Serdica Journal of Computing 16(1), 2022, pp. 24-38, 10.55630/sjc.2022.16.24-38
Published by Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

On the Problem Related to Reductive Attributes in the Incomplete Decision Tables

Janos Demetrovics[1], Nguyen Long Giang[2,*], Vu Duc Thi[3], Pham Viet Anh[4]

[1]Institute for Computer Science and Control (SZTAKI),
Hungarian Academy of Sciences, Budapest, Hungary
demetrovics@sztaki.mta.hu

[2]Institute of Information Technology,
Vietnam Academy of Science and Technology, Hanoi, Vietnam
nlgiang@ioit.ac.vn

[3]Information Technology Institute,
Vietnam National University, Hanoi, Vietnam
vdthi@vnu.edu.vn

[4]Hanoi University of Industry, Hanoi, Vietnam
anhpv@haui.edu.vn

Abstract

Attribute reduction is a key problem in the process of data mining and knowledge discovery. Up to now, many attribute reduction algorithms in incomplete decision tables have been proposed. However, the research results related to conditional attributes and reduct of incomplete decision tables are still limited. By relational database approach, this paper investigates some properties of conditional attributes and proposes an algorithm to determine all reductive attributes of consistent incomplete decision tables in polynomial time. The proposed algorithm is an effective tool to eliminate all redundant attributes in data pre-processing in order to improve the efficiency of data mining models.

Keywords: Incomplete Decision System, Attribute Reduction, Reduct, Reductive Attribute

ACM 2012 CCS Concepts: Information systems → Information systems applications → Decision support systems → Data analytics

Mathematics Subject Classification 2020: 68P15, 68T30, 68T37, 68U35

Received: May 28, 2022, *Accepted:* June 6, 2022, *Published:* July 11, 2022

[*]*Corresponding author*

Citation: Janos Demetrovics, Nguyen Long Giang, Vu Duc Thi, Pham Viet Anh, On the Problem Related to Reductive Attributes in the Incomplete Decision Tables, Serdica Journal of Computing 16(1), 2022, pp. 24-38, https://doi.org/10.55630/sjc.2022.16.24-38

1 Introduction

One of the most important technique in data preprocessing in data mining and machine learning is attribute reduction or feature selection. The attribute reduction process objectives are removing unnecessary and redundant attributes and keep the attribute reduction set (called as reduct). The objective of attribute reduction is to improve the data mining models efficiently. Pawlak [1] was introduced traditional Rough Set (RS), which is considered as an effective tool for discover the reduct of decision tables. Based on RS theory or expanded modeling of RS, various methods have been introduced to find the reduct of decision tables recently. However, these algorithms are heuristic based algorithms which finds the best reduct regarding the classification quality of the attribute set. In fact, studying properties of reducts and conditional attributes plays an important role in eliminating redundant attributes in decision tables.

In consistent complete decision tables, many scientist proposed new methods related to the properties of reduct and inferring knowledge by relational database theory approach in recent years [2–9]. In paper [2], authors proved that the time complexly of calculating all reducts is exponentials in the number of conditional attributes. In paper [3], authors proved some properties related to the time complexity of relatively reduced set search. In paper [4], the algorithm was proposed to find a set of entire reducts in a complete decision consistency table in polynomial time. Based on the proposed algorithm, an algorithm is constructed for a complete decision consistency table to inferring knowledge in term of functional dependencies. Furthermore, authors in paper [5] solved the inverse problem, they recommend an algorithm which propose a complete decision table based on a traditional of knowledge in term of functional dependencies. This result allows us to generate data for knowledge systems to improve the efficient training and testing knowledge models. In paper [6], authors discovered some properties related to conditional attributes and reduct. A polynomial algorithm was introduced in order to construct a novel reduction decision table from an already build decision table and also presented another new method to extract entire the functional dependencies from the consistent decision table. Authors in paper [7] proposed some algorithms to decrease the quantity of objects in consistent complete decision tables by discovering some features of reduct and reductive attributes in regard to relational database theory approach. The proposed algorithms can be effectively applied in the data preprocessing phase to improve data mining efficient and machine learning models.

In research [8] authors discovered some properties of reduct related to Sperner-system and state that the study of some properties on reduct is equivalent to the

study of some properties on the Sperner-system. This result opens a new research direction on building efficient attribute reduction models based on research on Sperner-system. On the problem related to reductive attributes, in paper [9], authors proposed two algorithms in polynomial time: the first algorithm to find all reductive attributes by relational database theory approach; the second algorithm also to find all reductive attributes by rough set theory. The authors also proved that the results of two algorithms are the same.

In practical problems, decision tables often miss values in the attribute value domain, known as incomplete decision tables. On these tables, Kryszkiewicz in paper [10] constructed a tolerance relationship on the attribute value domain and proposed tolerance RS model. From the modeling of tolerance RS, various heuristic algorithms about feature reduction have been proposed so far. However, studies refereed to reduct properties of incomplete decision tables are still limited. In the paper [11] authors discover some properties of reducts in incomplete decision tables and prove that the properties of reducts in incomplete decision tables are equivalent to properties of the Sperner-systems in the theory of relational database. By extending the results in the paper [9, 12], in this paper we developed an algorithm to find all reductive attributes of consistent incomplete decision tables in polynomial time. The proposed algorithm let us to eliminate all redundant attributes in incomplete decision tables before performing attribute reduction and rule extraction algorithms. The structure of this paper is as follows. Section 2 describes the basic definitions related to RS theory and relational database. Section 3 provides some combinational results in relational database. Section 4 proposes an algorithm to find all reductive attributes of consistent incomplete decision tables in polynomial time. The last section addresses the concluding and further research.

