BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **14**, No 4

Sofia • 2014

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2014-0001

On Finding All Reducts of Consistent Decision Tables

Demetrovics Janos[1], *Vu Duc Thi*[2], *Nguyen Long Giang*[3]

[1] *Institute for Computer and Control* (*SZTAKI*), *Hungarian Academy of Sciences*
[2] *Institute of Information Technology, VNU, Viet Nam*
[3] *Institute of Information Technology, VAST, Viet Nam*
Emails: *demetrovics@sztaki.mta.hu* *vdthi@vnu.edu.vn* *nlgiang@ioit.ac.vn*

Abstract: *The problem of finding reducts plays an important role in processing information on decision tables. The objective of the attribute reduction problem is to reject a redundant attribute in order to find a core attribute for data processing. The attribute reduction in decision tables is the process of finding a minimal subset of conditional attributes which preserve the classification ability of decision tables. In this paper we present the time complexity of the problem of finding all reducts of a consistent decision table. We prove that this time complexity is exponential with respect to the number of attributes of the decision tables. Our proof is performed in two steps. The first step is to show that there exists an exponential algorithm which finds all reducts. The other step is to prove that the time complexity of the problem of finding all reducts of a decision table is not less than exponential.*

Keywords: *Consistent decision table, attribute reduction, reduct, time complexity, relation, relation scheme, minimal key, antikey.*

1. Basic concepts

Let us give some necessary concepts that are used in this paper. The concepts and facts given in this section can be found in [1, 3].

Firstly, we summarize some basic concepts in rough set theory [3]. An Information System (IS) is defined as $\mathrm{IS} = (U,\ A,\ V,\ f)$ in which U is the finite and non-empty set of objects; A is the finite and non-empty set of attributes; $V = \bigcup_{a \in A} V_a$, where V_a is the value range of the attribute $a \in A$; $f : U \times A \to V_a$ is an

information function, where $\forall a \in A,\ u \in U,\ f(u,a) \in V_a$ hold.

For any $u \in U,\ a \in A$, we will denote the value of the attribute *a* on the object *u* by $a(u)$ instead of $f(u,a)$. If $B = \{b_1, b_2, ..., b_k\} \subseteq A$ is a subset of attributes, then the set of $b_i(u)$ is denoted as $B(u)$. Therefore, if *u, v* are two objects in *U*, then $B(u) = B(v)$ if and only if $b_i(u) = b_i(v)$ for any $i = 1, ..., k$.

Definition 1.1. A Decision table is an information System $\mathrm{DS} = (U, A, V, f)$ where $A = C \cup D,\ C$ is the set of condition attributes, D is the set of decision attributes and $C \cap D = \varnothing$.

Without loss of generality, suppose that *D* consists of only one decision attribute *d*. Therefore, from this time we consider the decision table $\mathrm{DS} = (U, C \cup \{d\}, V, f)$, where $\{d\} \notin C$. A decision table DS is consistent if and only if the functional dependency $C \to \{d\}$ is true; it means that for any $u,\ v \in U$ if $C(u) = C(v)$ then $d(u) = d(v)$. Conversely, DS is inconsistent**.**

Definition 1.2. Let $\mathrm{DS} = (U, C \cup \{d\}, V, f)$ be a consistent decision table. If $R \subseteq C$ satisfies the conditions below:

1) for any $u, v \in U$ if $R(u) = R(v)$ then $d(u) = d(v)$;

2) for any $B \subset R$, there exist $u, v \in U$ such that $B(u) = B(v)$ and $d(u) \neq d(v)$, then *R* is called a reduct of *C*.

According to P a w l a k [3], the reduct of Definition 1.2 is called a REDuct based on a positive region (RED). Let $\mathrm{RED}(C)$ be the set of all reducts of *C*. In the view point of relation database theory, R is a reduct of *C*, if R satisfies the functional dependency $R \to \{d\}$ and $\forall B \subset R,\ B \not\to \{d\}$.

In the next content, we introduce some basic concepts of relational database theory [1].

