the theory of monotone discrete functions developed earlier by the first author to represent and to compute decision rules. In particular we use monotone extensions, decision lists and dualization to compute classification rules that cover the whole input space. The theory is applied to the bankruptcy prediction problem.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HG 3760 HD 30.27	Bankruptcy Forecasting
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11 R 4	Production Management Transportation Systems
	C 53	Forecasting and other model applications
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	250 E	Forecasting
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34 85.20	Logistiek management Bestuurlijke informatie, informatieverzorging
	85.03	Methode en technieken, operations research
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Faillisementen, Voorspellen, Verzamelingen (wiskunde) Classificatietheorie,	
Free keywords	bankruptcy prediction, ordinal classification, rough sets, attributes selection, decision rules	

Bankruptcy Prediction with Rough Sets

Jan C. Bioch and Viara Popova

Dept. of Computer Science, Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam.
{bioch,popova}@few.eur.nl

Abstract. The bankruptcy prediction problem can be considered an ordinal classification problem. The classical theory of Rough Sets describes objects by discrete attributes, and does not take into account the ordering of the attributes values. This paper proposes a modification of the Rough Set approach applicable to monotone datasets. We introduce respectively the concepts of monotone discernibility matrix and monotone (object) reduct. Furthermore, we use the theory of monotone discrete functions developed earlier by the first author to represent and to compute decision rules. In particular we use monotone extensions, decision lists and dualization to compute classification rules that cover the whole input space. The theory is applied to the bankruptcy prediction problem.

Keywords: bankruptcy prediction, ordinal classification, rough sets, attributes selection, decision rules

1 The Bankruptcy Prediction Problem

The research in bankruptcy prediction has a long history dating back to the 1930s. A number of statistical methods were applied for developing models able to predict in advance whether a company will go bankrupt or not. The analysis is based on financial information about the company in the form of financial indicators and ratios obtained from the company's annual reports. The main goal is to describe the relationship between these indicators and bankruptcy using available data about companies that already went bankrupt and data about 'healthy' companies.

A major breakthrough in the research was achieved in the 1960s by applying the method of discriminant analysis for bankruptcy prediction. In 1968 Altman proposed multivariate discriminant analysis for developing the prediction model [1] and the approach has been further improved and tested in a number of studies since then. However, this method makes several assumptions that are not always present in real-life data. This encouraged the researchers to look for alternatives. One of them is the logistic analysis that was proposed in the 1980s. It was applied on bankruptcy data and gave very good results.

The success of the machine learning methods in a number of application domains suggested that they might be useful for predicting bankruptcy as well. Neural networks, decision trees, genetic algorithms, rough sets and other machine learning approaches were applied on bankruptcy data with promising results [2, 17, 20, 22, 23]. In a number of studies these methods are tested and compared to the traditional statistical techniques. Some of the articles suggest that the new approaches outperform the classical methods on datasets from a number of areas including bankruptcy prediction.

In this paper the bankruptcy prediction problem is interpreted as an ordinal classification problem and an extension of the rough sets theory is proposed for extracting classification rules from ordinal datasets.

2 Ordinal Classification

Ordinal classification refers to the category of problems, in which the attributes of the objects to be classified are ordered. Ordinal classification has been studied by a number of authors, e.g. [3, 18, 6, 21, 13]. The classical theory of Rough Sets does not take into account the ordering of the attribute values. While this is a general approach that can be applied on a wide variety of data, for specific problems we might get better results if we use this property of the problem. This paper proposes a modification of the Rough Sets approach applicable to monotone datasets. Monotonicity appears as a property of many real-world problems and often conveys important information. Intuitively it means that if we increase the value of a condition attribute in a decision table containing examples, this will not result in a decrease in the value of the decision attribute. Therefore, monotonicity is a characteristic of the problem itself and when analyzing the data we get more appropriate results if we use methods that take this additional information into account. Our approach uses the theory of monotone discrete functions developed earlier in [4]. We introduce respectively monotone decision tables/datasets, monotone discernibility matrices and monotone reducts in section 3 and consider some issues of complexity. In section 4 we introduce monotone discrete functions and show the relationship with Rough Set Theory. As a corollary we find an efficient alternative way to compute classification rules. In section 5 we discuss the application of the theory to the bankruptcy dataset earlier investigated in [13]. It appears that our method is more advantageous in several aspects. Conclusions are given in section 6.

3 Monotone Information Systems

An *information system* S is a tuple $S = \{U, A, V\}$ where: $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty, finite set of objects (observations, examples), $A = \{a_1, a_2, \ldots, a_m\}$ is a non-empty, finite set of attributes, and $V = \{V_1, V_2, \ldots, V_m\}$ is the set of domains of the attributes in A. A *decision table* is a special case of an information system where among the attributes in A we distinguish one called a *decision attribute*. The other attributes are called *condition attributes*. Therefore: $A = C \cup \{d\}$, $C = \{a_1, a_2, \ldots, a_m\}$ where a_i - condition attributes, d - decision attribute.

We call the information system $S = \{U, C \cup \{d\}, V\}$ *monotone* when for each couple $x_i, x_j \in U$ the following holds:

$$a_k(x_i) \geq a_k(x_j), \forall a_k \in C \Rightarrow d(x_i) \geq d(x_j) \ , \tag{1}$$

where $a_k(x_i)$ is the value of the attribute a_k for the object x_i. The following example will serve as a running example for this paper.

