Quick Attribute Reduction Based on Approximation Dependency Degree

Min Li
NanChang Institute of Technology, Nanchang, Jiangxi 330099, PR China
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, PR China
Email:liminghuadi@hotmail.com

ShaoBo Deng ¹, Shengzhong Feng ², Jianping Fan ²
¹ Key laboratory of intelligent Information Processing, Institute of Computing Technology, Chinese Academy of
Sciences, Beijing 100080, PR China
² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, PR
China
Email: houjiyuan2002@163.com,sz.feng@siat.ac.cn,jp. fan@siat.ac.cn

Abstract—Attribute reduction is one of the core research content of Rough sets theory. Many existing algorithms mainly are aimed at the reduction of consistency decision table, and very little work has been done for attribute reduction aimed at inconsistency decision table. In fact, the methods finding Pawlak reduction from consistent decision table are not suitable for inconsistency decision table. In this paper, we introduce the approximation dependency reduction modal and present the Quick Attribution Reduction based on Approximation Dependency Degree (Quick-ARADD), which can retain the original boundary region and the original positive region unchanged, and keep the approximation accuracy unchanged for all decision equivalence classes (the partition of universe on decision attributes) of a decision table. Theoretical analysis and experimental results show that the Quick-ARADD algorithm is effective and feasible.

Index Terms—Rough Set, Attribute Reduction, Approximation Precision, Approximation Dependency, Classification

I. INTRODUCTION

Rough set theory is a powerful mathematical tool for dealing with vague, imprecise, incomplete and uncertain data, and the theory has been successfully applied in a number of areas such as machine learning, expert system, pattern recognition, decision analysis, and knowledge discovery in databases [1-7,22,23]. Attribute reduction is one important part researched in rough set theory. The usual methods for finding out the relative reduct mainly includes those based on positive region [8-9], on discernibility matrix [10-11] and on entropy [12-14]. These attribute reduction methods are founded upon equivalence relations, suitable for discrete data. Recent years have seen more attention attached to the attribute reduction directly from mixture (discrete and continuous) dataset, such as the neighborhood-based reduction algorithm [15]. Since the neighborhood-based reduction algorithm requires repeated calculation and keeps the neighborhood of each object in universe, their complexities are no less than \( O(|C|^{2} |U|^{2}) \). They are thus unsuitable for large data sets, greatly limiting potential applications. This paper focuses on finding out a reduct from discrete data set,

The significance of a relative reduct lies in preserving the original classification power provided by the whole attribute set through deleting redundant attributes, which is established for the consistent decision table. However, for the inconsistent decision table, a relative reduct may degrade the original approximation accuracy [2] for the decision equivalence classes (the partition of universe on decision attributes). In reality, owing to incomplete information acquisition or some data preprocessing steps, such as discretization of continuous attributes, there exists the inconsistent table. The distribution reduct was proposed by [16] to the inconsistent table. As distribution reduction has strict requirements, Zhang et al. proposed the maximum distribution reduct [17], which is considered too lenient and may shrink the positive region for a decision table. In addition, the reasoning processes for both distribution reduction and maximum distribution reduction are quite complicated and Refs. [16, 17] do not given the corresponding valid formal algorithm. For such reasons, we introduce the approximation dependency reduction modal and the efficient reduction algorithm Quick-ARADD (Quick Attribution Reduction based on Approximation Dependency Degree), which can retain both the original boundary region and the original positive region unchanged, thus keeping the approximation accuracy unchanged. Furthermore, the time complexity of Quick-ARADD algorithm is no more than \( O(|C|^{2} |U| |\log_{2} |U||) \). Hence, it is relatively efficient and suitable for consistent and inconsistent decision tables, especially large ones.

The structure of the rest of this paper is organized as follows. Section 2 briefly introduces the basic notions of rough set. Section 3 presents the concepts and the relative properties of approximation reduction, followed by Section 4, an introduction of a reduction algorithm...
Quick-ARADD based on approximation dependency. In Section 5, we compare our algorithm with Quick-reduct [8] algorithm using the benchmark datasets. Section 6 is the conclusion

II. ROUGH SET THEORY

In this section, we briefly review basic concepts about rough set, which can be found in the Refs. [1,10]

The theory of rough set begins with the notion of indiscernibility. Let $I=(U,A)$ be an information system, where $U=\{u_1,u_2,\ldots,u_n\}$ is a finite and nonempty universe of objects and $A$ is a nonempty finite set of attributes such that $a:U \rightarrow V_a$ for every $a \in A$. With every subset of attributes $P \subseteq A$, there is an associated equivalence relation $Ind(P) = \{(x,y) \in U^2| \forall a \in P, a(x) = a(y)\}$. The equivalence relation $Ind(P)$ divides the universe $U$ into a family of disjoint classes, denoted by $U/Ind(P)$, where $X_i$ is an equivalence class induced by $Ind(P), i=1,2,\ldots,s$. Obviously, any two elements belonging to the same equivalence class $X_i \in U/Ind(P)$ are equivalent according to attribute set $P$.