2 Basic Concepts

2.1 The basic concepts in rough set theory

Some basic concepts in the rough set theory are performed in this paper [1]. The decision table is a set of four $DT = (U, C \cup D, V, f)$, in which $U = \{u_1, u_2, \ldots, u_n\}$ is non-empty set, containing finite objects; $C = \{c_1, c_2, \ldots, c_m\}$ is a gathering of condition attribute; D is a set of decision attributes where C and D are two separated sets, and $V = \bigcup V_a$ where V_a is the value set of the attribute $a \in A = C \cup D; f : U \times (C \cup D) \to V$ is the information function. For any $a \in C \cup D, u \in U$, function f has the value $f(u, a) \in V_a$. Not losing the comprehensive characteristics, hypothesis D only has one decision attribute

is d (If D has many attributes, it can be reduced to an attribute by using an encryption [10]). From this, we consider the decision table $DT = (U, C \cup d, V, f)$, in which $\{d\} \notin C$.

Each subset $P \subseteq C \cup \{d\}$ defines an indistinguishable relation, called equivalence relation: $IND\,(P) = \{(x, y) \in U \times U | \forall a \in P, f\,(x, a) = f\,(y, a)\}$. $IND(P)$ defines a partition on U, denoted by $U/P = \{P_1, P_2, \ldots, P_m\}$. One element in U/P is called an equivalence class. For any $B \subseteq C$ and $X \subseteq U$, $B-$ upper approximation of X is set $\overline{B}X = \{u \in U \mid [u]_B \cap X \neq \emptyset\}$, $B-$ lower approximation of X is set $\underline{B}X = \{u \in U \mid [u]_B \subseteq X\}$, $B-$ boundary region of X is set $\overline{B}X \backslash \underline{B}X$ and $B-$ the positive region of $\{d\}$ is the set $POS_B(\{d\}) = \cup_{X \in U/D}(\underline{B}X)$. The decision table DT is consistent only when $POS_C\,(\{d\}) = U$, or function dependency $C \to d$ is true, whereas DT is inconsistent. If DT is inconsistent, $POS_C(\{d\})$ is maximum subset of U that satisfies the function dependency $C \to d$.

Definition 2.1. *Let the decision table $DT = (U, C \cup d, V, f)$. If $B \subseteq C$ satisfies:*

(1) $POS_B(\{d\}) = POS_C(\{d\})$,

(2) $\forall B' \subset B(POS_{B'}(\{d\}) \neq POS_C(\{d\}))$,

then B is a reduct of C. If DT is consistent, the above definition shows that B is a reduct of C if it satisfies $B \to d$ and $\forall B' \subset B$, $B' \nrightarrow \{d\}$.

Definition 2.2. *Let $R = \{o_1, o_2, \ldots, o_n\}$ be a finite set of attributes and let $D\,(o_i)$ be the set of all possible values of attribute o_i. A relation r over R is the tuples' rally $\{b_1, \ldots, b_m\}$ where $b_j : R \to \bigcup_{o_i \in R} D\,(o_i)\,, 1 \leq j \leq m$ is a function that $b_j\,(o_i) \in D\,(o_i)$.*

Let $r = \{b_1, \ldots, b_m\}$ be a relation over $R = \{o_1, o_2, \ldots, o_n\}$. Any pair of attribute sets $\mathcal{X}, \mathcal{Y} \subseteq R$ is called the functional dependency (FD for short) over R, and denoted by $\mathcal{X} \to \mathcal{Y}$, if and only if

$$(\forall b_i b_j \in r)\,((\forall b \in \mathcal{X})\,(b_i\,(x) = b_j\,(x))) \Longrightarrow ((\forall y \in \mathcal{Y})\,(b_i\,(y) = b_j\,(y)))\,.$$

The set $F_r = \{(\mathcal{X}, \mathcal{Y}) : \mathcal{X}, \mathcal{Y} \subseteq R, \mathcal{X} \to \mathcal{Y}\}$ is called the full family of functional dependencies in r.

Definition 2.3. *Let $DT = (U, C \cup \{d\})$ be a decision table where $\{d\} \notin C$. DT is consistent if and only if the functional dependency $C \to \{d\}$ is true, it means that for any $s, t \in U$, if $C(s) = C(t)$ then $d(s) = d(t)$. Otherwise, DT is inconsistent.*

Definition 2.4. *Let $DT = (U, C \cup \{d\}, V, f)$ be a consistent decision table and an attribute set $\mathcal{R} \subseteq C$. $\mathcal{R}$ is called a reduct of DT if:*

(1) *For each* $s, t \in U$, *if* $\mathcal{R}(s) = \mathcal{R}(t)$ *then* $d(s) = d(t)$.

(2) *For each* $\mathcal{G} \subset \mathcal{R}$, *there exists* $s, t \in U$ *in which* $\mathcal{G}(s) = \mathcal{G}(t)$ *and* $d(s) \neq d(t)$.

The above reduct is called Pawlak reduct.

Definition 2.5. *Let* $r = \{b_1, \ldots, b_m\}$ *be a relation on* $R = \{o_1, \ldots, o_n\}$. *If* $\forall o_i \in R$ *has* $\mathcal{D}_{o_i}$ *and* $* \in \mathcal{D}_{o_i}$ *where* $*$ *is "missing value":*

$$b_j : R \to \cup \mathcal{D}_{o_i} \quad so \quad b_j(o_i) \in \mathcal{D}_{o_i}.$$

Definition 2.6. *Let* r *is the relation on* $R = \{o_1, o_2, \ldots, o_n\}$ *and* $A \subseteq R$. *In the case, we denote* $b_i \sim b_j(A)$ *if each* o *belongs to* A: $b_i(o) = b_j(o)$ *or* $b_i(o) = *$ *or* $b_j(o) = *$.

