Vietnam Journal of Science and Technology ***57*** *(4) (2019) 499-512*

doi:10.15625/2525-2518/57/4/13773

A DISTANCE BASED INCREMENTAL FILTER-WRAPPER ALGORITHM FOR FINDING REDUCT IN INCOMPLETE DECISION TABLES

Nguyen Ba Quang[1], Nguyen Long Giang[2, *], Dang Thi Oanh[3]

[1]*Hanoi Architectural University, Km 10 Nguyen Trai, Thanh Xuan, Ha Noi*

[2]*Institute of Information Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet, Cau Giay, Ha Noi*

[3]*University of Information and Communication Technology, Thai Nguyen University, Z115 Quyet Thang, Thai Nguyen*

[*]Email: *nlgiang@ioit.ac.vn*

Received: 21 April 2019; Accepted for publication: 9 May 2019

Abstract. Tolerance rough set model is an effective tool for attribute reduction in incomplete decision tables. In recent years, some incremental algorithms have been proposed to find reduct of dynamic incomplete decision tables in order to reduce computation time. However, they are classical filter algorithms, in which the classification accuracy of decision tables is computed after obtaining reduct. Therefore, the obtained reducts of these algorithms are not optimal on cardinality of reduct and classification accuracy. In this paper, we propose an incremental filter-wrapper algorithm to find one reduct of an incomplete desision table in case of adding multiple objects. The experimental results on some datasets show that the proposed filter-wrapper algorithm is more effective than some filter algorithms on classification accuracy and cardinality of reduct

Keywords: Tolerance rough set, distance, incremental algorithm, incomplete decision table, attribute reduction, reduct.

Classification numbers: 4.7.3, 4.7.4, 4.8.3.

1. INTRODUCTION

Rough set theory has been introduced by Pawlak [1] as an effective tool for solving attribute reduction problem in decision tables. In fact, decision tables often contain missing values for at least one conditional attribute and these decision tables are called incomplete decision tables. To solve attribute reduction problem and extract decision rules directly from incomplete decision tables, Kryszkiewicz [2] has extended the equivalence relation in traditional rough set theory to tolerance relation and proposed tolerance rough set model. Based on tolerance rough set, many attribute reduction algorithms in incomplete decision tables have been investigated. In real-world problems, decision tables often vary dynamically over time. When these decision tables change, traditional attribute reduction algorithms have to re-compute a

reduct from the whole new data set. As a result, these algorithms consume a huge amount of computation time when dealing with dynamic datasets. Therefore, researchers have proposed an incremental technique to update a reduct dynamically to avoid some re-computations. According to classical rough set approach, there are many research works on incremental attribute reduction algorithms in dynamic complete decision tables, which can be categorized along three variations: adding and deleting object set [3-8], adding and deleting conditional attribute set [9, 10], varying attribute values [11-13].

In recent years, some incremental attribute reduction algorithms in incomplete decision tables have been proposed based on tolerance rough set [14- 20]. Zhang *et al*. [16] proposed an incremental algorithm for updating reduct when adding one object. Shu *et al*. [15, 17] constructed incremental mechanisms for updating positive region and developed incremental algorithms when adding and deleting an object set. Yu *et al*. [14] constructed incremental formula for computing information entropy and they proposed incremental algorithms to find one reduct when adding and deleting multiple objects. Shu *el al*. [18] developed positive region based incremental attribute reduction algorithms in the case of adding and deleting a conditional attribute set. Shu *et al*. [19] also developed positive region based incremental attribute reduction algorithms when the values of objects are varying. Xie *et al*. [20] constructed inconsistency degree and proposed incremental algorithms to find reducts based on inconsistency degree with variation of attribute values. The experimental results show that the computation time of the incremental algorithms is much less than that of non-incremental algorithms. However, the above incremental algorithms are all filter algorithms. In this filter algorithms, the obtained reducts are the minimal subset of conditional attributes which keep the original measure. The classification accuracy of decision table is calculated after obtaining reduct. Consequently, the reducts of the filter incremental algorithms are not optimal on the cardinality of reduct and classification accuracy.

In this paper, we propose the incremental filter-wrapper algorithm IDS_IFW_AO to find one reduct of an incomplete decision table based on the distance in [21]. In proposed algorithm IDS_IFW_AO, the filter phase finds candidates for reduct when adding the most important attribute, the wrapper phase finds the reduct with the highest classification accuracy. The experimental results on sample datasets [22] show that the classification accuracy of IDS_IFW_AO is higher than that of the incremental filter algorithm IARM-I [15]. Furthermore, the cardinality of reduct of IDS_IFW_AO is much less than that of IARM-I. The rest of this paper is organized as follows. Section 2 presents some basic concepts. Section 3 constructs incremental formulas for computing distance when adding multiple objects. Section 4 proposes an incremental filter-wrapper algorithm to find one reduct. The experimental results of proposed algorithm are present in Section 5. Some conclusions and further research are drawn in Section 6.

2. PRELIMINARY

In this section, we present some basic concepts related to tolerance rough set model proposed by Kryszkiewicz [2].