Let $R = \{a_1, ..., a_n\}$ be a finite set of attributes and let $D(a_i)$ be the set of all possible values of attribute a_i, the relation *r* over *R* is the set of tuples $\{h_1, ..., h_m\}$ where $h_j : R \to \bigcup_{a_i \in R} D(a_i),\ 1 \le j \le m,$ is a function that $h_j(a_i) \in D(a_i)$.

Let $r = \{h_1, ..., h_m\}$ be a relation over $R = \{a_1, ..., a_n\}$. Any pair of attribute sets $A, B \subseteq R$ is called Functional Dependency (FD) over *R*, and it is denoted by $A \to B$ if and only if

$$(\forall h_i, h_j \in r)\big((\forall a \in A)(h_i(a) = h_j(a)) \Rightarrow (\forall b \in B)(h_i(b) = h_j(b))\big).$$

The set $F_r = \{(A, B) : A, B \subseteq R, A \to B\}$ is called a full family of functional dependencies in r. Let $P(R)$ be the power set of the attribute set R. A family $F = P(R) \times P(R)$ is called an *f*-family over R if and only if for all subsets of the attributes $A, B, C, D \subseteq R$ the following properties hold:

(1) $$(A, A) \in F.$$

(2) $$(A, B) \in F,\ (B, C) \in F \Rightarrow (A, C) \in F.$$

(3) $$(A, B) \in F,\ A \subseteq C, D \subseteq B \Rightarrow (C, D) \in F.$$

(4) $$(A, B) \in F,\ (C, D) \in F \Rightarrow (A \cup C, B \cup D) \in F.$$

Clearly, F_r is an *f*-family over R. It is also known that if F is an *f*-family over R, then there is a relation r such that $F_r = F$. Let us denote by F^+ the set of all FDs, which can be derived from F by using rules (1)-(4).

A pair $s = (R, F)$, where R is a set of attributes and F is a set of FDs on R, is called a *relation scheme*. For any $A \subseteq R$, the set $A^+ = \{a : A \to \{a\} \in F^+\}$ is called the *closure* of A on s. It is clear that $A \to B \in F^+$ if and only if $B \subseteq A^+$. Similarly, $A_r^+ = \{a : A \to \{a\} \in F^+\}$ is called the *closure of A* on relation r.

Let $s = (R, F)$ be a relation scheme over R and $a \in R$. The set

$$\mathcal{K}_a^s = \{A \subseteq R : A \to \{a\}, \nexists B : (B \to \{a\})(B \subset A)\}$$

is called a family of minimal sets of the attribute *a* over *s*. Similarly, the set

$$\mathcal{K}_a^r = \{A \subseteq R : A \to \{a\}, \nexists B \subseteq R : (B \to \{a\})(B \subset A)\}$$

is called a family of minimal sets of the attribute *a* over *r*.

Recall that a family $\mathcal{K} \subseteq P(R)$ is a Sperner-system on R if for any $A, B \in \mathcal{K}$ implies $A \not\subset B$. It is clear that $\mathcal{K}_a^s, \mathcal{K}_a^r$ are Sperner-systems over R. Let $\mathcal{K}$ be a Sperner-system. We defined the set $\mathcal{K}^{-1}$, as follows:

$$\mathcal{K}^{-1} = \{A \subset R : (B \in \mathcal{K}) \Rightarrow (B \not\subset A)\} \text{ and if } (A \subset C) \Rightarrow (\exists B \in \mathcal{K})(B \subseteq C).$$

It is easy to see that $\mathcal{K}^{-1}$ is a Sperner-system on R, too. If $\mathcal{K}$ is a Sperner-system over R as a set of all minimal keys of relation r (or the relation scheme s), then $\mathcal{K}^{-1}$ is the set of subsets of R, which does not contain the elements of $\mathcal{K}$ and which is maximal for this property, $\mathcal{K}^{-1}$ is called antikey. If $\mathcal{K}$ is a Sperner-system over R as the family of minimal sets of the attribute a over r (or s); in other words $\mathcal{K} = \mathcal{K}_a^r$ (or $\mathcal{K} = \mathcal{K}_a^s$), then $\mathcal{K}^{-1} = \{\mathcal{K}_a^r\}^{-1}$ (or $\mathcal{K}^{-1} = \{\mathcal{K}_a^s\}^{-1}$) is the

