ISSN No. 0976-5697
Volume 3, No. 7, Nov-Dec 2012
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Independence of Redundant Attributes in the Attribute Reduction Algorithm

Nguyen Duc Thuan
Information Systems Departement
Nha Trang University Nha Trang City,
Khanh Hoa Province, Vietnam
ngducthuan@ntu.edu.vn

Abstract: We proposed an attribute reduction algorithm of decision system. It based on a family covering rough set. In this Algorithm, the independence of redundant attributes is critical to the correctness and complexity of the algorithm. This paper presents removing a redundant attribute does not affect the property of a nonredundant attribute.

Keywords: redundant attribute, attribute reduction, decision system, covering rough sets, consistent decision system.

I. INTRODUCTION

Attribute reduction of an information system is a key problem in rough set theory and its application. It has been proven that finding the minimal reduct of an information system. In [2], Cheng Degang et al. have defined consistent and inconsistent covering decision system and their attribute reduction. They gave an algorithm to compute all the reducts of decision systems. Their method based on discernibility matrix. But, in rough set theory, it has been proved that finding all the reduct of information systems (decision tables) is NP-complete. Hence, sometime we only need to find an attribute reduction. Using some results of Chen Degang et al, we proposed an algorithm which is finding a minimal attribute reduct information decision system [1]. Removing a redundant attributes can affect the property of the remaining properties (e.g. nonredundant attribute X can become redundant attribute, after a redundant attribute Z removed because there are the relationships between the attributes). This paper show the independence of redundant attributes in attribute reduction algorithm based on family covering rough sets.

The remainder of this paper is structured as follows. In section 2 briefly introduces some relevant concepts and results. Section 3, we present our attribute reduction algorithm based on family covering rough sets. Section 4 presents two propositions about the independence of redundant attributes.

II. SOME RELEVANT CONCEPTS AND RESULTS

In this section, we first recall the concept of a cover and then review the existing research on covering rough sets of Cheng Degang et al. [2]

A. Covering rough sets and induced covers:

Definition 2.1 Let U be a universe of discourse, C a family of subsets of U. C is called a cover of U if no subset in C is empty and $\cup$C = U.

Definition 2.2 Let C = {C_1, C_2..., C_n} be a cover of U. For every x$\in$U, let $C_x = \cap\{C_j: C_j \in C, x\in C_j\}$. Cov(C) = {$C_x$: x$\in$U} is then also a cover of U. We call it induced over of C.

Definition 2.3 Let Δ= {C_i: i=1, m} be a family of covers of U. For every x$\in$U, let $\Delta x = \cap\{C_{ix}: C_{ix}\in \text{Cov}(C_i), x\in C_{ix}\}$ then Cov (Δ) = {Δ_x: x$\in$U} is also a cover of U. We call it the induced cover of Δ.

Clearly Δ_x is the intersection of all the elements in every C_i including x, so for every x$\in$U, Δ_x is the minimal set in Cov(Δ) including x. If every cover in Δ is an attribute, then $\Delta_x= \cap\{C_{ix}: C_{ix}\in\text{Cov}(C_i), x\in C_{ix}\}$ means the relation among C_{ix} is a conjunction. Cov(Δ) can be viewed as the intersection of covers in Δ. If every cover in Δ is a partition, then Cov(Δ) is also a partition and Δ_x is the equivalence class including x. For every x, y $\in$ U, if y $\in$ Δ_x, then $\Delta_x \supseteq \Delta_y$, so if y $\in\Delta_x$ and x $\in\Delta_y$, then Δ_x=Δ_y. Every element in Cov(Δ) can not be written as the union of other elements in Cov(Δ). We employ an example to illustrate the practical meaning of C_x and Δ_x.

For every X $\subseteq$ U, the lower and upper approximation of X with respect to Cov(Δ) are defined as follows:

$$\underline{\Delta}(X) = \cup\{\Delta_x : \Delta_x \subseteq X\},$$

$$\overline{\Delta}(X) = \cup\{\Delta_x : \Delta_x \cap X \neq \varnothing\}$$

The positive, negative and boundary regions of X relative to Δ are computed using the following formulas respectively:

$$POS_\Delta(X) = \underline{\Delta}(X), NEG_\Delta(U - \overline{\Delta}(X),$$

$$BN_\Delta(X) = \overline{\Delta}(X) - \underline{\Delta}(X)$$

Clearly in Cov(Δ), Δ_x is the minimal description of object x.