Definition 2.7. *Let* $r = \{b_1, \ldots, b_m\}$ *on* $R = \{o_1, o_2, \ldots, o_n\}$. *Then* $\mathcal{X}, \mathcal{Y} \subseteq R$ *and* $\mathcal{X}$ *tolerance determines* $\mathcal{Y}$ *denoted by* $\mathcal{X} \xrightarrow{t} \mathcal{Y}$ *if:*

$$(\forall b_i, b_j \in r) \ \ (if \ \ b_i \sim b_j(\mathcal{X}) \ \ \ then \ \ \ b_i \sim b_j(\mathcal{Y})).$$

Set $T_r = \left\{(\mathcal{X}, \mathcal{Y}) : \mathcal{X}, \mathcal{Y} \subseteq R \text{ and } \mathcal{X} \xrightarrow{t} \mathcal{Y}\right\}$. *It is easy to see that:*

(1) $(\mathcal{X}, \mathcal{X}) \in T_r \forall \mathcal{X} \subseteq R$,

(2) $(\mathcal{X}, \mathcal{Y}) \in T_r$ *then* $\mathcal{X} \subseteq \mathcal{C}, \mathcal{D} \subseteq Y$ *has* $(\mathcal{C}, \mathcal{D}) \in T_r$,

(3) $(\mathcal{X}, \mathcal{Y}) \in T_r, (Y, \mathcal{C}) \in T_r \Longrightarrow (\mathcal{X}, \mathcal{C}) \in T_r$.

Set $\mathcal{X}^+ = \left\{o \in R : \mathcal{X} \xrightarrow{t} \{o\}\right\}$.

Definition 2.8. *Incomplete decision table* $IDT = (U, C \cup d, V, f)$ *with* $* \notin \mathcal{D}_d$ *(it means that the value domain of the decision attribute* d *does not include* $*$*).* C *is the set of condition attributes.* DT *is the incomplete decision table which is consistent if* $C \xrightarrow{t} \{d\}$.

We can see that if IDT is inconsistent, we can test by using a polynomial time algorithm on elements of U to eliminate the elements, making DT consistently. After the elimination, we have the set $\mathbb{U}$ then $DT = (\mathbb{U}, C \cup d, V, f)$ is consistent.

Definition 2.9. *Let* $IDT = (U, C \cup d, V, f)$ *be a consistent incomplete decision table.* B *is a reduct of* IDT *if:* $B \subseteq C : B \xrightarrow{t} \{d\}$ *and* $\forall B' \subsetneq B$ *then* $B' \not\xrightarrow{t} \{d\}$ *(it means that* B' *is a proper subset of* B *then* B' *does not tolerance determine* d*). Set* $PRED(C) = \{B : B \text{ is reduct of } IDT\}$.

Definition 2.10. *Suppose that* $R = \{o_1, o_2, \ldots, o_n\}$. $\mathcal{K} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m\}$ *is the Sperner-system on* R *if*

$$\mathcal{A}_i \nsubseteq \mathcal{A}_j \ \forall i, j.$$

Definition 2.11. *Suppose that* $\mathcal{K} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m\}$ *is the Sperner-system on* R. *Set* $\mathcal{K}^{-1} = \{B \subsetneq R : (A \in \mathcal{K} \Longrightarrow \mathcal{A} \nsubseteq B \text{ and } B \subsetneq C) \text{ then } \exists \mathcal{A} \in \mathcal{K} : \mathcal{A} \subseteq C\}$. *Then* $\mathcal{K}^{-1}$ *is called the anti-key of* $\mathcal{K}$.

Definition 2.12. *Suppose that* $IDT = (U, C \cup d, V, f)$ *is a consistent incomplete decision table. Set* $r = u = \{u_1, \ldots, u_m\}$, $R = C \cup d$. *It is easy to see that* $PRED(C) = \mathcal{K}_d^t = \{\mathcal{A} \subseteq C : \mathcal{A} \xrightarrow{t} \{d\} \text{ and } \nexists B : B \xrightarrow{t} \{d\} \text{ and } B \subsetneq \mathcal{A}\}$ *and* $PRED(C)$ *is the Sperner-system.*

Definition 2.13. *Suppose that* $IDT = (U, C \cup d, V, f)$ *is the consistent incomplete decision table. Let* $r = U = \{u_1, \ldots, u_m\}$, $R = C \cup d$.

1. *From* r *we calculate the equivalent sets:* $\epsilon_r = \{E_{ij} : 1 \leqslant i \leqslant j \leqslant m\}$ *with*

$$E_{ij} = \{o \in R : o(u_i) = o(u_j) \ \text{ or } \ o(u_i) = * \ \text{ or } \ o(u_j) = *\}.$$

2. *From* ϵ_r, *set*

$$M_d = \{A \in \epsilon_r : A \neq R, d \notin A \ \text{ and } \ \nexists B \in \varepsilon_r : d \notin B \ \text{ and } \ A \subsetneq B\}.$$

Definition 2.14. *Suppose that* $IDT = (U, C \cup d, V, f)$ *is the consistent incomplete decision table. Attribute* $a \in C$ *is referred to as a reductive attribute if there is a reduct* $A \in PRED(C)$ *such that* $a \in A$.

Set $REAT(DT) = \{a \in C : a \text{ is reductive attribute}\}$.
It is easy to see that $REAT(C) = \cup_{A \in PRED(C)} A$.

3 Some Results in Relational Database

3.1 Algorithm finding the minimum key set from the set of anti-keys

Algorithm 3.1 ([13,14])**.** Find the minimum key set from the set of anti-keys.

Input: Let $\mathcal{K}$ be the Sperner-system playing the role of an anti-key set, $C = \{b_1, \ldots, b_n\} \subseteq R$ and $\mathcal{H}$ is the Sperner-system playing the role of key set $\left(\mathcal{K} = \mathcal{H}^{-1}\right)$ for $\exists B \in \mathcal{K} : B \subsetneq C$.