Given an arbitrary set $X \subseteq U$, $X$ can be approximated using only the information contained within $P$ by constructing two unions of elemental sets $P(X) = \bigcup_{x \in X} P(x)$ and $\overline{P}(X)$, where $P(X)$ and $\overline{P}(X)$ are called P-lower and P-upper approximations of $X$ in $I$. These definitions state that object $x \in P(X)$ belongs certainly to $X$, while objects $x \in \overline{P}(X)$ could belong to $X$. Obviously, there is $\overline{P}(X) \subseteq X \subseteq P(X)$. A set $X$ is said to be definable if $P(X) = \overline{P}(X)$, otherwise, $X$ is said to be rough. The difference between $P(X)$ and $\overline{P}(X)$ is called $P$-boundary region of $X$, denoted as $BN_p(X) = \overline{P}(X) - P(X)$.

A decision table is a four-tuple $DT = (U,C \cup D,V,f)$, where $U$ is a finite and nonempty universe of objects; $A = C \cup D$ a finite set of attributes, where $C$ and $D$ are called conditional attributes set and decision attributes set respectively; $V$ a set of values $\bigcup_{q \in C} V_q$, where $V_q$ is a value set of the attribute $q$; and $f:U \rightarrow A \rightarrow V$, an information function which specifies attribute value of $u \in U$. Given an arbitrary condition attribute subset $B \subseteq C$, then $POS_B(D) = \bigcup_{X \subseteq D} B(X)$ is called the positive region of $D$ with respect to $B$.

Definition 1. Given a decision table $DT = (U,C \cup D,V,f)$, we say $DT$ is consistent table if $POS_D(D) = U$; otherwise, $DT$ is inconsistent.

Definition 2. Given a decision table $DT = (U,C \cup D,V,f)$, we say $a \in B \subseteq C$ is relatively dispensible in $B$ if $POS_{B \setminus \{a\}}(D) = POS_B(D)$; otherwise, $a$ is said to be relatively indispensable in $B$. If every attribute in $B$ is relatively indispensable, we say that $B$ is relatively independent in $DT$.

One of the major applications of rough set theory is the attribute reduction, implying needless attributes are removed so that the reduced set provides the same quality of classification as the original. More precisely a relative reduct with respect to the decision $D$ is defined as follows:

Definition 3. Given a decision table $DT = (U,C \cup D,V,f)$, we say attribute set $B \subseteq C$ is a relative reduct, if $POS_B(D) = POS_C(D)$ and $B$ is relatively independent in $DT$.

Definition 4. Given a decision table $DT = (U,C \cup D,V,f)$, $X \subseteq U \land X \neq \emptyset, B \subseteq C$, the approximation quality of $X$ by $B$ is defined as follows

$$r_B(X) = \frac{|B(X)|}{|X|}$$

(1)

where $|\bullet|$ denotes the cardinality of set.

For most of the applications, only one relative reduct is required. Many literatures adopt forward greedy algorithm based on various significance degree of attribute, to search a relative reduct [8,18-20]. One of the most widely applied attribute significance degrees is defined as follows [16]

$$SIG(a,B,D) = \gamma_{B \setminus \{a\}}(D) - \gamma_B(D)$$

(2)

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}$$

(3)

The Equation (2) expresses the significance degree of attribute $a$ relative to attribute set $B$ with respect to the decision $D$. The Equation (3), called Positive Dependency Degree in the paper, defines the dependency degree of $D$ based on attribute set $B$. The Positive Dependency Degree is the ratio of the number of objects belonging to the positive region to the number of all objects in universe $U$. It is also the sum of the weighted approximation quality of each decision equivalence class, as illustrated in the Equation (4)

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|} = \sum_{X \subseteq D \subseteq U} \frac{|B(X)|}{|U|} = \sum_{X \subseteq D \subseteq U} \frac{|X|}{|U|} r_B(X)$$

(4)

The Equation (3) is widely applied in the practical application owing to having higher calculation efficiency than the Equation (4) does.

For a decision table, a relative reduct is an indispensable condition attribute subset, which preserves the original approximation power provided by the whole attribute set. This works for the consistency decision table. However, for the inconsistency decision table, the relative
reduce may change the approximation accuracy of the decision equivalence classes. The reason is the relative reduce is likely to enlarge the upper approximations for some decision equivalence classes.