B. Attribute reduction of consistent and inconsistent decision systems:

Definition 2.4 Let Δ = {C_i: i=1,..m} be a family of covers of U, D is a decision attribute, U/D is a decision partition on U. If for $\forall$x$\in$U, $\exists D_j \in$U/D such that $\Delta x \subseteq D_j$, then decision system (U,Δ,D) is called a consistent covering decision system, and denoted as Cov(Δ)$\leq$ U/D. Otherwise, (U,Δ,D) is called an inconsistent covering decision system. The positive region of D relative to Δ is defined as

$$POS_\Delta(D) = \bigcup_{X\in U/D} \underline{\Delta}(X)$$

Remark 2.1 Let D={d}, then d(x) is a decision function d: U $\rightarrow$ Vd of the universe U into value set Vd. For every x_i, $x_j \in U$, if $\Delta_{xi} \subseteq \Delta_{xj}$, then $d(x_i) = d([x_i]_D) = d(\Delta_{xi}) = d(\Delta_{xj}) = d(x_j) = d([x_j]_D)$. If $d(\Delta_{xi}) \neq d(\Delta_{xj})$, then $\Delta_{xi} \cap \Delta_{xj} = \emptyset$, i.e $\Delta_{xi} \not\subset \Delta_{xj}$ and $\Delta_{xj} \not\subset \Delta_{xi}$.

Definition 2.5 Let (U,Δ, D= {d}) be a consistent covering decision system. For $C_i \in \Delta$, if Cov(Δ-{C_i}) $\leq$ U/D, then C_i is called superfluous relative to D in Δ , otherwise C_i is called indispensable relative to D in Δ. For every P $\subseteq \Delta$ satisfying Cov(P) $\leq$U/D , if every element in P is indispensable, i.e., for every $C_i \in$P, Cov(Δ-{C_i}) $\leq$ U/D is not true, then P is called a reduct of D relative to D, relative reduct in short. The collection of all the indispensable elements in D is called the core of Δ relative to D, denoted as CoreD(Δ). The relative reduct of a consistent covering decision system is the minimal set of conditional covers (attributes) to ensure every decision rule still consistent. For a single cover C_i, we present some equivalence conditions to judge whether it is indispensable.

Definition 2.6 Suppose U is a finite universe and Δ = {C_i: i=1,..m} be a family of covers of U, $C_i \in \Delta$, D is a decision attribute relative Δ on U and d: U $\rightarrow V_d$ is the decision function V_d defined as d(x) = $[x]_D$. (U,Δ,D) is an inconsistent covering decision system, i.e., $POS_\Delta(D) \neq U$. If $POS_\Delta(D)=POS_{\Delta\text{-}\{Ci\}}(D)$, then C_i is superfluous relative to D in Δ. Otherwise C_i is indispensable relative to D in Δ. For every P$\subseteq\Delta$, if every element in P is indispensable relative to D, and $POS_\Delta(D)=POS_P(D)$, then P is a reduct of $POS_\Delta(D)=POS_{\Delta\text{-}\{Ci\}}(D)$ relative to D, called relative reduct in short. The collection of all the indispensable elements relative to D in Δ is the core of Δ relative to D, denoted by $Core_D(\Delta)$.

C. Some results of Chang et al:

Theorem 2.1 ([2]) Supposing U is a finite universe and Δ = {C_i: i=1,..m} be a family of covers of U, the following statements hold:

a. $\Delta_x = \Delta_y$ if and only if for every $C_i \in \Delta$ we have $C_{ix} = C_{iy}$.
b. $\Delta_x \supset \Delta_y$ if and only if for every $C_i \in \Delta$ we have $C_{ix} \supseteq C_{iy}$ and there is a $C_i \in \Delta$ such that $C_{i0\,x} \supset C_{i0\,y}$.
c. $\Delta_x \not\subset \Delta_y$ and $\Delta_y \not\subset \Delta_x$ hold if and only if there are C_i, $C_j \in \Delta$ such that $C_{ix} \subset C_{iy}$ and $C_{jx} \supset C_{jy}$ or there is a $C_{i0} \in \Delta$ such that $C_{i0\,x} \not\subset C_{i0\,y}$ and $C_{i0\,y} \not\subset C_{i0\,x}$.

