Electronic Notes in Theoretical Computer Science 82 No. 4 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume82.html 8 pages

On the Decision Table with Maximal Number of Reducts

Hung Son Nguyen [2]

Institute of Mathematics
Warsaw University
Warsaw, Poland

Abstract

Searching for reducts is a basic problem for many rough set methods like rule induction, classification, etc.. Many of them can not be realized in exact way because of existing possibly exponential number of (relative) reducts in decision tables. In this paper we investigate properties of the most malicious decision tables, i.e., tables with maximal number of reducts. We show that in such systems, the number of objects must be also exponential. The presented method is based on Boolean reasoning approach.

1 Introduction

Rough set theory offers a large collection of tools for knowledge discovery from data. Many of those methods, like decision rule induction, classifier construction, discretization, decision tree construction, representative association rule induction, etc., are based on computing the most relevant sets of attributes called reducts.

Research on reduct calculation is one of the fundamental investigations in rough set theory. It is well known that problem of searching for minimal reduct is NP-hard. Moreover, the potential number of all reducts existing in a given decision table, consisting with k attributes, is equal to $N(k) = \binom{k}{k/2}$. These facts cause the high computational complexity of all reduct-based rough set methods. They also explain the reason, for which we use in applications heuristics instead of exact algorithm.

[1] This work was supported by KBN grant 8T11C02519
[2] Email: son@mimuw.edu.pl

In this paper we investigate the structure of the such decision tables containing maximal number, i.e. $N(k)$, of reducts, where k is a number of attributes. We present a lower bound of the minimal number of objects consisting in such decision tables. Our evaluation means that we do not often meat such tables in practice, because they must contain as least exponential number of objects (with respect to number of attributes).

2 Basic notions

2.1 Rough set preliminaries

An *information system* is a pair $\mathbb{S} = (U, A)$, where U is a non-empty, finite set, called the *universe* and A is a non-empty, finite set, of *attributes*. Each $a \in A$ corresponds to the function $a : U \to V_a$, where V_a is called the *value set* of a. Elements of U are called *situations*, *objects* or *rows*, interpreted as, e.g., cases, states, patients, observations.

In the paper we also consider a special case of information systems called *decision tables*. In a decision table $\mathbb{S} = (U, A \cup \{d\})$, $d \notin A$ is a distinguished attribute called the *decision*. The elements of A are called *conditional attributes (conditions)*.

With any subset of attributes $B \subseteq A$, we associate the *information vector* for any object $x \in U$ by

$$inf_B(x) = \{(a, a(x)) : a \in B\}$$

An equivalence relation called the *B-indiscernibility relation* [3], denoted by $IND(B)$, is defined by

$$IND(B) = \{(x, y) \in U \times U : inf_B(x) = inf_B(y)\}$$

Objects x, y satisfying relation $IND(B)$ are indiscernible by attributes from B. By $[x]_{IND(B)}$ we denote the equivalence class of $IND(B)$ defined by x. A minimal subset B of A (with regard to inclusion) such that $IND(A) = IND(B)$ is called a *reduct* of $\mathbb{S}$.

If $\mathbb{S} = (U, A)$ is an information system, $B \subseteq A$ is a set of attributes and $X \subseteq U$ is a set of objects, then the sets

$$\underline{B}X = \left\{x \in U : [x]_{IND(B)} \subseteq X\right\} \quad \overline{B}X = \left\{x \in U : [x]_{IND(B)} \cap X \neq \emptyset\right\}$$

are called the *B-lower* and the *B-upper approximation* of X in $\mathbb{S}$, respectively.

If $\mathbb{S} = (U, A \cup \{d\})$ is a decision table and $B \subseteq A$ then we define a function $\partial_B : U \to 2^{\{1,..,r(d)\}}$, called the *generalized decision in* $\mathbb{S}$, by

$$\partial_B(x) = \{i : \exists_{x' \in U} \left[(x' IND(B) x) \wedge (d(x') = i)\right]\} = d\left([x]_{IND(B)}\right)$$

A decision table $\mathbb{S}$ is called *consistent (deterministic)* if $card\left(\partial_A(x)\right) = 1$ for any $x \in U$, otherwise $\mathbb{S}$ is *inconsistent (non-deterministic)*.

The set of attributes $B \subseteq A$ is called a "*relative reduct*" (or simply a *reduct*) of decision table $\mathbb{S}$ if and only if

(i) $\partial_B(x) = \partial_A(x)$ for all object $x \in U$.

