



Compacted decision tables based attribute reduction



Wei Wei^{a,b}, Junhong Wang^a, Jiye Liang^{a,*}, Xin Mi^a, Chuangyin Dang^b

^a Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

^b Department of System Engineering and Engineering Management, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:

Received 5 November 2014

Received in revised form 25 May 2015

Accepted 22 June 2015

Available online 25 June 2015

Keywords:

Feature selection
Attribute reduction
Rough set
Decision table

ABSTRACT

This paper first points out that the reducts obtained from a simplified decision table are different from those obtained from its original version, and from a simplified decision table, we cannot obtain the reducts in the sense of entropies. To solve these problems, we propose the compacted decision table that can preserve all the information coming from its original version. We theoretically demonstrate that the order preserving of attributes' inner significance and outer significance in the sense of positive region and two types of entropies after a decision table is compacted, which ensures that the reducts obtained from a compacted decision are identical to those obtained from its original version. Finally, several numerical experiments indicate the effectiveness and efficiency of the attribute reduction algorithms for a compacted decision table.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In many practical applications, the dimensions of data sets (the number of attributes) are becoming higher and higher [1,8,24,31]. For these high-dimensional data, attributes irrelevant to recognition tasks may deteriorate the performance of learning algorithms, and result in the high computing cost [11,33]. Therefore, feature selection has become an important preprocessing step in pattern recognition, data mining and machine learning [9,36].

Among existing feature selection algorithms, supervised feature selection algorithms are commonly employed to process the data with class labels, in which there are some representatives, such as feature selection algorithm with feature selection algorithm based on mRMR [32], sparsity-inducing norms [14], feature selection algorithm based on t-test [44,45], feature subset selection algorithm with ordinal optimization [5] and feature selection algorithm based on neighborhood multi-granulation fusion [25]. For the investigation of feature selection, one of critical issues is how to select feature subset, and filters, wrappers and embedded methods have been generally recognized as the most popular methods to solve the issue [2,8]. In filters methods [16,17], the selection of feature subsets has nothing to do with the chosen learning machine. In wrappers methods [18], the selection of feature

subsets depends on the learning machine that scores subsets of feature according to their predictive power. In embedded methods [8,18], the selection of feature subsets, which is a part of training process, is embedded in learning machines. General speaking, filters and embedded methods are more efficient than wrappers methods, and the wrappers and embedded methods are more effective than filters methods [8].

Attribute reduction is an important research area in rough set theory [4,28–30]. From the perspective of feature selection, attribute reduction is a specific kind of supervised feature selection method which adopts filters method. In recent years, researchers have introduced a lot of attribute reduction algorithms. Skowron and Rauszer [40], based on discernibility matrix, proposed an attribute reduction algorithm, by which all reducts can be obtained. Hu and Cercone [10] introduced discernibility matrix into decision tables. Ye and Chen [57] found out that only the reducts for a consistent decision table can be obtained by the method in [10], and proposed a modified discernibility matrix that is suitable for an inconsistent decision table. Yang [54], through considering the discernibility information in the consistent and inconsistent parts of a decision table respectively, proposed another decision-relative discernibility matrix, by which the time of computing reducts is significantly reduced. Wei et al. [51] proposed two discernibility matrices in the sense of Shannon entropy and complement entropy, which efficiently expands the application range of attribute reduction methods based on discernibility matrix. However, the problem of finding all reducts via using these discernibility matrices has been proved to be NP-hard [52,56].

* Corresponding author. Tel./fax: +86 0351 7018176.

E-mail addresses: weiwei@sxu.edu.cn (W. Wei), wjhwjhwj@sxu.edu.cn (J. Wang), ljjy@sxu.edu.cn (J. Liang), 707264128@qq.com (X. Mi), mecdang@cityu.edu.hk (C. Dang).

To solve the above problem, researchers introduced heuristic search strategy into the algorithms of finding reducts, which remarkably lessens their computational burden. Hu and Cercone [10] proposed a heuristic attribute reduction algorithm, in which the positive region is utilized to evaluate attribute significance and stop criterion. Slezak [38,39] first introduced an attribute reduction algorithm in the sense of Shannon entropy. Wang et al. [46,47] further improved the kind of algorithms in the sense of Shannon entropy. Sequently, Liang et al. [20–22,49], through introducing complement entropy to assess attribute significance and stop criterion, defined a new type of attribute reduction algorithms: the one based on complement entropy. To deal with hybrid data with numerical and categorical attributes, the attribute reduction algorithms based on fuzzy rough set and rough fuzzy set were proposed in [3,12,13,37,41,50]. Additionally, plenty of attribute reduction methods were introduced to process incomplete data [26,27]. Yao and Zhao proposed the attribute reduction methods in decision-theoretic rough sets [55], which can achieve the objective of minimize the cost of decisions [15]. Although these heuristic algorithms have speed up the process of finding reducts, the attribute reduction algorithms are still inefficient to deal with large data.

To further improve heuristic attribute reduction algorithms, Qian et al. [34] proposed an acceleration mechanism, in which the useless objects for finding reducts is progressively deleted in each iteration. The similar idea in [34] was developed to deal with incomplete data sets and hybrid data sets [35,48]. However, in [34,35,48], only the useless objects are gradually deleted from data sets. In fact, the number of attributes also largely affects the efficiency of attribute reduction algorithm. Based on this consideration, Liang et al. [23] developed a more effective attribute reduction algorithm, in which both the useless objects and the irrelevant attributes are progressively removed from data sets in the process of finding reducts.

However, all the objects in one equivalence class are dealt with one by one when running these algorithms mentioned above, though they have the same value on each condition attribute. Thus, it is obvious that the duplicated counting results in the unnecessary time-consuming. To address this issue, some researchers introduced several homomorphisms of an information system, by which a massive information system can be compacted into a relatively small-scale information system and all its reducts are unchanged under the condition of homomorphism [6,7,19,42,43]. Furthermore, to remove the redundancy of a decision table, Xu et al. [53] proposed the simplified decision table, in which all the objects in a condition equivalence class are represented by one of objects in the equivalence class. Thus, the attribute reduction algorithms based on the simplifying decision table become more efficient than the previous ones. But, it is worth noticing that for the objects in one condition equivalence class, their values on the decision attribute are possibly different. In other words, the simplification of a decision table in [53] could make a loss of the values on decision attributes. It is precisely the fault of the simplified decision table that motivates us to seek a new method which cannot only eliminate the repetition of condition attribute values, but also preserve all the information on decision attributes.

Based on the analysis mentioned above, in this paper, we first point out that the reducts obtained from a simplified decision table are different from those obtained from its original version. Then, we propose the compacted decision table, and demonstrate that the sequence preserving of inner significance and outer significance in the sense of positive region after a decision table is compacted. And then, we indicate that from a simplified decision table, the reducts in the senses of Shannon entropy and complement entropy cannot be acquired, and demonstrate that they are able

to be obtained from a compacted decision table. Sequently, we design three algorithms based on the proposed compacted decision table to find the reducts in the sense of positive region, Shannon entropy and complement entropy. Finally, several numerical experiments are carried out to verify that our proposed algorithms are more efficient than the existing algorithms.

The remainder of the paper is organized as follows. In Section 2, some preliminaries about the rough set theory and attribute reduction algorithms are reviewed. In Section 3, we point out the fault of the simplified decision table, propose the compacted decision table, demonstrate sequence preserving of inner significance and outer significance in the sense of positive region, and design a new positive region attribute reduction algorithm. In Section 4, based on the proposed compacted decision table, we demonstrate the sequence preserving of inner significance and outer significance in the sense of Shannon entropy and complement entropy, and give the corresponding attribute reduction algorithms. In Section 5, several numerical experiments are carried out to indicate the effectiveness and efficiency of the proposed algorithms. Section 6 concludes the paper with some remarks.

2. Preliminaries

2.1. Rough set

An information system (also known as a data table, an attribute-value system, a knowledge representation system) is a 4-tuple $S = (U, A, V, f)$ (for short $S = (U, A)$), where U is a non-empty and finite set of objects, called a universe, and A is a non-empty and finite set of attributes, V_a is the domain of the attribute a , $V = \bigcup_{a \in A} V_a$ and $f : U \times A \rightarrow V$ is a function $f(x, a) \in V_a$ for each $a \in A$ [28].

Each attribute subset $B \subseteq A$ derives an indiscernibility relation in the following way: $R_B = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \forall a \in B\}$, where $f(x, a)$ and $f(y, a)$ denote the values of objects x and y with respect to the attribute a , respectively. Moreover, the relation R_B partitions U into some equivalence classes given by $U/R_B = \{[x]_B \mid x \in U\}$, just U/B , where $[x]_B$ is the equivalence class determined by x with respect to B , i.e., $[x]_B = \{y \in U \mid (x, y) \in R_B\}$. Furthermore, for any $Y \subseteq U$, one defines that $(\bar{B}(Y), \underline{B}(Y))$ is the rough set of Y with respect to B , where the lower approximation $\underline{B}(Y)$ and the upper approximation $\bar{B}(Y)$ of Y [28] are described by $\underline{B}(Y) = \{x \mid [x]_B \subseteq Y\}$ and $\bar{B}(Y) = \{x \mid [x]_B \cap Y \neq \emptyset\}$.

The objects in $\underline{B}(Y)$ can be certainly classified as the members of Y on the basis of knowledge in B , while the objects in $\bar{B}(Y)$ can be only possibly classified as the members of Y on the basis of knowledge in B . The set $BN_B(Y) = \bar{B}(Y) - \underline{B}(Y)$ is called the B -boundary region of Y . A set is called a rough set (or a crisp set) if the boundary region is non-empty (or empty).

A classification problem can be represented by a decision table $DT = (U, C \cup D)$ with $C \cap D = \emptyset$, where an element of C is called a condition attribute, C is called a condition attribute set, an element of D is called a decision attribute, and D is called a decision attribute set. For the convenience of the latter discussion, we define $\partial([x]_B) = \{f(x_j, d) \mid x_j \in [x]_B\}$, where $D = \{d\}$.

For a given decision table $DT = (U, C \cup D)$, $B \subseteq C$, $U/D = \{Y_1, Y_2, \dots, Y_n\}$, we define the lower and upper approximations of the decision attribute set D with respect to B as

$$\underline{BD} = \{\underline{BY}_1, \underline{BY}_2, \dots, \underline{BY}_n\}, \quad \text{and} \quad \bar{BD} = \{\bar{BY}_1, \bar{BY}_2, \dots, \bar{BY}_n\}.$$

Furthermore, the positive region of D with respect to B is defined as $POS_B^U(D) = \bigcup_{i=1}^n \underline{BY}_i$, the boundary region is defined as $BND_B^U(D) = \bar{BD} - \underline{BD}$, and the negative region $NEC_B^U(D) = U - \bar{BD}$.

Furthermore, we define a partial relation \preceq on the family $\{U/B|B \subseteq A\}$ as follows: $U/A \preceq U/B$ if and only if for every $[x_i]_A$ there exists $[x_j]_B$ such that $[x_i]_A \subseteq [x_j]_B$, $x_i, x_j \in U$. In this case, we say that B is coarser than A (or A is finer than B). If $U/A \preceq U/B$ and $U/A \neq U/B$, we say that B is strictly coarser than A (or A is strictly finer than B), denoted by $U/A \prec U/B$ (or $U/B \succ U/A$). Moreover, if $U/C \preceq U/D$ in a decision table, we call the decision table is consistent, otherwise it is inconsistent.

2.2. Attribute significance

It is crucial how to measure the significance of one attribute when obtaining the reducts by a heuristic algorithm. Thus, for the development of this paper, we review three representative significance measures, which are based on positive region, Shannon condition entropy, complement condition entropy, respectively.

For a given decision table $DT = (U, C \cup D)$ and $B \subseteq C$, the three attribute significance measures are reviewed as follows.

- Attribute significance based on positive region (PR) [10]:
The inner significance of $a \in C$ is defined as

$$\text{Sig}_{PR}^{inner}(a, C, D, U) = \gamma_C^U(D) - \gamma_{C-\{a\}}^U(D),$$

The outer significance of $a \in C - B$ with B is defined as

$$\text{Sig}_{PR}^{outer}(a, B, C, D, U) = \gamma_{B \cup \{a\}}^U(D) - \gamma_B^U(D),$$

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

- Attribute significance based on Shannon condition entropy (SCE) [38]:

The inner significance of $a \in C$ is defined as

$$\text{Sig}_{SCE}^{inner}(a, C, D, U) = H^U(D|C - \{a\}) - H^U(D|C),$$

The outer significance of $a \in C - B$ with B is defined as

$$\text{Sig}_{SCE}^{outer}(a, B, C, D, U) = H^U(D|B) - H^U(D|B \cup \{a\}),$$

where $H^U(D|B) = -\sum_{i=1}^m \frac{|X_i|}{|U|} \sum_{j=1}^n \frac{|X_i \cap Y_j|}{|X_i|} \log \frac{|X_i \cap Y_j|}{|X_i|}$, $X_i \in U/B$, $Y_j \in U/D$, $m = |U/B|$ and $n = |U/D|$.

- Attribute significance based on complement condition entropy (CCE) [20]:

The inner significance of $a \in C$ is defined as

$$\text{Sig}_{CCE}^{inner}(a, C, D, U) = E^U(D|C - \{a\}) - E^U(D|C).$$

The outer significance of $a \in C - B$ with B is defined as

$$\text{Sig}_{CCE}^{outer}(a, B, C, D, U) = E^U(D|B) - E^U(D|B \cup \{a\}).$$

where $E^U(D|B) = \sum_{i=1}^m \sum_{j=1}^n \frac{|Y_j \cap X_i|}{|U|} \frac{|Y_j^c - X_i^c|}{|U|}$, $m = |U/B|$, $n = |U/D|$, and X_i^c and Y_j^c are the complements of $X_i \in U/B$ and $Y_j \in U/D$, respectively.

By means of the inner significance, core [20,28,34,46] can be defined as follows.

Let $DT = (U, C \cup D)$ be a decision table and $a \in C$. If $\text{Sig}_{\Delta}^{inner}(a, C, D, U) > 0$, then a is a core attribute of C with respect to D in the context of type Δ , where $\Delta \in \{PR, SCE, CCE\}$.

2.3. Heuristic attribute reduction algorithms

Many heuristic algorithms of obtaining reducts have been proposed, in which forward greedy search strategy is commonly adopted. This kind of algorithms start from core, and gradually add the attribute with the maximum outer significance into the candidate attribute subset in each iteration until the stop

criterion is satisfied. The description of an forward greedy attribute reduction algorithm is reviewed as follow.

Algorithm 1 ([10,34]). General forward greedy attribute reduction algorithm (GAR- Δ)

Input: Decision table $DT = (U, C \cup D)$;

Output: One reduct red .

Step 1: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;

Step 2: Compute $\text{Sig}_{\Delta}^{inner}(a_k, C, D, U)$, $k \leq |C|$;

Step 3: Put a_k into red , where $\text{Sig}_{\Delta}^{inner}(a_k, C, D, U) > 0$;

Step 4: While $EF_{\Delta}^U(red, D) \neq EF_{\Delta}^U(C, D)$ Do // This provides a stopping criterion.