Theorem 2.2 ([2]) Suppose Cov(Δ)$\leq$ U/D, $C_i \in \Delta$, C_i is then indispensable, i.e., Cov(Δ-{C_i}) $\leq$ U/D is not true if and only if there is at least a pair of x_i, $x_j \in U$ satisfying $d(D_{xi}) \neq d(D_{xj})$, of which the original relation with respect to Δ changes after C_i is deleted from Δ.

Theorem 2.3 ([2]) Suppose Cov(Δ) $\leq$ U/D,P $\subseteq \Delta$, then Cov(P) $\leq$ U/D if and only if for x_i, $x_j \in U$ satisfying $d(\Delta_{xi}) \neq d(\Delta_{xj})$, the relation between x_i and x_j with respect to Δ is equivalent to their relation with respect to P, i.e., $\Delta_{xi} \not\subset \Delta_{xj}$ and $\Delta_{xj} \not\subset \Delta_{xi} \Leftrightarrow P_{xi} \not\subset P_{xj}$, $P_{xj} \not\subset P_{xi}$.

Theorem 2.4 ([2]) Inconsistent covering decision system (U,Δ,D = {d}) have the following properties:

a. For $\forall x_i \in U$, if $\Delta_{xi} \subset POS_\Delta(D)$, then $\Delta_{xi} \subseteq [x_i]_D$; if $\Delta_{xi} \not\subset POS_\Delta(D)$, then for $\forall x_k \in U$, $\Delta_{xi} \subseteq [x_k]_D$ is not true.
b. For any P$\subseteq\Delta$, $POS_P(D)= POS_\Delta(D)$ if and only if $\underline{P}(X) = \underline{\Delta}(X)$ for $\forall X \in U/D$.
c. For any P$\subseteq\Delta$, $POS_P(D)= POS_\Delta(D)$ if and only if $\forall x_i \in U$, $\Delta xi \subseteq [x_i]_D \Leftrightarrow Pxi \subseteq [x_i]_D$.

III. ALGORITHM OF ATTRIBUTE REDUCTION

In this section, we propose a new algorithm of attribute reduction. Propositions 3.1 and 3.2 are theoretic foundation for our proposing. This algorithm finds an approximately minimal reduct.

A. Two propositions as a base for new algorithm:

Proposition 3.1 Let (U,Δ,D={d}) be a covering decision system. P $\subseteq \Delta$, then we have:

a. (U,Δ,D={d}) is a consistent covering decision system when it holds:
$$\sum_{x\in U}\frac{|\Delta_x \cap [x]_D|}{|\Delta_x|} = |U|$$
b. Suppose Cov(Δ)$\leq$ U/D, $C_i \in \Delta$, C_i is then indispensable, i.e., Cov(Δ-{C_i}) $\leq$ U/D is true if and only if
$$\sum_{xi\in U}\sum_{xj\in U}\left|(\Delta_{xi}\cap\Delta_{xj}\cup(P_{xi}\cap P_{xj})\right|\left|d(\Delta_{xi})-d(\Delta_{xj})\right| = 0$$

Where Cov(Δ-{C_i})={P_x : x$\in$U}, Cov(Δ)= {Δ_x : x $\in$U}

Proof:

a) By define of a consistent covering decision system, clearly for every x$\in$U, $\Delta_x \subseteq [x]_D$ is always true, thus we have
$$|\Delta_x \cap [x]_D| = |\Delta_x|$$
i.e
$$\sum_{x\in U}\frac{|\Delta_x \cap [x]_D|}{|\Delta_x|} = |U|$$