Example 1. The following decision table represents a monotone dataset:

Table 1. Monotone decision table

U	a	b	c	d
1	0	1	0	0
2	1	0	0	1
3	0	2	1	2
4	1	1	2	2
5	2	2	1	2

3.1 Monotone Reducts

Let $S = \{U, C \cup \{d\}, V\}$ be a decision table. In the classical rough sets theory, the discernibility matrix (DM) is defined as follows:

$$(c_{ij}) = \begin{cases} \{a \in A : a(x_i) \neq a(x_j)\} & \text{for } i,j : d(x_i) \neq d(x_j) \\ \emptyset & \text{otherwise} \ . \end{cases} \tag{2}$$

The variation of the DM proposed here is the *monotone discernibility matrix* $M_d(S)$ defined as follows:

$$(c_{ij}) = \begin{cases} \{a \in A : a(x_i) > a(x_j)\} & \text{for } i,j : d(x_i) > d(x_j) \\ \emptyset & \text{otherwise} \ . \end{cases} \tag{3}$$

Based on the monotone discernibility matrix, the *monotone discernibility function* can be constructed following the same procedure as in the classical Rough Sets approach. For each non-empty entry of the monotone M_d $c_{ij} = \{a_{k_1}, a_{k_2}, \ldots, a_{k_l}\}$ we construct the conjunction $C = a_{k_1} \wedge a_{k_2} \wedge \ldots \wedge a_{k_l}$. The disjunction of all these conjunctions is the monotone discernibility function:

$$f = C_1 \vee C_2 \vee \ldots \vee C_p \ . \tag{4}$$

The *monotone reducts* of the decision table are the minimal transversals of the entries of the monotone discernibility matrix. In other words the monotone reducts are the minimal subsets of condition attributes that have a non-empty intersection with each non-empty entry of the monotone discernibility matrix. They are computed by dualizing the Boolean function f, see [5, 4, 16]. In section

3.3 we give another equivalent definition for a monotone reduct described from a different point of view.

Example 2. Consider the decision table from example 1. The general and *monotone* discernibility matrix modulo decision for this table are respectively:

Table 2. General decision matrix

	1	2	3	4	5
1	$\emptyset$				
2	a, b	$\emptyset$			
3	b, c	a, b, c	$\emptyset$		
4	a, c	b, c	$\emptyset$	$\emptyset$	
5	a, b, c	a, b, c	$\emptyset$	$\emptyset$	$\emptyset$

Table 3. Monotone decision matrix

	1	2	3	4	5
1	$\emptyset$				
2	a	$\emptyset$			
3	b, c	b, c	$\emptyset$		
4	a, c	b, c	$\emptyset$	$\emptyset$	
5	a, b, c	a, b, c	$\emptyset$	$\emptyset$	$\emptyset$

The general discernibility function is $f(a, b, c) = ab \vee ac \vee bc$. Therefore, the general reducts of table 1 are respectively: $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$ and the core is empty. However, the *monotone* discernibility function is $g(a, b, c) = a \vee bc$. So the *monotone* reducts are: $\{a, b\}$ and$\{a, c\}$, and the *monotone* core is $\{a\}$. It can be proved that monotone reducts preserve the monotonicity property of the dataset.

Complexity Generating a reduct of minimum length is an NP-hard problem. Therefore, in practice a number of heuristics are preferred for the generation of only one reduct. Two of these heuristics are the "Best Reduct" method [14] and Johnson's algorithm [15]. The complexity of a total time algorithm for the problem of generating all minimal reducts (or dualizing the discernibility function) has been intensively studied in Boolean function theory, see [5, 11, 4]. Unfortunately, this problem is still unsolved, but a quasi-polynomial algorithm is known [12]. However, these results are not mentioned yet in the rough set literature, see e.g. [16].

3.2 Heuristics

As it was mentioned above, two of the more successful heuristics for generating one reduct are the Johnson's algorithm and the "Best reduct" heuristic. Strictly speaking these methods do not necessarily generate reducts, since the minimality requirement is not assured. Therefore, in the *sequel* we will make the distinction between reducts vs *minimal* reducts. A good approach to solve the problem is to generate the reduct and then check whether any of the subsets is also a reduct. The *Johnson heuristic* uses a very simple procedure that tends to generate a reduct with minimal length (which is not guaranteed, however). Given the discernibility matrix, for each attribute the number of entries where it appears is counted. The one with the highest number of entries is added to the future reduct. Then all the entries containing that attribute are removed and the procedure repeats until all the entries are covered. It is logical to start the procedure with simplifying the set of entries (removing the entries that contain strictly or non strictly other elements). In some cases the results with and without simplification might be different. The *"Best reduct"* heuristic is based on the significance of attributes measure. The procedure starts with the core and on each step adds the attribute with the highest significance, if added to the set, until the value reaches one. In many of the practical cases the two heuristics give the same result, however, they are not the same and a counter example can be given. The dataset discussed in section 4, for example, gives different results when the two heuristics are applied.

3.3 Rule Generation

The next step in the classical Rough Set approach [19, 16] is, for the chosen reduct, to generate the *value (object) reducts* using a similar procedure as for computing the reducts. A contraction of the discernibility matrix is generated based only on the attributes in the reduct. Further, for each row of the matrix, the object discernibility function is constructed - the discernibility function relative to this particular object. The object reducts are the minimal transversals of the object discernibility functions.

Using the same procedure but on the monotone discernibility matrix, we can generate the monotone object reducts. Based on them, the classification rules are constructed. For the monotone case we use the following format:

$$\text{if } (a_{i_1} \geq v_1) \wedge (a_{i_2} \geq v_2) \wedge \ldots \wedge (a_{i_l} \geq v_l) \text{ then } d \geq v_{l+1} \ . \tag{5}$$

It is also possible to construct the classification rules using the dual format:

$$\text{if } (a_{i_1} \leq v_1) \wedge (a_{i_2} \leq v_2) \wedge \ldots \wedge (a_{i_l} \leq v_l) \text{ then } d \leq v_{l+1} \ . \tag{6}$$

This type of rules can be obtained by the same procedure only considering the columns of the monotone discernibility matrix instead of the rows. As a result we get rules that cover at least one example of class smaller than the maximal class value and no examples of the maximal class.