A decision table is a pair $DS = (U, C \cup \{d\})$ where U is a finite, non-empty set of objects; C is a finite, non-empty set of conditional attribute; d is a decision attribute, $d \notin C$. Each attribute $a \in C$ determines a mapping: $a : U \to V_a$ where V_a is the value set of attribute $a \in C$. If V_a contains a missing value then DS is called as incomplete decision table, otherwise DS is

complete decision table. Furthermore, we will denote the missing value by '*'. Analogically, an incomplete decision table is denoted as $IDS = (U, C \cup \{d\})$ where $d \notin C$ and $'*' \notin V_d$.

Let us consider an incomplete decision table $IDS = (U, C \cup \{d\})$, for any subset $P \subseteq C$, we define a binary relation on U as follows:

$$SIM(P) = \{(u,v) \in U \times U \mid \forall a \in P, a(u) = a(v) \vee a(u) = '*' \vee a(v) = '*'\}$$

where $a(u)$ is the value of attribute a on object u. $SIM(P)$ is a tolerance relation on U as it is reflective, symmetrical but not transitive. It is easy to see that $SIM(P) = \cap_{a \in P} SIM(\{a\})$. For any $u \in U$, $S_P(u) = \{v \in U \mid (u,v) \in SIM(P)\}$ is called a tolerance class of object *u*. $S_P(u)$ is a set of objects which are indiscernibility with respect to u on tolerance relation $SIM(P)$. In special case, if $P = \varnothing$ then $S_\varnothing(u) = U$. For any $P \subseteq C$, $X \subseteq U$, P-*lower approximation* of X is $\underline{P}X = \{u \in U \mid S_P(u) \subseteq X\} = \{u \in X \mid S_P(u) \subseteq X\}$, P-*upper approximation* of X is $\overline{P}X = \{u \in U \mid S_P(u) \cap X \neq \varnothing\} = \cup\{S_P(u) \mid u \in U\}$, B-*Boundary region* of X is $BN_P(X) = \overline{P}X - \underline{P}X$. Then, $\langle \underline{P}X, \overline{P}X \rangle$ is called the tolerance rough set. For such approximation set, P-*positive region* with respect to D is defined as $POS_P(\{d\}) = \bigcup_{X \in U/\{d\}} (\underline{P}X)$.

Let us consider the incomplete decision table $IDS = (U, C \cup \{d\})$. For $P \subseteq C$ and $u \in U$, $\partial_P(u) = \{d(v) \mid v \in S_P(u)\}$ is called generalized decision in IDS. If $|\partial_C(u)| = 1$ for any $u \in U$ then IDS is consistent, otherwise it is inconsistent. According to the concept of positive region, IDS is consistent if and only if $POS_C(\{d\}) = U$, otherwise it is inconsistent.

Definition 1. Given an incomplete decision table $IDS = (U, C \cup D)$ where $U = \{u_1, u_2, ..., u_n\}$ and $P \subseteq C$. Then, the tolerance matrix of the relation $SIM(P)$, denoted by $M(P) = [p_{ij}]_{n \times n}$, is defined as

$$M(P) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

in which $p_{ij} \in \{0,1\}$. $p_{ij} = 1$ if $u_j \in S_P(u_i)$ and $p_{ij} = 0$ if $u_j \notin S_P(u_i)$ for $i, j = 1..n$

According to the representation of the tolerance relation $SIM(P)$ by the tolerance matrix $M(P)$, for any $u_i \in U$ we have $S_P(u_i) = \{u_j \in U \mid p_{ij} = 1\}$ and $|S_P(u_i)| = \sum_{j=1}^{n} p_{ij}$. It is easy to see that $S_{P \cup Q}(u) = S_P(u) \cap S_Q(u)$ for any $P, Q \subseteq C, u \in U$. Assuming that $M(P) = [p_{ij}]_{n \times n}$, $M(Q) = [q_{ij}]_{n \times n}$ are two tolerance matrices of $SIM(P)$, $SIM(Q)$ respectively, then the tolerance matrix on the attribute set $S = P \cup Q$ is defined as $M(S) = M(P \cup Q) = [s_{ij}]_{n \times n}$ where $s_{ij} = p_{ij} \cdot q_{ij}$.

Let us consider the incomplete decision table $IDS=(U,C\cup D)$ where $U=\{u_1,u_2,...,u_n\}$, $P\subseteq C$, $X\subseteq U$. Suppose that the object set X is represented by a one-dimensional vector $X=(x_1,x_2,...,x_n)$ where $x_i=1$ if $u_i\in X$ and $x_i=0$ if $u_i\notin X$. Then, $\underline{P}X=\{u_i\in U\,|\,p_{ij}\le x_j,\ j=1..n\}$ and $\overline{P}X=\{u_i\in U\,|\,p_{ij}.x_j\neq\varnothing,\ j=1..n\}$.

3. INCREMENTAL METHOD FOR UPDATING DISTANCE WHEN ADDING MULTIPLE OBJECTS

In [21], the authors have built a distance measure on attribute sets in incomplete decision tables. This section incrementally computes the distance measure in [21] when adding a single object and multiple objects. By using this incremental formulas, an incremental algorithm to find one reduct will be developed in Section IV.