b) Let Cov(Δ-{C_i})={P_x : x$\in$U} = Cov(P), Cov(Δ)= {Δ_x : x $\in$U, by theorem 2.3, P is a reduct or Ci is indispensable, for x_i, $x_j \in U$ satisfying $d(\Delta_{xi}) \neq d(\Delta_{xj})$, the relation between x_i and x_j with respect to Δ is equivalent to their relation with respect to P, i.e., $\Delta_{xi} \not\subset \Delta_{xj}$ and $\Delta_{xj} \not\subset \Delta_{xi} \Leftrightarrow P_{xi} \not\subset P_{xj}$, $P_{xj} \not\subset P_{xi}$. Follow remark 2.1, If $d(\Delta_{xi}) \neq d(\Delta_{xj})$, then $\Delta_{xi} \cap \Delta_{xj} = \emptyset$, i.e
$$\left|(\Delta_{xi}\cap\Delta_{xj})\cup(P_{xi}\cap P_{xj})\right| = 0$$

If x_i, $x_j \in U$ satisfying $d(\Delta_{xi}) = d(\Delta_{xj})$ then
$$\left|d(\Delta_{xi})-d(\Delta_{xj})\right| = 0$$

In other words, it holds:
$$\sum_{xi\in U}\sum_{xj\in U}\left|(\Delta_{xi}\cap\Delta_{xj}\cup(P_{xi}\cap P_{xj})\right|\left|d(\Delta)_{xi}-d(\Delta)_{xj}\right| = 0$$

This completes the proof.

Proposition 3.2 Let (U,Δ,D={d}) be an inconsistent covering decision system. P $\subseteq \Delta$, $POS_P(D) = POS_\Delta(D)$ if and only if $\forall$xi$\in$U,
$$\sum_{xi\in U}\left[\left|\frac{\Delta_{xi}\cap[x_i]_D}{\Delta_{xi}}\right| - \left|\frac{P_{xi}\cap[x_i]_D}{P_{xi}}\right|\right] = 0$$

Proof:

By theorem 2.4, from third condition $\forall x_i \in U$, $\Delta_{xi} \subseteq [xi]_D \Leftrightarrow P_{xi} \subseteq [xi]_D$ i.e $\forall x_i \in U$,
$$|\Delta_{xi} \cap [x]_D| = |\Delta_{xi}| \Leftrightarrow |P_{xi} \cap [x]_D| = |P_{xi}|$$
In other words, we have theorem above.

B. *Algorithm of attribute reduction in covering decision system:*

Input: A covering decision system
S= (U,Δ,D={d})
Output: One product RD of Δ.
Method
Step 1: Compute

$$CI = \sum_{x\in U} \frac{|\Delta_x \cap [x]_D|}{|\Delta_x|}$$

Step 2: If CI = |U| {S is a consistent covering decision system} then goto Step 3 else goto Step 5.
Step 3: Compute

$$\Delta_x, d(\Delta_x), \forall x \in U$$

Step 4: Begin
For each $C_i \in \Delta$ do
if

$$\sum_{xi\in U}\sum_{xj\in U} |(\Delta_{xi} \cap \Delta_{xj} \cup (P_{xi} \cap P_{xj})||d(\Delta_{xi}) - d(\Delta_{xj})| = 0$$

{Where Δ - {C_i} = {Px : x∈U}}
then Δ:= Δ - {C_i};
Endfor;
goto Step 6.
End;
Step 5: Begin
For each $C_i \in \Delta$ do
if

$$\sum_{xi\in U}\left[\left|\frac{\Delta_{xi} \cap [x_i]_D}{\Delta_{xi}}\right| - \left|\frac{P_{xi} \cap [x_i]_D}{P_{xi}}\right|\right] = 0$$

then Δ:= Δ - {C_i};
{Where Δ - {C_i} = {Px : x∈U}}
Endfor;
End;
Step 6: RD=Δ; the algorithm terminates.

By using this algorithm, the time complexity to find one reduct is polynomial.

At the first step, the time complexity to compute CI is O(|U|).

At the step 2, the time complexity is O(1).

At the step 3, the time complexity is O(|U|).

At the step 4, the time complexity to compute ∑∑() is O($|U|^2$), from i=1..|Δ|, thus the time complexity of this step is O($|\Delta||U|^2$).