It can be proved that in the monotone case it is not necessary to generate the value reducts for all the objects - the value reducts of the minimal vectors of each class will also cover the other objects from the same class. For the rules with the dual format we consider respectively the maximal vectors of each class. Tables 4 and 5 show the complete set of rules generated for the whole table.

A set of rules is called a *cover* if all the examples with class $d \geq 1$ are covered, and no example of class 0 is covered. The minimal covers (computed by solving a set-covering problem) for the full table are shown in tables 6 and 7. In this case the minimal covers correspond to the unique minimal covers of the reduced tables associated with respectively the monotone reducts {a,b} and {a,c}.

The set of rules with dual format is not an addition but rather an alternative to the set rules of the other format. If used together they may be conflicting in some cases. It is known that the decision rules induced by object reducts in

Table 4. Monotone decision rules

class $d \geq 2$	class $d \geq 1$
$a \geq 2$	$a \geq 1$
$b \geq 2$	
$a \geq 1 \wedge b \geq 1$	
$c \geq 1$	

Table 5. The dual format rules

class $d \leq 0$	class $d \leq 1$
$a \leq 0 \wedge b \leq 1$	$b \leq 0$
$a \leq 0 \wedge c \leq 0$	$c \leq 0$

Table 6. mincover ab

class $d \geq 2$	class $d \geq 1$
$a \geq 1 \wedge b \geq 1$	$a \geq 1$
$b \geq 2$	

Table 7. mincover ac

class $d \geq 2$	class $d \geq 1$
$c \geq 1$	$a \geq 1$

general do not cover the whole input space. Furthermore, the class assigned by these decision rules to an input vector is not uniquely determined. We therefore briefly discuss the concept of an *extension* of a discrete data set or decision table in the next section.

4 Monotone Discrete Functions

The theory of monotone discrete functions as a tool for data-analysis has been developed in [4]. Here we only briefly review some concepts that are crucial for our approach. A discrete function of n variables is a function of the form:

$$f : X_1 \times X_2 \times \ldots \times X_n \to Y \ ,$$

where $X = X_1 \times X_2 \times \ldots \times X_n$ and Y are finite sets. Without loss of generality we may assume: $X_i = \{0, 1, \ldots, n_i\}$ and $Y = \{0, 1, \ldots, m\}$. Let $x, y \in X$ be two discrete vectors. Least upper bounds and greatest lower bounds will be defined as follows:

$$x \vee y = v, \ \text{where } v_i = \max\{x_i, y_i\} \tag{7}$$

$$x \wedge y = w, \ \text{where } w_i = \min\{x_i, y_i\} \ . \tag{8}$$

Furthermore, if f and g are two discrete functions then we define:

$$(f \vee g)(x) = \max\{f(x), g(x)\} \tag{9}$$

$$(f \wedge g)(x) = \min\{f(x), g(x)\} \ . \tag{10}$$

Table 8. mincover ab (dual format)

class $d \leq 0$	class $d \leq 1$
$a \leq 0 \wedge b \leq 1$	$b \leq 0$

Table 9. mincover ac (dual format)

class $d \leq 0$	class $d \leq 1$
$a \leq 0 \wedge c \leq 0$	$c \leq 0$

(Quasi) complementation for X is defined as: $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$, where $\overline{x_i} = n_i - x_i$. Similarly, the complement of $j \in Y$ is defined as $\overline{j} = m - j$. The complement of a discrete function f is defined by: $\overline{f}(x) = \overline{f(x)}$. The *dual* of a discrete function f is defined as: $f^d(x) = \overline{f}(\overline{x})$. A discrete function f is called *positive (monotone non-decreasing)* if $x \leq y$ implies $f(x) \leq f(y)$.

4.1 Representations

Normal Forms Discrete variables are defined as:

$$x_{ip} = \text{if } x_i \geq p \text{ then } m \text{ else } 0, \text{ where } 1 \leq p \leq n_i,\ i \in (n] = \{1, \ldots, n\}\ . \tag{11}$$

Thus: $\overline{x}_{ip+1} = \text{if } x_i \leq p \text{ then } m \text{ else } 0$. Furthermore, we define $x_{in_i+1} = 0$ and $\overline{x}_{in_i+1} = m$. *Cubic* functions are defined as:

$$c_{v,j} = j.x_{1v_1} x_{2v_2} \cdots x_{nv_n}\ . \tag{12}$$

Notation: $c_{v,j}(x) =$ if $x \geq v$ then j else 0, $j \in (m]$.
Similarly, we define *anti-cubic* functions by:

$$a_{w,i} = i \vee x_{1w_1+1} \vee x_{2w_2+1} \cdots \vee x_{nw_n+1}\ . \tag{13}$$

Notation: $a_{w,i}(x) =$ if $x \leq w$ then i else m, $i \in [m) = \{0, \ldots, m-1\}$. Note, that $j.x_{ip}$ denotes the conjunction $j \wedge x_{ip}$, where $j \in Y$ is a constant, and $x_{ip}x_{jq}$ denotes $x_{ip} \wedge x_{iq}$. A cubic function $c_{v,j}$ is called a *prime implicant* of f if $c_{v,j} \leq f$ and $c_{v,j}$ is maximal w.r.t. this property. The DNF of f:

$$f = \bigvee_{v,j} \{c_{v,j} \mid v \in\ j \in (m]\}\ , \tag{14}$$

is a unique representation of f as a disjunction of all its prime implicants (v is a minimal vector of class $d \geq j$).

