An accelerator for attribute reduction based on perspective of objects and attributes

Jiye Liang, Junrong Mi, Wei Wei, Feng Wang

School of Management, Shanxi University, Taiyuan, 030006 Shanxi, China

Keywords: Feature selection, Accelerating algorithm, Attribute reduction, Rough set, Large scale data

Abstract

Feature selection is an active area of research in pattern recognition, machine learning and artificial intelligence, which greatly improves the performance of forecasting or classification. In rough set theory, attribute reduction, as a special form of feature selection, aims to retain the discernability of the original attribute set. To solve this problem, many heuristic attribute reduction algorithms have been proposed in the literature. However, these methods are computationally time-consuming for large scale datasets. Recently, an accelerator was introduced by computing reducts on gradually reducing the size of the universe. Although the accelerator can considerably shorten the computational time, it remains a challenging issue. To further enhance the efficiency of these algorithms, we develop a new accelerator for attribute reduction, which simultaneously reduces the size of the universe and the number of attributes at each iteration of the process of reduction. Based on the new accelerator, several representative heuristic attribute reduction algorithms are accelerated. Experiments show that these accelerated algorithms can significantly reduce computational time while maintaining their results the same as before.

© 2013 Published by Elsevier B.V.

1. Introduction

Feature selection is a preprocessing step in many applications including pattern recognition, machine learning, and data mining. Attribute reduction is regarded as a special form of feature selection in rough set theory and aims to retain the discriminatory power of the original attribute set [21–24,44]. In databases of practical applications (Image processing, Bioinformatics, Astronomy, Finance, etc.), the number of objects is very large and the dimension (the number of attributes) is very high as well [1,2,4,24]. It is well known that attributes irrelevant to recognition tasks may deteriorate the performance of learning algorithms [6,27]. In other words, storing and processing irrelevant attributes could be computationally very expensive. To address this issue, irrelevant attributes, as pointed out in [6,29], can be omitted, which will not severely affect the classification (recognition) accuracy. Therefore, the omission of some irrelevant attributes would be desirable relative to the costs involved [20].

According to how to combine the feature subset search with the construction of the classification model, feature selection techniques can be organized into three categories: wrapper strategy [11], filter strategy [4], and embedded strategy [30]. The wrapper strategy uses a classifier to assess feature subsets and train a learning machine for every feature subset considered. The interaction between feature subset search and classification model is its significant advantage. However, it thus is usually time-consuming [11,30]. The filter strategy employs another evaluation criterion different from the target classification scheme, and therefore usually does not involve any learning machine in the feature selection process [4]. The embedded strategy generates candidate subsets by means of the methods used in filter strategy, and searches an optimal subset of features based on the classifier construction. The embedded methods combine the advantages of wrapper methods and filter methods, that they include the interaction with the classification model, while at the same time being far less computational than wrapper methods [30]. This paper focuses on the filter strategy in order to pursue both computational efficiency and solution quality regardless of a classification scheme. In filter methods, some common feature selection criteria are introduced as stopping conditions, which include information gain [12], consistency [2], and dependency [18]. These criteria can be divided into two main categories: distance-based and consistency-based [6]. For consistency-based feature selection, attribute reduction in rough set theory offers a systematic theoretic framework, which does not attempt to maximize the class separability but rather to retain the discernible ability of original attribute sets for the objects from the universe [9,36].