$\{red \leftarrow red \cup \{a_0\}$, where $\text{Sig}_{\Delta}^{outer}(a_0, red, C, D, U)$
 $= \max\{\text{Sig}_{\Delta}^{outer}(a_k, red, C, D, U), a_k \in C - red\}$ };

Step 5: return red and end,

where $EF_{\Delta}^U(B, D) = EF_{\Delta}^U(C, D)$ is the stop criterion and $\Delta = \{PR, SCE, CCE\}$. For example, while the positive region is employed as the evaluation function, $EF_{PR}^U(B, D)$ is equal to $POS_B^U(D)$ and $EF_{PR}^U(C, D)$ is equal to $POS_C^U(D)$.

Based on Algorithm 1, Qian et.al. [34] proposed an accelerator (shown in Algorithm 2) for heuristic attribute reduction in which the current positive region is progressively removed in each iteration. Thus, the consuming time of finding a reduct is significantly reduced.

Algorithm 2 [34]. Accelerator for attribute reduction from the perspective of objects (ACC- Δ)

Input: Decision table $DT = (U, C \cup D)$;

Output: One reduct red .

Step 1: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;

Step 2: Compute $\text{Sig}_{\Delta}^{inner}(a_k, C, D, U)$, $k \leq |C|$;

Step 3: Put a_k into red , where $\text{Sig}_{\Delta}^{inner}(a_k, C, D, U) > 0$; // These attributes form the core of the given decision table

Step 4: $i \leftarrow 1$ and $U_1 \leftarrow U$;

Step 5: While $EF_{\Delta}^{U_i}(red, D) \neq EF_{\Delta}^{U_i}(C, D)$,

Do {Compute the positive region $POS_{red}^{U_i}(D)$,

$U_{i+1} = U_i - POS_{red}^{U_i}(D)$,

$red \leftarrow red \cup \{a_0\}$, where $\text{Sig}_{\Delta}^{outer}(a_0, red, C, D, U_{i+1})$

$= \max\{\text{Sig}_{\Delta}^{outer}(a_k, red, C, D, U_{i+1}), a_k \in C - red\}$,

$i \leftarrow i + 1$;

Step 6: return red and end,

where $EF_{\Delta}^{U_i}(B, D) = EF_{\Delta}^{U_i}(C, D)$ is the stop criterion, $\Delta = \{PR, SCE, CCE\}$. For example, while the positive region is employed as the evaluation function, $EF_{PR}^{U_i}(B, D)$ is equal to $POS_B^{U_i}(D)$ and $EF_{PR}^{U_i}(C, D)$ is equal to $POS_C^{U_i}(D)$. Based on the Ref. [34], we review the time complexity of Algorithm 2. The time complexity of Step 2 is $O(|U||C|(|C| - 1))$. The time complexity of Step 5 is $O(\sum_{i=1}^{|C|} |C||U_i|(|C| - i + 1))$. And the time complexity of other steps is constant. Therefore, the time complexity of Algorithm 2 is $O(|U||C|(|C| - 1) + \sum_{i=1}^{|C|} |U_i|(|C| - i + 1))$. Comparison with Algorithm 1, it is easy to know that Step 5 of Algorithm 2 is the key point to accelerate attribute reduction.

3. Simplified decision tables and compacted decision tables

In this section, we first point out that the sequence of attribute significance in a simplified decision table is inconsistent with that in its original version by means of a concrete example. To solve the issue, we propose a kind of new decision table: the compacted decision table. It preserves all the information that its corresponding original decision table has. We further demonstrate that the sequence of attribute significance can be remain after compacting a decision table. Finally, we design the positive region attribute reduction algorithm based on the proposed compacted table.

For the development of this section, the simplified decision table, positive region and negative region of a simplified decision table are first reviewed.

Definition 3.1 [53]. Given a decision table $DT = (U, C \cup D)$, $U/C = \{[x'_1]_C, [x'_2]_C, \dots, [x'_m]_C\}$, then the simplified decision table is defined as $DT' = (U', C \cup D)$, where $U' = \{x'_1, x'_2, \dots, x'_m\}$.

The following example gives a concrete simplified decision table.

Example 3.1. From Table 1, we can obtain $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\}$, $U/C = \{\{x_1, x_3\}, \{x_2, x_5\}, \{x_4\}, \{x_6, x_9, x_{11}\}, \{x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_{13}, x_{15}\}\}$. From the Definition 3.1, we get Table 1's simplified version, which is shown as Table 2.

To investigate the properties of a simplified decision table and the attribute significance for it, we will review positive region and negative region for a simplified decision table as follows.

Definition 3.2 [53]. Given a decision table $DT = (U, C \cup D)$ and its simplified version $DT' = (U', C \cup D)$, $B \subseteq C$, then the positive region of D with respect to B is defined as

$$POS_B^{U'}(D) = \{x \in X | X \in U'/B \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\},$$

where $U'_{POS} = \{x'_i | [x'_i]_C \in POS_C^U(D)\}$.

Table 1
A decision table.

| | a_1 | a_2 | a_3 | a_4 | D |
|----------|-------|-------|-------|-------|-----|
| x_1 | 1 | 1 | 1 | 1 | 0 |
| x_2 | 2 | 2 | 2 | 1 | 1 |
| x_3 | 1 | 1 | 1 | 1 | 0 |
| x_4 | 1 | 3 | 1 | 3 | 0 |
| x_5 | 2 | 2 | 2 | 1 | 1 |
| x_6 | 3 | 1 | 2 | 1 | 0 |
| x_7 | 2 | 2 | 3 | 2 | 2 |
| x_8 | 2 | 3 | 1 | 2 | 3 |
| x_9 | 3 | 1 | 2 | 1 | 1 |
| x_{10} | 2 | 2 | 3 | 2 | 2 |
| x_{11} | 3 | 1 | 2 | 1 | 1 |
| x_{12} | 2 | 3 | 1 | 2 | 3 |
| x_{13} | 4 | 3 | 4 | 2 | 1 |
| x_{14} | 2 | 2 | 3 | 2 | 2 |
| x_{15} | 4 | 3 | 4 | 2 | 2 |

Table 2
The decision table simplified from Table 1.

| | a_1 | a_2 | a_3 | a_4 | D |
|----------|-------|-------|-------|-------|-----|
| x_1 | 1 | 1 | 1 | 1 | 0 |
| x_2 | 2 | 2 | 2 | 1 | 1 |
| x_4 | 1 | 3 | 1 | 3 | 0 |
| x_6 | 3 | 1 | 2 | 1 | 0 |
| x_7 | 2 | 2 | 3 | 2 | 2 |
| x_8 | 2 | 3 | 1 | 2 | 3 |
| x_{13} | 4 | 3 | 4 | 2 | 1 |

Definition 3.3 [53]. Given a decision table $DT = (U, C \cup D)$ and its simplified version $DT' = (U', C \cup D)$, $B \subseteq C$, then the negative region of D with respect to B is defined as

$$NEG_B^{U'}(D) = \{X | X \in U'/B, X \subseteq U'_{NEG}\},$$

where $U'_{NEG} = U' - U'_{POS}$.

Based on the positive region and the negative region, the inner significance and the outer significance of a condition attribute in a simplified decision table are defined as:

Definition 3.4. Given a decision table $DT = (U, C \cup D)$ and its simplified version $DT' = (U', C \cup D)$, then the inner significance of $\forall a \in C$ is

$$Sig_{PR}^{inner}(a, C, D, U') = |POS_C^{U'}(D)| - |POS_{C-\{a\}}^{U'}(D)|.$$

Computing core, which is one key step in attribute reduction algorithms, is precisely based on the inner significance.

Definition 3.5 [53]. Given a decision table $DT = (U, C \cup D)$ and its simplified version $DT' = (U', C \cup D)$, $B \subseteq C$, then the outer significance of $\forall a \in (C - B)$ is

$$Sig_{PR}^{outer}(a, B, C, D, U') = |POS_{B \cup \{a\}}^{U'}(D) \cup NEG_{B \cup \{a\}}^{U'}(D)| - |POS_B^{U'}(D) \cup NEG_B^{U'}(D)|.$$

For a condition attribute, the value of its outer significance determines whether it is added into a candidate reduct. In other words, the sequence of all attributes' outer significance values in a certain iteration of an attribute reduction algorithm determines which attribute is added in the candidate reduct. Therefore, the sequence of outer significance values is crucial to the outcome of running an attribute reduction algorithm. However, we find out that the sequence of outer significance values of the condition attributes in a decision table is different from that in its simplified version, which will be illustrated by the following example.

Example 3.2. From Table 1, we have $U/C = \{\{x_1, x_3\}, \{x_2, x_5\}, \{x_4\}, \{x_6, x_9, x_{11}\}, \{x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_{13}, x_{15}\}\}$. Based on Table 1's simplified version (Shown as Table 2), we can obtain the simplified universe $U' = \{x_1, x_2, x_4, x_6, x_7, x_8, x_{13}\}$.

By computing on Table 1, we have that $U/\{a_1\} = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_8, x_{10}, x_{12}, x_{14}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, $U/\{a_1, a_2\} = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5, x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, and $U/\{a_1, a_3\} = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}, \{x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$. And, by computing on Table 2, we have that $U'_{POS} = \{x_1, x_2, x_4, x_7, x_8\}$, $U'_{NEG} = U'_C - U'_{POS} = \{x_6, x_{13}\}$, $U'/\{a_1\} = \{\{x_1, x_4\}, \{x_2, x_7, x_8\}, \{x_6\}, \{x_{13}\}\}$, $U'/\{a_1, a_2\} = \{\{x_1\}, \{x_4\}, \{x_2, x_7\}, \{x_8\}, \{x_6\}, \{x_{13}\}\}$, and $U'/\{a_1, a_3\} = \{\{x_1, x_4\}, \{x_2\}, \{x_7\}, \{x_8\}, \{x_6\}, \{x_{13}\}\}$.

For the convenience of illustration, we suppose $B = \{a_1\}$. by Definitions 3.2 and 3.3, we have that $U'_B = POS_B^{U'}(D) \cup NEG_B^{U'}(D) = \{x_6, x_{13}\}$, $POS_B^U = \emptyset$, $U'_{B \cup \{a_2\}} = POS_{B \cup \{a_2\}}^{U'}(D) \cup NEG_{B \cup \{a_2\}}^{U'}(D) = \{x_1, x_4, x_8, x_6, x_{13}\}$ and $POS_{B \cup \{a_2\}}^U = \{x_1, x_3, x_4, x_8, x_{12}\}$, $U'_{B \cup \{a_3\}} = POS_{B \cup \{a_3\}}^{U'}(D) \cup NEG_{B \cup \{a_3\}}^{U'}(D) = \{x_2, x_7, x_8, x_6, x_{13}\}$ and $POS_{B \cup \{a_3\}}^U = \{x_2, x_5, x_7, x_8, x_{12}, x_{10}, x_{14}\}$.

Therefore, $Sig_{PR}^{outer}(a_2, B, C, D, U') = |U'_{B \cup \{a_2\}} - U'_B| = |\{x_1, x_4, x_8\}| = 3$, $Sig_{PR}^{outer}(a_3, B, C, D, U) = |POS_{B \cup \{a_3\}}^U| - |POS_B^U| = |\{x_1, x_3, x_4, x_8, x_{12}\}| = 5$, $Sig_{PR}^{outer}(a_3, B, C, D, U') = |U'_{B \cup \{a_3\}} - U'_B| = |\{x_2, x_7, x_8\}| = 3$,

$$\text{Sig}^{\text{outer}}(a_3, B, C, D, U) = |\text{POS}_{B \cup \{a_3\}}^U| - |\text{POS}_B^U| = |\{x_2, x_5, x_7, x_8, x_{12}, x_{10}, x_{14}\}| = 7.$$

From the above analysis, we can find out that $\text{Sig}^{\text{outer}}(a_2, B, C, D, U') = \text{Sig}^{\text{outer}}(a_3, B, C, D, U')$, but $\text{Sig}^{\text{outer}}(a_2, B, C, D, U) < \text{Sig}^{\text{outer}}(a_3, B, C, D, U)$. It is evident that the sequence of attributes' outer significance values cannot be remained after a decision table is simplified.

To reduce the redundancy attributes in the simplified decision table, Xu et al. [53] designed an attribute reduction algorithm in the sense of positive region. For the convenience of comparing the algorithm in [53] with other algorithms in this paper, we add the step of computing core into it and rewrite it in the following algorithm.

Algorithm 3. Attribute reduction algorithm based on simplified decision tables in the sense of positive region (AR-ST-PR)

Input: Decision table $DT = (U, C \cup D)$;
Output: One reduct red .
Step 1: Compute $DT' = (U', C \cup D)$ by simplifying the decision table DT ;
Step 2: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;
Step 3: Compute $\text{Sig}_{PR}^{\text{inner}}(a_k, C, D', U')$, $k \leq |C|$;
Step 4: Put a_k into red , where $\text{Sig}_{PR}^{\text{inner}}(a_k, C, D', U'_C) > 0$; // These attributes form the core of the given decision table
Step 5: $i \leftarrow 1$ and $U'_i \leftarrow U'$;
Step 6: While $|\text{POS}_{red}^{U'_i}(D)| \neq |\text{POS}_C^{U'_i}(D)|$,
 Do {Compute the positive region $\text{POS}_{red}^{U'_i}(D)$,
 $U'_{i+1} = U'_i - \text{POS}_{red}^{U'_i}(D) \cup \text{NEC}_{red}^{U'_i}(D)$,
 $red \leftarrow red \cup \{a_0\}$, where $\text{Sig}_{PR}^{\text{outer}}(a_0, red, C, D, U'_{i+1})$
 $= \max\{\text{Sig}_{PR}^{\text{outer}}(a_k, red, C, D, U'_{i+1}), a_k \in C - red\}$,
 $i \leftarrow i + 1$ };
Step 7: return red and end.

Through adding the step of simplifying decision table (Step 1), Algorithm 3 works on a smaller scale of objects than Algorithm 2, which can significantly reduce its consuming time of computing reduct. Based on the Ref. [53], we review the time complexity of Algorithm 3. The time complexity of Step 1 is $O(|C||U|)$. The complexity of Step 3 is $O(|U'| |C| (|C| - 1))$. The time complexity of Step 5 is $O(\sum_{i=1}^{|C|} |C| |U'_i| (|C| - i + 1))$. And the time complexity of other steps is constant. Therefore, the time complexity of Algorithm 3 is $O(|U'| |C| (|C| - 1) + \sum_{i=1}^{|C|} |U'_i| (|C| - i + 1))$. However, the algorithm cannot generate the same reduct as Algorithms 1 and 2 do because of the problem pointed out in Example 3.2. To solve this problem, we propose the compacted decision table in the following definition.

Definition 3.6. Given a decision table $DT = (U, C \cup D)$, $U/C = \{[x'_1]_C, [x'_2]_C, \dots, [x'_m]_C\}$, $D = \{d\}$, $V_d = \{v_{d_1}, v_{d_2}, \dots, v_{d_n}\}$, then the compacted decision table is defined as $DT'' = (U', C \cup D')$, where $U' = \{x'_1, x'_2, \dots, x'_m\}$, $D' = \{d_1, d_2, \dots, d_n\}$, and $f(x'_j, d_i) = |\{x|f(x, d) = v_{d_i}, x \in [x'_j]_C\}|$.

The following example will give a concrete compacted decision table.