If x_{ip} is a discrete variable and $j \in Y$ a constant then $x_{ip}^d = x_{i\overline{p}+1}$ and $j^d = \overline{j}$. The dual of the positive function $f = \bigvee_{v,j} j.c_{v,j}$ equals $f^d = \bigwedge_{v,j} \overline{j} \vee a_{\overline{v},\overline{j}}$.

Example 3. Let f be the function defined by table 6 and let e.g. x_{11} denote the variable: if $a \geq 1$ then 2 else 0, etc. Then $f = 2.(x_{11}x_{21} \vee x_{22}) \vee 1.x_{11}$, and $f^d = 2.x_{12}x_{21} \vee 1.x_{22}$.

Decision Lists

In [4] we have shown that monotone functions can effectively be represented by decision lists of which the minlist and the maxlist representations are the most important ones. We introduce these lists here only by example. The minlist representation of the functions f and f^d of example 2 are respectively:

$f(x) =$ if $x \geq 11, 02$ then 2 else if $x \geq 10$ then 1 else 0, and
$f^d(x) =$ if $x \geq 21$ then 2 else if $x \geq 02$ then 1 else 0.

The meaning of the minlist of f is given by:
 if $(a \geq 1 \wedge b \geq 1) \vee b = 2$ then 2 else if $a \geq 1$ then 1 else 0.
The maxlist of f is obtained from the minlist of f^d by complementing the minimal vectors as well as the function values, and by reversing the inequalities. The maxlist representation of f is therefore:
 $f(x) =$ if $x \leq 01$ then 0 else if $x \leq 20$ then 1 else 2, or equivalently:
 if $a = 0 \wedge b \leq 1$ then 0 else if $b = 0$ then 1 else 2.
The two representations are equivalent to the following table that contains respectively the minimal and maximal vectors for each decision class of f. Each representation can be derived from the other by dualization.

Table 10. Two representations of f

minvectors	maxvectors	class
11, 02		2
10	20	1
	01	0

4.2 Extensions of Monotone Datasets

A *partially defined discrete function* (pdDf) is a function: $f : D \mapsto Y$, where $D \subseteq X$. We assume that a pdDf f is given by a decision table such as e.g. table 1. Although pdDfs are often used in practical applications, the theory of pdDfs is only developed in the case of pdBfs (partially defined Boolean functions). Here we discuss monotone pdDfs, i.e. functions that are monotone on D. If the function $\hat{f} : X \mapsto Y$, agrees with f on D: $\hat{f}(x) = f(x),\ x \in D$, then $\hat{f}$ is called an *extension* of the pdDf f. The collection of all extensions forms a lattice: for, if f_1 and f_2 are extensions of the pdDf f, then $f_1 \wedge f_2$ and $f_1 \vee f_2$ are also extensions of f. The same holds for the set of all monotone extensions. The lattice of all *monotone* extensions of a pdDf f will be denoted here by $\mathcal{E}(f)$. It is easy to see that $\mathcal{E}(f)$ is universally bounded: it has a greatest and a smallest element. The maxlist of the maximal element called the *maximal monotone extension* can be directly obtained from the decision table.

Definition 1 *Let f be a monotone pdDf. Then the functions $f_{\min}$ and $f_{\max}$ are defined as follows:*

$$f_{\min}(x) = \begin{cases} \max\{f(y) :\ y \in D \cap {\downarrow}x\} & \textit{if } x \in {\uparrow}D \\ 0 & \textit{otherwise} \end{cases} \tag{15}$$

$$f_{\max}(x) = \begin{cases} \min\{f(y) :\ y \in D \cap {\uparrow}x\} & \textit{if } x \in {\downarrow}D \\ m & \textit{otherwise .} \end{cases} \tag{16}$$

Lemma 1 *Let f be a monotone pdDf. Then*

a) $f_{\min},\ f_{\max}\ \in \mathcal{E}(f)$.
b) $\forall \hat{f} \in \mathcal{E}(f):\quad f_{\min} \leq \hat{f} \leq f_{\max}$.

Since $\mathcal{E}(f)$ is a distributive lattice, the minimal and maximal monotone extension of f can also be described by the following expressions:

$$f_{\max} = \bigvee\{\ \hat{f} \mid \hat{f}\ \in\ \mathcal{E}(f)\} \text{ and } f_{\min} = \bigwedge\{\ \hat{f} \mid \hat{f}\ \in\ \mathcal{E}(f)\}\ . \tag{17}$$

Notation: Let $T_j(f) := \{x \in D : f(x) = j\}$. A minimal vector v of class j is a vector such that $f(v) = j$ and no vector strictly smaller than v is also in $T_j(f)$. Similarly, a maximal vector w is a vector maximal in $T_j(f)$, where $j = f(w)$. The sets of minimal and maximal vectors of class j are denoted by $minT_j(f)$ and $maxT_j(f)$ respectively.

According to the previous lemma $f_{\min}$ and $f_{\max}$ are respectively the minimal and maximal monotone extension of f. Decision lists of these extensions can be directly constructed from f as follows. Let $D_j := D \cap T_j(f)$, then $\min T_j(f_{\min}) = \min D_j$ and $\max T_j(f_{\max}) = \max D_j$.

Example 4. Consider the pdDf given by table 1, then its maximal extension is:

$$\begin{aligned} f(x) = \ & \text{if } x \leq 010 \text{ then } 0 \\ & \quad \text{else if } x \leq 100 \text{ then } 1 \\ & \qquad \text{else } 2\ . \end{aligned}$$

As described in the last subsection, from this maxlist representation we can deduce directly the minlist representation of the dual of f and finally by dualization we find that f is:

$$f = 2.(x_{12} \vee x_{11}x_{21} \vee x_{22} \vee x_{31}) \vee 1.x_{11}\ . \tag{18}$$

However, f can be viewed as a representation of table 4! This suggests a close relationship between minimal monotone decision rules and the maximal monotone extension f_{max}. This relationship is discussed in the next section.The relationship with the methodology LAD (Logical Analysis of Data) is briefly discussed in subsection 3.5.