In recent years, many methods have been proposed and examined for finding reducts. Skowron [33] proposed an attribute reduction algorithm using a discernibility matrix, which can find...
all reducts. Kryszkiewicz and Lasek [10] proposed an approach to the discovery of minimal sets of attributes functionally determining a decision attribute. Hu and Cercone [8] proposed a heuristic attribute reduction method, called positive-region reduction, which remains the positive region of target decision unchanged. Furthermore, many researchers introduced various information entropies (Shannon’s entropy, complement entropy, combination entropy, etc.) to measure the uncertainty of an information table, and constructed the corresponding attribute reduction algorithms [13,14,16,17,28,38,39]. To handle hybrid data with numerical and categorical features, fuzzy rough set model and rough fuzzy set model were employed to obtain attribute reducts [5–7,31,32,37,42]. In addition, β-reduct proposed by Ziarko provides a suite of reduction methods in the variable precision rough set model [47]. By means of the tolerance rough set model, Parthalain and Shen presented a new approach to deal with real-valued data, which can retaining dataset semantics [19]. Yao and Zhao introduced attribute reduction in decision-theoretic rough set models in the context of different classification properties, which provided a new insight into the problem of attribute reduction [45]. These attribute reduction algorithms mentioned above can be divided into two categories: finding all reducts (or an optimal reduct) and finding one reduct [3,46]. However, it has been proved to be an NP-hard problem to find all reducts [42]. By contrast, heuristic algorithms (finding one reduct) can efficiently lessen the computational burden of attribute reduction [5,6,8,13,14,28,34,41]. In this paper, we efforts to further improve the efficiency of heuristic algorithms. For convenience of our further development, we classify these attribute reduction methods in terms of heuristics into four categories: positive region reduction [8,21–23], Shannon’s entropy reduction [34,35], complement entropy reduction [13,15] and combination entropy reduction [28]. Each of these heuristic methods can extract a single reduct from a given decision table and preserves the particular property of the decision table. Although these heuristic methods are much faster, attribute reduction still remains a computationally difficult problem when data sets are large. To overcome this difficulty, Qian and Liang [29] proposed an accelerator for attribute reduction based on positive approximation. The heuristic methods based on the accelerator can significantly decrease the time consuming and obtain the same attribute reduct as their original versions. In [26,40], this idea of accelerator was extended to incomplete data and hybrid data, and these corresponding accelerators can significantly improve the performance of attribute reduction algorithms. However, by means of the accelerator, only the insignificant objects are removed from datasets in each iteration of computing reducts. It has been observed that the number of attributes in datasets can also largely affect the efficiency of attribute reduction. This motivates the idea of this paper. In order to further improve the performance of the heuristic attribute reduction methods, we develop a new accelerator by gradually reducing not only the size of universe but also the number of attributes in each iteration of attribute reduction. By incorporating the new accelerator into each of the above four representative heuristic attribute reduction methods, we obtain their accelerating versions. Numerical experiments show that each of the improved methods can obtain the same attribute subset as its corresponding original method while greatly saving computational cost, especially for the large scale datasets.

The rest of study is organized as follows. A brief review of relative basic concepts in Section 2. In Section 3, through analyzing the rank preservation of four representative significant measures of attributes, we develop a new accelerator based on the perspective of objects and attributes. Experiments on ten datasets in UCI machine learning repository show that the four representative heuristic algorithms based on the proposed accelerator outperform their original counterparts in terms of time consuming in Section 4. Then, conclusion and future work come in Section 5.

2. Preliminaries

In this section, we review some basic concepts such as indiscernibility relation, partition, significance measures and forward attribute reduction algorithms.

2.1. Rough approximations

An information table 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 \( f: U \times A \to V \) is a function \( f(x, a) \in V_a \) for each \( a \in A \) [21–23].

An indiscernibility relation \( R_B = \{(x, y) \in U \times U | f(x, a) = f(y, a), \forall a \in B \} \) was determined by a non-empty subset \( B \subseteq A \). \( U/R_B = \{X_b | x \in U \} \) (just as \( U/B \)) indicates the partition of \( U \) resulted from \( R_B \), where \( x_b \) denotes the equivalence class determined by \( x \) with respect to \( B \), i.e., \( \{x \in U | (x, y) \in R_B \} \).

Furthermore, given an information table \( S = (U, A) \) and an object subset \( X \subseteq U, B \subseteq A \), one can construct a rough set of \( X \) on the universe by elemental information granules in the following definition:

\[ B^X = U \left\{ [x_b]_B | [x_b]_B \subseteq X \right\}, \text{ and } B^X \cup \emptyset = U \left\{ [x_b]_B | [x_b]_B \cap X = \emptyset \right\}. \]

where \( B^X \) and \( B^X \cup \emptyset \) are called \( B \)-lower approximation and \( B \)-upper approximation with respect to \( B \), respectively. The order pair \((B^X, B^X \cup \emptyset)\) is called a rough set of \( X \).

There are two kinds of attributes for a classification problem, which can be characterized 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.

Given a decision table \( DT = (U, C \cup D) \), \( B \subseteq C \), \( U/D = \{Y_1, Y_2, \ldots, Y_n\} \), the lower and upper approximations of the decision attribute set \( D \) are defined as:

\[ BD = \{B Y_1, B Y_2, \ldots, B Y_n\} \text{ and } B D = \{B Y_1, B Y_2, \ldots, B Y_n\}. \]

Let \( POS^D_{U/C}(D) = \bigcup_{b \in B} b Y_b \), which is called the positive region of \( D \) with respect to \( B \) in the decision table \( DT = (U, C \cup D) \).

