Distance based Attribute Reduction in Set-Valued Decision Tables

Nguyen Long Giang
Institute of Information Technology, VAST, Ha Noi, Viet Nam

Pham Minh Ngoc Ha
Academy of Finance, Ha Noi, Viet Nam

Nguyen Manh Hung
Military Technical Academy, Ha Noi, Viet Nam

ABSTRACT
Rough set based attribute reduction is an important problem in pre-processing step in data mining. However, most rough set based attribute reduction methods perform on single-value decision tables. In this paper, we solve attribute reduction problems in set-valued decision tables. Our method uses the distance measure which constructed between a conditional attribute set and decision attribute.

General Terms
Rough set, decision table, set-valued decision table, attribute reduction.

Keywords
Rough set, tolerance rough set, decision table, set-valued decision table, attribute reduction, reduct.

1. INTRODUCTION
The rough set theory proposed by Pawlak [6] is an effective tool to solve attribute reduction problems and to extract rules in single-valued information systems. In real problems, the attribute value of object might be a value set . For example, let us consider an information system in which the object "*Nguyen Van A*" with the attribute "Foreign Languages" contains the value "*English, French, Russian*"; that is Nguyen Van A can speak English, or French, or Russian. Such information system is called a set-valued information system.

Attribute reduction in decision systems is the process of choosing the minimum set of the conditional attribute set, preserving classification information of the decision systems. In single-valued decision tables, many attribute reduction methods have been proposed in recent years [1]. In set-valued decision tables, Yan Yong Guan et. al. [3] expanded the equivalence relation in traditional rough set to tolerance relation and developed tolerance rough set model by expanding lower approximation, upper approximation, positive domain, etc. based on tolerance relation. There are remarkable publications about tolerance based attribute reduction methods [5, 9, 12]. Nguyen Sinh Hoa et. al. [5] proposed contingency function based on discernibility function and constructed an attribute reduction method. However, the number of attributes in obtained reduct is larger than the number of attribute in the reduct based on generalized decision function in [4]. Using generalized discernibility function [9], Thi Thu Hien Phung proposed attribute reduction method in set-valued attribute reduction and proved that the reduct of this method is the same as the reduct in [4], it means that this reduct is more minimal than the reduct in [9]. The authors [12] proposed attribute reduction methods in dynamic set-valued decision tables.

In this paper, we propose a distance based attribute reduction method in a set-valued decision table. First, we define a distance determined by the object set *U* and a conditional attribute set *P* based on Jaccard distance. Then we construct a distance measure between a conditional attribute set and the decision attribute. Secondly, we propose a distance based attribute reduction method in a set-valued decision table. We also prove that our reduct is the same as the reduct in [9], it means that our reduct is more minimal than the reduct in [5]. Our distance based method is more effective than the method based on discernibility matrix in [9] about storage.

The structure of this paper is as follows. Section 2 presents some basic concepts in set-valued decision tables as well as the reduct. Section 3 presents the method to construct a distance between two attribute sets. In Section 4, we propose a distance based attribute reduction method in set-valued decision tables. The conclusions and future remarks are presented in the last section

2. BASIC CONCEPTS
In this section, we summarize some basic concepts about set-valued information systems which presented in [3].

An information system is defined as $IS=(U,A)$ in which U is a nonempty set of objects; A is a nonempty set of attributes. The attribute value $a\in A$ of the object $u\in U$ is denoted as $a(u)$, then each subset of attributes $P\subseteq A$ determines a binary indistinguishable relation as follows

$$IND(B)=\{(u,v)\in U\times U \mid \forall b\in B,\ b(u)=b(v)\}$$

It can be easily shown that $IND(P)$ is an equivalence relation on the set *U*. The relation $IND(P)$ constitutes a partition of *U*, which is denoted by U/P. Any element $[u]_P=\{v\in U \mid (u,v)\in IND(P)\}$ in U/P is called the equivalent class. It is easy to see that $[u]_B=\cap[u]_{\{b\}}$ for any $b\in B$.