4.3 The relationship between monotone decision rules and f_{max}

We first redefine the concept of a monotone reduct in terms of discrete functions. Let $X = X_1 \times X_2 \times \ldots \times X_n$ be the input space, and let $A = [1, \ldots, n]$ denote the set of attributes. Then for $U \subseteq A$, $x \in X$ we define the set $U.x$ respectively the vector $x.U$ by:

$$U.x = \{i \in U : x_i > 0\} \tag{19}$$

and

$$(x.U)_i = \begin{cases} x_i & \text{if } i \in U \\ 0 & \text{if } i \notin U\ . \end{cases} \tag{20}$$

Furthermore, the characteristic set U of x is defined by $U = A.x$.

Definition 2 *Suppose $f : D \to Y$ is a monotone pdDf, $w \in D$ and $f(w) = j$. Then $V \subseteq A$ is a monotone w-reduct iff $\forall x \in D :\ (f(x) < j \Rightarrow w.U \not\leq x.U)$.*

Note, that in this definition the condition $w.U \not\leq x.U$ is equivalent to $w.U \not\leq x$. The following lemma is a direct consequence of this definition.

Lemma 2 *Suppose f is a monotone pdDf, $w \in T_j(f)$. Then $V \subseteq A$ is a monotone w-reduct $\Leftrightarrow \forall x(f(x) < j \Rightarrow \exists i \in V$ such that $w_i > x_i)$.*

Corollary 1 *V is a monotone w-reduct iff $V.w$ is a monotone w-reduct. Therefore, w.l.o.g. we may assume that V is a subset of the characteristic set W of w: $V \subseteq W$.*

Monotone Boolean functions

We first consider the case that the dataset is Boolean: so the objects are described by condition and decision attributes taking one of two possible values $\{0,1\}$. The dataset represents a *partially defined Boolean function* (pdBf) $f : D \to \{0,1\}$ where $D \subseteq \{0,1\}^n$. As we have only two classes, we define the set of true vectors of f by $T(f) := T_1(f)$ and the set of false vectors of f by $F(f) := T_0(f)$.

Notation: In the Boolean case we will make no distinction between a set V and its characteristic vector v.

Lemma 3 *Let $f : D \to \{0,1\}$ be a monotone pdBf, $w \in D$, $w \in T(f)$. Suppose $v \leq w$. Then v is a w-reduct $\Leftrightarrow v \in T(f_{max})$.*

Proof: Since $v \leq w$, we have
v is a w-reduct $\Leftrightarrow \forall x(x \in D \cap F(f) \Rightarrow v \not\leq x) \Leftrightarrow v \in T(f_{max})$.

Theorem 1 *Suppose $f : D \to \{0,1\}$ is a monotone pdBf, $w \in D$, $w \in T(f)$. Then, for $v \leq w$, $v \in minT(f_{max}) \Leftrightarrow v$ is a minimal monotone w-reduct.*

Proof: Let $v \in minT(f_{max})$ and $v \leq w$ for some $w \in D$. Then v is a monotone w-reduct. Suppose $\exists u < v$ and u is a monotone w-reduct. Then by definition 2 we have: $u \in T(f_{max})$, which contradicts the assumption that $v \in minT(f_{max})$.

Conversely, let v be a minimal monotone w-reduct. Then by lemma 3 we have: $v \in T(f_{max})$. Suppose $\exists u < v : u \in T(f_{max})$. However, $v \leq w \Rightarrow u < w \Rightarrow U$ is a monotone w-reduct, which contradicts the assumption that v is a minimal w-reduct.

The results imply that the irredundant (monotone) decision rules that correspond to the object reducts are just the prime implicants of the maximal extension.

Corollary 2 *The decision rules obtained in rough set theory can be obtained by the following procedure: a) find the maximal vectors of class 1 (positive examples) b) determine the minimal vectors of the dual of the maximal extension and c) compute the minimal vectors of this extension by dualization. The complexity of this procedure is the same as for the dualization problem.*

Although the above corollary is formulated for monotone Boolean functions, results in [10] indicate that a similar statement holds for Boolean functions in general.

Monotone discrete functions

Lemma 4 *Suppose f is a monotone pdDf, $w \in T_j(f)$ and $v \leq w$. If $v \in T_j(f_{max})$ then the characteristic set V of v is a monotone w-reduct.*

Proof: $f_{max}(v) = j$ implies $\forall x(f(x) < j \Rightarrow v \not\leq x)$. Since $w \geq v$ we therefore have $\forall x(f(x) < j \Rightarrow \exists i \in V$ such that $w_i \geq v_j > x_i)$.

Remark: Even if in lemma 4 the vector v is minimal: $v \in minT_j(f_{max})$, then still $V = A.v$ is not necessarily a minimal monotone w-reduct.

Theorem 2 *Suppose f is a monotone pdDf and $w \in T_j(f)$. Then $V \subseteq A$ is a monotone w-reduct $\Leftrightarrow f_{max}(w.V) = j$.*

Proof: If V is a monotone w-reduct, then by definition $\forall x(f(x) < j \Rightarrow w.V \not\leq x)$. Since $w.V \leq w$ and $f(w) = j$ we therefore have $f_{max}(w.V) = j$.

Conversely, let $f_{max}(w.V) = j$, $V \subseteq A$. Then, since $w.V \leq w$ and the characteristic set of $w.V$ is equal to V, lemma 4 implies that V is a monotone w-reduct.