Example 3.3. Give a decision table $DT = (U, C \cup D)$ (Shown as Table 1), $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\}$, $U/C = \{\{x_1, x_3\}, \{x_2, x_5\}, \{x_4\}, \{x_6, x_9, x_{11}\}, \{x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\},$

$\{x_{13}, x_{15}\}\}$, $V_d = \{0, 1, 2, 3\}$. Then, based on the definition of the compacted decision table, we have that $U'_C = \{x_1, x_2, x_4, x_6, x_7, x_8, x_{13}\}$, $f(x_1, d_0) = |\{x|f(x, d) = 0, x \in [x_1]_C\}| = 2$. Similarly, we have $f(x_1, d_1) = f(x_1, d_2) = f(x_1, d_3) = 0$, $f(x_2, d_2) = 2$, $f(x_1, d_1) = f(x_2, d_3) = f(x_2, d_4) = 0$, $f(x_4, d_1) = 1$, $f(x_4, d_2) = f(x_4, d_3) = f(x_4, d_4) = 0$, $f(x_6, d_1) = 1$, $f(x_6, d_2) = 2$, $f(x_6, d_3) = f(x_6, d_4) = 0$, $f(x_7, d_1) = f(x_7, d_2) = 0$, $f(x_7, d_3) = 2$, $f(x_7, d_4) = 1$, $f(x_8, d_1) = f(x_8, d_2) = f(x_8, d_3) = 0$, $f(x_8, d_4) = 2$, $f(x_{13}, d_1) = 0$, $f(x_{13}, d_2) = 1$, $f(x_{13}, d_3) = 1$, and $f(x_{13}, d_4) = 0$.

Thus, we can obtain the compacted version of Table 1, i.e. Table 3.

Based on the proposed compacted decision table, the positive region, which is the basis of inner significance and outer significance, is given as follow.

Definition 3.7. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $U'/C = \{[x'_1]_C, [x'_2]_C, \dots, [x'_m]_C\}$, then the positive region of D with respect to C is defined as

$$\text{POS}_B^{U'}(D) = \{x \in X | x \in U'/B \wedge x \subseteq U'_{\text{POS}} \wedge \{d_i \in D' | f(x, d_i) \neq 0\} = 1\},$$

where $U'_{\text{POS}} = \{x'_i | [x'_i]_C \in \text{POS}_C^U(D)\}$, and $f(x, d_i) = \sum_{x'_j \in X'} f(x'_j, d_i)$.

Based on the positive region in a compacted decision table, we define the inner and the outer attribute significance in the following definition.

Definition 3.8. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, then the inner significance of an attribute $a \in C$ is

$$\text{Sig}_{PR}^{\text{inner}}(a, C, D', U') = \sum_{x'_j \in \text{POS}_C^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) - \sum_{x'_j \in \text{POS}_{C - \{a\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i).$$

Inner significance is the basis of computing the core, which is also the key step in the process of computing a reduct in a compacted decision table.

Definition 3.9. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, then the outer significance of an attribute $a \in C - B$ is

$$\text{Sig}_{PR}^{\text{outer}}(a, B, C, D', U') = \sum_{x'_j \in \text{POS}_{B \cup \{a\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) - \sum_{x'_j \in \text{POS}_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i).$$

To analyze how the sequence of attributes' inner and outer significance values will change after a decision table is compacted, we will introduce two theorems, which are based on the following lemma.

Table 3
The decision table compacted from Table 1.

| | a_1 | a_2 | a_3 | a_4 | d_0 | d_1 | d_2 | d_3 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| x_1 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 0 |
| x_2 | 2 | 2 | 2 | 1 | 0 | 2 | 0 | 0 |
| x_4 | 1 | 3 | 1 | 3 | 1 | 0 | 0 | 0 |
| x_6 | 3 | 1 | 2 | 1 | 1 | 2 | 0 | 0 |
| x_7 | 2 | 2 | 3 | 2 | 0 | 0 | 3 | 0 |
| x_8 | 2 | 3 | 1 | 2 | 0 | 0 | 0 | 2 |
| x_{13} | 4 | 3 | 4 | 2 | 0 | 1 | 1 | 0 |

Lemma 3.1. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, then

$$\sum_{x'_j \in POS_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) = |POS_B^U(D)|.$$

Proof. In order to prove the theorem, we suppose that $U = \{x_1, x_2, \dots, x_n\}$, $POS_C^U(D) = \{[x'_1]_C \cup [x'_2]_C \cup \dots \cup [x'_u]_C\}$, $U/C = \{[x'_1]_C, [x'_2]_C, \dots, [x'_m]_C\}$, $U' = \{x'_1, x'_2, \dots, x'_m\}$.

Then, from the definition of positive region in the compacted decision table, we have that

$$\begin{aligned} \sum_{x'_j \in POS_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) &= \sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} \sum_{d_i \in D'} f(x'_j, d_i) \\ &= \sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} |[x'_j]_C|. \end{aligned}$$

Furthermore, from the existing condition $B \subseteq C$, we can obtain that $U/B \succeq U/C$. Thus, without any loss of generalization, we suppose that $POS_C^U(D) = \{[x'_1]_C \cup [x'_2]_C \cup \dots \cup [x'_u]_C\}$, $U'_{POS} = \{x'_1, x'_2, \dots, x'_u\}$, $[x'_p]_B = [x'_p]_C \cup [x'_q]_C$, $[x'_i]_B = [x'_i]_C$ for $i \in [1, m], i \neq p, q$. To prove the lemma, there are four cases should be considered in the following.

- (1) $[x'_p]_C \in POS_C^U(D)$, $[x'_q]_C \in POS_C^U(D)$ and $\partial([x'_p]_C) = \partial([x'_q]_C)$.

From the existing condition, it is easy to know that $\{x'_j \in X | X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1\} = U'_{POS} = \{x'_1, x'_2, \dots, x'_u\}$. Thus,

$$\begin{aligned} \sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} |[x'_j]_C| &= \sum_{x'_j \in U'_{POS}} |[x'_j]_C| = \sum_{j=[1, u], j \neq p, q} |[x'_j]_C| + |[x'_p]_C| + |[x'_q]_C| \\ &= \sum_{j=[1, u], j \neq p, q} |[x'_j]_B| + |[x'_p]_B| = |POS_B^U(D)|. \end{aligned}$$

- (2) $[x'_p]_C \in POS_C^U(D)$, $[x'_q]_C \in POS_C^U(D)$ and $\partial([x'_p]_C) \neq \partial([x'_q]_C)$.

From the existing condition, it is easy to know that $\{x'_j \in X | X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1\} = U'_{POS} - \{x_p, x_q\}$. Thus,

$$\begin{aligned} \sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} |[x'_j]_C| &= \sum_{x'_j \in (U'_{POS} - \{x_p, x_q\})} |[x'_j]_C| = \sum_{j=[1, u], j \neq p, q} |[x'_j]_C| = \sum_{j=[1, u], j \neq p, q} |[x'_j]_B| \\ &= |POS_B^U(D)|. \end{aligned}$$

- (3) $[x'_p]_C \notin POS_C^U(D)$, $[x'_q]_C \in POS_C^U(D)$.

From the existing condition, it is easy to know that $\{x'_j \in X | X \in U'/B \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} = U'_{POS} - \{x_q\}$. Thus, by the similar method with Case(2), we have that

$$\sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} |[x'_j]_C| = |POS_B^U(D)|.$$

- (4) $[x'_p]_C \notin POS_C^U(D)$, $[x'_q]_C \notin POS_C^U(D)$.

From the existing condition, it is easy to know that $\{x'_j \in X | X \in U'/B \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} = U'_{POS}$. Thus, by the similar method with Case(1), we have that

$$\sum_{x'_j \in X \wedge X \in U'/B \wedge X \subseteq U'_{POS} \wedge \{d_i \in D' | f(X, d_i) \neq 0\} = 1} |[x'_j]_C| = |POS_B^U(D)|.$$

In all, based on the above analysis, we conclude that

$$\sum_{x'_j \in POS_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) = |POS_B^U(D)|. \quad \square$$

To illustrate Lemma 3.1, we employ the following example.

Example 3.4. Based on Tables 1 and 3, we suppose $B = \{a_1\}$, and can obtain that the equivalent classes in the decision table shown as Table 1 is $U/B = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_8, x_{10}, x_{12}, x_{14}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, and the partition of universe in the compacted Table 1 (shown as Table 3) is $U'/B = \{\{x_1, x_4\}, \{x_2, x_7, x_8\}, \{x_6\}, \{x_{13}\}\}$. Furthermore, it easy to know that $\sum_{x'_j \in POS_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) = \sum_{d_i \in D'} f(x_1, d_i) + \sum_{d_i \in D'} f(x_4, d_i) = 2 + 1 = 3$, and $|POS_B^U(D)| = |\{x_1, x_3, x_4\}| = 3$, i.e. $\sum_{x'_j \in POS_B^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) = |POS_B^U(D)|$.

To demonstrate the sequence preserving of attributes' inner significance and outer significance that are defined based on positive region is unchanged after a decision table is compacted, we introduce the following two theorems.

Theorem 3.1. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$. If $Sig_{PR}^{inner}(a, C, D', U') > Sig_{PR}^{inner}(b, C, D', U')$, then

$$Sig_{PR}^{inner}(a, C, D, U) > Sig_{PR}^{inner}(b, C, D, U),$$

where, $a, b \in C$.

The theorem is easy to be proved by means of Lemma 3.1.

Theorem 3.2. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$. If $Sig_{PR}^{outer}(a, B, C, D', U' - U'_B) > Sig_{PR}^{outer}(b, B, C, D', U' - U'_B)$, then

$$Sig_{PR}^{outer}(a, B, C, D, U) > Sig_{PR}^{outer}(b, B, C, D, U),$$

where $U'_B = POS_B^{U'}(D') \cup NEG_B^{U'}(D')$.

Proof. From the existing condition, it is evident that $U'_B/(B \cup \{a\}) \leq U'_B/B$. Thus, we have that $\{x \in X | X \in U'_B/(B \cup \{a\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} = \{y \in Y | Y \in U'_B/B \wedge Y \subseteq U'_{POS} \wedge |Y/D| = 1\}$.

Similarly, from $U'_B/(B \cup \{b\}) \leq U'_B/B$, we obtain that $\{x \in X | X \in U'_B/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} = \{y \in Y | Y \in U'_B/B \wedge Y \subseteq U'_{POS} \wedge |Y/D| = 1\}$.

Therefore, it is easy to know that

$$\begin{aligned} \{x \in X | X \in U'_B/(B \cup \{a\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} &= \{x \in X | X \in U'_B/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\}. \end{aligned}$$

From the above equation and the existing condition, we have that

$$\begin{aligned} Sig_{PR}^{outer}(a, B, C, D', U' - U'_B) &> Sig_{PR}^{outer}(b, B, C, D', U' - U'_B) \\ \iff |POS_{B \cup \{a\}}^{U' - U'_B}(D')| - |POS_B^{U' - U'_B}(D')| &> |POS_{B \cup \{b\}}^{U' - U'_B}(D')| - |POS_B^{U' - U'_B}(D')| \\ \iff |POS_{B \cup \{a\}}^{U' - U'_B}(D')| &> |POS_{B \cup \{b\}}^{U' - U'_B}(D')| \\ \iff \{x \in X | X \in U'/(B \cup \{a\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} &= \{x \in X | X \in U'/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} \\ &> \{x \in X | X \in U'/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} \\ &= \{x \in X | X \in U'/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} \\ \iff \{x \in X | X \in U'/(B \cup \{a\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} &= \{x \in X | X \in U'/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} \\ &> \{x \in X | X \in U'/(B \cup \{b\}) \wedge X \subseteq U'_{POS} \wedge |X/D| = 1\} \\ \iff \sum_{x'_j \in POS_{B \cup \{a\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) &> \sum_{x'_j \in POS_{B \cup \{b\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i). \end{aligned}$$

Furthermore, by the conclusion of Lemma 3.1, we have that

$$\begin{aligned} & \sum_{x'_j \in POS_{B \cup \{a\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) > \sum_{x'_j \in POS_{B \cup \{b\}}^{U'}(D')} \sum_{d_i \in D'} f(x'_j, d_i) \\ \iff & |POS_{B \cup \{a\}}^{U'}(D)| > |POS_{B \cup \{b\}}^{U'}(D)| \\ \iff & |POS_{B \cup \{a\}}^{U'}(D)| - |POS_B^{U'}(D)| > |POS_{B \cup \{b\}}^{U'}(D)| - |POS_B^{U'}(D)| \\ \iff & Sig_{PR}^{outer}(a, B, C, D, U) > Sig_{PR}^{outer}(b, B, C, D, U). \end{aligned}$$

Therefore, we have that $Sig_{PR}^{outer}(a, B, C, D, U) > Sig_{PR}^{outer}(b, B, C, D, U)$ if $Sig_{PR}^{outer}(a, B, C, D', U' - U'_B) > Sig_{PR}^{outer}(b, B, C, D', U' - U'_B)$. □

Based on the results of Theorems 3.1 and 3.2, we design a new attribute reduction algorithm.

Algorithm 4. Attribute reduction algorithm based on compacted decision tables in the sense of positive region (AR-CT-PR)

Input: Decision table $DT = (U, C \cup D)$;
Output: One reduct red .
Step 1: Compute $DT'' = (U', C \cup D')$ by compacting the decision table DT ;
Step 2: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;
Step 3: Compute $Sig_{PR}^{inner}(a_k, C, D', U')$, $k \leq |C|$;
Step 4: Put a_k into red , where $Sig_{PR}^{inner}(a_k, C, D', U') > 0$; // These attributes form the core of the given decision table;
Step 5: $i \leftarrow 1$ and $U'_i \leftarrow U'$;
Step 6: While $|POS_{red}^{U'_i}(D')| = 0$,
 Do {Compute the positive region $POS_{red}^{U'_i}(D')$,
 $U'_{i+1} = U'_i - POS_{red}^{U'_i}(D') \cup NEG_{red}^{U'_i}(D')$,
 $red \leftarrow red \cup \{a_0\}$, where $Sig_{PR}^{outer}(a_0, red, C, D', U'_{i+1})$
 $= \max\{Sig_{PR}^{outer}(a_k, red, C, D', U'_{i+1}), a_k \in C - red\}$,
 $i \leftarrow i + 1$ };
Step 7: return red and end.

Furthermore, we analyze the time complexity of Algorithm 4. Because the process of constructing the compacted table is in fact identical to the one of partitioning the universe U with all condition attribute and the time complexity of computing partition is $O(|C||U|)$ [53], we have that their time complexity of Step 1 is $O(|C||U|)$. In step 3, $Sig_{PR}^{inner}(a_k, C, D', U')$ is computed on the compacted decision table DT'' . Thus the complexity of this step is $O(|U' ||C|(|C| - 1))$. In Step 5, through progressively adding an attribute with the maximal significance into the candidate reduct in each iteration, a reduct can eventually be found. The time complexity of this step is $O(\sum_{i=1}^{|C|} |C| |U'_i| (|C| - i + 1))$. And the time complexity of other steps is constant. Therefore, the time complexity of Algorithm 4 is $O(|U' ||C|(|C| - 1) + \sum_{i=1}^{|C|} |U'_i| (|C| - i + 1))$. Comparison with the complexity of Algorithm 2, the difference between the universe U in a decision table and the universe U' in its compacted version is key point. Therefore,

the smaller the ratio of between U' and U is, the less time-consuming of Algorithm 3 is. To illustrate these differences, the time complexity of each step in Algorithms 2–4 is shown as Table 4.