Let us consider an information system $IS=(U,A)$, if existing $u\in U$ and $a\in A$ such that $a(u)$ contains at least two values, and then $IS=(U,A)$ is called set-valued

information system. For any $B \subseteq A$, a binary relation on U is defined as [3]

$$T_B = \{(u,v) \in U \times U \mid \forall b \in B, b(u) \cap b(v) \neq \varnothing\}$$

It is easy to see that T_B is not an equivalence relation. T_B is a tolerance relation and $T_B = \bigcap_{b \in B} T_b$. Set $T_B(u) = \{v \in U \mid (u,v) \in T_B\}$, $T_B(u)$ is called a tolerance class. The set of all tolerance classes determined by the relation T_B is denoted as $U / T_B = \{T_B(u) \mid u \in U\}$, then U / T_B constitutes a covering of *U* and $\bigcup_{u \in U} T_B(u) = U$. We can see that if $C \subseteq B$ then $T_B(u) \subseteq T_C(u)$ for any $u \in U$.

A set-valued decision table is a set-valued information system $DS = (U, C \cup \{d\})$ in which *C* is conditional attributes and *d* is decision attributes, with assumption that $d(u)$ includes one value for any $u \in U$. For any $u \in U$, $\partial_C(u) = \{d(v) \mid v \in T_C(u)\}$ is called generalized decision function of object *u* on the attribute set *C*. If $|\partial_C(u)| = 1$ for any $u \in U$ then *DS* is consistent, otherwise it is inconsistent.

As incomplete decision tables [4], a reduct of a set-valued decision table is defined as

Definition 1. Let $DS = (U, C \cup \{d\})$ be a set-valued decision table. If $R \subseteq C$ satisfies

(1) $\partial_R(u) = \partial_C(u)$ for any $u \in U$

(2) For any $\forall R' \subset R$, there exists $u \in U$ such that $\partial_{R'}(u) \neq \partial_C(u)$

then *R* is called a reduct of *DS* based on generalized decision function.

Example 1. Let us consider the set-valued decision table $DS = (U, C \cup \{d\})$ as Table 1 where $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ and $C = \{a_1, a_2, a_3, a_4\}$.

Table 1. An example of set-valued decision table

U	a_1	a_2	a_3	a_4	d
u_1	{1}	{1}	{1}	{0}	1
u_2	{0}	{0, 1}	{1}	{0}	1
u_3	{0, 1}	{0, 1}	{0}	{1}	0
u_4	{1}	{0, 1}	{1}	{1}	1
u_5	{0, 1}	{0, 1}	{1}	{1}	2
u_6	{0}	{1}	{1}	{0, 1}	1

For $u_1 \in U$ we have $T_{a_1}(u_1) = \{u_1, u_3, u_4, u_5\}$, $T_{a_2}(u_1) = U$, $T_{a_3}(u_1) = \{u_1, u_2, u_4, u_5, u_6\}$, $T_{a_4}(u_1) = \{u_1, u_2, u_6\}$. So

$$T_C(u_1) = T_{a_1}(u_1) \cap T_{a_2}(u_1) \cap T_{a_3}(u_1) \cap T_{a_4}(u_1) = \{u_1\}$$

Similarly, $T_C(u_2) = \{u_2, u_6\}, T_C(u_3) = \{u_3\}$, $T_C(u_4) = \{u_4, u_5\}, T_C(u_5) = \{u_4, u_5, u_6\}$, $T_C(u_6) = \{u_2, u_5, u_6\}$.

Furthermore, $\partial_C(u_1) = \partial_C(u_2) = \{1\}$, $\partial_C(u_3) = \{0\}$, $\partial_C(u_4) = \partial_C(u_5) = \partial_C(u_6) = \{1,2\}$. Consequently, *DS* is inconsistent.

In next content, we present the method to construct the distance between conditional attributes and decision attribute in a set-valued decision table.

3. CONSTRUCT A DISTANCE MEARSURE IN SET-VALUED DECISION TABLE

3.1 Partition distance and information measure

Let *U* be a finite set of objects and $X, Y \subseteq U$. The following coefficient

$$D(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

is called Jaccard distance between *X* and *Y*. Based on Jaccard distance, we construct a partition distance as follow.