Given an incomplete decision table $IDS=(U,C\cup\{d\})$ where $U=\{u_1,u_2,...,u_n\}$ Then the distance between C and $C\cup\{d\}$ is defined as [21]

$$D(C,C\cup\{d\})=\frac{1}{n^2}\sum_{i=1}^{n}\left(|S_C(u_i)|-|S_C(u_i)\cap S_{\{d\}}(u_i)|\right) \tag{3.1}$$

Assuming that $M(C)=[c_{ij}]_{n\times n}$, $M(\{d\})=[d_{ij}]_{n\times n}$ are tolerance matrices on C and d respectively. Then the distance is computed as:

$$D(C,C\cup\{d\})=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}-c_{ij}.d_{ij})$$

3.1. Incremental method for updating distance when adding a single object

Proposition 1. *Given an incomplete decision table* $IDS=(U,C\cup\{d\})$ *where* $U=\{u_1,u_2,...,u_n\}$. *Suppose that a new object u is added into* U. *Let* $M_{U\cup\{u\}}(C)=[c_{i,j}]_{(n+1)\times(n+1)}$ *and* $M_{U\cup\{u\}}(\{d\})=[d_{i,j}]_{(n+1)\times(n+1)}$ *be tolerance matrices on C and {d} respectively, where* $S_C(u)=\{u_j\in U\,|\,c_{n+1,j}=1\}$. *Then, the incremental formula to compute the distance is :*

$$D_{U\cup\{u\}}(C,C\cup\{d\})=\left(\frac{n}{n+1}\right)^2.D_U(C,C\cup\{d\})+\frac{2}{(n+1)^2}.\left(\sum_{i=1}^{n+1}(c_{n+1,i}-c_{n+1,i}.d_{n+1,i})\right)$$

Proof. We have $D_{U\cup\{u\}}(C,C\cup\{d\})=$

$$=\frac{1}{(n+1)^2}.\left(\sum_{i=1}^{n+1}(c_{1,i}-c_{1,i}.d_{1,i})+...+\sum_{i=1}^{n+1}(c_{n,i}-c_{n,i}.d_{n,i})+\left(|S_C(u)|-|S_C(u)\cap S_{\{d\}}(u)|\right)\right)$$

$$=\frac{1}{(n+1)^2}.\left(\sum_{i=1}^{n}(c_{1,i}-c_{1,i}.d_{1,i})+...+\sum_{i=1}^{n}(c_{n,i}-c_{n,i}.d_{n,i})+\left(|S_C(u)|-|S_C(u)\cap S_{\{d\}}(u)|\right)+\right.$$

$$\left.+(c_{1,n+1}-c_{1,n+1}.d_{1,n+1})+...+(c_{n,n+1}-c_{n,n+1}.d_{n,n+1})\right)$$

$$= \frac{1}{(n+1)^2}.\left(\sum_{i=1}^{n}\left(c_{1,i}-c_{1,i}.d_{1,i}\right)+...+\sum_{i=1}^{n}\left(c_{n,i}-c_{n,i}.d_{n,i}\right)+2.\left(\left|S_C(u)\right|-\left|S_C(u)\cap S_{\{d\}}(u)\right|\right)\right)$$

Otherwise,

$$\left(\sum_{i=1}^{n}\left(c_{1,i}-c_{1,i}.d_{1,i}\right)\right)+...+\left(\sum_{i=1}^{n}\left(c_{n,i}-c_{n,i}.d_{n,i}\right)\right)=\sum_{i=1}^{n}\left(\left|S_C(u)\right|-\left|S_C(u)\cap S_{\{d\}}(u)\right|\right)=n^2.D_U\left(C,C\cup\{d\}\right)$$

Consequently

$$D_{U\cup\{u\}}\left(C,C\cup\{d\}\right)=\left(\frac{n}{n+1}\right)^2.D_U\left(C,C\cup\{d\}\right)+\frac{2}{(n+1)^2}.\left(\sum_{i=1}^{n+1}\left(c_{n+1,i}-c_{n+1,i}.d_{n+1,i}\right)\right).$$

3.2. Incremental method for updating distance when adding multiple objects

Based on Proposition 1, we construct an incremental formula to compute the distance when adding multiple objects by the following Proposition 2.

Proposition 2. *Given an incomplete decision table* $IDS=\left(U,C\cup\{d\}\right)$ *where* $U=\{u_1,u_2,...,u_n\}$. *Assuming that* $\Delta U=\{u_{n+1},u_{n+2},...,u_{n+s}\}$ *is the incremental object set which added into* U *where* $s\geq 2$. *Let* $M_{U\cup\Delta U}(C)=\left[c_{i,j}\right]_{(n+s)\times(n+s)}$ *and* $M_{U\cup\Delta U}(\{d\})=\left[d_{i,j}\right]_{(n+s)\times(n+s)}$ *be the tolerance matrices on C and {d} respectively. Then the incremental formula to compute the distance is:*

$$D_{U\cup\Delta U}\left(C,C\cup\{d\}\right)=\left(\frac{n}{n+s}\right)^2.D_U\left(C,C\cup\{d\}\right)+\frac{2}{(n+s)^2}.\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{i,j}-c_{i,j}.d_{i,j}\right)$$

Proof: Assuming that $D_1,D_2,...,D_s$ are the distances between C and $C\cup\{d\}$ when adding $u_{n+1},u_{n+2},...,u_{n+s}$ into U respectively, and D_0 is the distance between C and $C\cup\{d\}$ on the original object set U. When adding object u_{n+1} into U, we have:

$$D_1=\left(\frac{n}{n+1}\right)^2.D_0+\frac{2}{(n+1)^2}.\left(\sum_{i=1}^{n+1}\left(c_{n+1,i}-c_{n+1,i}.d_{n+1,i}\right)\right)$$