Output: $D \in \mathcal{H}$.

Step 1: Set $\mathfrak{t}(0) = C$.

Step i+1: Set $\mathfrak{t}(i+1) = \mathfrak{t}(i) - b_{i+1}$ if $\forall B \in \mathcal{K}$ without $\mathfrak{t}(i+1) \subset B$; in the opposite case, set $\mathfrak{t}(i+1) = \mathfrak{t}(i)$.

Finally we set $D = \mathfrak{t}(n)$.

It is noticeable that the above algorithm's time complexity is polynomial with n and $|\mathcal{K}|$.

Theorem 3.1 ([13,15]). *Let $\mathcal{K} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ be the Sperner-system on $R = \{o_1, \ldots, o_n\}$. Suppose $\mathcal{K}^{-1} = \{B_1, B_2, \ldots, B_k\}$ then $\cup_{\mathcal{A}_i \in \mathcal{K}} \mathcal{A}_i = R \setminus \cap_{B_i \in \mathcal{K}^{-1}} B_i$.*

3.2 Algorithm finding the set of anti-keys from given Sperner-system

Algorithm 3.2 ([13,15]). Finding $\mathcal{K}^{-1}$ from a given Sperner-system $\mathcal{K}$.

Input: Let $\mathcal{K} = \{B_1, \ldots, B_m\}$ be a Sperner-system over $R = \{a_1, \ldots, a_n\}$.

Output: $\mathcal{K}^{-1}$.

Step 1: We set $\mathcal{K}_1 = \{R - \{a\} : a \in B_1\}$. It is obviously that $\mathcal{K}_1 = \{B_1\}^{-1}$.

Step q+1: $(q < m)$. Assume that $\mathcal{K}_q = F_q \cup \{X_1, \ldots, X_{tq}\}$, where $X_1, \ldots, X_{tq}$ are elements of $\mathcal{K}_q$ containing B_{q+1} and $F_q = \{A \in \mathcal{K}_q : B_{q+1} \nsubseteq A\}$. For all i $(i = 1, \ldots, t_q)$, we compute $\{B_{q+1}\}^{-1}$ on X_i in the same way as $\mathcal{K}_1$, which are the maximal subsets of X_i not containing B_{q+1}. We denote them by $A_1^i, \ldots, A_{ri}^i$. Let:

$$\mathcal{K}_{q+1} = F_q \cup \left\{A_p^i : A \in F_q \Rightarrow A_p^i \not\subset A, 1 \leq i \leq t_q, 1 \leq p \leq r_i\right\}$$

Finally, let $\mathcal{K}^{-1} = \mathcal{K}_m$.

Theorem 3.2 ([13,15]). *For any q $(1 \leq q \leq m)$, $\mathcal{K}_q = \{B_1, \ldots, B_q\}^{-1}$, that is $\mathcal{K}_m = \mathcal{K}^{-1}$. It is clear that $\mathcal{K}$, $\mathcal{K}^{-1}$ are unique and it is drawn the definition of $\mathcal{K}^{-1}$ that the order of the sequence $B_1, \ldots, B_m$ does not influence the Algorithm 1. Set $\mathcal{K}_q = \mathcal{F}_q \cup \{X_1, \ldots, X_{tq}\}$ and l_q $(1 \leq q \leq m-1)$ is the number of elements of K_q.*

Proposition 3.1 ([13,15]). *In the worst case, the Algorithm 3.1's is*

$$O\left(|R|^2 \sum_{q=1}^{m-1} t_q u_q\right)$$

where $u_q = I_q - t_q$ if $I_q > t_q$ and $u_q = 1$ if $I_q = t_q$.

It is apparent that in each step of the algorithm we have $\mathcal{K}_q$ is a Sperner-system over R. It is known that the size of any Sperner-system on R is not greater than $C_n^{[n/2]}$, where $n = |R|$. We can see that $C_n^{[n/2]}$ is roughly equal to $2^{n+1/2}/(\Pi.n^{1/2})$. Consequently, the time complexity of the algorithm can not be more than exponential in n. In cases of $I_q \leq I_m\,(\forall q : 1 \leq q \leq m-1)$, the Algorithm 1's time complexity is not more than $O\left(|R|^2 |\mathcal{K}| \left|\mathcal{K}^{-1}\right|^2\right)$. Therefore, in those cases the algorithm finds $\mathcal{K}^{-1}$ in polynomial time in $|R|, |\mathcal{K}|$ and $|\mathcal{K}|^{-1}$. Notably, when $|\mathcal{K}|, |\mathcal{K}|^{-1}$ is small, Algorithm 3.4 works.

Example 3.1. Let

$$\mathcal{K} = \{\{o_1, o_2, o_3\}, \{o_2, o_3, o_5\}, \{o_2, o_3, o_6\}, \{o_2, o_5, o_7, o_8\}\}$$

be the Sperner-system on $R = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}$, anti-keys set of $\mathcal{K}$ is determined as follows:

Step 1:

$$K_1 = \{\{o_2, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}\}.$$

Step 2: $K_1 = F_1 \cup X$

$$F_1 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}\}$$
$$X = \{\{o_2, o_3, o_4, o_5, o_6, o_7, o_8\}\}$$

It can be seen anti-keys of $\{o_2, o_3, o_5\}$ on set $X = \{\{o_2, o_3, o_4, o_5, o_6, o_7, o_8\}\}$ are $\{o_3, o_4, o_5, o_6, o_7, o_8\}$, $\{o_2, o_4, o_5, o_6, o_7, o_8\}$, $\{o_2, o_3, o_4, o_6, o_7, o_8\}$. From it we have:

$$K_2 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_6, o_7, o_8\}\}.$$

Step 3: $K_2 = F_2 \cup X$

$$F_2 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}\}$$
$$X = \{\{o_2, o_3, o_4, o_6, o_7, o_8\}\}$$