At the step 5, the time complexity is the same as step 4. It is O($|\Delta||U|^2$).

At the step 6, the time complexity is O(1).

Thus the time complexity of this algorithm is O($|\Delta||U|^2$) (Where we ignore the time complexity for computing Δ_{xi}, P_{xi}, i= 1..|Δ|).

IV. ILLUSTRATIVE EXAMPLES

A. *Example for a consistent covering decision system:*

Suppose U = {x_1, x_2, .., x_9}, Δ = {C_i, i=1..4}, and
C_1={{x_1, x_2, x_4, x_5, x_7, x_8},{x_2, x_3, x_5, x_6, x_8, x_9}},
C_2={{x_1, x_2, x_3, x_4, x_5, x_6},{x_4, x_5, x_6, x_7, x_8, x_9}},
C_3={{x_1, x_2, x_3},{x_4, x_5, x_6, x_7, x_8, x_9},{x_8, x_9}},
C_4={{x_1, x_2, x_4, x_5},{x_2, x_3, x_5, x_6},{x_7, x_8},{x_5, x_6, x_8, x_9}}
U/D={{x_1, x_2, x_3}, {x_4, x_5, x_6}, {x_7, x_8, x_9}}

where, Δ_i=Δ_{xi}, P_i is P_{xi} (for short)
Step 1:
Δ_1={x_1, x_2}, Δ_2={x_2}, Δ_3={x_2, x_3},
we have d(Δ_1) = d(Δ_2) = d(Δ_3) = 1,
because Δ_1, Δ_2, Δ_3 ⊆ {x_1, x2, x3},
Δ_4={x_4, x_5}, Δ5={x_5}, Δ6={x_5, x_6},
we have d(Δ_4) = d(Δ_5) = d(Δ_6) = 2,
because Δ_4, Δ_5, Δ_6 ⊆ {x_4, x_5, x_6},
Δ_7={x_7, x_8}, Δ_8={x_8}, Δ_9={x_8, x_9},
we have d(Δ_7) = d(Δ_8) = d(Δ_9) = 3,
because Δ_7, Δ_8, Δ_9 ⊆ {x_7,x_8, x_9}
CI = 9 ⇒ S is consistent system.
Step 2:
P - {C_1}:
P_1={x_1, x_2}, P_2={x_2}, P_3={x_2, x_3},
P_4={x_4,x_5}, P_5={x_5}, P_6={x_5, x_6},
P_7={x_7, x_8}, P_8={x_8}, P_9={x_8, x_9}

$$\sum_{xi\in U}\sum_{xj\in U} |(\Delta_{xi} \cap \Delta_{xj} \cup (P_{xi} \cap P_{xj})||d(\Delta_{xi}) - d(\Delta_{xj})| = 0$$

Δ=Δ - {C_1} = {C_2, C_3, C_4}.
Step 3:
P=Δ - {C_2}
P_1={x_1, x_2}, P_2={x_2}, P_3={x_2, x_3},
P_4={x_4, x_5}, P_5={x_5}, P_6={x_5, x_6},
P_7={x_7, x_8}, P_8={x_8}, P_9={x_8, x_9}

$$\sum_{xi\in U}\sum_{xj\in U} |(\Delta_{xi} \cap \Delta_{xj} \cup (P_{xi} \cap P_{xj})||d(\Delta_{xi}) - d(\Delta_{xj})| = 0$$

Δ=Δ - {C_2} = {C_3, C_4}
Step 4:
P= Δ - {C_3}:
P_1={x_1, x_2, x_4, x_5}, P_2={x_2}, P_3={x_2, x_3, x_5, x_6},
P_4={x_4, x_5}, P_5={x_5}, P_6={x_5, x_6},
P_7={x_4, x_5, x_7, x_8}, P_8={x_5, x_8}, P_9={x_5, x_6, x_8, x_9}

$$\sum_{xi\in U}\sum_{xj\in U} |(\Delta_{xi} \cap \Delta_{xj} \cup (P_{xi} \cap P_{xj})||d(\Delta_{xi}) - d(\Delta_{xj})| \neq 0$$