4. Shannon entropy and complement entropy attribute reduction based on compacted decision tables

From the analysis in the above section, we can see that the simplified decision table discards some decision values of objects, which results in not being able to compute the items $|X_i|$ and $|X_i \cap Y_j|$ in the expression of entropies. Therefore, Shannon entropy and complement entropy cannot be computed by means of a simplified decision table. To solve this problem, in this section, we propose Shannon condition entropy and complement condition entropy for a compacted decision table and design the corresponding attribute reduction algorithms.

4.1. Shannon entropy attribute reduction for a compacted decision table

In this subsection, we first define the Shannon entropy, inner significance and outer significance for a compacted decision table. Furthermore, some theorems, which ensure that sequence preserving of attribute's inner significance and outer significance defined by Shannon entropy after a decision table is compacted, are proposed. Finally, an effective attribute reduction algorithm is designed for a compacted decision table.

Definition 4.1. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, $U'/B = \{X'_1, X'_2, \dots, X'_l\}$, then the Shannon condition entropy in the compacted decision table is defined as

$$H^{U'}(D'|B) = - \sum_{j=1}^{m'} \frac{\sum_{i=1}^{n'} f(X'_j, d_i)}{|U'|} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)},$$

where $f(X'_j, d_i) = \sum_{x'_k \in X'_j} f(x'_k, d_i)$, $m' = |U'/B|$ and $n' = |D'|$.

To analyze how the sequence of attribute's inner significance and outer significance values will change after a decision table being compacted, we employ three theorems, which is basis on the following lemma.

Lemma 4.1. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, then

$$H^U(D|B) = \frac{|U'|}{|U|} H^{U'}(D'|B).$$

Proof. From the definition of the compacted decision table, it is easy to know that $U/C = \{[x_{j1}]_C, [x_{j2}]_C, \dots, [x_{jm}]_C\}$, $X_i = [x_{ji}]_C$, $U' = \{x'_1, x'_2, \dots, x'_m\}$, $x'_j \in [x_{j1}]_C$. Let $m = |U/B|$, $n = |U/D|$, $m' = |U'/B|$ and $n' = |D'|$. It is obvious that $m = m'$ and $n = n'$. Without any loss of generalization, by the definition of a compacted decision table, we suppose that $U/B = \{X_1, X_2, \dots, X_l\}$, $U'/B = \{X'_1, X'_2, \dots, X'_l\}$ and $x'_k \in X'_j \iff [x'_k]_C \in X_j$. Then, we obtain that

Table 4
The comparison between the time complexities of Algorithms 2–4.

| Algorithm | Step1 | Step3 | Step5 | Other steps |
|-------------|-------------|------------------------|--|-------------|
| Algorithm 2 | Constant | $O(U C (C - 1))$ | $O(\sum_{i=1}^{ C } C U_i (C - i + 1))$ | Constant |
| Algorithm 3 | $O(C U)$ | $O(U' C (C - 1))$ | $O(\sum_{i=1}^{ C } C U'_i (C - i + 1))$ | Constant |
| Algorithm 4 | $O(C U)$ | $O(U' C (C - 1))$ | $O(\sum_{i=1}^{ C } C U'_i (C - i + 1))$ | Constant |

$$f(X'_j, d_i) = \sum_{x'_k \in X'_j} f(x'_k, d_i) = \sum_{x'_k \in X'_j} |\{x|f(x, d) = v_{d_i}, x \in [x'_k]_C\}|$$

$$= \sum_{x'_k \in X'_j} |[x'_k]_C \cap Y_i| = |X_j \cap Y_i|, \text{ and } \sum_{i=1}^{n'} f(X'_j, d_i) = |X_j|.$$

And, by means of Definition 3.1, we have that

$$H^U(D|B) = - \sum_{j=1}^m \frac{|X_j|}{|U|} \sum_{i=1}^n \frac{|X_j \cap Y_i|}{|X_j|} \log \frac{|X_j \cap Y_i|}{|X_j|}$$

$$= - \sum_{j=1}^{m'} \frac{\sum_{i=1}^{n'} f(X'_j, d_i)}{|U|} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)}$$

$$= \frac{|U'|}{|U|} H^{U'}(D'|B),$$

where $m = |U/B|, n = |U/D|, m' = |U'/B|$ and $n' = |D'|$. □

From the lemma, we can see that relationship between the value of Shannon entropy in a decision table and the one in its compacted version. To better illustrate the Lemma 4.1, based on Tables 1 and 3, we give the following example.

Example 4.1. Based on Tables 1 and 3, we suppose $B = \{a_1\}$, and can obtain that $U/B = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_8, x_{10}, x_{12}, x_{14}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, and the partition of universe in the compacted Table 1 (shown as Table 3) is $U'/B = \{\{x_1, x_4\}, \{x_2, x_7, x_8\}, \{x_6\}, \{x_{13}\}\}$. By Definition 4.1, we have that

$$H^U(D|B) = - \sum_{i=1}^m \frac{|X_i|}{|U|} \sum_{j=1}^n \frac{|X_i \cap Y_j|}{|X_i|} \log \frac{|X_i \cap Y_j|}{|X_i|}$$

$$= - \frac{3}{15} \left(\frac{3}{3} \log \frac{3}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} \right)$$

$$- \frac{7}{15} \left(\frac{0}{7} \log \frac{0}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{3}{7} \log \frac{3}{7} \right)$$

$$- \frac{3}{15} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} \right)$$

$$- \frac{2}{15} \left(\frac{0}{2} \log \frac{0}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{0}{2} \log \frac{0}{2} \right).$$

$$H^{U'}(D'|B) = - \sum_{j=1}^{m'} \frac{\sum_{i=1}^{n'} f(X'_j, d_i)}{|U'|} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)}$$

$$= - \frac{3}{7} \left(\frac{3}{3} \log \frac{3}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} \right)$$

$$- \frac{7}{7} \left(\frac{0}{7} \log \frac{0}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{3}{7} \log \frac{3}{7} \right)$$

$$- \frac{3}{7} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} \right)$$

$$- \frac{2}{7} \left(\frac{0}{2} \log \frac{0}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{0}{2} \log \frac{0}{2} \right).$$

Therefore, we have that $H^U(D|B) = \frac{7}{15} H^{U'}(D'|B)$.

Base on the conclusions in Lemma 4.1, we investigate the change mechanism of shannon entropy for the compacted decision table in which the current positive region is removed. The following theorem gives the concrete analysis.

Theorem 4.1. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, then

$$H^U(D|B) = \frac{|U' - U''_B|}{|U|} H^{U' - U''_B}(D'|B),$$

where $U''_B(D) = POS^{U''}_B(D')$.

Proof. From the definition of the compacted decision table, it is easy to know that $U/C = \{[x_{j_1}]_C, [x_{j_2}]_C, \dots, [x_{j_m}]_C\}, X_i = [x_{j_i}]_C, U' = \{x'_1, x'_2, \dots, x'_m\}, x'_j \in [x_{j_i}]_C$. Without any loss of generalization, we suppose that $U/B = \{X_1, X_2, \dots, X_l\}, U'/B = \{X'_1, X'_2, \dots, X'_l\}$ and $x'_k \in X'_j \iff [x'_k]_C \in X_j, POS^{U'}_B(D') = \{X'_1, X'_2, \dots, X'_l\}, u \leq m$.

From the definition of the positive region in the compacted decision table, we have that $\{d_i \in D' | f(X'_j, d_i) \neq 0\} = 1$ for $\forall X'_j \in POS^{U'}_B(D')$. Thus, it is obvious that for $\forall X'_j \in POS^{U'}_B(D'), \exists d_p \in D'$, such that $f(X'_j, d_p) = \sum_{i=1}^n f(X'_j, d_i)$, and $f(X'_j, d_i) = 0$ for $\forall d_i \neq d_p$. Therefore, we have that $\sum_{i=1}^n \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)} = 0$, for $X'_j \in POS^{U'}_B(D')$. And, by means of Definition 4.1 and Lemma 4.1, we have that

$$H^U(D|B) = \frac{|U'|}{|U|} H^{U'}(D'|B)$$

$$= - \frac{|U'|}{|U|} \sum_{j=u+1}^m \frac{\sum_{i=1}^n f(X'_j, d_i)}{|U'|} \sum_{i=1}^n \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)}$$

$$= - \frac{|U' - U''_B|}{|U|} \sum_{j=u+1}^m \frac{\sum_{i=1}^n f(X'_j, d_i)}{|U' - U''_B|} \sum_{i=1}^n \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^n f(X'_j, d_i)}$$

$$= \frac{|U' - U''_B|}{|U|} H^{U' - U''_B}(D'|B). \quad \square$$

From the theorem, we can see that the value of Shannon condition entropy for a decision table is proportional to the one for its compacted version, which is the basis of constructing effective attribute reduction algorithm on compacted decision tables. To better illustrate Theorem 4.1, we give the following example.

Example 4.2. (Continued from Example 4.1) Based on Table 3, we have $U''_B = POS^{U''}_B(D') = \{x_1, x_4\}$ and $U' - U''_B = \{x_2, x_6, x_7, x_8, x_{13}\}$. Then,

$$H^{U' - U''_B}(D'|B) = - \sum_{j=1}^{m'} \frac{\sum_{i=1}^{n'} f(X'_j, d_i)}{|U' - U''_B|} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)} \log \frac{f(X'_j, d_i)}{\sum_{i=1}^{n'} f(X'_j, d_i)}$$

$$= - \frac{7}{5} \left(\frac{0}{7} \log \frac{0}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{2}{7} \log \frac{2}{7} + \frac{3}{7} \log \frac{3}{7} \right)$$

$$- \frac{3}{5} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} + \frac{0}{3} \log \frac{0}{3} + \frac{0}{3} \log \frac{0}{3} \right)$$

$$- \frac{2}{5} \left(\frac{0}{2} \log \frac{0}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2} + \frac{0}{2} \log \frac{0}{2} \right).$$

Combination with the result of $H^U(D|B)$ in Example 4.1, we have that $H^U(D|B) = \frac{5}{15} H^{U' - U''_B}(D'|B)$.

Lemma 4.1 and Theorem 4.1 are the basis of the following Theorems 4.2 and 4.3. Based on Shannon condition entropy in a compacted decision table, we will define the inner significance and investigate its change mechanism.

Definition 4.2. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT' = (U', C \cup D')$, then the inner significance of $\forall a \in C$ is

$$Sig_H^{inner}(a, C, D', U') = H^{U'}(D'|C - \{a\}) - H^{U'}(D'|C).$$

The inner significance is the basis of computing the core for a compacted decision table.

Theorem 4.2. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$. If $Sig_H^{inner}(a, C, D', U') > Sig_H^{inner}(b, C, D', U')$, then $Sig_H^{inner}(a, C, D, U) > Sig_H^{inner}(b, C, D, U)$.

This theorem is easy to be prove by means of [Theorem 4.1](#), and ensures that the core obtained from a decision table is identical to the one from its compacted version. The conclusion provides the theoretical foundation of computing core for [Algorithm 5](#).

Furthermore, in the compacted decision table, the outer attribute significance defined by Shannon entropy and its change mechanism will be given as follows.

Definition 4.3. Given a decision table $DT = (U, C \cup D)$, and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, then the significance of $\forall a \in (C - B)$ is

$$Sig_H^{outer}(a, B, C, D', U') = H^{U'}(D'|B) - H^{U'}(D'|B \cup \{a\}).$$

Based on [Definition 4.3](#), we investigate the change mechanism of outer significance in the following theorem.

Theorem 4.3. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $a, b \in C - B$ and $B \subset C$. If $Sig_H^{outer}(a, B, C, D', U' - U'_B) > Sig_H^{outer}(b, B, C, D', U' - U'_B)$, then $Sig_H^{outer}(a, B, C, D, U) > Sig_H^{outer}(b, B, C, D, U)$.

Proof. From the existing condition and [Theorem 4.1](#), we have that

$$\begin{aligned} Sig_H^{outer}(a, B, C, D', U' - U'_B) &= H^{U' - U'_B}(D'|B) - H^{U' - U'_B}(D'|B \cup \{a\}) \\ &= \frac{|U|}{|U' - U'_B|} (H^U(D|B) - H^U(D|B \cup \{a\})) \\ &= \frac{|U|}{|U' - U'_B|} Sig_H^{outer}(a, B, C, D, U). \end{aligned}$$

In similar, we obtain that $Sig_H^{outer}(b, B, C, D', U' - U'_B) = \frac{|U|}{|U' - U'_B|} Sig_H^{outer}(b, B, C, D, U)$. Therefore,

$$\begin{aligned} Sig_H^{outer}(a, B, C, D', U' - U'_B) &> Sig_H^{outer}(b, B, C, D', U' - U'_B) \\ &\iff Sig_H^{outer}(a, B, C, D, U) \\ &> Sig_H^{outer}(b, B, C, D, U). \quad \square \end{aligned}$$

From [Theorem 4.3](#), we can see that the sequence of attributes' outer significance defined by Shannon entropy is unchanged after a decision table is compacted, which ensures that the results of attribute reduction in its compacted version is identical to the ones in a decision table. To better illustrate the theorem, we give the following example.

Example 4.3. (Continued from [Example 3.4](#) and [4.2](#)) Based on [Tables 1](#) and [3](#), we obtain the partitions $U/\{a_1\} = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_8, x_{10}, x_{12}, x_{14}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, $U/\{a_1, a_2\} = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5, x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, $U/\{a_1, a_3\} = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}, \{x_7, x_{10}, x_{14}\}, \{x_8, x_{12}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, $U'/\{a_1\} = \{\{x_1, x_4\}, \{x_2, x_7, x_8\}, \{x_6\}, \{x_{13}\}\}$, $U'/\{a_1, a_2\} = \{\{x_1\}, \{x_4\}, \{x_2, x_7\}, \{x_8\}, \{x_6\}, \{x_{13}\}\}$ and $U'/\{a_1, a_3\} = \{\{x_1, x_4\}, \{x_2\}, \{x_7\}, \{x_8\}, \{x_6\}, \{x_{13}\}\}$. From [Example 3.4](#), we have $B = \{a_1\}$ and $U' - U'_B = \{x_2, x_6, x_7, x_8, x_{13}\}$.

By computing, we have

$$\begin{aligned} H^U(D|B) &= 0.3141, \quad H^U(D|B \cup \{a_2\}) = 0.1929, \quad H^U(D|B \cup \{a_3\}) \\ &= 0.0954, \quad H^{U' - U'_B}(D'|B) = 0.9423, \quad H^{U' - U'_B}(D'|B \cup \{a_2\}) \\ &= 0.5786 \quad \text{and} \quad H^{U' - U'_B}(D'|B \cup \{a_3\}) = 0.2863. \end{aligned}$$

Furthermore, by the definition of outer significance, we have

$$\begin{aligned} Sig_H^{outer}(a_2, B, C, D, U) &= H^U(D|B) - H^U(D|B \cup \{a_2\}) \\ &= 0.3141 - 0.1929 = 0.1212. \end{aligned}$$

$$\begin{aligned} Sig_H^{outer}(a_3, B, C, D, U) &= H^U(D|B) - H^U(D|B \cup \{a_3\}) \\ &= 0.3141 - 0.0954 = 0.2187. \end{aligned}$$

Therefore, $Sig_H^{outer}(a_2, B, C, D, U) < Sig_H^{outer}(a_3, B, C, D, U)$.