Theorem 3 *Let f be a monotone pdDf and $w \in T_j(f)$. If $V \subseteq A$ is a minimal monotone w-reduct, then $\exists u \in minT_j(f_{max})$ such that $V = A.u$.*

Proof: Since V is a monotone w-reduct, theorem 2 implies that $f_{max}(w.V) = j$. Therefore, $\exists u \in minT_j(f_{max})$ such that $u \leq w.V$. Since $A.u \subseteq V$ and $A.u$ is a monotone w-reduct (by lemma 4), the minimality of V implies $A.u = V$.

Theorem 3 implies that the minimal decision rules obtained by monotone w-reducts are not shorter than the minimal vectors (prime implicants) of f_{max}. This suggests that we can optimize a minimal decision rule by minimizing the attribute values to the attribute values of a minimal vector of f_{max}. For example, if V is a minimal monotone w-reduct and $u \in minT_j(f_{max})$ such that $u \leq w.V$ then the rule: 'if $x_i \geq w_i$ then j', where $i \in V$ can be improved by using the rule: 'if $x_i \geq u_i$ then j', where $i \in V$. Since $u_i \leq w_i$, $i \in V$, the second rule is applicable to a larger part of the input space X.

The results so far indicate the close relationship between minimal monotone decision rules obtained by the rough sets approach and by the approach using f_{max}. To complete the picture we make the following observations:

Observation 1: The minimal vector u (theorem 3) is not unique.

Observation 2: Lemma 4 implies that the length of a decision rule induced by a minimal vector $v \leq w$, $v \in minT_j(f_{max})$ is not necessarily smaller than that of a rule induced by a minimal w-reduct. This means that there may exist an $x \in X$ that is covered by the rule induced by v but not by the decision rules induced by the minimal reducts of a vector $w \in D$.

Observation 3: There may be minimal vectors of f_{max} such that $\forall w \in D$ $v \not\leq w$. In this case if $x \geq v$ then $f_{max}(x) = m$ but x is not covered by a minimal decision rule induced by a minimal reduct.

In the next two subsections we briefly compare the rough set approach and the discrete function approach with two other methods.

4.4 Monotone Decision Trees

Ordinal classification using decision trees is discussed in [3, 6, 21]. A decision tree is called monotone if it represents a monotone function. A number of algorithms are available for generating and testing the monotonicity of the tree [6, 21]. Here we demonstrate the idea with an example.

Example 5. A monotone decision tree corresponding to the pdDf given by table 1 and example 3 is represented in figure 1.

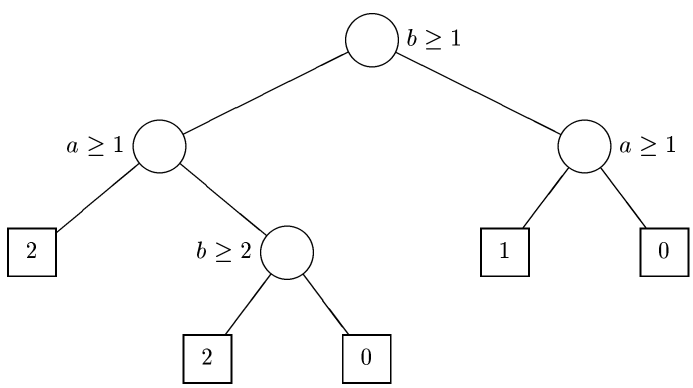

Fig. 1. Monotone decision tree representation of f

It can be seen that the tree contains information both on the corresponding extension and its complement (or equivalently its dual). Therefore the decision list representation tends to be more compact since we only need the information about the extension - the dual can always be derived if necessary.

4.5 Rough Sets and Logical Analysis of Data

The Logical Analysis of Data methodology (LAD) was presented in [10] and further developed in [9, 7, 8]. LAD is designed for the discovery of structural information in datasets. Originally it was developed for the analysis of Boolean datasets using partially defined Boolean functions. An extension of LAD for the analysis of numerical data is possible through the process of binarization. The building concepts are the supporting set, the pattern and the theory.

A set of variables (attributes) is called a *supporting set* for a partially defined Boolean function f if f has an extension depending only on these variables. A *pattern* is a conjunction of literals such that it is 0 for every negative example and 1 for at least one positive example. A subset of the set of patterns is used to form a *theory* - a disjunction of patterns that is consistent with all the available data and can predict the outcome of any new example. The theory is therefore an extension of the partially defined Boolean function.

Our research suggests that the LAD and the RS theories are similar in several aspects (for example, the supporting set corresponds to the reduct in the binary case and a pattern with the induced decision rule). The exact connections will be a subject of future research.

5 Experiments

5.1 The Bankruptcy Dataset

The dataset used in the experiments is discussed in [22, 13]. The sample consists of 39 objects denoted by $F1$ to $F39$ - firms that are described by 12 financial parameters. To each company a decision value is assigned - the expert evaluation of its category of risk for the year 1988. The condition attributes denoted by $A1$ to $A12$ take integer values from 0 to 4.

The decision attribute is denoted by d and takes integer values in the range 0 to 2 where: 0 means *unacceptable*, 1 means *uncertainty* and 2 means *acceptable*.

The data was first analyzed for monotonicity. The problem is obviously monotone (if one company outperforms another on all condition attributes then it should not have a lower value of the decision attribute). Nevertheless, one noisy example was discovered, namely $F24$. It was removed from the dataset and was not considered further.

5.2 Reducts and Decision Rules

The minimal reducts have been computed using our program 'the Dualizer'. There are 25 minimal general reducts (minimum length 3) and 15 monotone reducts (minimum length 4). We have also compared the heuristics to approximate a minimum reduct: the best reduct method (for general reducts) and the Johnson strategy (for general and monotone reducts).