Let $IS = (U, A)$ be an information system, suppose that $K(P) = U / P = \{P_1, ..., P_k\}$ is the partition determined by the attribute set $P \subseteq A$ and $K(\delta) = \{\delta_1, ..., \delta_k\}$ where $\delta_i = U, i = 1..k$. Then, the partition distance between $K(\delta)$ and $K(P)$, called the partition distance determined by the object set *U* and the attribute set *P*, is calculated by the sum of average distances among elements in $K(\delta)$ and $K(P)$ as follows

$$d(K(\delta), K(P)) = \frac{1}{k} \sum_{i=1}^{k} \left(1 - \frac{|U \cap P_i|}{|U \cup P_i|}\right) \quad (3.1)$$

Proposition 1. *Let* $IS = (U, A)$ *be an information system where* $P \subseteq A$ *and* $U = \{u_1, ..., u_n\}$. *Suppose that*

$K(P)=\{P_1,...,P_k\}$, $K(\delta)=\{\delta_1,...,\delta_k\}$ *where* $\delta_i=U, i=1..k$ *. Then we have*

1) $d(K(\delta),K(P))=1-\frac{1}{k}$

2) $d(K(\delta),K(P))$ *achieves the maximum value* $1-\frac{1}{n}$ *when* $K(P)=\omega=\{\{u_1\},...,\{u_n\}\}$. $d(K(\delta),K(P))$ *achieves the minimum value 0 when* $K(P)=\delta=\{U\}$.

Proof.

1) According to the formula (3.1), we have:

$$d(K(\delta),K(P))=\frac{1}{k}\sum_{i=1}^{k}\left(1-\frac{|P_i|}{|U|}\right)=\frac{1}{k}\left(k-\frac{|P_1|+...+|P_k|}{|U|}\right)=\frac{k-1}{k}=1-\frac{1}{k}$$

2) It is easy to see that $d(K(\delta),K(P))$ achieves the maximum value when $\frac{1}{k}$ achieves the minimum value, it means that $k=n$ or $K(P)=\omega=\{\{u_1\},...,\{u_n\}\}$. $d(K(\delta),K(P))$ achieves the minimum value when $k=1$, it means that $K(P)=\delta=\{U\}$.

Based on the above partition distance determined by the object set *U* and the attribute set *P*, we construct a distance between the conditional attribute set and the decision attribute $\{d\}$ in a set-valued decision table.

3.2 Construct a distance measure in a set-valued decision table

Let $DS=(U,C\cup\{d\})$ be a set-valued decision table where $U=\{u_1,...,u_n\}$ and an attribute set $P\subseteq C$. For any tolerance class $T_P(u_i), u_i\in U$, we denote that $K_P^i(\{d\})=T_P(u_i)/\{d\}=\{T_1^i,T_2^i,...,T_{k_P^i}^i\}$ is the partition of the tolerance class $T_P(u_i)$ on the decision attribute $\{d\}$, and $K_P^i(\delta)=\{\delta_1^i,\delta_2^i,...,\delta_{k_P^i}^i\}$ where $\delta_j^i=T_P(u_i),\ j=1..k_P^i$. Then, partition distance determined by the tolerance class $T_P(u_i)$ and the decision attribute $\{d\}$ is defined as:

$$d(K_P^i(\delta),K_P^i(\{d\}))=1-\frac{1}{k_P^i}$$

Let $DS=(U,C\cup\{d\})$ be a set-valued decision table and $U=\{u_1,...,u_n\}$, for $P\subseteq C$ we have that $U/T_P=\{T_P(u_i)|u_i\in U,i=1..n\}$ is a covering of *U*. Then, the distance between the conditional attribute set and the decision attribute $\{d\}$, denoted as $D(P,\{d\})$, is calculated by the average of the sum of partition distances determined tolerance classes $T_P(u_i)$ and $\{d\}$. This distance is defined as:

$$D(P,\{d\})=\frac{1}{n}\sum_{i=1}^{n}d(K_P^i(\delta),K_P^i(\{d\}))=$$

$$=\frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{1}{k_P^i}\right)=1-\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{k_P^i}\right) \quad (3.2)$$

Where *n* is the number of objects of the set-valued decision table and k_P^i is the number of equivalence classes of the partition $T_P(u_i)/\{d\}$, $u_i\in U$.