Similarly, we have

$$\begin{aligned} Sig_H^{outer}(a_2, B, C, D', U' - U'_B) &= H^{U' - U'_B}(D'|B) - H^{U' - U'_B}(D'|B \cup \{a_2\}) \\ &= 0.9423 - 0.5786 = 0.3637. \end{aligned}$$

$$\begin{aligned} Sig_H^{outer}(a_3, B, C, D', U' - U'_B) &= H^{U' - U'_B}(D'|B) - H^{U' - U'_B}(D'|B \cup \{a_3\}) \\ &= 0.9423 - 0.2863 = 0.6560. \end{aligned}$$

Therefore,

$$Sig_H^{outer}(a_2, B, C, D', U' - U'_B) < Sig_H^{outer}(a_3, B, C, D', U' - U'_B).$$

Based on the conclusions about shannon condition entropy and outer significance for a compacted decision table, we design an attribute reduction algorithm as follow.

Algorithm 5. Attribute reduction algorithm based on compacted decision tables in the sense of Shannon entropy (AR-CT-SCE)

Input: Decision table $DT = (U, C \cup D)$;

Output: One reduct red .

Step 1: Compute $DT'' = (U', C \cup D')$ by compacting the decision table DT ;

Step 2: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;

Step 3: Compute $Sig_{SCE}^{inner}(a_k, C, D', U')$, $k \leq |C|$;

Step 4: Put a_k into red , where $Sig_{SCE}^{inner}(a_k, C, D', U') > 0$; // These attributes form the core of the given decision table;

Step 5: $i \leftarrow 1$ and $U'_i \leftarrow U'$;

Step 6: While $H^U(D|C) = \frac{|U'_i - U'_{red}|}{|U|} H^{U'_i - U'_{red}}(D'|red)$

Do {Compute the positive region $POS_{red}^{U'_i}(D')$,

$U'_{i+1} = U'_i - POS_{red}^{U'_i}(D')$,

$red \leftarrow red \cup \{a_0\}$, where $Sig_{SCE}^{outer}(a_0, red, C, D', U'_{i+1}) =$

$\max\{Sig_{SCE}^{outer}(a_k, red, C, D', U'_{i+1}), a_k \in C - red\}$,

$i \leftarrow i + 1$;

Step 7: return red and end.

This algorithm is similar to [Algorithm 4](#), except for the inner significance, outer significance and the stop criterion being defined based on Shannon condition entropy. Thus, the time complexity of [Algorithm 5](#) is equal to the one of [Algorithm 4](#), and we omit its analysis here.

4.2. Complement entropy attribute reduction for compacted decision tables

In this subsection, we first define the complement entropy, inner significance and outer significance for a compacted decision table. Furthermore, some theorems are proposed to ensure that the sequence of attribute's inner significance and outer significance defined by complement entropy is unchanged after a decision table is compacted. Finally, an effective attribute reduction algorithm is designed for a compacted decision table.

Definition 4.4. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, then the complement entropy for a compacted decision table is defined as

$$E^{U'}(D'|C) = \sum_{j=1}^{m'} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{|U|} \frac{\sum_{i=1}^{n'} f(X'_j, d_i) - f(X'_j, d_i)}{|U|},$$

where $f(X'_j, d_i) = \sum_{x'_k \in X'_j} f(x'_k, d_i)$, $m' = |U'/B|$ and $n' = |D'|$.

Sequently, we investigate the relationship between complement entropy in a decision table and that in its compacted version, which is the basis of designing an attribute reduction algorithm based on a compacted decision table. The following lemma and theorems will be employed to solve the issue.

Lemma 4.2. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $B \subseteq C$, then

$$E^U(D|B) = \frac{|U'|^2}{|U|^2} E^{U'}(D'|B).$$

We omit the proof of Lemma 4.2, because it is similar with the one of Lemma 4.1. To better illustrate the Lemma 4.2, we give the following example.

Example 4.4. Based on Tables 1 and 3, we suppose $B = \{a_1\}$, and can obtain that $U/B = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_8, x_{10}, x_{12}, x_{14}\}, \{x_6, x_9, x_{11}\}, \{x_{13}, x_{15}\}\}$, and the partition of universe in the compacted Table 1 (shown as Table 3) is $U'/B = \{\{x_1, x_4\}, \{x_2, x_7, x_8\}, \{x_6\}, \{x_{13}\}\}$. By Definition 4.4, we have that

$$\begin{aligned} E^U(D|B) &= \sum_{i=1}^m \sum_{j=1}^n \frac{|Y_j \cap X_i|}{|U|} \frac{|Y_j^c - X_i^c|}{|U|} \\ &= \sum_{i=1}^m \sum_{j=1}^n \frac{|Y_j \cap X_i|}{|U|} \frac{|X_i - X_i \cap Y_j|}{|U|} \\ &= \frac{1}{15^2} (3 \times 0 + 0 \times 3 + 0 \times 3 + 0 \times 3) \\ &\quad + \frac{1}{15^2} (0 \times 7 + 2 \times 5 + 2 \times 5 + 3 \times 4) \\ &\quad \times \frac{1}{15^2} (1 \times 2 + 1 \times 1 + 0 \times 3 + 0 \times 3) \\ &\quad + \frac{1}{15^2} (0 \times 2 + 1 \times 1 + 1 \times 1 + 0 \times 2), \text{ and} \end{aligned}$$

$$\begin{aligned} E^{U'}(D'|B) &= \sum_{j=1}^{m'} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{|U'|} \frac{\sum_{i=1}^{n'} f(X'_j, d_i) - f(X'_j, d_i)}{|U'|} \\ &= \frac{1}{7^2} (3 \times 0 + 0 \times 3 + 0 \times 3 + 0 \times 3) \\ &\quad + \frac{1}{7^2} (0 \times 7 + 2 \times 5 + 2 \times 5 + 3 \times 4) \\ &\quad + \frac{1}{7^2} (1 \times 2 + 1 \times 1 + 0 \times 3 + 0 \times 3) \\ &\quad + \frac{1}{7^2} (0 \times 2 + 1 \times 1 + 1 \times 1 + 0 \times 2). \end{aligned}$$

Therefore, we have that $E^U(D|B) = \frac{7^2}{15^2} E^{U'}(D'|B)$.

Base on the results in Lemma 4.2, we investigate the change of complement entropy for a compacted decision table when the current positive region is removed from it. The following theorem gives a concrete analysis.

Theorem 4.4. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, then

$$E^U(D|B) = \frac{|U' - U'_B|^2}{|U|^2} E^{U'-U'_B}(D'|B).$$

We omit the proof of Theorem 4.4, because it is very similar with the one of Theorem 4.1.

Example 4.5. (Continued from Example 4.1) Based on Table 3 and Definition 3.7, we have $U'_B = POS'_B(D') = \{x_1, x_4\}$. Thus $U' - U'_B = \{x_2, x_6, x_7, x_8, x_{13}\}$. Then,

$$\begin{aligned} E^{U'}(D'|C) &= \sum_{j=1}^{m'} \sum_{i=1}^{n'} \frac{f(X'_j, d_i)}{|U'|} \frac{\sum_{i=1}^{n'} f(X'_j, d_i) - f(X'_j, d_i)}{|U'|} \\ &= \frac{1}{7^2} (0 \times 7 + 2 \times 5 + 2 \times 5 + 3 \times 4) \\ &\quad + \frac{1}{7^2} (1 \times 2 + 1 \times 1 + 0 \times 3 + 0 \times 3) \\ &\quad + \frac{1}{7^2} (0 \times 2 + 1 \times 1 + 1 \times 1 + 0 \times 2). \end{aligned}$$

Combination with the result of $E^U(D|B)$ in Example 4.4, we have that $E^U(D|B) = \frac{5^2}{15^2} E^{U'-U'_B}(D'|B)$.

Based on the complement condition entropy proposed above, the inner attribute significant in the sense of complement entropy and its change mechanism are given as follows.

Definition 4.5. Given a decision table $DT = (U, C \cup D)$, and its compacted version $DT' = (U', C \cup D')$, then the inner significance of $\forall a \in C$ is

$$Sig_E^{inner}(a, C, D', U') = E^{U'}(D'|C - \{a\}) - E^{U'}(D'|C).$$

For a compacted decision table, the inner attribute significance is the basis of computing the core of condition attributes with respect to decision attributes. Therefore, it is important to investigate the change mechanism of inner significance.

Theorem 4.5. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $a, b \in C$. If $Sig_H^{inner}(a, C, D', U') > Sig_H^{inner}(b, C, D', U')$, then $Sig_H^{inner}(a, C, D, U) > Sig_H^{inner}(b, C, D, U)$.

This theorem is easy to be proved by Lemma 4.2, and ensures that the core in the sense of complement entropy obtained from a decision table is equal to the one from its compacted version. This theorem provides the theoretical foundation of computing core in Algorithm 6.

Furthermore, we define the outer significance in the sense of complement entropy, which is the basis of determining which attribute is added into the candidate reduct in each iteration of attribute reduction algorithms.

Definition 4.6. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT'' = (U', C \cup D')$, $a, b \in C - B$, $B \subset C$, then the outer significance of $\forall a \in (C - B)$ is

$$Sig_E^{outer}(a, B, C, D', U') = E^{U'}(D'|C) - E^{U'}(D'|C \cup \{a\}).$$

Based on the definition, we investigate the change mechanism of outer attribute significance after a decision table is compacted.

Theorem 4.6. Given a decision table $DT = (U, C \cup D)$ and its compacted version $DT' = (U', C \cup D')$, $U/B \succ U/C$. If $Sig^{outer}(a, B, C, D', U') > Sig^{outer}(b, B, C, D', U')$, then $Sig^{outer}(a, B, C, D, U) > Sig^{outer}(b, B, C, D, U)$.

We omit the proof of this theorem, because it is similar with the one of Theorem 4.3.

From this theorem, we can obtain the sequence preserving of attribute significance in the sense of complement entropy after a decision table is compacted, which ensures that the reducts obtained from a compacted decision table is identical to those obtained from its original version.

Based on the theoretical results mentioned above, we design an attribute reduction algorithm (called as Algorithm 6: Attribute reduction algorithm for a compacted decision table in the sense of complement condition entropy (AR-CT-CCE)). The algorithm is similar with Algorithms 4 and 5, but the inner significance, outer significance and stop criterion are replaced by those in the sense of complement entropy. Thus, we omit the description of Algorithm AR-CT-CCE. Because the time complexity of Algorithm 6 is identical to the one of Algorithm 4, we also omit its analysis here.

5. Experimental analysis

To verify the theoretical results mentioned above, in this section, we carry out several comparative experiments between ACC-PR, AR-ST-PR and AR-CT-PR, between ACC-SCE and AR-CT-SCE, and between ACC-CCE and AR-CT-CCE. The hardware used in these experiments is a personal computer equipped with

Intel Core i3 and 2 GB Memory, and the operation system and software are Windows 7 and C#, respectively. Twelve data sets in UCI repository of machine learning databases are employed in experiments and shown in Table 5 in which the size of data sets, the dimension of the data sets and the number of decision value of every data set vary widely. In addition, to meet the needs of the experiment, we convert the incomplete data sets into the complete ones, and also discretize the numerical data sets.

5.1. Experiments about ACC-PR, AR-ST-PR and AR-CT-PR

We carry out the experiments on first ten data sets in Table 5 to compare ACC-PR with AR-ST-PR, and the results are shown in Table 6. From this table, we can see that the bold figures 12, 13 in the reduct of 'letter' are different from 13, 12 in the reduct of the simplified 'letter', and the phenomenon also appears in data sets 'Spect', 'Wine_lis', 'Ticdata2000' and 'Molecula', which indicates that the reducts obtained from a simplified decision table is different from the ones obtained from its original version.

Table 7 shows the concrete process of running ACC-PR and AR-CT-PR on the data set Ticdata2000. The profile of Ticdata2000 is given in the first row of Table 7 (Row 'Initialization'), in which Column 'Att' represents attribute, Column 'U' represents universe, Column 'POS' represents positive region, Column 'NEG' represents negative region, and Column 'Time' represents the time of reading data. The second row in Table 7 shows the profile of the compacted Ticdata2000 and the consuming time of compacting Ticdata2000, and the columns about ACC-PR in this row are Null because ACC-PR has no the step. The third row in Table 7 shows the consuming time of finding core by means of ACC-PR and AR-CT-PR. The rows from Loop1 to Loop14 show in each loop, the profiles of ticdata2000 and the compacted ticdata2000 and the consuming time of ACC-PR and AR-CT-PR, respectively. From the Table 7, we can see that the number of objects (5158) in the compacted Ticdata2000 is less than the one (5822) in Ticdata2000, and thus the consuming time of finding the core and the reduct by AR-CT-PR is correspondingly less than the one by ACC-PR. The experimental results indicate that AR-CT-PR is more efficient than ACC-PR if a decision can be significantly compacted.

Furthermore, Table 8 lists the experimental results of running ACC-PR and AR-CT-PR on all data sets in Table 5. In Table 8, Column 'Algorithm' represents the employed algorithm, Column 'Object' represents the numbers of objects in a data set and the one in its compacted version, Column 'Ratio_C' represents the ratio of the compacted data sets and the original data sets, Columns 'Ini_T', 'Com_T', 'Cor_T' and 'Red_T' represent the time of reading data, the time of compacting data, the time of computing core and the time of computing reduct, Column 'Total_T' is the sum of them, and Column 'Ratio_T' represents the ratio of the consuming time of ACC-PR and the one of AR-CT-PR. From the Table 8, we can see that

Table 5
The description of experimental data sets.

| Data sets | Preprocessing | Number of objects | Number of attributes | Number of classes |
|---------------|-------------------------|-------------------|----------------------|-------------------|
| 1 Cancer | – | 683 | 9 | 2 |
| 2 Letter | – | 20,000 | 16 | 26 |
| 3 Monks | – | 1711 | 6 | 2 |
| 4 Spect | – | 267 | 22 | 2 |
| 5 HVWNT | Discretization | 600 | 99 | 5 |
| 6 Wine_lis | Discretization | 178 | 13 | 3 |
| 7 Ticdata2000 | Discretization | 5822 | 85 | 2 |
| 8 Molecula | – | 3190 | 60 | 3 |
| 9 Kr-vs-kp | – | 3196 | 36 | 2 |
| 10 Mushroom | Deleting missing values | 5644 | 22 | 2 |
| 11 BlogData_T | Discretization | 13,099 | 280 | 2 |
| 12 GFE | Discretization | 27,936 | 300 | 2 |

*HVWNT represents the data set 'Hill_Valley_without_noise_Testing', BlogData_T represents the top quarter of data set 'BlogData.Train', and GFE represents the data set 'Grammatical_facial_expression'.