In other words, although a relative reduction maintains the lower approximation of each decision equivalence class unchanged, it may expand the boundary region of those inconsistent decision equivalence classes. Consequently, it reduces approximation accuracies (Equation 5). This is exemplified as follows:

Example 1. Let \( DT = (U, C \cup D, V, f) \) be a decision table, where \( U = \{u_1, u_2, \ldots, u_{20}\} \), \( U \setminus D = \{X_1, X_2, X_3\} \), and \( U \setminus C = \{E_1, E_2, E_3, E_4, E_5\} \). Here \( X_1 = \{u_1, u_2, u_3, u_4, u_5\} \), \( X_2 = \{u_6, u_7, u_8, u_9, u_{10}\} \), \( X_3 = \{u_11, u_{12}, u_{13}, u_{14}, u_{15}\} \), \( E_1 = \{u_1, u_2, u_3, u_4, u_5, u_6\} \), \( E_2 = \{u_6, u_7, u_8\} \), \( E_3 = \{u_1, u_2, u_3, u_4, u_5\} \), \( E_4 = \{u_6, u_7, u_8, u_9, u_{10}\} \), and \( E_5 = \{u_1, u_3, u_11, u_{12}, u_{13}\} \). Suppose there is a condition attribute set \( B \subseteq C \) and \( U \setminus B = \{Z_1, Z_2, Z_3, Z_4\} \), where \( Z_1 = E_1 \), \( Z_2 = E_2 \cup E_3 \), \( Z_3 = E_4 \), and \( Z_4 = E_5 \). Then we have \( \mathcal{B}(X_1) = \mathcal{C}(X_1) = E_1 \), \( \mathcal{B}(X_2) = \mathcal{C}(X_2) = E_4 \), and \( \mathcal{B}(X_3) = \mathcal{C}(X_3) = E_5 \). If \( B \) is relatively independent of \( D \), then by the definition 3, \( B \) is a relative reduce. Since \( \mathcal{C}(X_1) = E_1 \cup E_2 \), \( \mathcal{B}(X_1) = E_1 \cup E_4 \cup E_5 \), \( \mathcal{C}(X_2) = E_2 \cup E_4 \), and \( \mathcal{B}(X_2) = E_2 \cup E_4 \cup E_5 \), we then have \( BN_{\beta}(X_1) \supseteq BN_{\alpha}(X_1) \) and \( BN_{\beta}(X_2) \supseteq BN_{\alpha}(X_2) \), i.e., the relative reduce \( B \) gives rise to enlarging the boundary regions of decision classes \( X_1 \) and \( X_2 \). As we know, the uncertainty of the collection is caused by the boundary region. The larger the boundary region is, the greater uncertainty of the collection is. Therefore, the relative reduce \( B \) enlarges the uncertainty of decision classes \( X_1 \) and \( X_2 \).

Most applications require only one reduce. Usually, greedy search strategy is used to find out a relative reduce. For example, according to Equation (2), a forward selection hill-climbing search starts with an empty set \( \text{red} = \emptyset \), evaluates each attribute individually to find out a single attribute which results in the max \( \text{SIG}(\alpha, \text{red}, D) \), and put the single attribute into variable \( \text{red} \). The search then tries the remaining attributes in conjunction with the attribute set \( \text{red} \) to find the best single attribute and put it into \( \text{red} \) once again. This process continues until \( \text{POS}_{\alpha}(D) = \text{POS}_e(D) \). However, the methods above cannot make a good attribute choice when the significance degrees of each candidate attribute are the same. Furthermore, the stop criterion \( \text{POS}_{\alpha}(D) = \text{POS}_e(D) \) may enlarge the boundary regions of the decision classes. Therefore, on the basis of approximation precision, we introduce a new definition attribute significance degree, which preserves both lower and upper approximations to be unchanged for all decision equivalence classes, thus to keep the original approximation precision for a decision table.

### III. APPROXIMATION ACCURACY-BASED REDUCTION

By the Equation (2), the significance degree of a candidate attribute is related only to the concept of lower approximation in rough set theory. However, for an arbitrary object set \( X \subseteq U \), two knowledge sets may result in the same lower approximation and one subset may produce a smaller upper approximation. This subset will be more useful as there is less uncertainty concerning objects within the boundary region. The approximation accuracy [2], a measure degree of the classification power of a knowledge set concerning an object set \( X \subseteq U \), takes into synthetic consideration the lower and upper approximations. For any \( X \subseteq U (X \neq \emptyset) \) and \( B \subseteq C \), the approximation accuracy of \( X \) by \( B \) is defined as follows

\[
\alpha_b(X) = \frac{|B(X)|}{|B(X)|} \tag{5}
\]

Obviously, \( 0 \leq \alpha_b(X) \leq 1 \); If \( \alpha_b(X) = 1 \), then \( B(X) = B(X) \) and the boundary of \( X \) is empty, and \( X \) can be definable; if \( \alpha_b(X) < 1 \), then \( B(X) \subset B(X) \) and the boundary of \( X \) is non-empty, and \( X \) cannot be definable.

Theorem 1. Given a decision table \( DT = (U, C \cup D, V, f) \), \( \forall X \subseteq U \) and \( B \subseteq P \subseteq C \), we have \( \alpha_b(X) \geq \alpha_b(X) \).

Proof. For an arbitrary \( X \subseteq U \) and \( B \), we have \( B(X) \subseteq P(X) \subseteq X \subseteq B(X) \), consequently \( |B(X)| \leq |P(X)| \leq |X| \leq |B(X)| \), holds, which implies \( \frac{|P(X)|}{|B(X)|} \leq \frac{|B(X)|}{|B(X)|} \). Hence \( \alpha_b(X) \geq \alpha_b(X) \).