It can be seen anti-keys of $\{o_2, o_3, o_6\}$ on set $X = \{\{o_2, o_3, o_4, o_6, o_7, o_8\}\}$ are $\{o_3, o_4, o_6, o_7, o_8\}$, $\{o_2, o_4, o_6, o_7, o_8\}$, $\{o_2, o_3, o_4, o_7, o_8\}$. From it we have:

$$K_3 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_7, o_8\}\}.$$

Step 4: $K_3 = F_3 \cup X$

$$F_3 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_7, o_8\}\}$$
$$X = \{\{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}\}$$

It can be seen anti-keys of $\{o_2, o_5, o_7, o_8\}$ on set $X = \{\{o_1, o_2, o_4, o_5, o_6, o_7, o_8\}\}$ are

$$\{o_1, o_4, o_5, o_6, o_7, o_8\}, \{o_1, o_2, o_4, o_6, o_7, o_8\},$$
$$\{o_1, o_2, o_4, o_5, o_6, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7\}.$$

From it we have:

$$K_4 = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7\}\}.$$

From it we have a set of anti-keys of $\mathcal{K}$ is:

$$\mathcal{K}^{-1} = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7\}\}.$$

Example 3.2. Let $R = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}$ and the set of antikeys

$$K^{-1} = \{\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\}, \{o_2, o_3, o_4, o_7, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_8\}, \{o_1, o_2, o_4, o_5, o_6, o_7\}\}.$$

Consider $C = \{o_1, o_2, o_3, o_4, o_5, o_7, o_8\}$. Then the minimum key set is determined as follows:

Step 1: Set $\mathfrak{t}(0) = C = \{o_1, o_2, o_3, o_4, o_5, o_7, o_8\}$.

Step 2: Set $\mathfrak{t}(1) = \mathfrak{t}(0) \setminus \{o_1\} = \{o_2, o_3, o_4, o_5, o_7, o_8\}$.

$$\forall B \in \mathcal{K}^{-1} \text{ without } \mathfrak{t}(1) \subset B \to \mathfrak{t}(1) = \{o_2, o_3, o_4, o_5, o_7, o_8\}.$$

Step 3: Set $\mathfrak{t}(2) = \mathfrak{t}(1) \setminus \{o_2\} = \{o_3, o_4, o_5, o_7, o_8\}$. Because

$$\{o_1, o_3, o_4, o_5, o_6, o_7, o_8\} \in \mathcal{K}^{-1}$$

and

$$\mathfrak{t}(2) \subset \{o_1, o_3, o_4, o_5, o_6, o_7, o_8\} \to \mathfrak{t}(2) = \{o_2, o_3, o_4, o_5, o_7, o_8\}.$$

Step 4: Set $\mathfrak{t}(3) = \mathfrak{t}(2) \setminus \{o_3\} = \{o_2, o_4, o_5, o_7, o_8\}$.

$$\forall B \in \mathcal{K}^{-1} \text{ without } \mathfrak{t}(3) \subset B \to \mathfrak{t}(3) = \{o_2, o_4, o_5, o_7, o_8\}.$$

Step 5: Set $\mathfrak{t}(4) = \mathfrak{t}(3) \setminus \{o_4\} = \{o_2, o_5, o_7, o_8\}$.

$$\forall B \in \mathcal{K}^{-1} \text{ without } \mathfrak{t}(4) \subset B \to \mathfrak{t}(4) = \{o_2, o_5, o_7, o_8\}.$$

Step 6: Set $\mathfrak{t}(5) = \mathfrak{t}(4) \setminus \{o_5\} = \{o_2, o_7, o_8\}$.

$$\forall B \in \mathcal{K}^{-1} \text{ without } \mathfrak{t}(5) \subset B \to \mathfrak{t}(5) = \{o_2, o_7, o_8\}.$$

Step 7: *Set* $\mathfrak{t}(6) = \mathfrak{t}(5) \setminus \{o_7\} = \{o_2, o_8\}$. Because $\{o_2, o_3, o_4, o_7, o_8\} \in \mathcal{K}^{-1}$ and

$$\mathfrak{t}(6) \subset \{o_2, o_3, o_4, o_7, o_8\} \to \mathfrak{t}(6) = \{o_2, o_7, o_8\}.$$

Step 8: Set $\mathfrak{t}(7) = \mathfrak{t}(6) \setminus \{o_8\} = \{o_2, o_7\}$. Because $\{o_2, o_3, o_4, o_7, o_8\} \in \mathcal{K}^{-1}$ and

$$\mathfrak{t}(7) \subset \{o_2, o_3, o_4, o_7, o_8\} \to \mathfrak{t}(7) = \{o_2, o_7, o_8\}.$$

Hence, $D = \{o_2, o_7, o_8\}$ is a minimum key set from the set of anti-keys.

4 Propose an Algorithm for Finding all Reductive Attributes in an Incomplete Decision Table

The results of reductive attributes in the incomplete decision table are demonstrated in this section.

Theorem 4.1. *Suppose that* $IDT = (U, C \cup d, V, f)$ *is a consistent incomplete decision table. Set* $r = U = \{u_1, \ldots, u_m\}$, $R = C \cup d$.

- *From* r *we calculate the same set* $\varepsilon_r = \{E_{ij} : 1 \leq i \leq j \leq m\}$ *with* $E_{ij} = \{o \in R : o(u_i) = o(u_j) \;\; or \;\; o(u_i) = * \;\; or \;\; o(u_j) = *\}$.
- *From* ε_R *we set* $M_d = \{\mathcal{A} \in \varepsilon_r : \mathcal{A} \neq R, d \notin \mathcal{A} \;\; and \;\; \nexists B \in \varepsilon_r : d \notin B \;\; and \;\; \mathcal{A} \subsetneq B\}$.

$$\mathcal{K}_d^t = \{\mathcal{A} \subseteq C : \mathcal{A} \xrightarrow{t} \{d\} \;\; and \;\; \nexists B : B \xrightarrow{t} \{d\} \;\; and \;\; B \subsetneq \mathcal{A}\}.$$

Then $M_d = (\mathcal{K}_d^t)^{-1}$.