Proposition 2. *Let* $DS=(U,C\cup\{d\})$ *be a set-valued decision table and* $P,Q\subseteq C$. *If* $P\subseteq Q$ *then* $D(P,\{d\})\geq D(Q,\{d\})$. $D(P,\{d\})=D(Q,\{d\})$ *if and only if* $\partial_P(u)=\partial_Q(u)$ *for any* $u\in U$.

Proof.

Let us consider the set-valued decision table $DS=(U,C\cup\{d\})$ where $U=\{u_1,...,u_n\}$. If $P\subseteq Q$ then $T_Q(u_i)\subseteq T_P(u_i)$ for any $u_i\in U$. Suppose that for $u_i\in U$ we have

$T_P(u_i)/\{d\}=\{T_1^i,T_2^i,...,T_{k_P^i}^i\}$,

$T_Q(u_i)/\{d\}=\{T_1^i,T_2^i,...,T_{k_Q^i}^i\}$, it is clear that $k_Q^i\leq k_P^i$. Consequently, $1-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_P^i}\geq 1-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_Q^i}$, it means that $D(P,\{d\})\geq D(Q,\{d\})$.

$D(P,\{d\})=D(Q,\{d\})$ if and only if $k_P^i=k_Q^i$ for any $u_i\in U$, according to the definition of generalized decision function we have $|\partial_P(u_i)|=|\partial_Q(u_i)|$ for any $u_i\in U$. Since $T_Q(u_i)\subseteq T_P(u_i)$ we have $\partial_P(u_i)=\partial_Q(u_i)$ for any $u_i\in U$.

Proposition 2 shows that the bigger the attribute set P is, the smaller the distance is, and vice versa. The Proposition 2 is the background to construct distance based attribute reduction methods.

Proposition 3. *Let* $DS = \left(U, C \cup \{d\}\right)$ *be a set-valued decision table and* $P \subseteq C$. *Then we have:*

1) $D\left(P, \{d\}\right)$ *achieves the maximum value* $1 - \frac{1}{n}$ *when* $\left|\partial_P(u_i)\right| = n$ *for any* $u_i \in U$.

2) $D\left(P, \{d\}\right)$ *achieves the minimum value* 0 *when* $\left|\partial_P(u_i)\right| = 1$ *for any* $u_i \in U$

Proof.

1) From the fomular (3.2) we can see that $D\left(P, \{d\}\right)$ *achieves the maximum value* when k_P^i *achieves the maximum value n* for any $u_i \in U$, when $T_P(u_i) = U$ and the partition $T_P(u_i)/\{d\} = \left\{\{u_i\} | u_i \in U\right\}$, it means that $\left|\partial_P(u_i)\right| = n$. Then, the maximum value is $1 - \frac{1}{n}\left(\sum_{i=1}^{n} \frac{1}{n}\right) = 1 - \frac{1}{n}$.

2) Similarly, $D\left(P, \{d\}\right)$ *achieves the minimum value* when k_P^i *achieves the minimum value* 1 for any $u_i \in U$, when the partition $T_P(u_i)/\{d\} = \left\{T_P(u_i)\right\}$ (block partition), it means that $\left|\partial_P(u_i)\right| = 1$ for any $u_i \in U$, then *DS* is consistent on the conditional attribute set *P*.

4. DISTANCE BASED ATTRIBUTE REDUCTION IN SET-VALUED DECISION TABLES

In this section, we present a heuristic attribute reduction method in a set-valued decision table based on the distance in section 3. As others heuristic methods, our method consists of steps: the definition of a reduct based on distance, the definition of the significance of attribute based on distance and construction a heuristic algorithm to find the best reduct according to the significance of attribute.

Definition 2. Let $DS = \left(U, C \cup \{d\}\right)$ be a set-valued decision table and an attribute set $R \subseteq C$. If

1) $D(R, \{d\}) = D(C, \{d\})$

2) $\forall r \in R,\ D(R - \{r\}, \{d\}) \neq D(C, \{d\})$

then *R* is a reduct of *C* based on distance.

From Proposition 2 we can conclude that the reduct based on distance is the same as the reduct based on generalized decision function.