Theorem 1 implies the approximation accuracy increases monotonously as knowledge (the condition attributes) increases.

We use the approximation accuracy to replace the approximation quality in Equation (4), and we get the definition of Approximation Dependency Degree.

Definition 5. Given a decision table \( DT = (U, C \cup D, V, f) \) and \( B \subseteq C \), Approximation Dependency Degree of \( D \) concerning \( B \) is defined as

\[
\text{DEP}_b(D) = \sum_{X \in U/D} \frac{|X|}{|U|} \alpha_b(X) \tag{6}
\]

Approximation Dependency Degree is the sum of weighted approximation accuracy of each decision equivalence class. The bigger the weight of the decision equivalence class is, the bigger influence the approximation accuracy has on the Approximation Dependency Degree. Obviously, \( 0 \leq \text{DEP}_b(D) \leq 1 \). If \( \alpha_b(X) = 1 \) holds for \( \forall X \in U \), then \( \text{DEP}_b(D) \) obtains the maximum value 1; if \( \alpha_b(X) = 0 \) holds for...
∀\(X_i \in U / D\), then \(DEP_{\alpha}(D)\) obtain the minimum value 0. \(DEP_{\alpha}(D)\) can be considered as to what degree the concept of decision \(D\) can be interpreted according to the knowledge set \(B\): the bigger the value is, the better interpretation \(B\) has on \(D\).

Theorem 2. \(DT = (U, C \cup D, V, f)\) is a consistent decision table, if and only if \(DEP_{\alpha}(D) = 1\).

Proof. ⇒ If \(DT\) is a consistent decision table, then \(POS_{\alpha}(D) = U\), we have \(C(Xi) = \overline{C}(Xe) = X_i\) holds for \(\forall X_i \in U / D\) (otherwise \(POS_{\alpha}(D) \neq U\)); hence \(DEP_{\alpha}(D) = \sum_{X_i \in U / D} |X_i| = 1\) holds.

⇐ If \(DEP_{\alpha}(D) = 1\), then by the definition 5, we have \(\alpha(Xi) = 1\) holds for \(\forall X_i \in U / D\) (otherwise \(DEP_{\alpha}(D) < 1\)), i.e., \(\overline{C}(X_i) = \overline{C}(X_i) = X_i\) holds for \(\forall X_i \in U / D\); thus \(POS_{\alpha}(D) = U\) holds and \(DT\) is a consistent decision table.

Theorem 2 implies that the Approximation Dependency Degree equals to 1 for a consistent decision table but less than 1 for an inconsistent decision table. Theorem 3 implies the Approximation Dependency Degree increases monotonously as knowledge increases.

Definition 6. Given a decision table \(DT = (U, C \cup D, V, f)\), \(B \subseteq C\), we say attribute set \(B\) is an approximation reduct if (1) \(DEP_{\alpha}(D) = DEP_{\alpha}(D)\), and (2) \(\forall\alpha \in B, DEP_{\alpha}(D) < DEP_{\alpha}(D)\).

The first condition guarantees that the approximation reduct has the same Approximation Dependency Degree as the whole condition attribute set, and the second condition guarantees there is no redundant or superfluous attribute in the reduct.

Theorem 4. Given a consistent decision table \(DT = (U, C \cup D, V, f)\), \(B \subseteq C\), then the attribute set \(B\) is an approximation reduct if and only if \(B\) is a relative reduct.

Proof. ⇒ To prove \(B\) is a relative reduct, we only need to prove (1) \(POS_{\alpha}(D) = POS_{\alpha}(D)\) and (2) \(\forall\alpha \in B, POS_{\alpha}(D) \subseteq POS_{\alpha}(D)\).

Since \(DT\) is a consistent decision table, then \(POS_{\alpha}(D) = U\) holds. If attribute set \(B\) is an approximation reduct, then \(DEP_{\alpha}(D) = DEP_{\alpha}(D)\) and \(DEP_{\alpha}(D) = 1\) holds (\(DT\) is a consistent decision table). \(DEP_{\alpha}(D) = 1\) if and only if \(B(X_i) = \overline{B}(X_i) = X_i\) for \(\forall X_i \in U / D\); and \(POS_{\alpha}(D) = U = POS_{\alpha}(D)\) holds. In addition, in terms of the Definition 6, we have \(DEP_{\alpha}(B) < DEP_{\alpha}(D)\) holds for \(\forall\alpha \in B\), which implies \(DEP_{\alpha}(B) = 1\) for \(\forall\alpha \in B\). However \(DEP_{\alpha}(B) = 1\) if and only if \(POS_{\alpha}(B) = U\). And \(POS_{\alpha}(B) = U\) holds, implying \(POS_{\alpha}(B) \subseteq POS_{\alpha}(B)\) holds for \(\forall\alpha \in B\). Hence \(B\) is a relative reduct.