Proof. If $\forall \mathcal{A} \in M_d$, we can see that $\mathcal{A} = \mathcal{A}^+$ because if $\mathcal{A} \subsetneq \mathcal{A}^+$ then there is $e \in \mathcal{A}^+$ and $e \notin \mathcal{A}$. Because $\mathcal{A}$ is the equivalent maximum set then $\exists i, j$ $(1 \leq i < j \leq m)$ for $E_{ij} = \mathcal{A}$ and according to the definition of set $\mathcal{A}^+$, we have $\mathcal{A} \xrightarrow{t} \{e\}$ and from the definition of set E_{ij} then $e \in E_{ij}$. Therefore, $\mathcal{A} = \mathcal{A}^+$ and $d \notin \mathcal{A}$ then $d \notin \mathcal{A}^+$. Thus, $\mathcal{A} \overset{t}{\not\to} \{d\}$ ($\mathcal{A}$ does not tolerance determine d).

If we have B with $\mathcal{A} \subsetneq B$, based on the definition of set $\mathcal{A}$ if $d \notin B$ then $\forall i, j$ $(1 \leq i < j \leq m)$ we have $b_i \sim b_j\,(B)$, which is incorrect. Therefore, according to the definition of tolerance determination, we have $B \xrightarrow{t} R$. The case $d \in B$ then it is easy to see that $d \in B^+$. Therefore, both cases have $\forall B : \mathcal{A} \subsetneq B \Rightarrow B^+ \xrightarrow{t} \{d\}$. Therefore, according to the definition of $\mathcal{K}_d^t$ then $C \in \mathcal{K}_d^t$ so $C \subseteq B$. According to the definition of set $(\mathcal{K}_d^t)^{-1}$ we have $\mathcal{A} \in (\mathcal{K}_d^t)^{-1}$.

Oppositely, if $\mathcal{A} \in (\mathcal{K}_d^t)^{-1}$ then $\mathcal{A}^+ = \mathcal{A}$. Because if $\mathcal{A} \subsetneq \mathcal{A}^+$ then according to the definition of anti-key set we have $C \in (\mathcal{K}_d^t)$ for $C \subseteq \mathcal{A}^+$, means $\mathcal{A}^+ \xrightarrow{t} \{d\}$, leading to $\mathcal{A} \xrightarrow{t} \{d\}$. According to the definition of $(\mathcal{K}_d^t)^{-1}$ then $\mathcal{A}$ does not tolerance determine $\{d\}$ ($\mathcal{A} \overset{t}{\not\to} \{d\}$). Thus $\mathcal{A}^+ = \mathcal{A}$.

The definitions of sets M_d and $(\mathcal{K}_d^t)^{-1}$ (is the set of largest sets do not tolerance determine d), means $\mathcal{A} \in M_d$. Therefore $M_d = (\mathcal{K}_d^t)^{-1}$. □

Based on Theorem 4.1 and Theorem 3.2, the proposed algorithm is described as follows:

Car	Price	Mileage	Size	Max-speed	Decision
u_1	$\mathbb{H}$	$\mathbb{H}$	$*$	$*$	$\mathbb{P}$
u_2	$\mathbb{L}$	$*$	$\mathbb{F}$	$\mathbb{L}$	$\mathbb{G}$
u_3	$\mathbb{L}$	$\mathbb{L}$	$\mathbb{C}$	$\mathbb{H}$	$\mathbb{P}$
u_4	$\mathbb{M}$	$\mathbb{H}$	$\mathbb{C}$	$\mathbb{H}$	$\mathbb{G}$
u_5	$\mathbb{M}$	$\mathbb{H}$	$\mathbb{C}$	$*$	$\mathbb{G}$

Table 1: The first example of incomplete decision table.

Algorithm 4.1. The algorithm for finding all reductive attributes in an incomplete decision table.

Input: Let $DT = (U, C \cup d, V, f)$ be the consistent incomplete decision table. Set $r = U = \{u_1, \ldots, u_m\}$, $R = C \cup d$.

Output: $REAT(DT)$.

Steps:

1. From r we calculate the equivalent sets: $\varepsilon_r = \{E_{ij} : 1 \leq i \leq j \leq m\}$ with $E_{ij} = \{o \in R : o(u_i) = o(u_j) \text{ or } o(u_i) = * \text{ or } o(u_j) = *\}$.

2. From ε_R we set

$$M_d = \{\mathcal{A} \in \varepsilon_r : \mathcal{A} \neq R, d \notin \mathcal{A} \text{ and } \nexists B \in \varepsilon_r : d \notin B \text{ and } \mathcal{A} \subsetneq B\}.$$

3. Suppose $M_d = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k\}$. Let $G = \cap_{\mathcal{A}_i \in M_d} \mathcal{A}_i$.

4. Let $REAT(DT) = C \backslash G$.

Remark 4.1. Because all steps are calculated using a polynomial-time algorithm. Therefore, Algorithm 4.1 has polynomial time complexity with m and $|C|$. Based on Algorithm 4.1 and Theorem 4.1 we have the consequent below:

Corollary 4.1. *Let $IDT = (U, C \cup d, V, f)$ be an incomplete decision table, then there is one algorithm for finding all reductive attributes of IDT with polynomial time complexity.*

Example 4.1. Suppose that $IDT = (U, C \cup d, V, f)$ is an incomplete decision table as Table 1 where $U = \{u_1, u_2, \ldots, u_5, u_6\}$, $C = \{o_1, o_2, \ldots, o_4\}$.