(ii) any proper subset of B does not satisfy the previous condition.

i.e., B is a minimal subset (with respect to the inclusion relation $\subseteq$) of the attribute set satisfying the property $\forall_{x \in U} \partial_B(x) = \partial_A(x)$.

There are two problems related to the notion of "reduct", which have been intensively explored in rough set theory by many researchers (see e.g. [1,7,2]). The first problem is related to searching for "shortest reducts" (i.e. reducts with the minimal cardinality). The second problem is related to searching for all reducts. It has been shown that the first problem is NP-hard (see [5]) and second is at least NP-hard. Some heuristics have been proposed for those problems. Here we present the approach based on Boolean reasoning as proposed in [5].

2.2 Boolean reasoning approach

By Boolean function we denote any function $f : \{0,1\}^n \to \{0,1\}$. Boolean functions can be described by boolean formulas, i.e. expressions constructed by variables from a set $VAR = \{x_1, ..., x_k\}$, and boolean operators like conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). Let us remind some special types of boolean formulas:

- *Literal* is a simplest formula, which is either variable or negation of a variable. If $VAR = \{x_1, ..., x_k\}$ is a set of k variables, then we have $2k$ literals: $x_1, \neg x_1, ..., x_k, \neg x_k$.
- *Term* (or monomial) is a conjunction of some literals. We denote by $\mathbf{T}_X$ the following term:
$$\mathbf{T}_X = \bigwedge_{l \in X} l$$
where X is a set of literals. For example $\mathbf{T}_{\{x_1, \neg x_3, x_4\}} = x_1 \wedge \neg x_3 \wedge x_4$.
- *Clause* is a disjunction of literals. We denote by $\mathbf{C}_X$ the following clause:
$$\mathbf{C}_X = \bigvee_{l \in X} l$$
where X is a set of literals. For example $\mathbf{C}_{\{x_1, \neg x_3, x_4\}} = x_1 \vee \neg x_3 \vee x_4$.
- *CNF*[3] is a conjunction of some clauses.
- *DNF*[4] is a disjunction of some terms.

For any boolean expression ϕ, we denote by $VAR(\phi)$ the set of boolean variables occurring in the formula ϕ, and by $LIT(\phi)$ the set of literals occurring in ϕ. By those notations we have the following equalities:

$$\mathbf{T} = \mathbf{T}_{LIT(\mathbf{T})} \qquad \text{and} \qquad \mathbf{C} = \mathbf{C}_{LIT(\mathbf{C})}$$

for any term $\mathbf{T}$ and clause $\mathbf{C}$. The boolean function f is called "*monotone*" if

$$\forall_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} (\mathbf{x} \leqslant \mathbf{y}) \Rightarrow (f(\mathbf{x}) \leqslant f(\mathbf{y}))$$

[3] Conjunctive Normal Form
[4] Disjunctive Normal Form

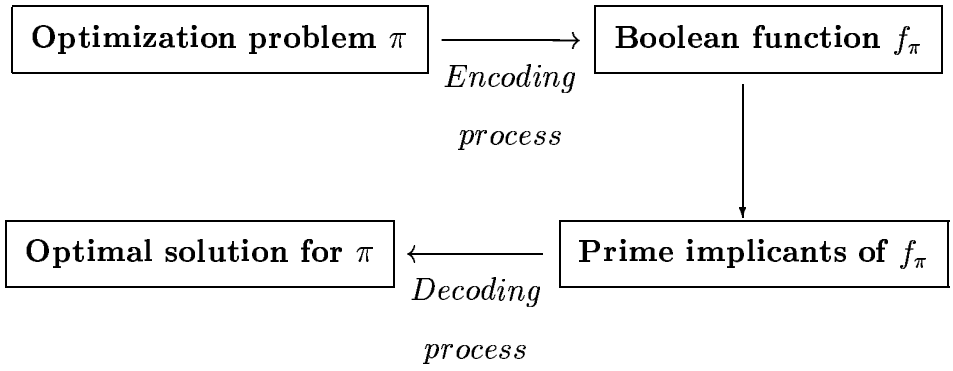

Fig. 1. The Boolean reasoning scheme for solving optimization problems.

It has been shown that monotone functions can be represented by boolean expression without negations [6].

The term $\mathbf{T}$ is called *implicant* of a boolean function f if $\mathbf{T}(\mathbf{x}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in \{0,1\}^n$. The implicant $\mathbf{T}$ is called *prime implicant* of f if for any $X \subsetneqq VAR(\mathbf{T})$, the term $\mathbf{T}_X$ is not implicant of f.