⇐ If \(B\) is a relative reduct, then \(POS_{\alpha}(B) = POS_{\alpha}(B) = U\) (since \(DT\) is a consistent table), and \(B(X_i) = X_i = \overline{B}(X_i) = \overline{C}(X_i) = \overline{C}(X_i)\) holds for \(\forall X_i \in U / D\); hence \(DEP_{\alpha}(B) = DEP_{\alpha}(B) = 1\). In addition, since \(B\) is a relative reduct, \(POS_{\alpha}(B) = POS_{\alpha}(B)\) holds for \(\forall\alpha \in B\), i.e., \(POS_{\alpha}(B) \subseteq U\) holds for \(\forall\alpha \in B\) ; thus \(\exists X_i \in U / D \land (\exists a \in B) \land (B' = B - a) \rightarrow B'(X_i) \subseteq X_i\) holds, which implies \(\alpha(B) < \alpha(X_i)\) holds, guaranteeing \(DEP_{\alpha}(B) < DEP_{\alpha}(B)\). Therefore, \(B\) is an approximation reduct.

Theorem 5. For an inconsistent decision table \(DT = (U, C \cup D, V, f)\), \(B \subseteq C\). If attribute set \(B\) is an approximation reduct, then \(B(X_i) = \overline{B}(X_i)\) and \(\overline{C}(X_i) = \overline{C}(X_i)\) holds for \(\forall X_i \in U / D\).

Proof. Since \(B\) is an approximation reduct, \(DEP_{\alpha}(B) = DEP_{\alpha}(B)\) holds. Then by the Definition 6 and Theorem 1, \(\alpha(B) = \alpha(X_i)\) holds (otherwise \(DEP_{\alpha}(B) < DEP_{\alpha}(B)\)) for \(\forall X_i \in U / D\).

In addition, \(B(X_i) \subseteq C(X_i) \subseteq X_i \subseteq \overline{C}(X_i) \subseteq \overline{B}(X_i)\) holds for \(\forall X_i \in U / D\), and \(\forall X_i \in U / D\), which implies \(|B(X_i)| = |C(X_i)| = |X_i| = |\overline{C}(X_i)| = |\overline{B}(X_i)|\). Then in terms of the Equation (5), \(\alpha(B) = \alpha(X_i)\) is established if and only if \(B(X_i) = \overline{C}(X_i)\). That is proved.

Theorem 5 shows for an inconsistent decision table, an approximation reduct \(B \subseteq C\) produces identical lower and upper approximations as the entire condition attribute set \(C\) concerning decision \(D\), which is quite different from a relative reduct only preserving original positive regions. In other words, the family of the decision class has identical approximation description on an approximation reduct and entire condition attribute set. Obviously, the same conclusion is true for a consistent decision table.

Theorem 6. Given a decision table \(DT = (U, C \cup D, V, f)\), \(B \subseteq C\). If attribute set \(B\) is an approximation reduct, then \(B\) contains a relative reduct.

Proof. If \(DT\) is a consistent decision table and \(B\) is an approximation reduct, then by the Theorem 4, we conclude \(B\) is a relative reduct.

If \(DT\) is an inconsistent decision table and \(B\) is an approximation reduct, then by the Theorem 5 we have \(B(X_i) = \overline{C}(X_i)\) holds for \(\forall X_i \in U / D\); thus \(POS_{\alpha}(B) = POS_{\alpha}(B)\), and \(B\) contains a relative reduct. This is proved.

It can also be proved that \(B\) is not necessarily a relative reduct if attribute set \(B\) is an approximation.
reduct. Only when \( \forall a \in B \land \exists X_i \in U / D \rightarrow \alpha_{a}(X_i) = 1 \land \alpha_{B \setminus a}(X_i) \neq 1 \) holds, \( B \) is a relative reduct.

Definition 6. Given a decision table \( DT = (U, C \cup D, V, f) \), \( B \subseteq C \), \( \forall a \in C - B \), the significance of \( a \) concerning \( B \) can be defined as

\[
SIG' (a, B, D) = DEP_{(a \setminus D)} (D) - DEP_a (D)
\]  

(8)

As \( 0 \leq DEP_{a} (D) \leq 1 \) and \( DEP_{a} (D) \geq DEP_{B \setminus a} (D) \) for \( \forall a \in B \), we have \( 0 \leq SIG' (a, B, D) \leq 1 \). We say attribute \( a \) is insignificant to \( B \) with respect to \( D \) if \( SIG' (a, B, D) = 0 \). Otherwise, the significance degree is \( SIG' (a, B, D) \).

IV. APPROXIMATION DEPENDENCY DEGREE-BASED REDUCTION ALGORITHM

We propose a forward greedy algorithm based on Approximation Dependency Degree according to the Equation (8). It starts off with an empty set and adds, in turn, one at each time, those attributes that result in the greatest increase in Approximation Dependency Degree, until the Approximation Dependency Degree does not increase. The calculation of Equation (8) involves the calculation of Approximation Dependency Degree between condition attribute set and decision \( D \) by Equation (6). The calculation of Approximation Dependency Degree by Equation (6) needs calculation of the approximation accuracy of all decision equivalence classes. This requires calculating the cardinality of both lower and upper approximations of all decision equivalence classes, which determines the efficiency of finding out an approximation reduct. Here, we present a quick algorithm for computing Approximation Dependency Degree of decision \( D \) on condition attribute set \( B \). The pseudo code is as follows:

Algorithm 1. Quick Approximation Dependency Degree algorithm

Input: Decision Table \( DT = (U, C \cup D, V, f) \), \( B \subseteq C \).