• According to Algorithm 4.1, we need to do the following steps to find all reductive attributes:

1. Calculate M_d:

   $E_{12} = \{o_2, o_3, o_4\}$, $E_{13} = \{o_3, o_4, d\}$, $E_{14} = \{o_2, o_3, o_4\}$, $E_{15} = \{o_2, o_3, o_4\}$,

   $E_{23} = \{o_1, o_2\}$, $E_{24} = \{o_2, d\}$, $E_{25} = \{o_2, o_4, d\}$,

   $E_{34} = \{o_3, o_4\}, E_{35} = \{o_3, o_4\}$, $E_{45} = \{o_1, o_2, o_3, o_4, d\}$.

2. According to the step condition of Algorithm 4.2, there are $\mathcal{A}_1 = \{o_2, o_3, o_4\}$ and $\mathcal{A}_2 = \{o_1, o_2\}$ satisfy the condition of M_d. Thus,

$$M_d = \{\{o_2, o_3, o_4\}, \{o_1, o_2\}\}.$$

3. Calculate $G = \cap_{\mathcal{A}_i \in M_d} \mathcal{A}_i$, $G = \{o_2, o_3, o_4\} \cap \{o_1, o_2\} = \{o_2\}$.

4. Calculate $REAT(DT) = C \backslash G = \{o_1, o_2, o_3, o_4\} \backslash \{o_2\} = \{o_1, o_3, o_4\}$.

Thus, the set of reductive attributes in Table 1 is $REAT(DT) = \{o_1, o_3, o_4\}$.

• According to Algorithm 4.1 and the result M_d calculated from Step 2 above, consider $C = \{o_1, o_2, o_3, o_4\}$, the set of reductive attributes is determined as the follows:

Step 1: Set $\mathfrak{t}(0) = C = \{o_1, o_2, o_3, o_4\}$.

Step 2: Set $\mathfrak{t}(1) = \mathfrak{t}(0) \setminus \{o_1\} = \{o_2, o_3, o_4\}$. Because $\{o_2, o_3, o_4\} \in M_d$ and

$$\mathfrak{t}(1) \subset \{o_2, o_3, o_4\} \rightarrow \mathfrak{t}(1) = \{o_1, o_2, o_3, o_4\}.$$

Step 3: Set $\mathfrak{t}(2) = \mathfrak{t}(1) \setminus \{o_2\} = \{o_1, o_3, o_4\}$.

$$\forall B \in M_d \text{ without } \mathfrak{t}(2) \subset B \rightarrow \mathfrak{t}(2) = \{o_1, o_3, o_4\}.$$

Step 4: Set $\mathfrak{t}(3) = \mathfrak{t}(2) \setminus \{o_3\} = \{o_1, o_4\}$.

$$\forall B \in M_d \text{ without } \mathfrak{t}(3) \subset B \rightarrow \mathfrak{t}(3) = \{o_1, o_4\}.$$

Step 5: Set $\mathfrak{t}(4) = \mathfrak{t}(3) \setminus \{o_4\} = \{o_1\}$. Because $\{o_1, o_2\} \in M_d$ and

$$\mathfrak{t}(4) \subset \{o_1, o_2\} \rightarrow \mathfrak{t}(1) = \{o_1, o_4\}.$$

Hence, $D = \{o_1, o_4\}$ is a reduct in Table 1.

• We already know that $K_1 = \{o_1, o_4\} \in K$. Set $D_1 = \{\{o_1, o_4\}\}$. Then we have $D_1^{-1} = \{\{o_1, o_2, o_3\}, \{o_2, o_3, o_4\}\}$.

Name	Years (exp)	Employed	Pre-employ	Level	Top-tier school	Interned	Hired
o_1	$*$	Y	$*$	BS	N	N	Y
o_2	2	N	1	BS	$*$	Y	Y
o_3	7	N	$*$	$*$	N	$*$	N
o_4	2	$*$	1	MS	Y	N	Y
o_5	$*$	N	2	PhD	Y	$*$	N

Table 2: The second example of incomplete decision table's sample.

Because $\{o_1, o_2, o_3\} \in D_1^{-1}$ and $\{o_1, o_2, o_3\} \not\subseteq M_{dj}$ for all $M_{dj} \in M_d$ we consider $C = \{o_1, o_2, o_3\}$. Then by Algorithm 4.1, we obtain:

$$\mathfrak{t}(0) = C = \{o_1, o_2, o_3\}, \mathfrak{t}(1) = \{o_1, o_2, o_3\}, \mathfrak{t}(2) = \{o_1, o_3\}, \mathfrak{t}(3) = \{o_1, o_3\}.$$

- Thus, $K_2 = \{o_1, o_3\} \in K$. We set $D_2 = D_1 \cup K_2 = \{\{o_1, o_4\}, \{o_1, o_3\}\}$. Then we have $D_2^{-1} = \{\{o_2, o_3, o_4\}, \{o_1, o_2\}\} = M_d$.
- Because there isn't any $X \in D_2^{-1}$, in which $X \not\subseteq M_{d_j}$ so we set $K = D_2$. Therefore, the set of all reductive attributes in Table 1 is:

$$D = \{\{o_1, o_4\}, \{o_1, o_3\}\}.$$

Example 4.2. Let $IDT = (U, C \cup d, V, f)$ be an incomplete decision table as Table 2 where $U = \{u_1, u_2, \ldots, u_5, u_6\}$, $C = \{o_1, o_2, \ldots, o_6\}$.