When adding object u_{n+2} into U, we have:

$$D_2=\left(\frac{n+1}{n+2}\right)^2.D_1+\frac{2}{(n+2)^2}.\left(\sum_{i=1}^{n+2}\left(c_{n+2,i}-c_{n+2,i}.d_{n+2,i}\right)\right)$$

$$D_2=\left(\frac{n}{n+2}\right)^2.D_0+\frac{2}{(n+2)^2}.\left(\sum_{i=1}^{n+1}\left(c_{n+1,i}-c_{n+1,i}.d_{n+1,i}\right)\right)+\frac{2}{(n+2)^2}.\left(\sum_{i=1}^{n+2}\left(c_{n+2,i}-c_{n+2,i}.d_{n+2,i}\right)\right)$$

Similarly, when adding object u_{n+s} into U, we have:

$$D_s=\left(\frac{n}{n+s}\right)^2.D_0+\frac{2}{(n+s)^2}.A_s$$

where

$$A_s=\sum_{i=1}^{n+1}\left(c_{n+1,i}-c_{n+1,i}.d_{n+1,i}\right)+\sum_{i=1}^{n+2}\left(c_{n+2,i}-c_{n+2,i}.d_{n+2,i}\right)+...+\sum_{i=1}^{n+s}\left(c_{n+s,i}-c_{n+s,i}.d_{n+s,i}\right)=\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{ij}-c_{ij}.d_{ij}\right)$$

Consequently, we have

$$D_s = \left(\frac{n}{n+s}\right)^2 .D_0 + \frac{2}{(n+s)^2}.\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{ij} - c_{ij}.d_{ij}\right)$$

as the result

$$D_{U\cup\Delta U}\left(C, C\cup\{d\}\right) = \left(\frac{n}{n+s}\right)^2 .D_U\left(C, C\cup\{d\}\right) + \frac{2}{(n+s)^2}.\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{ij} - c_{ij}.d_{ij}\right).$$

4. AN INCREMENTAL FILTER-WRAPPER ALGORITHM TO FIND ONE REDUCT WHEN ADDING MULTIPLE OBJECTS

In [21], authors proposed a distance based filter algorithm to find one reduct of an incomplete decision table. In this approach, the obtained reduct is the minimal attribute set which keeping original distance $D\left(C, C\cup\{d\}\right)$, the evaluation of classification accuracy is performed after finding out reduct. Based on the incremental formula to compute distance in Subsection 3.2, in this section we develop an incremental filter-wrapper algorithm to find one reduct from a dynamic incomplete decision tables when adding multiple objects. In proposed filter-wrapper algorithm, the filter phase finds candidates for reduct when adding the most important attribute, the wrapper phase finds the reduct with the highest classification accuracy. Firstly, we present the definition of reduct and significance of attribute based on distance.

Definition 1. [21] Given an incomplete decision table $IDS = \left(U, C\cup\{d\}\right)$ where $B \subseteq C$. If

1) $D\left(B, B\cup\{d\}\right) = D\left(C, C\cup\{d\}\right)$

2) $\forall b \in B,\ D\left(B-\{b\}, \{B-\{b\}\}\cup\{d\}\right) \neq D\left(C\cup\{d\}\right)$

then B is a reduct of C based on distance.

Definition 2. [21] Given an incomplete decision table $IDS = \left(U, C\cup\{d\}\right)$ where $B \subset C$ and $b \in C - B$. Significance of attribute b with respect to B is defined as

$$SIG_B(b) = D\left(B, B\cup\{d\}\right) - D\left(B\cup\{b\}, B\cup\{b\}\cup\{d\}\right)$$

Significance of attribute $SIG_B(b)$ characterizes the classification quality of attribute *b* with respect to *d* and it is treated as the attribute selection criterion in our heuristic algorithm for attribute reduction.

Proposition 3. *Given an incomplete decision table* $IDS = \left(U, C\cup\{d\}\right)$ *where* $U = \{u_1, u_2, ..., u_n\}$, $B \subseteq C$ *is a reduct of IDS based on distance. Suppose that the incremental object set* $\Delta U = \{u_{n+1}, u_{n+2}, ..., u_{n+s}\}$ *is added into* U *where* $s \geq 1$. *Then we have:*

if $S_B(u_{n+i}) \subseteq S_{\{d\}}(u_{n+i})$ *for any* $i = 1..s$ *then B is a reduct of* $IDS_1 = \left(U\cup\Delta U, C\cup\{d\}\right)$.

Proof. Suppose that $M_{U\cup\Delta U}(C)=\left[c_{i,j}\right]_{(n+s)\times(n+s)}$, $M_{U\cup\Delta U}(B)=\left[b_{i,j}\right]_{(n+s)\times(n+s)}$ are tolerance matrices on C and B of IDS_1 respectively. If $S_B(u_{n+i})\subseteq S_{\{d\}}(u_{n+i})$ for any $i=1..s$ then $S_C(x_{n+i})\subseteq S_B(x_{n+i})\subseteq S_{\{d\}}(x_{n+i})$, then we have:

1) For any $i=n+1..n+s$, $j=1..i$, from $S_B(u_i)\subseteq S_{\{d\}}(u_i)$ we have $b_{i,j}\le d_{i,j}$, or

$b_{i,j}-b_{i,j}.d_{i,j}=b_{i,j}-b_{i,j}=0$. So $\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(b_{i,j}-b_{i,j}.d_{i,j}\right)=0$.