Output: Approximation Dependency Degree \( DEP_{a} (D) \).

Step 1: Sort decision table \( DT \) in an ascending (or descending) order according to attribute set \( B \); 

Step 2: For each decision equivalence class \( X_j \in U / D \), define a lower approximation counting variable \( Lower_{a_j} \) and a upper approximation counting variable \( Upper_{a_j} \) respectively, and set their initial value \( Lower_{a_j} = 0 \) and \( Upper_{a_j} = 0 \);

Step 3:

Step3.1: Set \( s=1 \), \( g=1 \), and \( list = \phi \);

Add \( D(u_i) \) to list ; // Here \( D(u_i) \) //denotes the decision value (label) of \( u_i \);

Step3.2: for \( i=2 \) to \( |U| \) do

if \( u_i \) and \( u_s \) have the same values for all attributes in \( B \), then

Set \( g=g+1 \);

if \( D(u_i) \) \( \notin \) list ,

Add \( D(u_i) \) to list ;

end if

else // \( u_i \) and \( u_s \) is not identical ; in whole attribute set \( B \); if \( |list| = 1 \)

for the decision value (label) in \( list \), let the corresponding equivalence class be \( X_j \in U / D \), and update its lower and upper approximations counting variable as follows:

Set \( Upper_{a_j} = Upper_{a_j} + g \);

Set \( Lower_{a_j} = Lower_{a_j} + g \);

else for each decision value in \( list \), let the corresponding equivalence class be \( X_j \in U / D \), and just update its upper approximation counting variable as follows:

Set \( Upper_{a_j} = Upper_{a_j} + g \);

end if

Set \( s=i \) and \( g=1 \);

Set \( list = \phi \);

Add \( D(u_i) \) to the set \( list \);

end if

Step 4: On the basis of the results of lower and upper approximation counting variables of all decision equivalence classes obtained in Step 3, directly calculate \( DEP_{a} (D) \) by Equation (6).

In the following part, we estimate the time complexity of the Algorithm 1 for calculating Approximation Dependency Degree of decision \( D \) on the attribute set \( B \). The amount of time Step 1 takes is \( O(|B||U|\log_2|U|) \), where \( U \) denotes all objects in \( DT \), and \( |\bullet| \) is the cardinality of the set. Step 2 takes \( O(|U|/D) \) time. Step 3.1 takes constant \( O(|1|) \) time. Step 3.2 needs to traverse decision table \( DT \) one time, and it takes \( O(|B||U|) \) time. Step 4 needs to visit all lower and upper approximation counting variables, and it takes \( O(2|U|/D) \) time.

To sum up, the overall time complexity of the Algorithm 1 depends mainly on Step 1, which takes \( O(|B||U|\log_2|U|) \) time, the same amount of time as in the quick algorithm for calculating \( POS_{a} (D) \) (or Positive Region Dependency) [8]. However, Algorithm 1 calculates all cardinalities of lower and upper approximations for all decision equivalence classes rather than just the cardinality of positive region. We can easily analyze that the space complexity is \( O(|U||C\cup D|) \), which is the capacity of the whole
Once the basis of Algorithm 1 and the concept of attribute significance degree by Definition 6, a quick attribute reduction algorithm based on Approximation Dependency Degree is formulated as follows.

Algorithm 2. Quick Attribution Reduction based on Approximation Dependency Degree (Quick-ARADD)

Input: Decision Table $DT = (U, C \cup D, Y, f)$

Output: One approximation reduct $red$.

Step 1: set $red = \phi$;

Step 2: For each $a_i \in C - red$

Compute $SIG'(a_i, red, D) = DEP_{red \cup \{a_i\}}(D) - DEP_{red}(D)$;

// By convention, $DEP_{\phi}(D) = 0$

Step 3: Select the attribute $a_i$ which satisfies $a_i = \arg \max_{a_i \in C - red} (SIG'(a_i, red, D))$

Step 4: If $SIG'(a_i, red, D) > 0$, set $red = red \cup a_i$, go to Step 2

else return;

Step 5: end;

Steps 1, 3, 4, and 5 take $O(1)$ time. By Algorithm 1, the once calculation of $SIG'(a_i, red, D)$ takes $O(|red\cup\{a_i\}|\log |U|)$ time. However, by the way of gradual increase [8], it takes $O(|U|\log |U|)$ time. In the worst case, Quick-ARADD algorithm needs to calculate $SIG'(a_i, red, D)$ for $|C| + |C - red| + \ldots + 1 = |C|(|C| + 1)/2$ times, so the worst search time for an approximation reduct can be estimated as $O(C^2 |U| \log |U|)$. The space complexity of this algorithm is $O(|U|\log |C|\log |D|)$.