(we can see (Δ_1∩Δ_4)=∅, but (P_1∩P_4)≠∅, |d(Δ_1)-d(Δ_4)|≠0)
Δ= {C_3,C_4}.
Step 5:
P= Δ - {C_4}
P_1={x_1, x_2, x_3}, P_2={x_1, x_2, x_3}, P_3={x_1, x_2, x_3},
P_4={x_4, x_5, x_6, x_7, x_8, x_9}, P_5={x_4, x_5, x_6, x_7, x_8, x_9}, P_6={x_4,x_5,x_6,x_7,x_8,x_9}
P_7={x_7, x_8, x_9}, P_8={x_7, x_8, x_9}, P_9={x_7, x_8, x_9}

$$\sum_{xi\in U}\sum_{xj\in U} |(\Delta_{xi} \cap \Delta_{xj} \cup (P_{xi} \cap P_{xj})||d(\Delta_{xi}) - d(\Delta_{xj})| \neq 0$$

(we can see (Δ_6∩Δ_7) =∅, but (P_6∩P_7)≠∅, |d(Δ_6)-d(Δ_7)|≠0)
Δ= {C_3,C_4}.
Step 6:
RD= {C_3,C_4} is a reduct. i.e. attributes with respect to C_1, C_2 are deleted.

B. *Example for a inconsistent covering decision system:*

Suppose U={x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10}} and {C_i, i=1..4}
C_1={{x_1,x_2,x_3,x_4,x_6,x_7,x_8,x_9,x_{10}},{x_3,x_4,x_6,x_7},{x_3,x_4,x_5,x_6,x_7}}
C_2={{x_1,x_2,x_3,x_4,x_5,x_6,x_7},{x_6,x_7,x_8,x_9},{x_{10}}}
C_3={{x_1,x_2,x_3,x_6,x_8,x_9,x_{10}},{x_2,x_3,x_4,x_5,x_6,x_7,x_9}}
C_4={{x_1,x_2,x_3,x_6},{x_2,x_3,x_4,x_5,x_6,x_7},{x_6,x_8,x_9,x_{10}},{x_6,x_7,x_9}}

U/D={{x_1,x_2,x_3,x_6}, {x_4,x_5,x_7}, {x_8,x_9,x_{10}}}

Step 1:

Δ_1={ x_1,x_2,x_3,x_6}; Δ_2={ x_2,x_3,x_6}; Δ_3={ x_3,x_6};
Δ_4={ x_3,x_4,x_6,x_7}; Δ_5={ x_3,x_4,x_5,x_6,x_7};Δ_6={ x_6};
Δ_7={ x_6,x_7}; Δ_8={ x_6,x_8,x_9}; Δ_9={ x_6,x_9}; Δ_{10}={ x_{10}};

CI $\neq$ 9 $\Rightarrow$ S is an inconsistent system.

Step 2: P – {C_1}:

P_1={x_1,x_2,x_3,x_6}; P_2=P_3={x_2,x_3,x_6};
P_4=P_5={ x_2,x_3,x_4,x_5,x_6,x_7};
P_6={ x_6}; P_7={ x_6,x_7}; P_8={ x_6,x_8,x_9};
P_9= { x_6,x_9}; P_{10}={ x_{10}};

$$\sum_{xi\in U}\left[\left|\frac{\Delta_{xi}\cap[x_i]_D}{\Delta_{xi}}\right|-\left|\frac{P_{xi}\cap[x_i]_D}{P_{xi}}\right|\right]=0$$

Δ=Δ - {C_1}={C_2,C_3,C_4}. C_1 is dispensable.

Step 3: P – {C_2}

P_1={x_1,x_2,x_3,x_6}; P_2=P_3={x_2,x_3,x_6};
P_4=P_5={ x_2,x_3,x_4,x_5,x_6,x_7}; P_6={x_6};
P_7={x_2,x_3,x_4,x_5,x_6,x_7}; P_8={x_6,x_8,x_9, x_{10}};
P_9= { x_6,x_9}; P_{10}={ x_6,x_8,x_9, x_{10}}

$$\sum_{xi\in U}\left[\frac{|\Delta_{xi}\cap[x_i]_D|}{|\Delta_{xi}|}-\frac{|P_{xi}\cap[x_i]_D|}{|P_{xi}|}\right]\neq 0$$

C_2 is in dispensable. Δ={C_2,C_3,C_4}.