According to Proposition 2 we have

$$D_{U\cup\Delta U}\left(B,B\cup\{d\}\right)=\left(\frac{n}{n+s}\right)^2.D_U\left(B,B\cup\{d\}\right) \qquad (*)$$

2) Similarly, for any $i=n+1..n+s$, $j=1..i$, from $S_C(u_i)\subseteq S_{\{d\}}(u_i)$ we have $c_{i,j}\le d_{i,j}$, or $c_{i,j}-c_{i,j}.d_{i,j}=c_{i,j}-c_{i,j}=0$. So $\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{i,j}-c_{i,j}.d_{i,j}\right)=0$. According to Proposition 2 we have:

$$D_{U\cup\Delta U}\left(C,C\cup\{d\}\right)=\left(\frac{n}{n+s}\right)^2.D_U\left(C,C\cup\{d\}\right) \qquad (**)$$

.

Otherwise, as B is a reduct of IDS,. $D_U\left(B,B\cup\{d\}\right)=D_U\left(C,C\cup\{d\}\right)$ From (*) and (**) we can obtain

$$D_{U\cup\Delta U}\left(B,B\cup\{d\}\right)=D_{U\cup\Delta U}\left(C,C\cup\{d\}\right).$$

Furthermore, $\forall b\in B, D_U\left(\left(B-\{b\}\right),\left(B-\{b\}\right)\cup\{d\}\right)\ne D_U\left(C,C\cup\{d\}\right)$, from (*) and (**) we can obtain $\forall b\in B, D_{U\cup\Delta U}\left(\left(B-\{b\}\right),\left(B-\{b\}\right)\cup\{d\}\right)\ne D_{U\cup\Delta U}\left(C,C\cup\{d\}\right)$. Consequently, B is a reduct of $IDS_1=\left(U\cup\Delta U,C\cup\{d\}\right)$.

Based on Proposition 3, a distance based incremental filter-wrapper algorithm to find one reduct of an incomplete decision table when adding multiple object is described as follows:

Algorithm IDS_IFW_AO

Input: An incomplete decision table $IDS=\left(U,C\cup\{d\}\right)$ where $U=\{u_1,u_2,...,u_n\}$, a reduct $B\subseteq C$, tolerance matrices $M_U(B)=\left[b_{i,j}\right]_{n\times n}$, $M_U(C)=\left[c_{i,j}\right]_{n\times n}$, $M_U(\{d\})=\left[d_{i,j}\right]_{n\times n}$, an incremental object set $\Delta U=\{u_{n+1},u_{n+2},...,u_{n+s}\}$.

Output: A reduct B_{best} of $IDS_1=\left(U\cup\Delta U,C\cup\{d\}\right)$

Step 1: *Initialization*

1. $T:=\varnothing$
2. Compute tolerance matrices on $U\cup\Delta U$:

$M_{U\cup\Delta U}(B)=\left[b_{i,j}\right]_{(n+s)\times(n+s)}$, $M_{U\cup\Delta U}(\{d\})=\left[d_{i,j}\right]_{(n+s)\times(n+s)}$

Step 2: *Check the incremental object set*

3. Set $X := \Delta U$
4. For $i = 1$ to s do
5. If $S_B(u_{n+i}) \preceq S_{\{d\}}(u_{n+i})$ then $X := X - \{u_{n+i}\}$;
6. If $X = \varnothing$ then Return B;
7. Set $\Delta U := X; s := |\Delta U|$;

Step 3: Implement the algorithm to find one reduct

8. Compute original distances $D_U(B, B \cup \{d\}); D_U(C, C \cup \{d\})$
9. Compute distances by incremental formulas $D_{U \cup \Delta U}(B, B \cup \{d\}); D_{U \cup \Delta U}(C, C \cup \{d\})$;

// Filter phase, finding candidates for reduct

10. While $D_{U \cup \Delta U}(B, B \cup \{d\}) \neq D_{U \cup \Delta U}(C, C \cup \{d\})$ do
11. Begin
12. For each $a \in C - B$ do
13. Begin
14. Compute $D_{U \cup \Delta U}(B \cup \{a\}, B \cup \{a\} \cup \{d\})$ by the incremental formula;
15. Compute $SIG_B(a) = D_{U \cup \Delta U}(B, B \cup \{d\}) - D_{U \cup \Delta U}(B \cup \{a\}, B \cup \{a\} \cup \{d\})$
16. End;
17. Select $a \in C - B$ such that $SIG_B(a_m) = \underset{a \in C-B}{Max}\{SIG_B(a)\}$;
18. $B := B \cup \{a_m\}$;
19. $T := T \cup B$;
20. End;

// Wrapper phase, finding the reduct with the highest classification accuracy

21. Set $t := |T|$ // $T = \left\{B \cup \{a_{i_1}\}, B \cup \{a_{i_1}, a_{i_2}\}, ..., B \cup \{a_{i_1}, a_{i_2}, ..., a_{i_t}\}\right\}$;
22. Set $T_1 := B \cup \{a_{i_1}\}; T_2 := B \cup \{a_{i_1}, a_{i_2}\}; ...; T_t := B \cup \{a_{i_1}, a_{i_2}, ..., a_{i_t}\}$
23. For j = 1 to t
24. Begin
25. Compute the classification accuracy on T_j by a classifier based on the 10-fold cross validation;
26. End
27. $B_{best} := T_{jo}$ where T_{jo} has the highest classification accuracy.
28. Return B_{best};