family of maximal subsets of R which are not the family of minimal sets of the attribute a, defined as [1]:

$$\left\{\mathcal{K}_a^r\right\}^{-1} = \left\{A \subseteq R : A \to \{a\} \notin F_r^+, A \subset B \Rightarrow B \to \{a\} \in F_r^+\right\},$$

$$\left\{\mathcal{K}_a^s\right\}^{-1} = \left\{A \subseteq R : A \to \{a\} \notin F^+, A \subset B \Rightarrow B \to \{a\} \in F^+\right\}.$$

2. Results

In real problems, the decision tables often contain inconsistent objects which have the same values on the conditional attribute, but different values on the decision attribute. These decision tables are called inconsistent decision tables. However, based on a data preprocessing step, we can convert the inconsistent decision tables to consistent decision tables by deleting inconsistent objects in an inconsistent decision.

Algorithm 2.1 [1]. Finding a minimal key from the set of antikeys.

Input: Let K be a Sperner-system, H be a Sperner-system and $C = \{c_1, ..., c_m\} \subseteq R$, such that $H^{-1} = K$ and there exists $B \subseteq K : B \subseteq C$.

Output: $D \subseteq H$.

Step 0. We set $T(0) = C$.

Step *i*+1. Set

$$T(i+1) = \begin{cases} T(i) - c_i & \text{if} \quad \forall B \in K, T(i) - c_i \not\subset B, \\ T(i) & \text{otherwise.} \end{cases}$$

Then set $D = T(m)$

Lemma 2.1 [1]. If K is the set of antikeys, then $T(m) \in H$.

Theorem 2.1 [1]. Let H be a Sperner-system over R and $H^{-1} = \{B_1, ..., B_m\}$ is a set of antikeys of H, $T \in H$. Then $T \subset H$, $T \neq \varnothing$ if and only if there exists $B \subseteq R$ such that $B \subseteq T^{-1}$, $B \not\subset B_i$ for any $1 \le i \le m$.

Based on Lemma 2.1 and Theorem 2.1 we have the following

Algorithm 2.2 [1]. Finding the set of minimal keys from the set of antikeys.

Input: Let $K = \{B_1, ..., B_m\}$ be a Sperner-system over R.

Output: H such that $H^{-1} = K$.

Step 1. Using Algorithm 2.1, we calculate A_1. We set $K_1 = A_1$.

Step *i*+1. If there is a $B \in K_i^{-1}$ such that $B \not\subseteq B_j \ (\forall j : 1 \le j \le m)$, then using Algorithm 2.1 we calculate A_{i+1}, where $A_{i+1} \in H$, $A_{i+1} \subseteq B$. We set $K_{i+1} = K_i \cup A_{i+1}$. In the converse case we set $H = K_i$.

Lemma 2.2 [1]. The set *H*, obtained by Algorithm 2.2 satisfies $H^{-1} = K$ and the time complexity of Algorithm 2.2 is exponential to the number of attributes of *R*.

Algorithm 2.3 [5]. Finding the set of all reducts in a consistent decision table.

Input: Let $\mathrm{DS} = \left(U, C \cup \{d\}, V, f\right)$ be a consistent decision table, $C = \{c_1, c_2, ..., c_n\}$, $U = \{u_1, u_2, ..., u_m\}$.

Output: $\mathrm{RED}(C)$.

Let us consider the decision table *DS* as the relation $r = \{u_1, u_2, ..., u_m\}$ over the set of attributes $R = A \cup \{d\}$.

Step 1. From r we construct the equality system $\mathcal{E}_r = \{E_{ij} : 1 \le i < j \le m\}$ where $E_{ij} = \{a \in R : a(u_i) = a(u_j)\}$.

Step 2. From $\mathcal{E}_r$ we construct

$\mathcal{M}_d = \{A \in \mathcal{E}_r : d \notin A \not\exists B \in \mathcal{E}_r : d \notin B, A \subset B\}$.