Step 4: P – {C_3}

P_1={ x_1,x_2,x_3,x_6}; P_2=P_3={ x_2,x_3,x_6};
P_4=P_5={x_2,x_3,x_4,x_5,x_6,x_7}; P_6={ x_6};
P_7={x_6,x_7}; P_8=P_9= {x_6,x_8,x_9 }; P_{10}={ x_{10}}

$$\sum_{xi\in U}\left[\left|\frac{\Delta_{xi}\cap[x_i]_D}{\Delta_{xi}}\right|-\left|\frac{P_{xi}\cap[x_i]_D}{P_{xi}}\right|\right]=0$$

Δ=Δ - {C_3}={C_2,C_4}. C_3 is dispensable

Step 5: P – {C_4}

P_1= P_2=P_3= P_4=P_5={ x_1, x_2,x_3,x_4,x_5,x_6,x_7}
P_6= P_7={ x_6,x_7}; P_8=P_9= { x_6, x_7,x_8,x_9 }; P_{10}={ x_{10}}

$$\sum_{xi\in U}\left[\frac{|\Delta_{xi}\cap[x_i]_D|}{|\Delta_{xi}|}-\frac{|P_{xi}\cap[x_i]_D|}{|P_{xi}|}\right]\neq 0$$

C_4 is in dispensable. Δ={C_2,C_4}.

Step 6:

RD= {C_2,C_4} is a reduct. i.e. attributes with respect to C_1, C_3 are deleted.

Table I. Comparision with results of Chen Degang et al

Algorithm of Chen Degang et al	New Algorithm
Example 1	
Red(Δ) = {{C_3, C_4}, {C_2, C_3}}	RD= {C_3,C_4}
Example 2	
Red(Δ) = {{C_2, C_4}, {C_2, C_3}}	RD= {C_2,C_4}

Note: Where Red(Δ) = Collection all reducts of Δ; RD is a reduct of Δ

V. INDEPENDENCE OF REDUNDANT ATTRIBUTES

In this section, we show the independence of redundant attributes in the algorithms above. This property is presented through problem:

Is there a conversion of a nonredundant covering into a redundant covering when a redundant covering removed?

We have two propositions:

Proposition 4.1 Let T= (U,Δ,D={d}) be a consistent covering decision system has

a. is a family covering Δ={C_1, C_2, .., C_n}
b. Cov(Δ)$\leq$U/D

Consider 2 family covering P^1, P^2 statisfy:

$P^2\subseteq P^1\subseteq\Delta$, Cov($P^i$)$\leq$U/D, i=1,2.

Then $\forall C_k\in P^2\subseteq P^1$, if C_k is nonredundant in P^1 then C_k is nonredundant in P^2 (*)

Proof:

We need to prove that if C_k is redundant in P^2 then C_k is redundant in P^1.

Let P^{11} = P^1-{C_k}, P^{22} = P^2-{C_k}.

Suppose C_k is nonredundant in P^1 , then Cov(P^1-{C_k})$\leq$U/D is not true.

Cov(P^1-{C_k})$\leq$U/D is not true $\Leftrightarrow$ $\exists x_{i0},x_{j0}\in$U such that $d(P^1x_{i0})\neq d(P^1x_{j0})$, $P^1x_{i0}\cap P^1x_{j0}=\varnothing$ but $P^{11}x_{i0}\cap P^{11}x_{j0}\neq\varnothing$

If C_k is redundant in P^2 $\Leftrightarrow$ Cov(P^2-{C_k})$\leq$U/D. $\Leftrightarrow$ $\forall x_i,x_j\in$U, $d(P^2x_i)\neq d(P^2x_j)$, we get $P^2x_i\cap P^2x_j=\varnothing$ and $P^{22}x_i\cap P^{22}x_j=\varnothing$

Since $P^2\subseteq P^1\subseteq\Delta$, Cov($P^i$)$\leq$U/D, i=1,2, so $\forall x_i,x_j\in$U, $d(P^1x_i)\neq d(P^1x_j)$ implies that $P^1x_i\cap P^1x_j=\varnothing$ and $P^2x_i\cap P^2x_j=\varnothing$

Clearly, $P^2\subseteq P^1$ implies that $\forall x_i\in$U, $\forall C_k\in P^2\subseteq P^1$: $P^{11}x_i\subseteq P^{22}x_i$,

Combining (1)(2)(3)(4) gives a contradiction: $\varnothing\neq P^{11}x_{i0}\cap P^{11}x_{j0}\subseteq P^{22}x_{i0}\cap P^{22}x_{j0}=\varnothing$.