Definition 3. Let $DS = \left(U, C \cup \{d\}\right)$ be a set-valued decision table, $B \subset C$ and $b \in C - B$. The significance of attribute *b* with respect to the attribute set B is defined as

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

According to Proposition 2, $D\left(B, \{d\}\right) \geq D\left(B \cup \{b\}, \{d\}\right)$, so $SIG_B(b) \geq 0$. $SIG_B(b)$ is measured by the changes of the distance between B and {d} when *b* is added to *B*. The bigger value of $SIG_B(b)$ is, the more important attribute *b* is. This significance of attribute is a attribute selection criterion in our heuristic attribute reduction algorithm.

In order to find the best reduct, first, we start with the empty set $R = \varnothing$; then the most important attribute is chosen from searching space and added into the set *R*. The above processes are done until we get the reduct. Our algorithm uses adding-deleting methods [11].

Algorithm 1. The heuristic algorithm to find the best reduct based on distance.

Input: The set-valued decision table $DS = \left(U, C \cup \{d\}\right)$

Output: The best reduct R.

1. $R = \varnothing$;
2. Calculate the distance $D\left(R, \{d\}\right)$ và $D\left(C, \{d\}\right)$;
3. While $D\left(R, \{d\}\right) \neq D\left(C, \{d\}\right)$ do
4. Begin
5. For $a \in C - R$ calculate

$$SIG_R(a) = D\left(R, \{d\}\right) - D\left(R \cup \{a\}, \{d\}\right);$$

6. Select $a_m \in C - R$ such that $SIG_R(a_m) = \underset{a \in A-R}{Max}\left\{SIG_R(a)\right\}$;
7. $R = R \cup \{a_m\}$;
8. Calculate the distance $D\left(R, \{d\}\right)$;
9. End;
10. For each $a \in R$ do
11. Begin
12. Calculate the distance $D\left(R - \{a\}, \{d\}\right)$;

13. If $D(R-\{a\},\{d\})=D(R,\{d\})$ then

$R=R-\{a\}$;

14. End;

15. Return R;

Let us consider *While* loop from command line 3 to command line 9, to calculate $SIG_R(a)$ we need to calculate $D(R\cup\{a\},\{d\})$ because $D(R,\{d\})$ have already been calculated in the previous step, it means that we need to calculate $T_{R\cup\{a\}}(u_i)$ and the partition $T_{R\cup\{a\}}(u_i)/\{d\}$. It is easy to see that the time complexity to calculate $T_{R\cup\{a\}}(u_i)$ for any $u_i\in U$ when $T_R(u_i)$ calculated is $O(|U|^2)$, the time complexity to calculate the partition $T_{R\cup\{a\}}(u_i)/\{d\}$ for any $u_i\in U$ is $O(|U|^2)$. So, the time complexity to calculate all $SIG_R(a)$ at command line 5 is:

$$(|C|+(|C|-1)+...+1)*|U|^2=(|C|*(|C|-)/2)*|U|^2=O(|C|^2|U|^2)$$

where $|C|$ is the number of conditional attributes and $|U|$ is the number of objects. the time complexity to choose maximum for significance of attribute at command line 6 is:

$$|C|+(|C|-1)+...+1=|C|*(|C|-1)/2=O(|C|^2)$$

So, the time complexity of While loop is $O(|C|^2|U|^2)$. Similarly, the time complexity of For loop from command line 10 to command line 14 is $O(|C|^2|U|^2)$. Consequently, the time complexity of Algorithm 1 is $O(|C|^2|U|^2)$.

Example 2. Let us consider the set-valued decision table $DS=(U,C\cup\{d\})$ as Table 2 where $U=\{u_1,u_2,u_3,u_4,u_5,u_6\}$, $C=\{a_1,a_2,a_3,a_4\}$.