The same boolean function can be presented by many boolean formulas, particularly in both CNF and DNF forms. All prime implicants of monotone functions can be found by its transformation from CNF to DNF [6]. Unfortunately, many optimization and searching problems related to prime implicants are NP-complete, even for monotone functions.

The boolean reasoning method is based on encoding the investigated optimization problem π by a corresponding Boolean function f_π in such a way that any prime implicant of f_π states a solution of π (see Figure 1).

Many problems in rough set theory (e.g. reduct finding, rule extraction, discretization [2]) has been successively solved by Boolean reasoning approach.

2.3 Boolean reasoning approach for reduct problem

We illustrate this approach for the reduct problem. Given a decision table $\mathbb{S} = (U, A \cup \{d\})$, where $U = \{u_1, u_2, ..., u_n\}$, $A = \{a_1, ..., a_k\}$, by discernibility matrix of the decision table $\mathbb{S}$ we mean the $(n \times n)$ matrix

$$\mathbf{M}(\mathbb{S}) = [C_{i,j}]_{ij=1}^{n}$$

such that $C_{i,j}$ is the set of attributes discerning u_i and u_j. Formally:

$$C_{i,j} = \begin{cases} \{a_m \in A : a_m(x_i) \neq a_m(x_j)\} & \text{if } d(x_i) \neq d(x_j) \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $x_1, ..., x_n$ be boolean variables corresponding to attributes $a_1, ..., a_k$, and let $X_{i,j} = \{x_m : a_m \in C_{i,j}\}$. One can define the *discernibility function* $f_{\mathbb{S}}$ (as a Boolean function in CNF) as follows:

$$f_{\mathbb{S}}(x_1, ..., x_k) = \bigwedge_{i,j} (\mathbf{C}_{X_{i,j}})$$

One can show that prime implicants of $f_{\mathbb{S}}(a_1^*, ..., a_k^*)$ correspond exactly to reducts in $\mathbb{S}$.

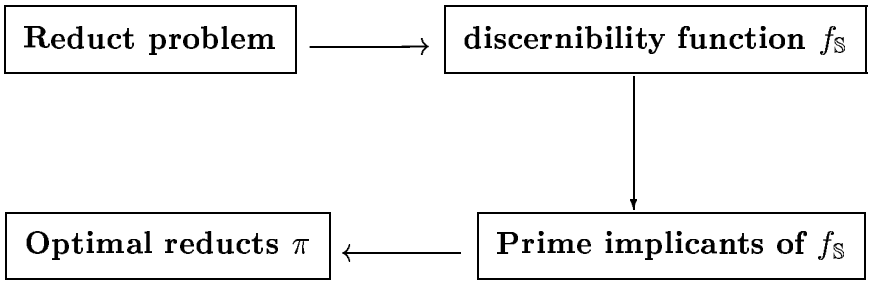

Fig. 2. The Boolean reasoning scheme for solving reduct problem

3 Malicious decision table

In this section we characterize the structure decision table with maximal number of reducts.

Let $\mathbb{S} = (U, A \cup \{d\})$ be an arbitrary decision table containing:

- k attributes, i.e. $A = \{a_1, ..., a_k\}$;
- n objects, i.e. $U = \{u_1, ..., u_n\}$;

and let $\mathbf{M}(\mathbb{S}) = [C_{i,j}]_{ij=1}^{n}$ be the discernibility matrix of $\mathbb{S}$.

We denote by $RED(\mathbb{S})$ the set of all relative reducts in the decision table $\mathbb{S}$. Let us remind some properties of the set $RED(\mathbb{S})$:

Fact 3.1 *If $B_1 \in RED(\mathbb{S})$ is a reduct of the system $\mathbb{S}$, then there is no such reduct $B_2 \in RED(\mathbb{S})$ that $B_1 \subset B_2$.*

Fact 3.2 *The elements of $RED(\mathbb{S})$ create an antichain with respect to the inclusion between subsets of A. Moreover, if $|A| = k = 2k'$ is an even positive integer, then*

$$\mathbf{C} = \{B \subset A : |B| = k'\}$$

is the only antichain, that contains maximal number of subsets of A. If $|A| = k = 2k' + 1$ is an odd positive integer, then we have two antichains containing maximal number of subsets:

$$\mathbf{C}_1 = \{B \subset A : |B| = k'\}; \quad \mathbf{C}_2 = \{B \subset A : |B| = k' + 1\}$$