Suppose that $|C|, |U|, |\Delta U|$ are the number of conditional attributes, the number of objects, the number of incremental objects respectively. At command line 2, the time complexity to compute the tolerance matrix $M_{U \cup \Delta U}(B)$ when $M_U(B)$ computed is $O(|\Delta U| * (|U| + |\Delta U|))$. The time complexity of For loop at command line 4 is $O(|\Delta U| * (|U| + |\Delta U|))$. In the best case, the

algorithm finishes at command line 6 (the reduct is not changed). Then, the time complexity of IDS_IFW_AO is $O\left(|\Delta U|*\left(|U|+|\Delta U|\right)\right)$.

Otherwise, let us consider While loop from command line 10 to 20, to compute $SIG_B(a)$ we have to compute $D_{U+\Delta U}\left(B\cup\{a\},B\cup\{a\}\cup\{d\}\right)$ as $D_{U\cup\Delta U}\left(B,B\cup\{d\}\right)$ has already computed in the previous step. The time complexity to compute $D_{U+\Delta U}\left(B\cup\{a\},B\cup\{a\}\cup\{d\}\right)$ is $O\left(|\Delta U|*\left(|U|+|\Delta U|\right)\right)$. Therefore, the time complexity of While loop is $O\left(\left(|C|-|B|\right)^2|\Delta U|*\left(|U|+|\Delta U|\right)\right)$ and the time complexity of filter phase is $O\left(\left(|C|-|B|\right)^2|\Delta U|*\left(|U|+|\Delta U|\right)\right)$. Suppose that the time complexity of the classifier is $O(T)$, then the time complexity of wrapper phase is $O\left(\left(|C|-|B|\right)*T\right)$. Consequently, the time complexity of IDS_IFW_AO is $O\left(\left(|C|-|B|\right)^2*|\Delta U|*\left(|U|+|\Delta U|\right)\right)+O\left(\left(|C|-|B|\right)*T\right)$. If we perform a non-incremental filter-wrapper algorithm on the incomplete decision table with object set $U\cup\Delta U$ directly, the time complexity is $O\left(|C|^2*\left(|U|+|\Delta U|\right)^2\right)+O\left(|C|*T\right)$. As the results, IDS_IFW_AO significantly reduces the time complexity, especially when $|U|$ is large or $|B|$ is large.

5. EXPERIMENTAL ANALYSIS

In this section, some experiments have been conducted to evaluate the efficiency of proposed filter-wrapper incremental algorithm IDS_IFW_AO compared with filter incremental IARM-I [15]. The evaluation was performed on the cardinality of reduct, classification accuracy and runtime. IARM-I [15] is state-of-the-art incremental filter algorithm to find one reduct based on position region when adding multiple objects. The experiments were performed on six missing value data sets from UCI [22] (see Table 1). Each dataset in Table 1 was randomly divided into two parts of approximate equal size: the original dataset (denoted as U_0) and the incremental dataset (see the 4th and 5th columns of Table 1). The incremental dataset was randomly divided into five parts of equal size: U_1,U_2,U_3,U_4,U_5

To conduct experiments two algorithms IDS_IFW_AO, IARM-I [15], firstly we performed two algorithms on the original dataset as incremental data set. Next, we performed two algorithms when adding from the first part (U_1) to the fifth part (U_5) of the incremental dataset. C4.5 classifier was employed to evaluate the classification accuracy based on the 10-fold cross validation. All experiments have been run on a personal computer with Inter(R) Core(TM) 2 i3-2120 CPU, 3.3 GHz and 4 GB memory.

The cardinality of reduct (denoted as $|R|$) and the classification accuracy (denoted as Acc) of IDS_IFW_AO and IARM-I are shown in Table 2. As shown in Table 2, the classification accuracy of IDS_IFW_AO is higher than IARM-I on almost data sets because the wrapper phase of IDS_IFW_AO finds the reduct with the highest classification accuracy. Furthermore, the cardinality of reduct of IDS_IFW_ is much less than IARM-I, especially on Advertisements data set with large number of attributes. Therefore, the computational time and the generalization of classification rules on the reduct of IDS_IFW_AO are better than IARM-I.

*Table 1***.** Description of the datasets.

1	**Data sets**	**Number of objects**	**Original data sets**	**Incremental data sets**	**Number of attributes**	**Classes**
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
1	Audiology	226	111	115	69	24
2	Soybean-large	307	152	155	35	2
3	Congressional Voting Records	435	215	220	16	2
4	Arrhythmia	452	227	225	279	16
5	Anneal	798	398	400	38	6
6	Advertisements	3279	1639	1640	1558	2

*Table 2***.** The cardinality of reduct and the accuracy of IDS_IFW_AO and IARM-I.