Table 11 shows the two sets of decision rules obtained by computing the object (value)- reducts for the monotone reduct $(A1, A3, A7, A9)$. Both sets of rules have minimal covers, of which the ones with minimum length are shown in table 12. A minimum cover can be transformed into an extension if the rules are considered as minimal/maximal vectors in a decision list representation. In this sense the minimum cover of the first set of rules can be described by the following function:

$$f = 2.x_{73}x_{93} \vee 1.(x_{33} \vee x_{73} \vee x_{11}x_{93} \vee x_{32}x_{72}) \ . \tag{21}$$

The maximal extension corresponding to the monotone reduct $(A1, A3, A7, A9)$ is represented in table 13.

Table 11. The rules for $(A1, A3, A7, A9)$

class $d \geq 2$	class $d \geq 1$
$A1 \geq 3$	$A1 \geq 3$
$A7 \geq 4$	$A3 \geq 3$
$A9 \geq 4$	$A7 \geq 3$
$A1 \geq 2 \wedge A7 \geq 3$	$A9 \geq 4$
$A3 \geq 2 \wedge A7 \geq 3$	$A1 \geq 1 \wedge A3 \geq 2$
$A7 \geq 3 \wedge A9 \geq 3$	$A1 \geq 1 \wedge A9 \geq 3$
	$A3 \geq 2 \wedge A7 \geq 2$
	$A3 \geq 2 \wedge A7 \geq 1 \wedge A9 \geq 3$
class $d \leq 0$	**class $d \leq 1$**
$A7 \leq 0$	$A7 \leq 2$
$A9 \leq 1$	$A9 \leq 2$
$A1 \leq 0 \wedge A3 \leq 0$	
$A1 \leq 0 \wedge A3 \leq 2 \wedge A7 \leq 1$	
$A1 \leq 0 \wedge A3 \leq 1 \wedge A7 \leq 2$	
$A1 \leq 0 \wedge A3 \leq 2 \wedge A9 \leq 2$	
$A3 \leq 0 \wedge A9 \leq 2$	
$A3 \leq 1 \wedge A7 \leq 2 \wedge A9 \leq 2$	
$A3 \leq 2 \wedge A7 \leq 1 \wedge A9 \leq 2$	

Table 12. The minimum covers for $(A1, A3, A7, A9)$

class $d \geq 2$	class $d \geq 1$
$A7 \geq 3 \wedge A9 \geq 3$	$A3 \geq 3$
	$A7 \geq 3$
	$A1 \geq 1 \wedge A9 \geq 3$
	$A3 \geq 2 \wedge A7 \geq 2$
class $d \leq 0$	**class $d \leq 1$**
$A1 \leq 0 \wedge A3 \leq 2 \wedge A7 \leq 1$	$A7 \leq 2$
$A1 \leq 0 \wedge A3 \leq 1 \wedge A7 \leq 2$	$A9 \leq 2$
$A3 \leq 1 \wedge A7 \leq 2 \wedge A9 \leq 2$	

Table 13. The maximal extension for $(A1, A3, A7, A9)$

class $d = 2$	class $d = 1$
$A1 \geq 3$	$A3 \geq 3$
$A3 \geq 4$	$A7 \geq 3$
$A7 \geq 4$	$A1 \geq 1 \wedge A3 \geq 2$
$A9 \geq 4$	$A1 \geq 1 \wedge A9 \geq 3$
$A1 \geq 2 \wedge A7 \geq 3$	$A3 \geq 2 \wedge A7 \geq 2$
$A3 \geq 2 \wedge A7 \geq 3$	$A3 \geq 2 \wedge A7 \geq 1 \wedge A9 \geq 3$
$A7 \geq 3 \wedge A9 \geq 3$	

The function f or equivalently its minlist we have found consists of only 5 decision rules (prime implicants). They cover the whole input space. Moreover, each possible vector is classified as $d = 0, 1$ or 2 and not as $d \geq 1$ or $d \geq 2$ like in [13]. The latter paper uses both the formats shown in table 11 to describe a minimum cover, resulting in a system of 11 rules. Using both formats at the same time can result in much (possibly exponential) larger sets of rules. Another difference between our approach and [13] is our use of the monotone discernibility matrix. Therefore, we can compute all the monotone reducts and not only a generalization of the 'best reduct' as in [13].

6 Discussion and Further Research

Our approach using the concepts of monotone discernibility matrix/function and monotone (object) reduct and using the theory of monotone discrete functions has a number of advantages summarized in the discussion on the experiment with the bankruptcy dataset in section 4. Furthermore , it appears that there is close relationship between the decision rules obtained using the rough set approach and the prime implicants of the maximal extension. Although this has been shown for the monotone case this also holds at least for non-monotone *Boolean* datasets. We have discussed how to compute this extension by using dualization. The relationship with two other possible approaches for ordinal classification is discussed in subsections 3.4 and 3.5. We also computed monotone decision trees [6, 21] for the datasets discussed in this paper. It appears that monotone decision trees are larger because they contain the information of both an extension and its dual! The generalization of the discrete function approach to non-monotone datasets and the comparison with the theory of rough sets is a topic of further research. Finally, the sometimes striking similarity we have found between Rough Set Theory and Logical Analysis of Data remains an interesting research topic.