Compared with the classical forward attribute reduction method based on attribute significance degree with respect to the positive region [8, 19, 20], Quick-ARADD algorithm has the same time complexity; for a consistent decision table, either Quick-ARADD or a classical method can find out a relative reduct. However, for an inconsistent decision table, a relative reduct only keeps the lower approximation unchanged, while Quick-ARADD algorithm holds both lower and upper approximations unchanged for all decision equivalence classes.

The distribution reduction proposed by Kryzkiewicz [16], sensitive to noise, needs to maintain unchanged the membership degree of all objects in a decision table, which is very demanding. The maximum distribution reduction holds unchanged the maximum distribution decision class of each object. This is too lenient and may shrink the positive region. As for an approximation reduction, although the membership degree may change for those objects in boundary region of the decision equivalence classes, it retains both the original boundary region and the original positive region unchanged. Thereby, it keeps unchanged the approximation accuracy of each decision equivalence class. So, an approximation reduct preserves considerably the original classification power. It has good robustness and will work well in the environment with noise. In addition, the process of finding out a distribution reduct or a maximum distribution reduct is rather complex, but with Quick-ARADD algorithm, it is simple, clear, and of high efficiency.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to validate the effectiveness of the Quick-ARADD algorithm, the experiment is conducted on 8 UCI (machine learning data repository, University of California at Irvine) datasets, which are described in Table 1. All experiments are run on a PC equipped with Windows XP operating system, Pentium IV 2.2 GHz CPU, and 1G memory. The objective of these experiments is to show the classification power of the relative reduct and the approximation reduct. Here, we find out a relative reduct and an approximation reduct for each dataset by implementing the attribute reduction algorithm based on Positive Dependency Degree [8] (to be referred to as POS-RED) and the Quick-ARADD algorithm, respectively. Both algorithms can just deal with discrete attributes, and here we employ the CACM [21] discretization algorithm to transform the continuous data into discrete data.

As for each discretized dataset in Table 1, the Approximation Dependency Degree of decision attribute concerning the whole condition attribute set is displayed in the second column of Table 2. When the Approximation Dependency Degree equals to 1, the discretized dataset is consistent. Otherwise it is inconsistent. In Table 2, the fourth column presents the numbers of attributes of the unreduced discretized dataset and the reduction discretized dataset respectively, and the fifth column lists the run time for a particular reduction algorithm to find out a reduct of each dataset. It is found that, in general, the number of attributes in an approximation reduct found by Quick-ARADD algorithm is equal to or just a little bigger than the number of attributes in a relative reduct discovered by POS_RED algorithm, with only one exception of the WDBC dataset; The spending time for both Quick-ARADD and POS_RED algorithms is very close, simply depending on the cardinality of the reduction they found respectively.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Approximation dependency</th>
<th>Reduction Algorithm</th>
<th>Attributes</th>
<th>C4.5</th>
<th>3NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td>1</td>
<td>POS_RED</td>
<td>13</td>
<td>94.97±4.11</td>
<td>96.08±3.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>6</td>
<td>94.93±5.61</td>
<td>97.19±2.96</td>
</tr>
<tr>
<td>Sonar</td>
<td>1</td>
<td>POS_RED</td>
<td>60</td>
<td>81.76±7.96</td>
<td>86.55±5.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>2.112</td>
<td>78.83±0.36</td>
<td>81.26±5.66</td>
</tr>
<tr>
<td>WDBC</td>
<td>0.9895</td>
<td>POS_RED</td>
<td>14</td>
<td>95.08±2.96</td>
<td>94.41±5.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>2.229</td>
<td>79.83±9.15</td>
<td>81.74±5.38</td>
</tr>
<tr>
<td>WPBC</td>
<td>1</td>
<td>POS_RED</td>
<td>30</td>
<td>95.08±2.16</td>
<td>96.66±1.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>2.221</td>
<td>95.78±1.88</td>
<td>96.48±2.04</td>
</tr>
<tr>
<td>Sat</td>
<td>0.9933</td>
<td>POS_RED</td>
<td>36</td>
<td>86.51±0.75</td>
<td>89.59±0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>1.296</td>
<td>77.29±6.74</td>
<td>76.34±9.58</td>
</tr>
<tr>
<td>Glass</td>
<td>0.937</td>
<td>POS_RED</td>
<td>30</td>
<td>82.81±2.10</td>
<td>86.51±1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>30</td>
<td>86.76±1.39</td>
<td>89.32±0.88</td>
</tr>
<tr>
<td>Heart</td>
<td>0.9353</td>
<td>POS_RED</td>
<td>9</td>
<td>73.86±9.97</td>
<td>70.66±8.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>0.118</td>
<td>71.02±6.84</td>
<td>71.97±8.25</td>
</tr>
<tr>
<td>Ions</td>
<td>0.8885</td>
<td>POS_RED</td>
<td>13</td>
<td>72.92±5.06</td>
<td>72.04±7.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quick-AR ADD</td>
<td>2.093</td>
<td>80.37±7.42</td>
<td>79.63±8.05</td>
</tr>
</tbody>
</table>