It follows that:

Fact 3.3 *The maximal number of reducts for a given decision table $\mathbb{S}$ is less or equal to*

$$N(k) = \begin{pmatrix} k \\ k/2 \end{pmatrix}.$$

Definition 3.4 The decision table $\mathbb{S}$ is called *malicious* if it contains $N(k)$ reducts (see Fact 3.3)

Let

$$f_{\mathbb{S}} = \mathbf{C}_1 \wedge \mathbf{C}_2 \wedge \ldots \wedge \mathbf{C}_M$$

be the discernibility function of decision table $\mathbb{S}$, where $\mathbf{C}_1, ..., \mathbf{C}_M$ are clauses defined on boolean variables from $VAR = \{x_1, ..., x_k\}$ corresponding to attributes $a_1, ..., a_k$ (see Section 2.3).

From Fact 3.2 we have the following proposition:

Proposition 3.5 *The decision table $\mathbb{S}$ is malicious if and only if the discernibility function $f_{\mathbb{S}}$ has $N(k)$ prime implicants. In particular,*

- *if k is even, then $f_{\mathbb{S}}$ can be transformed to the form:*

$$f^* = \bigvee_{X \subset VAR: |X| = k/2} \mathbf{T}_X$$

- *if k is odd, then $f_{\mathbb{S}}$ can be transformed to one of the forms:*

$$f_1^* = \bigvee_{X \subset VAR: |X| = (k-1)/2} \mathbf{T}_X$$

or

$$f_2^* = \bigvee_{X \subset VAR: |X| = (k+1)/2} \mathbf{T}_X$$

From Proposition 3.5 we have:

Proposition 3.6 *For any integer $k > 0$, there exists a malicious decision table with k attributes.*

We are going to prove the main thesis of this paper:

Proposition 3.7 *If decision table $\mathbb{S}$ is malicious, then its discernibility function function $f_{\mathbb{S}}$ must consist at least $\Omega(N(k))$ clauses.*

Proof. Let us assume that

$$f_{\mathbb{S}} = \mathbf{C}_1 \wedge \mathbf{C}_2 \wedge \ldots \wedge \mathbf{C}_M$$

is an irreducible form of discernibility function $f_{\mathbb{S}}$, i.e. $VAR(\mathbf{C}_i) \subsetneq VAR(\mathbf{C}_j)$ for any $i, j \in \{1, ..., M\}$. We will prove the following facts:

FACT 1: The term $\mathbf{T}_X$ is an implicant of $f_{\mathbb{S}}$ if and only if $X \cap VAR(\mathbf{C}_m) \neq \emptyset$ for any $m \in \{1, ..., M\}$.

Proof. ($\Rightarrow$) Let us assume that $X \cap VAR(\mathbf{C}_m) = \emptyset$ for some $m \in \{1, ..., M\}$. Let us define a vector $\mathbf{v}_X = (v_1, ..., v_k) \in \{0, 1\}^n$ by: $v_i = 1 \Leftrightarrow x_i \in X$. We have $\mathbf{T}_X(\mathbf{v}_X) = 1$ and $C_m(\mathbf{v}_X) = 0$, hence $f_{\mathbb{S}}(\mathbf{v}_X) = 0$, i.e. $\mathbf{T}_X$ is not implicant of $f_{\mathbb{S}}$. The second implication can be proved in a similar way.

FACT 2: If k is an even integer, then $|VAR(\mathbf{C}_m)| > k/2$ for any $m \in \{1, ..., M\}$.

Proof. Let us assume that $|VAR(\mathbf{C}_m)| \leqslant k/2$ for some $m \in \{1, ..., M\}$, we will show that $f_{\mathbb{S}} \neq f^*$ (what is contradictory to the Proposition 3.5). In fact, because $|VAR \setminus VAR(\mathbf{C}_m)| \geqslant k/2$ then there exists a set of variables

$X \subset VAR \setminus VAR(\mathbf{C}_m)$ such that $|X| = k/2$. Because $X \cap VAR(\mathbf{C}_m) = \emptyset$, hence $\mathbf{T}_X$ is not implicant of $f_{\mathbb{S}}$ (see **F1**), and $f_{\mathbb{S}} \neq f^*$.