Step 3. Using Algorithm 2.2, we calculate $\mathcal{K}$ from $\mathcal{M}_d$ $\left(\mathcal{M}_d = \mathcal{K}^{-1}\right)$.

Step 4. We set $\mathrm{RED}(C) = \mathcal{K} - \{d\}$.

Lemma 2.2 [5]. The set $\mathrm{RED}(C)$ obtained by Algorithm 2.3 satisfies $\mathrm{RED}(C) = \mathcal{K} - \{d\}$. The time complexity of Algorithm 2.3 is exponential to the number of conditional attributes of the decision tables.

Lemma 2.3 [4]. Let $\mathrm{DS} = \left(U, C \cup \{d\}, V, f\right)$ be a consistent decision table where $C = \{c_1, c_2, ..., c_n\}$, $U = \{u_1, u_2, ..., u_m\}$. Let us consider $r = \{u_1, u_2, ..., u_m\}$ on the attribute set $R = C \cup \{d\}$.

We set $\mathcal{E}_r = \{E_{ij} : 1 \le i < j \le m\}$ where $E_{ij} = \{a \in R : a(u_i) = a(u_j)\}$.

We set $\mathcal{M}_d = \{A \in \mathcal{E}_r : d \notin A, \not\exists B \in \mathcal{E}_r : d \notin B, A \subset B\}$.

Then we have $\mathcal{M}_d = \left(\mathcal{K}_d^r\right)^{-1}$ where $\mathcal{K}_d^r$ is called a family of minimal sets of the attribute *d* over relation *r*.

Lemma 2.4. If $\mathrm{DS} = \left(U, C \cup \{d\}, V, f\right)$ is a consistent decision table then $\left(K_d^r\right)^{-1}$ is a Sperner-system over *C*. Conversely, if *K* is a Sperner-system over *C* then there esists a consistent decision table $\mathrm{DS} = \left(U, C \cup \{d\}, V, f\right)$, such that $K = \left(K_d^r\right)^{-1}$.

P r o o f: We have $\left\{\mathcal{K}_a^r\right\}^{-1} = \left\{A \subseteq R : A \to \{a\} \notin F_r^+, A \subset B \Rightarrow B \to \{a\} \in F_r^+\right\}$. It is obvious that $\left(K_d^r\right)^{-1}$ is a Sperner-system. Conversely, if *K* is a Sperner-system over *C*, supposing that $K = \{A_1, ..., A_m\}$, we construct a decision tables $\mathrm{DS} = \left(U, C \cup \{d\}, V, f\right)$ as follows.

We set $U = \{u_0, u_1, ..., u_m\}$, $R = C \cup \{d\}$:

1) for all $c \in C$ we set $u_0(c) = 0$; set $u_0(d) = 0$;

2) for all $i = 1, ..., m$ we set $u_i(c) = 0$ if $c \in A_i$; $u_i(c) = i$ otherwise; set $u_i(d) = i$ for all $i = 1, ..., m$.

We set $\mathcal{E}_r = \left\{E_{ij} : 1 \le i < j \le m\right\}$, $E_{ij} = \left\{a \in R : a(u_i) = a(u_j)\right\}$.

We set $\mathcal{M}_d = \left\{A \in \mathcal{E}_r : d \notin A, \nexists B \in \mathcal{E}_r : d \notin B, A \subset B\right\}$..

We can see that $\mathcal{M}_d = \{A_1, ..., A_m\}$. According to Lemma 2.3 we have $\mathcal{M}_d = \left(K_d^r\right)^{-1}$. Consequently, $K = \left(K_d^r\right)^{-1}$.

Lemma 2.5. Given $R = \{a_1, ..., a_n\}$ be a non-empty attribute set. Then there exists a Sperner-system *K* over *R* such that the number of elements in *K* is exponential with respect to n and the number of elements in K^{-1} is polynomial with respect to *n*.