For each dataset in Table 1, we use the corresponding discretized dataset and the discretized & reduced dataset generated by Quick-ARADD and POS_RED algorithms respectively to train C4.5 and KNN (here K=3) classifiers, so as to evaluate the effects of generated reducts on the performance of the classification algorithm. All of the results are obtained with 10-fold cross validation method, listed in the last two columns of Table 2. The comparison results show that, in general, the performances of classification generated on both two reduction algorithms are closer or better than the performances of classification induced from the whole discretized condition attributes. As to datasets of Wine, Sonar, WDBC, WPBC and Sat, the Approximation Degree is 1 or very close to 1. For such a reason, those datasets are considered as consistent decision datasets, and the performances of classification generated on the Quick-ARADD algorithm are slightly better than the performances of classification induced by POS_RED algorithm, with the exception of the WPBC dataset obtaining distinct advantages (see Figures 1-2).

For the other three datasets of Glass, Heart and Ions, the Approximation Degree is relatively less than 1 and these three datasets are considered as inconsistent datasets. The Quick-ARADD algorithm obtained obvious advantage of the performance of classification over the POS_RED algorithm (see Figures 3-4). This reinforces the fact that for the classification task on an inconsistent table, the approximation reduct is better than the relative reduct.

![Figure 1. C4.5 Classification performance on reduce subspace of those consistent or close consistent datasets](image1)

![Figure 2. KNN (K=3) Classification performance on reduce subspace of those consistent or close consistent datasets](image2)

![Figure 3. C4.5 Classification performance on reduce subspace of those inconsistent datasets](image3)

![Figure 4. KNN (K=3) Classification performance on reduce subspace of those inconsistent datasets](image4)
VI. CONCLUSIONS

Unlike the most widely applied Positive Dependency Degree, the concept of Approximation Dependency Degree and approximation reduction is introduced in this paper. The theoretical analyses show that for a consistent table, approximation reduct is a relative reduct; for an inconsistent table, the approximation reduct retains the original boundary regions to be unchanged and keeps the approximation accuracy unchanged for all decision equivalence classes. We also present and develop an efficient algorithm Quick-ARADD to find out an approximation reduct for the decision table. The time complexities of the Quick-ARADD algorithm is no more than \( O(|C|^2 |U| \log_2 |U|) \), and thus suitable for large data sets. We perform experiments on 8 discretized UCI datasets, including consistent and inconsistent decision datasets. The experiment results show that for a consistent table, the classification power of the approximation reduct and relative reduct is very close. Whereas for an inconsistent table, the approximation reduct is superior to the relative reduct on the classification performance. This shows the approximation reduct has good robustness and works well in inconsistent data sets. Another point is that the Approximation Dependency Degree proposed in this paper is naturally derived from the concepts of the Positive Dependency Degree and the rough approximation space in rough set theory.

ACKNOWLEDGEMENT

This work was supported by grants from the National High Technology Research and Development Program of China (863 Programs) (No. 2007AA120502), by the funds from Jiangxi Provincial Department of Education (No. GJJ11631) of China, and by the funds from Jiangxi Provincial Natural Science Foundation (No. 2011ZBAB201006) of China.

REFERENCES


© 2013 ACADEMY PUBLISHER
Min Li received the master degree of computer software and theory from NanChang University, NanChang, China in 2005. Now he is a PhD candidate in Shenzhen institute of Advanced Technology, Chinese Academy of Sciences, China. His research interests include data mining, knowledge discovery in database with fuzzy and rough techniques.

Shaobo Deng received the master degree of computer software and theory from NanChang University, NanChang, China in 2005. Now he is a PhD candidate in Institute of Computing Technology, Chinese Academy of Sciences, China. His research interests include Fuzzy and Rough set theory, mathematical logic, uncertainty reasoning.

Shengzhong Feng is a professor at the Shenzhen Institute of Advanced Technology, CAS. His research focuses on parallel algorithms, grid computing and bioinformatics. Specially, now his interests are in developing novel and effective methods for digital city modeling and application. Before came to SIAT, CAS, he had been working in the Institute of Computing Technology, CAS, and participated in the Dawning supercomputer research and development. He graduated from the University of Science and Technology of China in 1991, and received his PhD from Beijing Institute of Technology in 1997.

Jianping Fan is a professor at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His research interests include high performance computing, Grid computing, and computer architecture. He took part in and developed Dawning I, Dawning 1000, Dawning 3000, Dawning 4000 and other series of Dawning supercomputers. He has published more than 70 papers and 1 book. He also acquired 11 pending or issued patents. Professor Fan has received many awards, such as outstanding award of CAS science and technology progress, first price of national science and technology progress, first price of Beijing science and technology progress, and the outstanding young scientist of CAS. He has actively participated in various professional activities. He served as editor of journal of computer research and development, Advisor of 11th five years science and technology development plan of ministry of information industry, and general chair of HPC China2007 and GCC2008.