Seq	**Data sets**	**Original, incremental data sets**	**Number of objects**	**Total objects**	**IDS_IFW_AO**		**IARM-I**	
					$\|R\|$	**Acc**	$\|R\|$	**Acc**
1	Audiology	U_0	111	111	5	76.18	8	74.29
		U_1	23	134	5	76.18	9	75.12
		U_2	23	157	6	81.26	12	78.26
		U_3	23	180	6	81.26	12	78.26
		U_4	23	203	7	78.84	14	78.17
		U_5	23	**226**	7	**78.84**	**15**	**76.64**
2	Soybean-large	U_0	152	152	5	96.12	7	95.46
		U_1	31	183	5	96.12	7	95.46
		U_2	31	214	6	96.72	9	95.04
		U_3	31	245	7	95.18	9	95.04
		U_4	31	276	7	95.18	10	94.19
		U_5	31	**307**	**8**	**94.58**	**11**	**94.28**
3	Congressional Voting Records	U_0	215	215	4	92.48	9	91.17
		U_1	44	259	5	92.76	10	91.45
		U_2	44	303	7	94.48	14	92.28
		U_3	44	347	7	94.48	14	92.28
		U_4	44	391	9	94.12	16	92.06
		U_5	44	**435**	**9**	**94.12**	**17**	**92.88**
4	Arrhythmia	U_0	227	227	6	70.08	14	69.16
		U_1	45	272	7	72.45	17	72.05

		U_2	45	317	7	72.45	17	72.05
		U_3	45	362	8	74.18	21	73.23
		U_4	45	407	8	74.18	21	73.23
		U_5	45	**452**	**9**	**76.04**	**24**	**73.08**
5	Anneal	U_0	398	398	4	84.18	8	84.06
		U_1	80	478	5	89.06	8	84.06
		U_2	80	558	5	89.06	8	84.06
		U_3	80	638	6	91.28	9	88.48
		U_4	80	718	6	91.28	9	88.48
		U_5	80	**798**	**6**	**91.28**	**10**	**90.06**
6	Advertisements	U_0	1639	1639	12	93.01	23	92.16
		U_1	328	1967	14	91.18	28	90.48
		U_2	328	2295	14	91.18	28	90.48
		U_3	328	2623	17	91.65	32	91.17
		U_4	328	2951	18	92.82	36	92.06
		U_5	328	**3279**	**19**	**92.90**	**45**	**92.46**

Table 3. The runtime of IDS_IFW_AO and IARM-I.

Seq	**Data sets**	**Original, increm. data sets**	**Number of objects**	**Total objects**	**IDS_IFW_AO**		IARM-I	
					Runtime (s)	**Total runtime (s)**	**Runtime (s)**	**Total runtime (s)**
1	Audiology	U_0	111	111	6.08	6.08	5.82	5.82
		U_1	23	134	0.61	6.69	0.51	6.33
		U_2	23	157	0.35	7.04	0.26	6.59
		U_3	23	180	0.64	7.68	0.42	7.01
		U_4	23	203	0.34	8.02	0.28	7.29
		U_5	23	**226**	0.44	**8.46**	0.35	**7.64**
2	Soybean-large	U_0	152	152	3.04	3.04	2.86	2.86
		U_1	31	183	0.64	3.68	0.42	3.28
		U_2	31	214	0.34	4.02	0.22	3.52
		U_3	31	245	0.73	4.75	0.54	4.06
		U_4	31	276	0.43	5.18	0.34	4.40
		U_5	31	**307**	0.68	**5.86**	0.40	**4.80**

3	Congressional Voting Records	U_0	215	215	5.86	5.86	5.03	5.03
		U_1	44	259	0.56	6.42	0.39	5.42
		U_2	44	303	0.61	7.03	0.46	5.88
		U_3	44	347	0.53	7.56	0.37	6.25
		U_4	44	391	0.47	8.03	0.31	6.56
		U_5	44	**435**	0.55	**8.58**	0.32	**6.88**
4	Arrhythmia	U_0	227	227	35.48	35.48	28.72	28.72
		U_1	45	272	1.58	37.06	1.42	30.14
		U_2	45	317	3.12	40.18	2.26	32.40
		U_3	45	362	2.50	42.68	2.03	34.43
		U_4	45	407	1.36	44.04	1.15	35.58
		U_5	45	452	2.14	**46.18**	1.84	**37.42**
5	Anneal	U_0	398	398	7.48	7.48	6.05	6.05
		U_1	80	478	0.58	8.06	0.38	6.43
		U_2	80	558	0.81	8.95	0.63	7.06
		U_3	80	638	0.53	9.48	0.34	7.40
		U_4	80	718	0.77	10.25	0.56	7.96
		U_5	80	**798**	0.80	**11.05**	0.59	**8.55**
6	Advertisements	U_0	1639	1639	96.74	96.74	82.05	82.05
		U_1	328	1967	5.69	102.43	4.84	86.89
		U_2	328	2295	6.13	108.56	5.18	92.07
		U_3	328	2623	5.70	114.26	4.26	96.33
		U_4	328	2951	3.86	118.12	2.54	98.87
		U_5	328	**3279**	4.74	**122.86**	2.98	**101.85**

Table 3 presents the results of the runtime of IDS_IFW_AO and IARM-I (s). The runtime of IDS_IFW_AO and IARM-I is the average time after 10 times of running on our experimental environment. The results shown in Table 3 indicate that the runtime of IDS_IFW_AO is larger than IARM-I on all data sets because IDS_IFW_AO has more runtime to implement the classifier in the wrapper stage.