P r o o f: We partition *R* into groups, each group consists of two elements. So we have:

$$R = X_1 \cup X_2 \cup ... \cup X_m \cup W \quad \text{where } |X_i| = 2, X_i \cap X_j = \varnothing, \ m = \lfloor n/2 \rfloor,$$

$$|W| = \begin{cases} 0 & \text{if } n \text{ even,} \\ 1 & \text{if } n \text{ odd.} \end{cases}$$

Let us consider:

$$H = \left\{A : |A \cap X_i| = 1, \forall i\right\} \text{ if } |W| = 0, \text{ and}$$

$$H = \left\{A : |A \cap X_i| = 1 \quad \text{if} \quad 1 \le i \le m+1, \quad \text{and} \quad |A \cap (X_m \cup W)| = 1\right\}$$

$$\text{if } |W| = 1.$$

We can see that $|H| = C_n^{[n/2]}$. We know that $C_n^{[n/2]}$ is approximately $2^{(n+1)/2}/(3.14 \times n^{1/2})$. So we can conclude that $|H| > 2^{[n/4]}$.

We set $K = \left\{B : |B| = n-2, \ B \cap X_i = \varnothing \text{ for any } i, 1 \le i \le m\right\}$ if $|W| = 0$ and

$$K = \{B : |B| = n-2,\ B \cap X_i = \varnothing \text{ for any } i,\ 1 \le i \le m$$
$$\text{or } |B| = n-3,\ B \cap (X_m \cup W) = \varnothing\} \text{ if } |W| = 1.$$

We have $|K| \le m-1$ and $K = H^{-1}$.

Theorem 2.2. Let $\mathrm{DS} = (U, C \cup \{d\}, V, f)$ be a consistent decision, then the time complexity of the problem of finding all reducts of DS is exponential with respect to the number of attributes of DS.

P r o o f: We prove two steps.

1) There exists an exponential algorithm which finds all reducts of a decision table. According to Algorithm 2.3, we obtain all reducts of a decision table. The time complexity of Algorithm 2.3 is exponential with respect to the number of the attributes of DS.

2) The time complexity of the problem of finding all reducts is not less than exponential with respect to the number of attributes of DS. Suppose that $C = \{a_1, ..., a_n\}$. We construct the partition on C according to Lemma 2.5. We construct H and K according to Lemma 2.5. We can see that $|H| > 2^{[n/4]}$, $|K| \le m+1|$ and $K = H^{-1}$ where m = [n/2]. Suppose that $K = \{A_1, ..., A_p\}$. We construct a decision table $\mathrm{DS} = (U, C \cup \{d\}, V, f)$ where $U = \{u_0, u_1, ..., u_p\}$ as follows:

i) for all $c \in C$ we set $u_0(c) = 0$; set $u_0(d) = 0$;

ii) for all $i = 1, ..., p$ we set $u_i(c) = 0$ if $c \in A_i$; $u_i(c) = i$ otherwise; set $u_i(d) = i$ for all $i = 1, ..., p$.

According to Lemma 2.4 we have $K = (K_d^r)^{-1}$, so that $H = K_d^r$. Consequently, $\mathrm{RED}(C) = H - \{d\}$.

Therefore, according to Lemmas 2.4 and 2.5, for any non-empty attribute set $A = C \cup \{d\}$ we always have a consistent decision table $\mathrm{DS} = (U, C \cup \{d\}, V, f)$ such that the number of $\mathrm{RED}(C)$ is exponential with respect to the number of the attributes of A. On the other hand, the number of the objects in U is polynomial with respect to the number of the attributes of A. Theorem 2.2 was proved. ■

References

1. D e m e t r o v i c s, J., V. D. T h i. Some Remarks on Generating Armstrong and Inferring Functional Dependencies Relation. – Acta Cybernetica, Vol. **12**, 1995, 167-180.

2. D e m e t r o v i c s, J., V. D. T h i, N. L. G i a n g. An Efficient Algorithm for Determining the Set of All Reductive Attributes in Incomplete Decision Tables. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 4, 118-126.
3. P a w l a k, Z. Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, 1991.
4. T h i, V. D., N. L. G i a n g. A Method to Construct Decision Table from Relation Scheme. – Cybernetics and Information Technologies, Vol. **11**, 2011, No 3, 32-41.
5. T h i, V. D., N. L. G i a n g. A Method for Extracting Knowledge from Decision Tables in Terms of Functional Dependencies. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, 73-82.