FACT 3: If k is an even integer, then for any set of variables $X \subset VAR$ such that $|X| = k/2+1$, there exists $m \in \{1, ..., M\}$ such that $VAR(\mathbf{C}_m) = X$.
Proof. Suppose there exists such X that $|X| = k/2+1$ and $X \neq VAR(\mathbf{C}_m)$ for any $m \in \{1, ..., M\}$. Let $Y = VAR \setminus X$, we have $|Y| = k/2-1$. Moreover, for any $m \in \{1, ..., M\}$, we have

$$Y = VAR \setminus X \neq VAR \setminus VAR(\mathbf{C}_m)$$

From **FACT 2** we have $|VAR(\mathbf{C}_m)| > k/2$, hence $|Y| + |VAR(\mathbf{C}_m)| \geq k$. Because $|Y| + |VAR(\mathbf{C}_m)| \geq k$ and $Y \neq VAR \setminus VAR(\mathbf{C}_m)$, hence we have

$$Y \cap VAR(\mathbf{C}_m) \neq \emptyset$$

Therefore $\mathbf{T}_Y$ is implicant of $f_{\mathbb{S}}$, what contradicts Proposition 3.5.

FACT 4: If k is an odd integer and $f_{\mathbb{S}}$ is transformable to f_1^*, then for any subset of variables $X \subset VAR$ such that $|X| = (k-1)/2 + 2$, there exists $m \in \{1, ..., M\}$ such that $VAR(\mathbf{C}_m) = X$;

FACT 5: If k is an odd integer and $f_{\mathbb{S}}$ is transformable to f_2^*, then for any subset of variables $X \subset VAR$ such that $|X| = (k-1)/2 + 1$, there exists $m \in \{1, ..., M\}$ such that $VAR(\mathbf{C}_m) = X$;

The proofs of **FACT 4** and **FACT 5** are analogical to the proof of **FACT 3**. From **FACT 3**, **FACT 4** and **FACT 5** we have:

$$M \geqslant \begin{cases} \binom{k}{k/2+1} = \frac{k}{k+2}N(k) & \text{if } f_{\mathbb{S}} \text{ is transformable to } f^* \\ \binom{k}{(k+1)/2+1} = \frac{k-1}{k+3}N(k) & \text{if } f_{\mathbb{S}} \text{ is transformable to } f_1^* \\ \binom{k}{(k+1)/2} = N(k) & \text{if } f_{\mathbb{S}} \text{ is transformable to } f_2^* \end{cases}$$

Therefore $M \geqslant \Omega(N(k))$ in every cases. □

From Proposition 3.7 we obtain:

Corollary 3.8 *If a decision table* $\mathbb{S}$ *is malicious, then it consists at least* $\Omega(\sqrt{N(k)})$ *objects.*

Proof. Let n be the number of objects in $\mathbb{S}$, we have $n \cdot (n-1)/2 \geqslant M$. From Proposition 3.7 we have $M \geqslant \Omega(N(k))$, therefore $n \geqslant \Omega(\sqrt{N(k)})$. □

4 Conclusion

We have presented an application of boolean reasoning approach to analyzing structure of malicious decision tables. The results is showing that the decision

tables, with maximal number of reducts, must also contain exponential number of objects. This means that it is not easy to meet such decision tables in practice.

In the next papers, we plan to apply the presented method to design randomized algorithms for solving the reduct problems.

References

[1] Bazan J.: A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables. In: L. Polkowski and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery* **1**, Physica-Verlag, Heidelberg, 1998, pp. 321–365.

[2] Nguyen H.Son, Skowron A.: Boolean reasoning for feature extraction problems. In: Z.W. Raś and A.Skowron (Eds.): Proceedings of Tenth International Symposium on Foundation of Intelligent Systems, ISMIS'97, NC, USA, *Foundation of Intelligent Systems* LNAI **1325**, Springer Verlag, 1997, pp. 117-126.

[3] Pawlak Z., Rough sets: Theoretical aspects of reasoning about data, Kluwer Dordrecht, 1991.

[4] Polkowski L., Skowron A. (eds.), Rough Sets in Knowledge Discovery vol. 1–2, Physica-Verlag, Heidelberg, 1998

[5] Skowron A., Rauszer C.: The discernibility matrices and functions in information systems. In: Słowiński R. (ed.). Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 311–362

[6] Wegener I.: *The complexity of boolean functions.* Wiley, Stuttgart, 1987.

[7] Wróblewski J.: Covering with reducts - a fast algorithm for rule generation. In L. Polkowski and A. Skowron (Eds.): Proc. of RSCTC'98, Warsaw, Poland, 1998. Springer-Verlag, Berlin Heidelberg, pp. 402–407.