4. CONCLUSIONS

It is shown that incremental attribute reduction algorithms in incomplete decision tables which have been proposed are filter algorithms. The reducts of these filter algorithms are not

optimal on the cardinality of reduct and classification accuracy. In this paper, we constructed an incremental formula to compute the distance in [21] when adding multiple objects into incomplete decision tables. By using the incremental distance, we proposed the incremental filter-wrapper algorithm IDS_IFW_AO to find one reduct of an incomplete decision table in order to reduce the cardinanity of reduct and improve the classification accuracy. The experimental results on six data sets show that the classification accuracy of incremental filter-wrapper algorithm IDS_IFW_AO is higher than the incremental filter algorithm IARM-I [15]. Furthermore, the cardinality of reduct of IDS_IFW_AO is much less than IARM-I. Therefore, the execution time and the generalization of classification rules on the reduct of IDS_IFW_AO are better than IARM-I. Further research is to propose incremental filter-wrapper algorithms when adding and deleting conditional attribute sets.

Acknowledgements. This research is funded by the project NVKHK.02/2017 "Xay dung co so du lieu truc tuyen phuục vu phat trien kinh te, xa hoi tinh Thai Nguyen".

REFERENCES

1. Pawlak Z. - Rough sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publisher, London, 1991.
2. Kryszkiewicz M. - Rough set approach to incomplete information systems, Information Science **112** (1998) 39-49.
3. Demetrovics Janos, Vu Duc Thi, Nguyen Long Giang - Metric Based Attribute Reduction in Dynamic Decision Tables, Annales Univ. Sci. Budapest., Sect. Comp. **42** (2014) 157-172.
4. Ma F.M., Ding M.W. , Zhang T.F., Cao J. - Compressed binary discernibility matrix based incremental attribute reduction algorithm for group dynamic data, Neurocomputing **344** (2019) 20-27.
5. Wang L. N., Yang X., Chen Y., Liu L., An S. Y., Zhuo P. - Dynamic composite decision-theoretic rough set under the change of attributes, Int. J. Comput. Intell.Syst. **11** (2018) 355-370.
6. Nguyen Thi Lan Huong, Nguyen Long Giang. - Incremental algorithms based on metric for finding reduct in dynamic decision tables, Journal on Research and Development on Information and Communications Technology**E-3** (9) (2016) 26-39.
7. Shua W. H., Qian W. B., Xie Y. H. - Incremental approaches for feature selection from dynamic data with the variation of multiple objects, Knowledge-Based Systems **163** (2019) 320-331.
8. Wei W., Song P., Liang J. Y., Wu X. Y. - Accelerating incremental attribute reduction algorithm by compacting a decision table, International Journal of Machine Learning and Cybernetics, Springer (2018) 1-19.
9. Demetrovics Janos, Nguyen Thi Lan Huong, Vu Duc Thi, Nguyen Long Giang. - Metric Based Attribute Reduction Method in Dynamic Decision Tables, Cybernetics and Information Technologies **16** (2) (2016) 3-15.
10. Lang G., Li Q., Cai M., Yang T., Xiao Q. - Incremental approaches to knowledge reduction based on characteristic matrices, Int. J. Mach. Learn. Cybern. **8** (1) (2017) 203-222.

11. Yang C. J., Ge H., Li L. S., Ding J. - A unified incremental reduction with the variations of the object for decision tables, Soft Computing **23** (15) (2019) 6407-6427.
12. Wei W., WuX. Y., Liang J. Y., Cui J. B., Sun Y. J. - Discernibility matrix based incremental attribute reduction for dynamic data, Knowledge-Based Systems **140** (2018) 142-157.
13. Jing Y., Li T., Huang J., Chen H. M., Horng S. J. - A Group Incremental Reduction Algorithm with Varying Data Values, International Journal of Intelligent Systems **32** (9) (2017) 900-925.
14. Yu J., Sang L., Dong H. - Based on Attribute Order for Dynamic Attribute Reduction in the Incomplete Information System, IEEE IMCEC (2018) 2475-2478.
15. Shu W. H., Qian W. B. - An incremental approach to attribute reduction from dynamic incomplete decision systems in rough set theory, Data and Knowledge Engineering **100** (2015) 116-132.
16. Zhang D. D., Li R. P., Tang X. T., Zhao Y. S. - An incremental reduct algorithm based on generalized decision for incomplete decision tables, IEEE 3rd International Conference on Intelligent System and Knowledge Engineering (2008) 340-344.
17. Shu W. H., Shen H. - A rough-set based incremental approach for updating attribute reduction under dynamic incomplete decision systems, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (2013) 1-7.
18. Shu W.H., Shen H. - Updating attribute reduction in incomplete decision systems with the variation of attribute set, International Journal of Approximate Reasoning **55** (3) (2014) 867-884.
19. Shu W.H., Shen H. - Incremental feature selection based on rough set in dynamic incomplete data, Pattern Recognition **47** (2014) 3890-3906.
20. Xie X. J., Qin X. L.- A novel incremental attribute reduction approach for dynamic incomplete decision systems, International Journal of Approximate Reasoning **93** (2018) 443-462.
21. Long Giang Nguyen, Hung Son Nguyen. - Metric Based Attribute Reduction in Incomplete Decision Tables, Proceedings of 14th International Conference, Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC 2013, Halifax, NS, Canada, Lecture Notes in Computer Science, SpingerLink **8170** (2013) 99-110.
22. The UCI machine learning repository, http://archive.ics.uci.edu/ml/datasets.html. 22/06/2019.