Table 2. The set-valued decision table of Example 2.

	a_1	a_2	a_3	a_4	d
u_1	1	1	1	0	1
u_2	0	{0,1}	1	0	1
u_3	{0,1}	{0,1}	0	1	0
u_4	1	{0,1}	1	1	1
u_5	{0,1}	{0,1}	1	1	2
u_6	0	1	1	{0,1}	1

According to steps of Algorithm 1, we have

Initial $R=\varnothing$

$S_R(u_1)=S_R(u_2)=S_R(u_3)=S_R(u_4)=S_R(u_5)=S_R(u_6)=U$

$S_R(u_1)/\{d\}=S_R(u_2)/\{d\}=S_R(u_3)/\{d\}=S_R(u_4)/\{d\}=S_R(u_5)/\{d\}=S_R(u_6)/\{d\}$

$=U/\{d\}=\{\{u_1,u_2,u_4,u_6\},\{u_3\},\{u_5\}\}$. So

$$D(R,\{d\})=1-\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{k_P^i}\right)=\frac{2}{3}.$$

We have: $T_C(u_1)=\{u_1\}$, $T_C(u_2)=\{u_2,u_6\}$, $T_C(u_3)=\{u_3\}$, $T_C(u_4)=\{u_4,u_5\}$, $T_C(u_5)=\{u_4,u_5,u_6\}$, $T_C(u_6)=\{u_2,u_5,u_6\}$. So,

$$D(C,\{d\})=1-\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{k_P^i}\right)=\frac{1}{4}$$

Therefore, $D(R,\{d\})\neq D(A,\{d\})$. Perform the While loop. Similarly, we have

$D(R\cup\{a_1\},\{d\})=D(\{a_1\},\{d\})=\frac{2}{3}$. So,

$SIG_R(a_1)=D(R,\{d\})-D(R\cup\{a_1\},\{d\})=2/3-2/3=0$

Similarly,

$SIG_R(a_2)=D(R,\{d\})-D(R\cup\{a_2\},\{d\})=2/3-2/3=0$

$SIG_R(a_3)=D(R,\{d\})-D(R\cup\{a_3\},\{d\})=2/3-5/12=1/4$

$SIG_R(a_4)=D(R,\{d\})-D(R\cup\{a_4\},\{d\})=2/3-4/9=2/9$

So, $SIG_R(a_3)$ is maximal and $R=R\cup\{a_3\}=\{a_3\}$.

Calculate $D(\{a_3\},\{d\})=5/12$

Perform While loop at command line 3.

$SIG_R(a_1)=D(R,\{d\})-D(R\cup\{a_1\},\{d\})=5/12-5/12=0$

$SIG_R(a_2)=D(R,\{d\})-D(R\cup\{a_2\},\{d\})=5/12-5/12=0$

$SIG_R(a_4)=D(R,\{d\})-D(R\cup\{a_4\},\{d\})=5/12-1/4=1/6$

$SIG_R(a_4)$ is maximal, so we have $R=R\cup\{a_4\}=\{a_3,a_4\}$, calculate $D(R,\{d\})=1/4$

Check $D(R,\{d\})=D(C,\{d\})$, stop While loop.

Consequently, $R=\{a_3,a_4\}$. Do command line 10 to command line 14 to check the set *R*.

We have $D(R-\{a_4\},\{d\})=5/12$, so $D(R-\{a_4\},\{d\})\neq D(R,\{d\})$

We have $D(R-\{a_3\},\{d\})=4/9$, so $D(R-\{a_4\},\{d\})\neq D(R,\{d\})$

As the result, the best reduct of *DS* is $R=\{a_3,a_4\}$

5. EXPERIMENTS

The experiments on PC (Pentium Dual Core 2.13 GHz, 1GB RAM, WINXP) are performed on 8 data sets obtained from UCI Machine Learning Repository [13], then we choose Algorithm based on generalized discernibility function [9] (called Algorithm GDF) compared with Algorithm 1. Thus we obtain the results of *reduct* comparison in Table 3, where $|U|$, $|C|$, $|R|$ are the numbers of objects, condition attributes, and after reduction respectively, and *t* is the time of operation (calculated by second). Condition attributes will be denoted by 1, 2,…, $|C|$.