References

1. Altman, E.I.: Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance 4 (1968) 589–609

2. Back, B., Laitinen, T., Sere, K., Van Wezel, M.: Choosing Bankruptcy Predictors Using Discriminant Analysis, Logit Analysis, and Genetic Algorithms. Turku Centre for Computer Science, Technical Report No 40 (1996)
3. Ben-David, A.: Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms. Machine Learning 19 (1995) 29–43
4. Bioch, J.C.: Dualization, Decision Lists and Identification of Monotone Discrete Functions. Annals of Mathematics and Artificial Intelligence 24 (1998) 69–91
5. Bioch, J.C., Ibaraki, T.: Complexity of Identification and Dualization of Positive Boolean Functions. Information and Computation 123 (1995) 50–63
6. Bioch, J.C., Potharst, R.: Decision Trees for Monotone Classification. In: K. van Marcke and W. Daelmans (eds), Proceedings of the Dutch Artificial Conference on Artificial Intelligence (NAIC'97), Antwerpen (1997) 361–369
7. Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A.: Logical Analysis of Numerical Data. RUTCOR Research Report RRR 04-97, RUTCOR, Rutgers University (1997)
8. Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Logical Analysis of Numerical Data. Mathematical Programming 79 (1997) 165–190
9. Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I.:An Implementation of Logical Analysis of Data. RUTCOR Research Report RRR 22-96, RUTCOR, Rutgers University (1996)
10. Crama, Y., Hammer, P.L., Ibaraki, T.: Cause-Effect Relationships and Partially Defined Boolean Functions. Annals of Operations Research 16 (1988) 299–326
11. Eiter, T., Gottlob, G.: Identifying the Minimal Transversals of a Hypergraph and Related Problems. SIAM Journal on Computing 24 (1995) 1278–1304
12. Fredman, M., Khachiyan, L.: On the Complexity of Dualization of Monotone Disjunctive Normal Forms. Journal of Algorithms 21 (1996) 618–628
13. Greco, S., Matarazzo, B., Slowinski, R.: A New Rough Set Approach to Evaluation of Bankruptcy Risk. In: C. Zopounidis (ed.), Operational Tools in the Management of Financial Risks, Kluwer, Dordrecht (1998) 121–136
14. Hu, X., Cercone, N.: Learning in Relational Databases: a Rough Set Approach. Computational Intelligence 11 (1995) 323–338
15. Johnson, D.S.: Approximation Algorithms for Combinatorial Problems. Journal of Computer and System Sciences 9 (1974) 256–278
16. Komorowski, J., Polkowski, L., Skowron, A.: Rough Sets: A Tutorial. http://www.esslli.let.uu.nl/Courses/skowron/skowron.ps
17. Leshno, M., Spector, Y.: Neural Network Prediction Analysis: The Bankruptcy Case. Neurocomputing 10 (1996) 125–147
18. Makino, K., Suda, T., Yano, K., Ibaraki, T.: Data Analysis by Positive Decision Trees. In: Proceedings International symposium on cooperative database systems for advanced applications (CODAS), Kyoto (1996) 282–289
19. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
20. Pompe, P.P.M., Feelders, A.J.: Using Machine Learning, Neural Networks, and Statistics to Predict Corporate Bankruptcy. Microcomputers in Civil Engineering 12 (1997) 267–276
21. Potharst, R., Bioch, J.C.: Decision Trees for Ordinal Classification. Intelligent Data Analysis 4 (2000) 1–15
22. Slowinski, R., Zopounidis, C.: Application of the Rough Set Approach to Evaluation of Bankruptcy Prediction. Intelligent Systems in Accounting, Finance and Management 4 (1995) 27–41

23. Sung, T.K., Chank, N., Lee, G.: Dynamics of Modelling in Data Mining: Interpretive Approach to Bankruptcy Prediction. Journal of Management Information Systems 16 (1999) 63–85

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2001

Bankruptcy Prediction with Rough Sets
Jan C. Bioch & Viara Popova
ERS-2001-11-LIS

2000

A Greedy Heuristic for a Three-Level Multi-Period Single-Sourcing Problem
H. Edwin Romeijn & Dolores Romero Morales
ERS-2000-04-LIS

Integer Constraints for Train Series Connections
Rob A. Zuidwijk & Leo G. Kroon
ERS-2000-05-LIS

Competitive Exception Learning Using Fuzzy Frequency Distribution
W-M. van den Bergh & J. van den Berg
ERS-2000-06-LIS

Models and Algorithms for Integration of Vehicle and Crew Scheduling
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-14-LIS

Managing Knowledge in a Distributed Decision Making Context: The Way Forward for Decision Support Systems
Sajda Qureshi & Vlatka Hlupic
ERS-2000-16-LIS

Adaptiveness in Virtual Teams: Organisational Challenges and Research Direction
Sajda Qureshi & Doug Vogel
ERS-2000-20-LIS

Assessment of Sustainable Development: a Novel Approach using Fuzzy Set Theory
A.M.G. Cornelissen, J. van den Berg, W.J. Koops, M. Grossman & H.M.J. Udo
ERS-2000-23-LIS

Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-31-LIS

An NPV and AC analysis of a stochastic inventory system with joint manufacturing and remanufacturing
Erwin van der Laan
ERS-2000-38-LIS

Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms
Shan-Hwei Nienhuys-Cheng, Wim Van Laer, Jan Ramon & Luc De Raedt

---

* A complete overview of the ERIM Report Series Research in Management:
http://www.ers.erim.nl

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Decision Making in Marketing Management
F&A Financial Decision Making and Accounting
STR Strategic Renewal and the Dynamics of Firms, Networks and Industries

ERS-2000-39-LIS

Classification and Target Group Selection bases upon Frequent Patterns
Wim Pijls & Rob Potharst
ERS-2000-40-LIS

Average Costs versus Net Present Value: a Comparison for Multi-Source Inventory Models
Erwin van der Laan & Ruud Teunter
ERS-2000-47-LIS

Fuzzy Modeling of Client Preference in Data-Rich Marketing Environments
Magne Setnes & Uzay Kaymak
ERS-2000-49-LIS

Extended Fuzzy Clustering Algorithms
Uzay Kaymak & Magne Setnes
ERS-2000-51-LIS

Mining frequent itemsets in memory-resident databases
Wim Pijls & Jan C. Bioch
ERS-2000-53-LIS

Crew Scheduling for Netherlands Railways. "Destination: Curstomer"
Leo Kroon & Matteo Fischetti
ERS-2000-56-LIS