2.2. Four representative significance measures of attributes

In heuristic attribute reduction methods, attribute significance measure is a crucial factor. Therefore, we will introduce four representative significance measures here, which are based on positive region, Shannon’s conditional entropy, complement conditional entropy and combination conditional entropy.

- Positive region (PR) was first employed in a heuristic attribute reduction algorithm, called positive region reduction, which keeps the positive region of target decision unchanged [8].
- Shannon’s conditional entropy (SCE) was introduced to search reducts of a decision table [34,38]. This reduction algorithm calls Shannon’s entropy reduction, which remains the conditional entropy of target decision. Shannon’s conditional entropy of \( B \) with respect to \( D \) in \( DT = (U, C \cup D) \) is denoted as:

\[ H\left(\frac{U/C}{D}\right)(B) = \sum_{i=1}^{n} p(X_i) \sum_{j=1}^{m} p(Y_j | X_i) \log(p(Y_j | X_i)). \]

where \( p(X_i) = \frac{|X_i|}{|U|} \) and \( p(Y_j | X_i) = \frac{|X_i \cap Y_j|}{|X_i|} \), and \( X \) is a non-empty set.
• Complement conditional entropy (PCE) was defined to measure the uncertainty and applied to reduce redundant attribute of a decision table [13,14]. The reduction method based on the entropy is called complement entropy reduction, which can preserve the conditional entropy of a given decision table. The conditional entropy of $B$ with respect to $D$ in $DT = (U,C \cup D)$ is denoted as

$$E_{\text{UC}}(D|C) = \sum_{i=1}^{C} \sum_{j=1}^{D} \frac{|Y_j \cap X_i|}{|U|} \frac{|Y_j - X_i|}{|U|},$$

where $Y_j$ and $X_i$ are the complements of $Y_j$ and $X_i$, respectively.

• Combination conditional entropy (CCE) is based on the intuitionistic knowledge content nature of information gain, which can be used to obtain attribute reduces [28]. The reduction method can remain combination conditional entropy of a given decision table. The conditional entropy of $B$ with respect to $D$ in $DT = (U,C \cup D)$ is defined as

$$CE_{\text{UC}}(D|C) = \sum_{i=1}^{C} \sum_{j=1}^{D} \frac{|X_i|}{|U|} \frac{C_{X_i}^{2}}{C_{X_i}^{2}},$$

$$CE_{\text{UC}}(D|C) = \sum_{i=1}^{C} \sum_{j=1}^{D} \frac{|X_i|}{|U|} \frac{C_{X_i}^{2}}{C_{X_i}^{2}},$$

where $C_{X_i}^{2} = \sum_{j \in X_i} (X_i \cup U) - C_{X_i}^{2}$ denotes the number of pairs of the objects which are not distinguishable from each other in the equivalence class $X_i$.

The corresponding significance measures based on the measures mentioned above are given as follows.

Let $DT = (U,C \cup D)$ be a decision table and $B \subseteq C$. For $\forall a \in B$, the inner significance measures of a based on positive region, Shannon’s conditional entropy, complement conditional entropy and combination conditional entropy are respectively defined as

$$\text{Siginner}_1(a,B,C,D,U) = \frac{E_{\text{UC}}(D|C) - E_{\text{UC}}(B\cup\{a\}|C)}{E_{\text{UC}}(B\cup\{a\}|C)},$$

$$\text{Siginner}_2(a,B,C,D,U) = H_{\text{UC}}(D|B) - H_{\text{UC}}(D|B \cup \{a\}),$$

$$\text{Siginner}_3(a,B,C,D,U) = E_{\text{UC}}(B|\{a\}) - E_{\text{UC}}(B|B \cup \{a\}),$$

where $E_{\text{UC}}(B'|C) = \frac{\text{Pos}(B'|C)}{|U|}$.

By means of the inner significant measures, the definition of core [13,21,28,38] can be denoted as follows:

Let $S = (U,C \cup D)$ be a decision table and $a \in C$. If $\text{Siginner}_1(a,C,D,U) > 0$ and $\text{Siginner}_2(a,C,D,U) > 0$, then $a$ is a core attribute of $S$ in the context of type $A$.