Table 3. The results of Algorithm 1 and Algorithm GDF

Data sets	$\|U\|$	$\|C\|$	Algorithm GDF		Algorithm 1	
			$\|R\|$	*t*	$\|R\|$	*t*
Tic-tac-toe.data	958	9	8	8.343	8	5.937
Hepatitis.data	155	19	3	0.484	3	0.312
Lung-cancer.data	32	56	4	0.78	4	0.62
Automobile.data	205	25	6	3.921	6	2.562
Liver-disorders	345	6	3	0.796	3	0.531
Iris	150	4	3	0.93	3	0.78

Table 4. The reducts of Algorithm 1 and Algorithm GDF

Data sets	The *reducts* of Algorithm GDF	The *reducts* of Algorithm 1
Tic-tac-toe.data	{1, 2, 4, 5, 7, 8, 9}	{1, 2, 4, 5, 7, 8, 9}
Hepatitis.data	{2, 15, 16}	{2, 15, 16}
Lung-Cancer.data	{3, 4, 9, 43}	{3, 4, 9, 43}
Automobile.data	{1, 2, 7, 14, 20, 21}	{1, 2, 7, 14, 20, 21}
Liver-disorders	{1, 2, 5	{1, 2, 5}
Iris	{1, 2, 3	{1, 2, 3}

The experimental results in Table 3, Table 4 show that the *reduct* of Algorithm 1 is the same as that of the Algorithm GDF. However, the time of operation in Algorithm 1 is faster than that in the Algorithm GDF. It means that Algorithm 1 is more effective.

6. CONCLUSIONS

In this paper, we constructed a distance measure between a conditional attribute set and the decision attribute based on partition distance. Based on proposed distance, we proposed a heuristic attribute reduction method in set-valued decision tables. We have proved theoretically and experimentally that the reduct of our method is the same as the reduct in [9] and more effective than the reduct in [5]. Furthermore, our method is more effective than the method based on matrix in [9] about storage. We are planning to work on the more efficient attribute reduction methods in set-valued decision tables and the application in real problems.

7. REFERENCES

[1] Nguyen Long Giang, Rough Set Based Data Mining Methods, Doctor of Thesis, Institute of Information Technology, 2012.

[2] Chen Z. C, Shi P., Liu P. G., Pei Z., Criteria Reduction of Set-Valued Ordered Decision System Based on Approximation Quanlity, International Journal of Innovative Computing, Information and Control, Vol 9, N 6, 2013, pp. 2393-24-4.

[3] Guan Y. Y., Wang H. K., Set-valued information systems, Information Sciences 176, 2006, pp. 2507–2525.

[4] Kryszkiewicz M., Rough set approach to incomplete information systems, Information Science, Vol. 112, 1998, pp. 39-49.

[5] Nguyen Sinh Hoa, Phung Thi Thu Hien, Efficient Algorithms for Attribute Reduction on Set-valued Decision Systems, Lecture Notes of Computer Science (LNCS) Series, Springer, Volume 8170, 2013, pp 87-98.

[6] Pawlak Z., Rough sets, International Journal of Information and Computer Sciences, 11(5), 1982, pp. 341-356.

[7] Pawlak Z., Rough sets: Theoretical Aspects of Reasoning About Data, Kluwer Aca-demic Publishers, 1991.

[8] Qian Y. H., Dang C. Y., Liang J. Y., Tang D. W., Set-valued ordered information systems, Information Sciences 179, 2009, pp. 2809-2832.

[9] Thi Thu Hien Phung, Generalized Discernibility Function based Attribute Reduction in Set-valued Decision Systems, Proceedings of 3rd World Congress on Information and Communication Technologies (WICT2013), IEEE 2013, pp. 225-230. http://www.mirlabs.net/wict13/proceedings/html/toc.html

[10] Y. H. Qian Y. H. , Liang J. Y., On Dominance Relations in Disjunctive Set-Valued Ordered Information Systems, International Journal of Information Technology & Decision Making, Vol. 9, No. 1, 2010, pp. 9–33.

[11] Yao Y.Y., Zhao Y., Wang J., On reduct construction algorithms, Proceedings of International Conference on Rough Sets and Knowledge Technology, 2006, pp. 297-304.

[12] Zhang J. B., Li T. R., Ruan D., Liu D., Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, International Journal of Approximate Reasoning 53, 2012, pp. 620–635.

[13] The UCI machine learning repository, <http://archive.ics.uci.edu/ml/datasets.html>