According to Algorithm 4.1, we need to do the following steps to find all reductive attributes of Table 2:

1. Calculate M_d:

   $E_{12} = \{o_1, o_3, o_4, o_5, d\}$, $E_{13} = \{o_1, o_3, o_4, o_5, o_6\}$, $E_{14} = \{o_1, o_2, o_3, o_6, d\}$,
   $E_{15} = \{o_1, o_3, o_6\}$, $E_{23} = \{o_2, o_3, o_4, o_5, o_6\}$, $E_{24} = \{o_1, o_2, o_3, o_5, d\}$,
   $E_{25} = \{o_1, o_2, o_5, o_6\}$, $E_{34} = \{o_2, o_3, o_4, o_6\}$, $E_{35} = \{o_1, o_2, o_3, o_4, o_6, d\}$,
   $E_{45} = \{o_1, o_2, o_5, o_6\}$.

2. According to the step condition of Algorithm 4.1, there are:

   $$\mathcal{A}_1 = \{o_1, o_3, o_4, o_5, o_6\}, \ \mathcal{A}_2 = \{o_2, o_3, o_4, o_5, o_6\} \ \text{ and } \ \mathcal{A}_3 = \{o_1, o_2, o_5, o_6\}$$

satisfy the condition of M_d. Thus

$$M_d = \{\{o_1, o_3, o_4, o_5, o_6\}, \{o_2, o_3, o_4, o_5, o_6\}, \{o_1, o_2, o_5, o_6\}\}.$$

3. Calculate $G = \cap_{\mathcal{A}_i \in M_d} \mathcal{A}_i$:

$$G = \{o_1, o_3, o_4, o_5, o_6\} \cap \{o_2, o_3, o_4, o_5, o_6\} \cap \{o_1, o_2, o_5, o_6\} = \{o_5, o_6\}.$$

4. Calculate

$$\text{REAT}(DT) = C \backslash G = \{o_1, o_2, o_3, o_4, o_5, o_6\} \setminus \{o_5, o_6\} = \{o_1, o_2, o_3, o_4\}.$$

Thus attribute reductions in Table 2 is $REAT(DT) = \{o_1, o_2, o_3, o_4\}$.

5 Conclusions

In the context of growing on the current data volume, it is urgent to research and propose efficient algorithms for eliminating redundant attributes to improve the efficiency of data mining or machine learning models. Up to now, there have been many heuristic algorithms to find reducts of incomplete decision tables. However, the research results related to conditional attribute and reduct are still limited. In this paper, we discovered some properties of conditional attribute and proposed an algorithm to determine all reductive attributes of consistent incomplete decision tables in polynomial time. The proposed algorithm is an effective tool to eliminate all redundant attributes in incomplete decision tables before performing attribute reduction and rule extraction algorithms in data mining and machine learning. Further research is to study more properties on reducts to propose more efficient attribute reduction models.

References

[1] Z. Pawlak, *Rough Sets: Theoritical Aspets of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[2] J. Demetrovics, N.L. Giang, V.D.Thi, "On Finding All Reducts of Consistent Decision Tables", *Cybernetics and Information Technologies*, 14(4), 2014, pp. 3-10.

[3] J. Demetrovics, V.D. Thi, N.L. Giang, T.H. Duong, "On the Time Complexity of the Problem Related to Reducts of Consistent Decision Tables", *Serdica Journal of Computing*, 9(2), 2015, pp. 167-176.

[4] V.D. Thi, N.L. Giang, "A Method for Extracting Knowledge from Decision Tables in Terms of Functional Dependencies", *Cybernetics and Information Technologies*, 13(1), 2013, pp. 73-82.

[5] V.D. Thi, N.L. Giang, "A Method to Construct Decision Table from Relation Scheme", *Cybernetics and Information Technologies*, 11(3), 2011, pp. 32-41.

[6] N.L. Giang, V.D. Thi, "Some Problems Concerning Condition Attributes and Reducts in Decision Tables", *Proceeding of the fifth National Symposium "Fundamential and Applied Information Technology Research" (FAIR)*, 2011, pp. 142-152.

[7] J. Demetrovics, H.M. Quang, V.D. Thi, N.V. Anh, "An Efficient Method to Reduce the Size of Consistent Decision Tables", *Acta Cybernetica*, 23(4), 2018, pp. 1039-1054.

[8] N.L. Giang, J. Demetrovics, V.D. Thi, P.D. Khoa, "Some Properties Related to Reduct of Consistent Decision Systems", *Cybernetics and Information Technologies*, 21(2), 2021, pp. 3-9.

[9] L.G. Nguyen, H.S. Nguyen, "Searching for Reductive Attributes in Decision Tables", *Transactions on Rough Sets XIX*, Lecture Notes in Computer Science, Springer, 2015, pp. 51-64.

[10] M. Kryszkiewicz, "Rough set approach to incomplete information systems", *Information Sciences*, 112, 1998, pp. 39-49.

[11] D.T. Khanh, V.D. Thi, N.L. Giang, L.H. Son, "Some Problems Related to Reducts of Consistent Incomplete Decision Tables", *International Journal of Mathematical, Engineering and Management Sciences*, 7(2), 2022, pp. 288-298.

[12] J. Demetrovics, V.D. Thi, N.L. Giang, "An Efficient Algorithm for Determining the Set of All Reductive Attributes in Incomplete Decision Tables", *Cybernetics and Information Technologies*, 13(4), 2013, pp. 118-126.

[13] V.D. Thi, "Minimal keys and antikeys", *Acta Cybernetica*, 7(4), 1986, pp. 361-371.

[14] J. Demetrovics, V.D. Thi, "Some remarks on generating Armstrong and inferring functional dependencies relation", *Acta Cybernetica*, 12(2), 1995, pp. 167-180.

[15] J. Demetrovics, V.D. Thi, "Relations and minimal keys", *Acta Cybernetica*, 8(3), 1988, pp. 279-285.