Table 6
The comparison of the reducts obtained by Algorithms ACC-PR and AR-ST-PR.

| Data sets | Reducts obtained by ACC-PR | Reducts obtained by AR-ST-PR |
|--------------------|---|---|
| Cancer | 6,3,5,1 | 6,3,5,1 |
| Letter | 4,8,15,9,11, 13,12 ,10,16,3,14 | 4,8,15,9,11, 12,13 ,10,16,3,14 |
| Monks | 1,2,3,4,5,6 | 1,2,3,4,5,6 |
| Spect | 1,3,4,7,8,9,10,13, 14 ,16,19,20, 21 ,22,6,5 | 1,3,4,8,9,10,13,16,19,20, 22,11,2,21,5,7,14 |
| HVWNT | 1,38 | 1,38 |
| Wine_lis | 10, 13,7,2 | 10,7, 13,2 |
| Ticdata2000 | 2,5,43,44,47,55,59,68,80,83,31,18,9,39,54,49,64, 57,28,61,48,63,45,58 | 2,5,43,44,47,55,59,68,80,83,31,18,9,39,54,49,64, 28,57,61,45,48,58,63 |
| Molecula | 50,38,30,35,29,24,26, 58,18,11 | 50,38,30,35,29,24,26, 19,18,2 |
| Kr-vs-kp | 1,3,4,5,6,7,10,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36,11,9 | 1,3,4,5,6,7,10,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36,11,9 |
| Mushroom | 5,20,3 | 5,20,3 |

Table 7

The process of running ACC-PR and AR-CT-PR on the dataset Ticdata2000.

| | ACC-PR | | | | | AR-CT-PR | | | | |
|----------------|--------|------|------|-----|----------|----------|------|------|-----|----------|
| | Att | U | POS | NEG | Time | Att | U | POS | NEG | Time |
| Initialization | 85 | 5822 | 5701 | 121 | 0.2132 | 85 | 5822 | 5701 | 121 | 0.2132 |
| Com-table | – | – | – | – | – | 85 | 5158 | 5107 | 51 | 2.4960 |
| Core | 10 | 5822 | 5701 | 121 | 204.0536 | 10 | 5158 | 5107 | 50 | 160.6959 |
| Loop 1 | 11 | 2292 | 2171 | 121 | 4.6748 | 11 | 1906 | 1856 | 44 | 3.6504 |
| Loop 2 | 12 | 893 | 772 | 121 | 1.1024 | 12 | 702 | 658 | 23 | 0.8112 |
| Loop 3 | 13 | 337 | 216 | 121 | 0.2652 | 13 | 213 | 190 | 13 | 0.1664 |
| Loop 4 | 14 | 214 | 93 | 121 | 0.1560 | 14 | 97 | 84 | 8 | 0.0728 |
| Loop 5 | 15 | 190 | 69 | 121 | 0.1456 | 15 | 70 | 62 | 5 | 0.0520 |
| Loop 6 | 16 | 167 | 46 | 121 | 0.1248 | 16 | 47 | 42 | 3 | 0.0364 |
| Loop 7 | 17 | 155 | 34 | 121 | 0.1144 | 17 | 34 | 31 | 1 | 0.0260 |
| Loop 8 | 18 | 146 | 25 | 121 | 0.1248 | 18 | 24 | 23 | 1 | 0.0208 |
| Loop 9 | 19 | 140 | 19 | 121 | 0.1144 | 19 | 18 | 17 | 0 | 0.0156 |
| Loop 10 | 20 | 135 | 14 | 121 | 0.1196 | 20 | 12 | 12 | 0 | 0.0156 |
| Loop 11 | 21 | 131 | 10 | 121 | 0.1144 | 21 | 8 | 8 | 0 | 0.0104 |
| Loop 12 | 22 | 128 | 7 | 121 | 0.1092 | 22 | 6 | 6 | 0 | 0.0052 |
| Loop 13 | 23 | 125 | 4 | 121 | 0.1196 | 23 | 4 | 4 | 0 | 0.0052 |
| Loop 14 | 24 | 123 | 2 | 121 | 0.1144 | 24 | 2 | 2 | 0 | 0.0052 |
| Total time | | | | | 211.6664 | | | | | 168.2983 |

* ‘–’ represents Null.

Table 8

The comparative experiments about ACC-PR and AR-CT-PR on the data sets in Table 5.

| Dataset | Algorithm | Object | Ratio_C (%) | Cor/Red | Ini_T | Com_T | Cor_T | Red_T | Total_T | Ratio_T (%) |
|-------------|-----------|--------|-------------|---------|--------|----------|-------------|------------|-------------|-------------|
| Cancer | ACC-PR | 683 | 100.00 | 1/4 | 0.0052 | – | 0.0676 | 0.0208 | 0.0936 | 100.00 |
| | AR-CT-PR | 449 | 65.74 | 1/4 | 0.0052 | 0.0104 | 0.0312 | 0.0156 | 0.0624 | 66.67 |
| Letter | ACC-PR | 20,000 | 100.00 | 3/11 | 0.1091 | – | 25.3040 | 10.0308 | 35.4439 | 100.00 |
| | AR-CT-PR | 18,659 | 93.30 | 3/11 | 0.1091 | 1.5132 | 18.9852 | 8.6112 | 29.2187 | 82.44 |
| Monks | ACC-PR | 1711 | 100.00 | 6/6 | 0.0047 | – | 0.1352 | 0.0000 | 0.1399 | 100.00 |
| | AR-CT-PR | 432 | 25.25 | 6/6 | 0.0047 | 0.0052 | 0.0233 | 0.0000 | 0.0332 | 23.73 |
| Spect | ACC-PR | 267 | 100.00 | 14/16 | 0.0033 | – | 0.1397 | 0.0127 | 0.1557 | 100.00 |
| | AR-CT-PR | 219 | 82.02 | 14/16 | 0.0033 | 0.0104 | 0.0983 | 0.0020 | 0.1140 | 73.22 |
| HVVNT | ACC-PR | 600 | 100.00 | 0/2 | 0.0294 | – | 5.8276 | 0.1040 | 5.9610 | 100.00 |
| | AR-CT-PR | 78 | 13.00 | 0/2 | 0.0294 | 0.0897 | 0.4166 | 0.0223 | 0.5590 | 9.38 |
| Wine-lis | ACC-PR | 178 | 100.00 | 1/4 | 0.0020 | – | 0.0323 | 0.0090 | 0.0433 | 100.00 |
| | AR-CT-PR | 126 | 70.79 | 1/4 | 0.0020 | 0.0052 | 0.0200 | 0.0027 | 0.0299 | 69.05 |
| Ticdata2000 | ACC-PR | 5822 | 100.00 | 10/24 | 0.2132 | – | 204.0536 | 7.3996 | 211.6664 | 100.00 |
| | AR-CT-PR | 5158 | 88.59 | 10/24 | 0.2132 | 2.4960 | 160.6959 | 4.8932 | 168.2983 | 79.51 |
| Molecular | ACC-PR | 3190 | 100.00 | 0/10 | 0.0780 | – | 21.6372 | 7.0876 | 28.8028 | 100.00 |
| | AR-CT-PR | 3005 | 94.20 | 0/10 | 0.0780 | 0.3692 | 19.1880 | 7.0148 | 26.6500 | 92.53 |
| Kr-vs-kp | ACC-PR | 3196 | 100.00 | 27/29 | 0.0400 | – | 14.7829 | 0.0052 | 14.8281 | 100.00 |
| | AR-CT-PR | 3196 | 100.00 | 27/29 | 0.0400 | 0.4400 | 14.7829 | 0.0052 | 15.2681 | 102.97 |
| Mushroom | ACC-PR | 5644 | 100.00 | 0/3 | 0.0533 | – | 11.7412 | 1.6276 | 13.4221 | 100.00 |
| | AR-CT-PR | 5644 | 100.00 | 0/3 | 0.0533 | 0.6117 | 11.7412 | 1.6276 | 14.0338 | 104.56 |
| BlogData_T | ACC-PR | 13,099 | 100.00 | 0/90 | 0.9516 | – | 32945.1473 | 225.6699 | 33171.7688 | 100.00 |
| | AR-CT-PR | 8771 | 66.96 | 0/90 | 0.9516 | 118.6858 | 5853.0885 | 145.7193 | 6118.4452 | 18.44 |
| GFE | ACC-PR | 27,936 | 100.00 | 183/204 | 2.4960 | – | 202971.4707 | 11310.6804 | 214284.6471 | 100.00 |
| | AR-CT-PR | 18,156 | 64.99 | 183/204 | 2.4960 | 767.9113 | 83720.5454 | 244.6396 | 84735.5923 | 39.54 |

the consuming time of AR-CT-PR is much less than the one of consuming time of ACC-PR on the most of the data sets in Table 5. However, AR-CT-PR consumes more time than ACC-PR on the two data sets Kr-vs-kp and Mushroom, which is caused by the reason that AR-CT-PR need the time to compact data though these data sets are unchanged after being compacted. It is should be pointed that we can see based on result in Table 8, the ratio of the time consuming of ACC-PR and AR-CT-PR has some correlation with the ratio of the size of a data set and that of its compacted version.

5.2. Experiments about ACC-SCE and AR-CT-SCE

The comparative experiments between Algorithm ACC-SCE and Algorithm AR-CT-SCE on the data sets in Table 5 are listed

in Tables 9 and 10. Table 9 shows the concrete process of running ACC-SCE and AR-CT-SCE on data set Ticdata2000. In Table 9, the first row (Row ‘Initialization’) indicates the profile of Ticdata2000, which exhibits the information of attribute (Column ‘Att’), universe (Column ‘U’), positive region (Column ‘POS’), negative region (Column ‘NEG’), and the time of reading Ticdata2000 (Column ‘Time’); the second row shows the profile of the compacted Ticdata2000 and the consuming time of compacting Ticdata2000, and the columns about ACC-SCE in this row are Null because there is on the step in ACC-SCE; the third row shows the profiles of Ticdata2000 and the compacted Ticdata2000 after deleting the positive region of core with respect to decision attribute, and the consuming time of computing core by ACC-SCE and AR-CT-SCE. The rows between Loop 1

Table 9

The process of running ACC-SCE and AR-CT-SCE on the dataset Ticdata2000.

| | ACC-SCE | | | | | AR-CT-SCE | | | | |
|----------------|---------|------|------|-----|----------|-----------|------|------|-----|----------|
| | Att | U | POS | NEG | Time | Att | U | POS | NEG | Time |
| Initialization | 85 | 5822 | 5701 | 121 | 0.2132 | 85 | 5822 | 5701 | 51 | 0.2132 |
| Com-table | – | – | – | – | – | 85 | 5158 | 5107 | 51 | 2.4960 |
| Core | 10 | 2292 | 2171 | 121 | 208.1096 | 10 | 1907 | 1856 | 51 | 164.7675 |
| Loop 1 | 11 | 893 | 772 | 121 | 4.8932 | 11 | 709 | 658 | 51 | 3.6816 |
| Loop 2 | 12 | 337 | 216 | 121 | 1.1804 | 12 | 241 | 190 | 51 | 0.8372 |
| Loop 3 | 13 | 214 | 93 | 121 | 0.2808 | 13 | 135 | 84 | 51 | 0.1612 |
| Loop 4 | 14 | 190 | 69 | 121 | 0.1664 | 14 | 113 | 62 | 51 | 0.0780 |
| Loop 5 | 15 | 167 | 46 | 121 | 0.1456 | 15 | 93 | 42 | 51 | 0.0468 |
| Loop 6 | 16 | 155 | 34 | 121 | 0.1404 | 16 | 82 | 31 | 51 | 0.0416 |
| Loop 7 | 17 | 146 | 25 | 121 | 0.1248 | 17 | 74 | 23 | 51 | 0.0260 |
| Loop 8 | 18 | 140 | 19 | 121 | 0.1248 | 18 | 68 | 17 | 51 | 0.0260 |
| Loop 9 | 19 | 135 | 14 | 121 | 0.1248 | 19 | 63 | 12 | 51 | 0.0156 |
| Loop 10 | 20 | 131 | 10 | 121 | 0.1196 | 20 | 59 | 8 | 51 | 0.0156 |
| Loop 11 | 21 | 128 | 7 | 121 | 0.1248 | 21 | 57 | 6 | 51 | 0.0156 |
| Loop 12 | 22 | 125 | 4 | 121 | 0.1196 | 22 | 55 | 4 | 51 | 0.0052 |
| Loop 13 | 23 | 123 | 2 | 121 | 0.1248 | 23 | 53 | 2 | 51 | 0.0052 |
| Loop 14 | 24 | 121 | 0 | 121 | 0.1196 | 24 | 51 | 0 | 51 | 0.0052 |
| Total time | | | | | 216.1124 | | | | | 172.4375 |

Table 10

The comparative experiments about ACC-SCE and AR-CT-SCE on the data sets in Table 5.

| Dataset | Algorithm | Object | Ratio_C (%) | Cor/Red | Ini_T | Com_T | Cor_T | Red_T | Total_T | Ratio_T (%) |
|-------------|-----------|--------|-------------|---------|--------|----------|-------------|------------|-------------|-------------|
| Cancer | ACC-SCE | 683 | 100.00 | 1/4 | 0.0052 | – | 0.0710 | 0.0233 | 0.0995 | 100.00 |
| | AR-CT-SCE | 449 | 65.74 | 1/4 | 0.0052 | 0.0104 | 0.0383 | 0.0157 | 0.0696 | 69.95 |
| Letter | ACC-SCE | 20,000 | 100.00 | 3/11 | 0.1091 | – | 24.3238 | 9.7500 | 34.1829 | 100.00 |
| | AR-CT-SCE | 18,659 | 93.30 | 3/11 | 0.1091 | 1.5132 | 19.3752 | 8.3938 | 29.3913 | 85.98 |
| Monks | ACC-SCE | 1711 | 100.00 | 6/6 | 0.0047 | – | 0.1327 | 0.0000 | 0.1374 | 100.00 |
| | AR-CT-SCE | 432 | 25.25 | 6/6 | 0.0047 | 0.0052 | 0.0203 | 0.0000 | 0.0302 | 21.98 |
| Spect | ACC-SCE | 267 | 100.00 | 14/16 | 0.0033 | – | 0.1147 | 0.0120 | 0.1300 | 100.00 |
| | AR-CT-SCE | 219 | 82.02 | 14/16 | 0.0033 | 0.0083 | 0.0917 | 0.0017 | 0.1050 | 80.77 |
| HWNT | ACC-SCE | 600 | 100.00 | 0/2 | 0.0294 | – | 6.0843 | 0.4163 | 6.5300 | 100.00 |
| | AR-CT-SCE | 78 | 13.00 | 0/2 | 0.0294 | 0.0897 | 0.4370 | 0.0430 | 0.5591 | 9.17 |
| Wine-lis | ACC-SCE | 178 | 100.00 | 1/4 | 0.0020 | – | 0.0340 | 0.0107 | 0.0467 | 100.00 |
| | AR-CT-SCE | 126 | 70.79 | 1/4 | 0.0020 | 0.0052 | 0.0233 | 0.0053 | 0.0358 | 76.66 |
| Ticdata2000 | ACC-SCE | 5822 | 100.00 | 10/24 | 0.2132 | – | 208.1096 | 7.7896 | 216.1124 | 100.00 |
| | AR-CT-SCE | 5158 | 88.59 | 10/24 | 0.2132 | 2.4960 | 164.7675 | 4.9608 | 172.4375 | 79.79 |
| Molecular | ACC-SCE | 3190 | 100.00 | 0/11 | 0.0780 | – | 23.4272 | 7.6874 | 31.1926 | 100.00 |
| | AR-CT-SCE | 3005 | 94.20 | 0/11 | 0.0780 | 0.3692 | 19.5794 | 6.7906 | 26.8172 | 85.97 |
| Kr-vs-kp | ACC-SCE | 3196 | 100.00 | 27/29 | 0.0400 | – | 15.2685 | 0.0087 | 15.3172 | 100.00 |
| | AR-CT-SCE | 3196 | 100.00 | 27/29 | 0.0400 | 0.4400 | 15.2685 | 0.0087 | 15.7572 | 102.87 |
| Mushroom | ACC-SCE | 5644 | 100.00 | 0/4 | 0.0533 | – | 12.7507 | 4.1062 | 16.9102 | 100.00 |
| | AR-CT-SCE | 5644 | 100.00 | 0/4 | 0.0533 | 0.6117 | 12.7507 | 4.1062 | 17.5219 | 103.61 |
| BlogData_T | ACC-PR | 13,099 | 100.00 | 0/14 | 0.9516 | – | 29256.3127 | 206.5131 | 29463.7774 | 100.00 |
| | AR-CT-PR | 8771 | 66.96 | 0/14 | 0.9516 | 118.6858 | 5883.9605 | 45.6206 | 6049.2185 | 20.53 |
| GFE | ACC-SCE | 27,936 | 100.00 | 183/204 | 2.4960 | – | 240439.2682 | 11035.2007 | 251476.9649 | 100.00 |
| | AR-CT-SCE | 18,156 | 64.99 | 183/204 | 2.4960 | 767.9113 | 98933.9096 | 239.0860 | 99943.4029 | 39.74 |

and Loop14 show in each loop, the profiles of Ticdata2000 and the compacted Ticdata2000 and the consuming time of ACC-SCE and AR-CT-SCE. From Table 9, we can see that the number of objects (5158) in the compacted Ticdata2000 is less than the one (5822) in Ticdata2000, meanwhile the time of computing core and reduct by AR-CT-SCE is much less than the one by ACC-SCE. These experimental results indicate that AR-CT-SCE is more efficient than ACC-SCE, if a decision can be significantly compacted.