In other words, we have (*). The proof is complete.

Proposition 4.2 Let T= (U,Δ,D={d}) be an inconsistent covering decision system has

a. Δ is a family covering Δ={C_1, C_2, .., C_n}
b. $POS_\Delta(D)\neq U$

Consider 2 family covering P^1, $P^2\subseteq\Delta$ statisfy:

a) $P^2\subseteq P^1$
b) $POS_{P^1}(D)=POS_{P^2}(D)\neq U$

Then $\forall C_k\in P^2\subseteq P^1$, if C_k is nonredundant in P^1 then C_k is nonredundant in P^2 (*)

Proof:

In the same way as in Proposition 4.1, we need to prove that if C_k redundant in P^2 then C_k redundant in P^1.

Let P^{11}=P^1-{C_k}, P^{22}=P^2-{C_k}. If C_k is redundant in P^2 then

$$POS_{P^{22}}(D)=POS_{P^2}(D)$$

$$\Leftrightarrow\forall x_i\in U:P^{22}_{x_i}\subseteq[x_i]_D\Leftrightarrow P^2_{x_i}\subseteq[x_i]_D$$

Suppose C_k is nonredundant in P^1 , we have :

$$POS_{P^{11}}(D)\neq POS_{P^1}(D)$$

$$\Leftrightarrow\exists x_0\in U:P^{11}_{x_0}\not\subset[x_0]_D \text{ và } P^1_{x_0}\subseteq[x_0]_D$$

Since $POS_{P^1}(D)=POS_{P^2}(D)\neq U$, it follows that

$$P^1_{x_0}\subseteq[x_0]_D\Leftrightarrow P^2_{x_0}\subseteq[x_0]_D$$

By (α)(β)(γ), we get

$$\exists x_0\in U:P^1_{x_0}\subseteq[x_0]_D,P^2_{x_0}\subseteq[x_0]_D,P^{22}_{x_0}\subseteq[x_0]_D,P^{11}_{x_0}\not\subset[x_0]_D$$

Since $P^{22}\subseteq P^{11}$, it follows that $P^{11}_{xo}\subseteq P^{22}_{x0}$ which contradicts with

$$P^{11}_{x_0}\not\subset[x_0]_D,P^{22}_{x_0}\subseteq[x_0]_D$$

In other words, we have (*). The proof is complete.

VI. CONCLUSION

Independence of redundant attributes in the Attribute reduction algorithm based on a family covering rough sets allows we process only one time to remove redundant attributes. This determines the performance of the algorithm above.

VII. REFERENCES

[1] Nguyen Duc Thuan, “A family of covering rough sets based algorithm for reduction attributes”, International journal of Computer Theory and Engineering, vol.2, No.2, pp.181–184, April 2010.

[2] Chen Degang, Wang Changzhong, Hu Quinghua, “A new approach to attribute reduction of consistent and inconsistent covering rough sets”, Information Sciences, 177 (2007) 3500-3518

[3] Guo-Yin Wang, Jun Zhao, Jiu-Jiang An, Yu Wu, “Theoretical study on attribute reduction of rough set theory: Comparision of algebra and information views”, Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI’04), 2004

[4] Ruizhi Wang, Duoqian Miao, Guirong Hu, “Descernibility Matrix Based Algorithm for Reduction of Attributes”, IAT Workshops 2006: 477- 480

[5] Yiyu Yao, Yan Zhao, Jue Wang, “On Reduct Construction Algorithms, Rough sets and Knowledge Technology”, First International Conference, RSKT2006, Proceedings, LNAI 4062, pp 207-304, 2006