Furthermore, the comparison of running ACC-SCE and AR-CT-SCE on all data sets in Table 5 is listed in Table 10. In this table, Column ‘Algorithm’ represents the employed algorithms, Column ‘Object’ represents the number of objects in the original and the compacted data sets, Column ‘Ratio_C’ shows the ratio of the number of objects in the compacted data sets and the one in

its original version, Columns ‘Ini_T’, ‘Com_T’, ‘Cor_T’ and ‘Red_T’ represent the read time, the time of compacting data sets, the time of computing core and the time of computing reduct respectively, Column ‘Total_T’ represents the sum of them, Column ‘Ratio_T’ represents the ratio of the consuming time of running ACC-SCE and the one of running AR-CT-SCE. From Table 10, we can see that the consuming time of finding reducts by AR-CT-SCE is much less than the one by ACC-SCE on the most of the data sets in Table 5. However, for data sets Kr-vs-kp and Mushroom, AR-CT-SCE consumes more than the one of running ACC-SCE, because these two data sets are unchanged after the two data sets are compacted. It should be pointed that we can see based on result in Table 10, the ratio of the time consuming of ACC-SCE and AR-CT-SCE has some correlation with the ratio of the size of a data set and that of its compacted version.

5.3. Experiments about ACC-CCE and AR-CT-CCE

In this section, Tables 11 and 12 show the experimental results of the comparison between ACC-CCE and AR-CT-CCE on the data sets in Table 5. Table 11 indicates the concrete process of running ACC-CCE and AR-CT-CCE on the data set Ticdata2000. In Table 11, the first row (Row 'Initialization') shows the initial status of Ticdata2000, in which Column 'Att' represents attribute, Column 'U' represents universe, Column 'POS' represents positive region, Column 'NEG' represents negative region, and Column 'Time' represents the time of reading Ticdata2000; the second row shows the profile of the compacted Ticdata2000 and the consuming time of compacting Ticdata2000, and the columns about ACC-CCE in this row are Null because there is no the step of compacting data set in

ACC-CCE; the third row shows the consuming time of computing core by ACC-CCE and AR-CT-CCE; the rows between Loop1 and Loop14 show in each loop, the profiles of Ticdata2000 and the compacted Ticdata2000 and the consuming time of ACC-CCE and AR-CT-CCE. From Table 11, we can see that the number of objects (5158) in the compacted Ticdata2000 is less than the one (5822) in Ticdata2000, and the time of computing core and reduct by means of AR-CT-CCE is much less than the one by means of ACC-CCE. The results indicate that AR-CT-CCE is more efficient than ACC-CCE if a decision can be significantly compacted.

Furthermore, the experimental results of running ACC-CCE and AR-CT-CCE on all data sets in Table 5 are listed in Table 12. In this table, Column 'Algorithm' represents the employed algorithms, Column 'Object' represents the number of objects in the original

Table 11
The process of running ACC-CCE and AR-CT-CCE on the dataset Ticdata2000.

| | ACC-CCE | | | | | AR-CT-CCE | | | | |
|----------------|---------|------|------|-----|----------|-----------|------|------|-----|----------|
| | Att | U | POS | NEG | Time | Att | U | POS | NEG | Time |
| Initialization | 85 | 5822 | 5701 | 121 | 0.2036 | 85 | 5822 | 5701 | 51 | 0.2036 |
| Com-table | - | - | - | - | - | 85 | 5158 | 5107 | 51 | 2.5024 |
| Core | 10 | 2292 | 2171 | 121 | 208.9198 | 10 | 1907 | 1856 | 51 | 165.0301 |
| Loop 1 | 11 | 893 | 772 | 121 | 4.9036 | 11 | 709 | 658 | 51 | 3.7603 |
| Loop 2 | 12 | 337 | 216 | 121 | 1.1705 | 12 | 241 | 190 | 51 | 0.8384 |
| Loop 3 | 13 | 214 | 93 | 121 | 0.2028 | 13 | 135 | 84 | 51 | 0.1895 |
| Loop 4 | 14 | 190 | 69 | 121 | 0.2874 | 14 | 113 | 62 | 51 | 0.0770 |
| Loop 5 | 15 | 167 | 46 | 121 | 0.1478 | 15 | 93 | 42 | 51 | 0.0489 |
| Loop 6 | 16 | 155 | 34 | 121 | 0.1331 | 16 | 82 | 31 | 51 | 0.0328 |
| Loop 7 | 17 | 146 | 25 | 121 | 0.1259 | 17 | 74 | 23 | 51 | 0.0298 |
| Loop 8 | 18 | 140 | 19 | 121 | 0.1249 | 18 | 68 | 17 | 51 | 0.0171 |
| Loop 9 | 19 | 135 | 14 | 121 | 0.1232 | 19 | 63 | 12 | 51 | 0.0157 |
| Loop 10 | 20 | 131 | 10 | 121 | 0.1351 | 20 | 59 | 8 | 51 | 0.0147 |
| Loop 11 | 21 | 128 | 7 | 121 | 0.1302 | 21 | 57 | 6 | 51 | 0.0104 |
| Loop 12 | 22 | 125 | 4 | 121 | 0.1314 | 22 | 55 | 4 | 51 | 0.0063 |
| Loop 13 | 23 | 123 | 2 | 121 | 0.1193 | 23 | 53 | 2 | 51 | 0.0127 |
| Loop 14 | 24 | 121 | 0 | 121 | 0.1249 | 24 | 51 | 0 | 51 | 0.0013 |
| Total time | | | | | 216.9931 | | | | | 172.8006 |

Table 12
The comparative experiments about ACC-CCE and AR-CT-CCE on the data sets in Table 5.

| Dataset | Algorithm | Object | Ratio_C (%) | Cor/Red | Ini_T | Com_T | Cor_T | Red_T | Total_T | Ratio_T (%) |
|-------------|-----------|--------|-------------|---------|--------|----------|-------------|------------|-------------|-------------|
| Cancer | ACC-CCE | 683 | 100.00 | 1/4 | 0.0052 | - | 0.0713 | 0.0253 | 0.1018 | 100.00 |
| | AR-CT-CCE | 449 | 65.74 | 1/4 | 0.0052 | 0.0104 | 0.0383 | 0.0120 | 0.0659 | 60.96 |
| Letter | ACC-CCE | 20,000 | 100.00 | 3/12 | 0.1091 | - | 24.0524 | 9.6816 | 33.8431 | 100.00 |
| | AR-CT-CCE | 18,659 | 93.30 | 3/12 | 0.1091 | 1.5132 | 19.0974 | 8.6600 | 29.3797 | 86.81 |
| Monks | ACC-CCE | 1711 | 100.00 | 6/6 | 0.0047 | - | 0.1410 | 0.0000 | 0.1457 | 100.00 |
| | AR-CT-CCE | 432 | 25.25 | 6/6 | 0.0047 | 0.0052 | 0.0200 | 0.0000 | 0.0299 | 20.52 |
| Spect | ACC-CCE | 267 | 100.00 | 14/16 | 0.0033 | - | 0.1083 | 0.0123 | 0.1239 | 100.00 |
| | AR-CT-CCE | 219 | 82.02 | 14/16 | 0.0033 | 0.0083 | 0.0880 | 0.0017 | 0.1013 | 81.76 |
| HVVNT | ACC-CCE | 600 | 100.00 | 0/3 | 0.0294 | - | 6.0056 | 0.4007 | 6.4357 | 100.00 |
| | AR-CT-CCE | 78 | 13.00 | 0/3 | 0.0294 | 0.0897 | 0.4370 | 0.0367 | 0.5928 | 9.21 |
| Wine-lis | ACC-CCE | 178 | 100.00 | 1/4 | 0.0020 | - | 0.0333 | 0.0097 | 0.0450 | 100.00 |
| | AR-CT-CCE | 126 | 70.79 | 1/4 | 0.0020 | 0.0052 | 0.0230 | 0.0047 | 0.0349 | 77.56 |
| Ticdata2000 | ACC-CCE | 5822 | 100.00 | 10/24 | 0.2132 | - | 208.9198 | 7.8601 | 216.9931 | 100.00 |
| | AR-CT-CCE | 5158 | 88.59 | 10/24 | 0.2132 | 2.4960 | 164.7675 | 5.0549 | 172.8006 | 79.63 |
| Molecular | ACC-CCE | 3190 | 100.00 | 0/10 | 0.0780 | - | 22.8773 | 7.1311 | 30.0864 | 100.00 |
| | AR-CT-CCE | 3005 | 94.20 | 0/10 | 0.0780 | 0.3692 | 20.5758 | 6.6440 | 27.6670 | 91.96 |
| Kr-vs-kp | ACC-CCE | 3196 | 100.00 | 27/29 | 0.0400 | - | 15.3958 | 0.0093 | 15.4451 | 100.00 |
| | AR-CT-CCE | 3196 | 100.00 | 27/29 | 0.0400 | 0.4400 | 15.3958 | 0.0093 | 15.8851 | 102.84 |
| Mushroom | ACC-CCE | 5644 | 100.00 | 0/4 | 0.0533 | - | 12.6547 | 4.1039 | 16.8119 | 100.00 |
| | AR-CT-CCE | 5644 | 100.00 | 0/4 | 0.0533 | 0.6117 | 12.6547 | 4.1039 | 17.4236 | 103.63 |
| BlogData_T | ACC-PR | 13,099 | 100.00 | 0/14 | 0.9516 | - | 29730.0842 | 212.9091 | 29943.9449 | 100.00 |
| | AR-CT-PR | 8771 | 66.96 | 0/14 | 0.9516 | 118.6858 | 5880.9273 | 47.485,716 | 6048.0504 | 20.19 |
| GFE | ACC-PR | 27,936 | 100.00 | 183/204 | 2.4960 | - | 222024.0392 | 11231.7701 | 233258.3053 | 100.00 |
| | AR-CT-CCE | 18,156 | 64.99 | 183/204 | 2.4960 | 767.9113 | 81408.8343 | 263.1304 | 82442.3450 | 36.67 |

Table 13

The comparative experiments about AR-mRMR-PR and AR-CT-PR on the data sets in Table 5.

| Dataset | N-attribute | AR-mRMR-PR | | AR-CT-PR | |
|-------------|-------------|------------|-------------|----------|------------|
| | | N-reduct | T-reduct | N-reduct | T-reduct |
| Cancer | 9 | 4 | 0.2530 | 4 | 0.0624 |
| Letter | 16 | 14 | 84.1218 | 11 | 29.2187 |
| Monks | 6 | 6 | 0.1310 | 6 | 0.0332 |
| Spect | 22 | 21 | 0.4290 | 16 | 0.1140 |
| HVWNT | 99 | 2 | 0.3031 | 2 | 0.5590 |
| Wine-lis | 13 | 8 | 0.0624 | 4 | 0.0299 |
| Ticdata2000 | 85 | 82 | 1796.006 | 24 | 168.2983 |
| Molecular | 60 | 13 | 22.4743 | 10 | 26.6500 |
| Kr-vs-kp | 36 | 35 | 26.9615 | 29 | 15.2681 |
| Mushroom | 22 | 9 | 6.9888 | 3 | 14.0338 |
| BlogData_T | 280 | 127 | 13464.1868 | 90 | 6118.4452 |
| GFE | 300 | 266 | 154539.3754 | 204 | 84735.5923 |

Table 14

The comparative experiments about AR-mRMR-SCE and AR-CT-SCE on the data sets in Table 5.

| Dataset | N-attribute | AR-mRMR-SCE | | AR-CT-SCE | |
|-------------|-------------|-------------|-------------|-----------|------------|
| | | N-reduct | T-reduct | N-reduct | T-reduct |
| Cancer | 9 | 4 | 0.1500 | 4 | 0.0696 |
| Letter | 16 | 14 | 72.6491 | 12 | 29.3913 |
| Monks | 6 | 6 | 0.1290 | 6 | 0.0302 |
| Spect | 22 | 21 | 0.4470 | 16 | 0.1050 |
| HVWNT | 99 | 2 | 0.2770 | 2 | 0.5591 |
| Wine-lis | 13 | 8 | 0.0440 | 4 | 0.0358 |
| Ticdata2000 | 85 | 82 | 1842.3973 | 24 | 172.4375 |
| Molecular | 60 | 13 | 20.6622 | 11 | 26.8172 |
| Kr-vs-kp | 36 | 35 | 26.0074 | 29 | 15.7572 |
| Mushroom | 22 | 9 | 6.7704 | 3 | 17.5219 |
| BlogData_T | 280 | 128 | 13500.8465 | 14 | 6049.2185 |
| GFE | 300 | 294 | 169433.3550 | 204 | 99943.4029 |

and the compacted data sets, Column ‘Ratio_C’ shows the ratio of the number of objects in the compacted data sets and the one in its original version, Columns ‘Ini_T’, ‘Com_T’, ‘Cor_T’ and ‘Red_T’ represent the read time, the time of compacting data sets, the time of computing core and the time of computing reduct respectively, Column ‘Total_T’ represents the sum of them, Column ‘Ratio_T’ represents the ratio of the consuming time of running ACC-CCE and the one of running AR-CT-CCE. From the Table 12, we can see that the consuming time of AR-CT-CCE is much less than the one of ACC-CCE on the most of the data sets in Table 5. However, AR-CT-CCE consumes more time than ACC-CCE on data sets Kr-vs-kp and Mushroom, which is caused by the reason that AR-CT-SCE need to compact the two data sets but they are unchanged after being compacted. It is should be pointed that we can see based on result in Table 12, the ratio of the time consuming of ACC-CCE and AR-CT-CCE has some correlation with the ratio of the size of a data set and that of its compacted version.

5.4. Comparative experiments between the attribute reduction algorithm based on mRMR and our algorithms

To better indicate the effectiveness of our algorithms, in this section, we will carry out the comparative experiment between the attribute reduction algorithm based on mRMR (minimal-redundancy-maximal-relevance criterion) and our proposed algorithms. In [32], the feature selection algorithm base on mRMR was proposed, which is representative feature selection algorithm. The algorithm aims at selecting the feature subset that can leads to

Table 15

The comparative experiments about AR-mRMR-CCE and AR-CT-CCE on the data sets in Table 5.

| Dataset | N-attribute | AR-mRMR-PR | | AR-CT-PR | |
|-------------|-------------|------------|-------------|----------|------------|
| | | N-reduct | T-reduct | N-reduct | T-reduct |
| Cancer | 9 | 4 | 0.1100 | 4 | 0.0659 |
| Letter | 16 | 14 | 72.7031 | 11 | 29.3797 |
| Monks | 6 | 6 | 0.1290 | 6 | 0.0299 |
| Spect | 22 | 21 | 0.4250 | 16 | 0.1013 |
| HVWNT | 99 | 2 | 0.0450 | 2 | 0.5928 |
| Wine-lis | 13 | 8 | 0.2600 | 4 | 0.0349 |
| Ticdata2000 | 85 | 82 | 1822.7658 | 24 | 172.8006 |
| Molecular | 60 | 13 | 20.6402 | 10 | 27.6670 |
| Kr-vs-kp | 36 | 35 | 25.6295 | 29 | 15.8851 |
| Mushroom | 22 | 9 | 7.1604 | 3 | 17.4236 |
| BlogData_T | 280 | 128 | 13452.3675 | 14 | 6048.0504 |
| GFE | 300 | 294 | 168997.1145 | 204 | 82442.3450 |

promising improvement on classification accuracy. Since this paper focus on investigating the attribute reduction algorithms that aim at preserving the discernibility of data sets in some sense (for example, in the sense of positive region, Shannon entropy and complement entropy), we need to modify the feature selection algorithm in [32] to make it become comparable with our proposed algorithms. The modified algorithm adapts the filter strategy, forward greedy search, and the same stop criteria as those in our algorithms, whose description is given as follow.

Algorithm 7. Attribute reduction algorithm based on mRMR (AR-mRMR-Δ)

Input: Decision table $DT = (U, C \cup D)$;

Output: One reduct red .

Step 1: $red \leftarrow \emptyset$; // red is the pool to conserve the selected attributes;

Step 2: While $EF_{\Delta}^U(red, D) \neq EF_{\Delta}^U(C, D)$ Do// This provides a stopping criterion

$$\{red \leftarrow red \cup \{a_0\}, \text{ where } (I(a_0, D) - \frac{1}{m-1} \sum_{a_i \in red} I(a_0, a_i)) = \max\{(I(a_k, D) - \frac{1}{m-1} \sum_{a_i \in red} I(a_k, a_i)), a_k \in C - red\};$$

Step 3: return red and end,

where $I(a_i, a_j) = \sum_{i=1}^m \sum_{j=1}^n \frac{|X_i \cap Y_j|}{|U|} \log_2 \frac{|X_i \cap Y_j|}{|X_i| |Y_j|}$, $X_i \in U/a_i$, $Y_j \in U/a_j$, $EF_{\Delta}^U(B, D) = EF_{\Delta}^U(C, D)$ is the stopping criterion, $\Delta = \{PR, SCE, CCE\}$. For example, while the positive region is employed as the evaluation function, $EF_{PR}^U(B, D)$ is equal to $POS_B^U(D)$ and $EF_{PR}^U(C, D)$ is equal to $POS_C^U(D)$.

The results of comparative experiments between AR-mRMR-PR and AR-CT-PR, between AR-mRMR-SCE and AR-CT-SCE, and between AR-mRMR-CCE and AR-CT-CCE are shown in Tables 13–15, respectively. From Table 13, we can see that the reducts obtained by Algorithm AR-CT-PR are more optimal than that of Algorithm AR-mRMR-PR (from the perspective of the number of attributes in the selected reducts), whilst the time consuming of conducting Algorithm AR-CT-PR is shorter than that of conducting Algorithm AR-mRMR-PR on the most of data sets (except for HVWNT, Molecular and Mushroom). The similar experimental results appear in the Tables 14 and 15, which indicate, in most cases, the Algorithms AR-CT-SCE and AR-CT-CCE are also more optimal than Algorithms AR-mRMR-SCE and AR-mRMR-CCE respectively, in the aspect of the number of selected attributes and time-consuming. However, it is should be pointed that our

proposed Algorithms are only superior than that based on mRMR from the perspective of acquiring reducts.

6. Conclusion

In this paper, we first pointed out that the attribute reduction algorithm for a simplified decision table has two key faults as follows: (1) The reducts obtained from a simplified decision table are different with the ones obtained from its original version; (2) The reducts in the sense of Shannon entropy and complement entropy cannot be obtained from a simplified decision table. We further found out that the reason that results in these two faults is essentially the lose of the values on decision attributes while a decision table is simplified. To solve these two issues, we proposed the compacted table which preserves all the information coming from a decision table. Several theorems are introduced to theoretically demonstrate the sequence preserving of inner and outer significance after a decision table is compacted, and three new attribute reduction algorithms based on compacted decision tables are designed to find the reducts in the sense of positive region, Shannon entropy and complement entropy. Finally, we carried out several numerical experiments to indicate the effectiveness and efficiency of these proposed algorithms.

Acknowledgements

The research was supported by the National Natural Science Foundation of China (Nos. 61303008, 61202018, 61432011, and U1435212), the National Key Basic Research and Development Program of China (973) (No. 2013CB329404), and the Natural Science Foundation of Shanxi Province, China (No. 2013021018-1), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi, China (No. 2013102).

References

- [1] V.S. Babu, P. Viswanath, Rough-fuzzy weighted k-nearest leader classifier for large datasets, *Pattern Recogn.* 42 (2009) 1719–1731.
- [2] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, *AI* 97 (1–2) (1997) 245–271.
- [3] D.G. Chen, Q.H. Hu, Y.P. Yang, Parameterized attribute reduction with Gaussian kernel based fuzzy rough sets, *Inf. Sci.* 181 (23) (2011) 5169–5179.
- [4] T.Q. Deng, C.D. Yang, X.F. Wang, A reduct derived from feature selection, *Pattern Recogn. Lett.* 33 (2012) 1638–1646.
- [5] D.C. Feng, F. Chen, W.L. Xu, Supervised feature subset selection with ordinal optimization, *Knowl.-Based Syst.* 56 (2014) 123–140.
- [6] J.W. Grzymala-Busse, Algebraic properties of knowledge representation systems, in: *Proc. of the ACM SIGART International Symposium on Methodologies for Intelligent Systems, 1986*, pp. 432–440.
- [7] J.W. Grzymala-Busse, W.A.J. Sedelow, On rough sets and information system homomorphism, *Bull. Pol. Acad. Sci. Tech. Sci.* 36 (1988) 233–239.
- [8] I. Guyon, A. Elisseeff, An introduction to variable feature selection, *J. Mach. Learn. Res.* 3 (2003) 1157–1182.
- [9] S. Gunal, R. Edizkan, Subspace based feature selection for pattern recognition, *Inf. Sci.* 178 (2008) 3716–3726.
- [10] X.H. Hu, N. Cercone, Learning in relational databases: a rough set approach, *Int. J. Comput. Intell.* 11 (2) (1995) 323–338.
- [11] Q.H. Hu, D.R. Yu, Z.X. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, *Pattern Recogn. Lett.* 27 (2006) 414–423.
- [12] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, *Pattern Recogn.* 40 (2007) 3509–3521.
- [13] Q.H. Hu, J.F. Liu, D.R. Yu, Mixed feature selection based on granulation and approximation, *Knowl.-Based Syst.* 21 (4) (2008) 294–304.
- [14] R. Jenatton, J.Y. Audibert, F. Bach, Structured variable selection with sparsity-inducing norms, *J. Mach. Learn. Res.* 12 (2011) 2777–2824.
- [15] X.Y. Jia, W.H. Liao, Z.M. Tang, L. Shang, Minimum cost attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 219 (2013) 151–167.
- [16] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: *Proceedings AAAI-92*, San Jose, CA, MIT Press, Cambridge, MA, 1992, pp. 129–134.
- [17] K. Kim, L.A. Rendell, A practical approach to feature selection, in: *Proceedings 9th International Conference on Machine Learning*, Aberdeen, Scotland, Morgan Kaufman, Los Altos, CA, 1992.
- [18] R. Kohavi, G.H. John, Wrappers for feature subset selection, *Artif. Intell.* 97 (1–2) (1997) 273–324.
- [19] D.Y. Li, Y.C. Ma, Invariant characters of information systems under some homomorphisms, *Inf. Sci.* 129 (2000) 211–220.
- [20] J.Y. Liang, K.S. Chin, C.Y. Dang, C.M. Yam Richid, A new method for measuring uncertainty and fuzziness in rough set theory, *Int. J. Gen. Syst.* 31 (4) (2002) 331–342.
- [21] J.Y. Liang, Z.Z. Shi, D.Y. Li, M.J. Wierman, The information entropy, rough entropy and knowledge granulation in incomplete information table, *Int. J. Gen. Syst.* 35 (6) (2006) 641–654.
- [22] J.Y. Liang, J.H. Wang, Y.H. Qian, A new measure of uncertainty based on knowledge granulation for rough sets, *Inf. Sci.* 179 (4) (2009) 458–470.
- [23] J.Y. Liang, J.R. Mi, W. Wei, F. Wang, An accelerator for attribute reduction based on perspective of objects and attributes, *Knowl.-Based Syst.* 44 (2013) 90–100.
- [24] F.Y. Lin, D.R. Liang, C.C. Yeh, J.C. Huang, Novel feature selection methods to financial distress prediction, *Expert Syst. Appl.* 41 (5) (2014) 2472–2483.
- [25] Y.J. Lin, J.J. Li, P.R. Lin, G.P. Lin, J.K. Chen, Feature selection via neighborhood multi-granulation fusion, *Knowl.-Based Syst.* 67 (2014) 162–168.
- [26] Z.C. Lu, Zh. Qin, Y.Q. Zhang, J. Fang, A fast feature selection approach based on rough set boundary regions, *Pattern Recogn. Lett.* 36 (2014) 81–88.
- [27] Z.Q. Meng, Z.Z. Shi, A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets, *Inf. Sci.* 179 (2009) 2774–2793.
- [28] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Boston, 1991.
- [29] Z. Pawlak, A. Skowron, Rudiments of rough sets, *Inf. Sci.* 177 (1) (2007) 3–27.
- [30] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, *Inf. Sci.* 177 (1) (2007) 41–73.
- [31] W. Pedrycz, G. Vukovich, Feature analysis through information granulation and fuzzy sets, *Pattern Recogn.* 35 (2002) 825–834.
- [32] H.C. Peng, F.H. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (2005) 1226–1238.
- [33] Y.H. Qian, J.Y. Liang, D.Y. Li, H.Y. Zhang, C.Y. Dang, Measures for evaluating the decision performance of a decision table in rough set theory, *Inf. Sci.* 178 (2008) 181–202.
- [34] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, *Artif. Intell.* 174 (2010) 597–618.
- [35] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, *Pattern Recogn.* 44 (2011) 1658–1670.
- [36] R. Sikora, S. Piramuthu, Framework for efficient feature selection in genetic algorithm based data mining, *Eur. J. Oper. Res.* 180 (2) (2007) 723–737.
- [37] Q. Shen, R. Jensen, Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring, *Pattern Recogn.* 37 (2004) 1351–1363.
- [38] D. Slezak, Approximate reducts in decision tables, Research report, Institute of Computer Science, Warsaw University of Technology, 1995.
- [39] D. Slezak, Approximate entropy reducts, *Fundamenta Informaticae* 53 (3–4) (2002) 365–390.
- [40] A. Skowron, C. Rauszer, The discernibility matrices and functions in information tables, *Intell. Decis. Support: Handbook Appl. Adv. Rough Set Theory* (1992) 331–362.
- [41] E.C.C. Tsang, D.G. Chen, D.S. Yeung, X.Z. Wang, J. Lee, Attributes reduction using fuzzy rough sets, *IEEE Trans. Fuzzy Syst.* 16 (5) (2008) 1130–1141.
- [42] C.Z. Wang, D.G. Chen, C. Wu, Q.H. Hu, Data compression with homomorphism in covering information systems, *Int. J. Approx. Reason.* 52 (2011) 519–525.
- [43] C.Z. Wang, D.G. Chen, Q.H. Hu, Fuzzy information systems and their homomorphisms, *Fuzzy Sets Syst.* 249 (2014) 128–138.
- [44] D.Q. Wang, H. Zhang, R. Liu, W.F. Lv, Feature selection based on term frequency and T-test for text categorization, in: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012*, pp. 1482–1486.
- [45] D.Q. Wang, H. Zhang, R. Liu, W.F. Lv, D.T. Wang, t-Test feature selection approach based on term frequency for text categorization, *Pattern Recogn. Lett.* 45 (2014) 1–10.
- [46] G.Y. Wang, H. Yu, D.C. Yang, Decision table reduction based on conditional information entropy, *Chinese J. Comput.* 25 (7) (2002) 759–766.
- [47] G.Y. Wang, J. Zhao, J.J. An, A comparative study of algebra viewpoint and information viewpoint in attribute reduction, *Fundamenta Informaticae* 68 (3) (2005) 289–301.
- [48] W. Wei, J.Y. Liang, Y.H. Qian, F. Wang, An attribute reduction approach and its accelerated version for hybrid data, in: *The 8th IEEE International Conference on Cognitive Informatics, 2009*, pp. 167–173.
- [49] W. Wei, J.Y. Liang, Y.H. Qian, F. Wang, C.Y. Dang, Comparative study of decision performance of decision tables induced by attribute reductions, *Int. J. Gen. Syst.* 39 (8) (2010) 813–838.
- [50] W. Wei, J.Y. Liang, Y.H. Qian, A comparative study of rough sets for hybrid data, *Inf. Sci.* 190 (2012) 1–16.
- [51] W. Wei, J.Y. Liang, J.H. Wang, Y.H. Qian, Decision-relative discernibility matrixes in the sense of entropies, *Int. J. Gen. Syst.* 42 (7) (2013) 721–738.
- [52] S.K.M. Wong, W. Ziako, On optimal decision rules in decision tables, *Bull. Polish Acad. Sci.* 33 (1985) 693–696.
- [53] Z.Y. Xu, Z.P. Liu, B.R. Yang, et al., A quick attribute reduction algorithm with complexity of max, *Chinese J. Comput.* 29 (3) (2006) 611–615.

- [54] M. Yang, An incremental updating algorithm of the computation of a core based on the improved discernibility matrix, *Chinese J. Comput.* 29 (3) (2006) 407–413.
- [55] Y.Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 178 (17) (2008) 3356–3373.
- [56] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, *Inf. Sci.* 179 (2009) 867–882.
- [57] D.Y. Ye, Z.J. Chen, A new discernibility matrix and the computation of a core, *Acta Electron. Sinica* 30 (7) (2002) 1086–1088.