Furthermore, we suppose $S = (U,C \cup D)$ be a decision table and $B \subseteq C$. For $\forall a \in B - C$, the outer significance measures of a based on positive region, Shannon’s conditional entropy, complement conditional entropy and combination conditional entropy are respectively defined as

$$\text{Sigouter}_1(a,B,C,D,U) = \frac{E_{\text{UC}}(D|B) - E_{\text{UC}}(D|B \cup \{a\})}{E_{\text{UC}}(B|\{a\})},$$

$$\text{Sigouter}_2(a,B,C,D,U) = E_{\text{UC}}(B|\{a\}) - E_{\text{UC}}(B|B \cup \{a\}),$$

$$\text{Sigouter}_3(a,B,C,D,U) = CE_{\text{UC}}(D|B) - CE_{\text{UC}}(D|B \cup \{a\}),$$

where $E_{\text{UC}}(B'|C) = \frac{\text{Pos}(B'|C)}{|U|}$.

2.3. Forward attribute reduction algorithms

In rough set theory, many heuristic attribute reduction algorithms have been designed to achieve efficiently attribute reduces, in which forward greedy search strategy is common [5,6,8,13,15,29,34]. In general, starting with an attribute with the maximal inner significance measure, a forward greedy attribute reduction approach takes an attribute with the maximal outer importance into the attribute reduct in each loop until this subset satisfies the stopping criterion, which yields an attribute reduce. Formally, a forward greedy attribute reduction algorithm can be written as follows.

**Algorithm 1 (8,29,38).** General forward greedy attribute reduction algorithm

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

**Output:** One reduce red.

**Step 1:** red $= \emptyset$; red is the pool to conserve the selected attributes

**Step 2:** Compute $\text{Siginner}(a_i,C,D,U), 1 \leq i \leq |C|$;

**Step 3:** Put $a_i$ into red, where $\text{Siginner}(a_i,C,D,U) > 0$;

**Step 4:** While $E_{\text{UC}}(\text{red},D) \neq E_{\text{UC}}(C,D)$ Do//This provides a stopping criterion.

$$\{\text{red} \rightarrow \text{red} \cup \{a_i\}, \text{where } \text{Siginner}(a_i,\text{red},C,D,U) = \max(\text{Siginner}(a_i,\text{red},C,D,U), a_i \in C \in \text{red})\};$$

**Step 5:** Return red and end.

3. Rank preservation of significance measures of attributes

It is well known that each of the significance measures of attributes provides some heuristic information for forward attribute reduction algorithms. In this section, to further improve the performance of these attribute reduction algorithms, we will focus on the rank preservation of the four significance measures of attributes from the perspective of decreasing the number of objects and attributes simultaneously.

In order to prove the rank preservation of a significance measure of attributes, we need the following lemma.

**Lemma 3.1.** Let $0 \leq a_i, b_i \leq 1, i = 1, 2, \ldots, n, \sum_{i=1}^{n} a_i = 1$, and $\sum_{i=1}^{n} b_i = 1$. If $\sum_{i=1}^{n} a_i b_i = 0$, then $\exists 1 \leq i \leq n$ such that $a_i = b_i = 1$ and $a_i = b_i = 0$ for $\forall k \neq i$.

**Proof.** By means of the existing conditions, we have that

$$\sum_{i=1}^{n} a_i b_i = \sum_{i=1}^{n} \left( a_i + \sum_{j=1}^{n} b_j \right) = \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} \left( a_i + \sum_{j=1}^{n} b_j \right) = 1 - \sum_{i=1}^{n} \left( a_i + \sum_{j=1}^{n} b_j \right).$$

Thus, one has

$$\sum_{i=1}^{n} a_i b_i = 1 \iff 1 - \sum_{i=1}^{n} \left( a_i + \sum_{j=1}^{n} b_j \right) = 1$$

$$\iff \sum_{i=1}^{n} \left( a_i + b_i \right) = 0$$

$$\iff a_i = 0 \text{ or } b_i = 0, \text{ for } \forall i \leq n$$

Furthermore, because of $\sum_{i=1}^{n} a_i = 1$, there exists $u \leq n$ such that $a_u \neq 0$. Therefore $\sum_{i=1}^{n} b_i = 0$, i.e., $b_u = 0$, for $\forall k \neq u$. And because of $\sum_{i=1}^{n} b_i = 1$, we can obtain $b_u = 1$.

Then, we have $\sum_{i=1}^{n} a_i b_i = 1 = 0$, i.e., $a_u = 0$, for $\forall k \neq u$, i.e., $\sum_{i=1}^{n} a_i b_i = 0$. So, it is obvious that $a_u = 0$.

That is to say, $\exists u \leq n$ such that $a_u = b_u = 1$ and $a_i = b_i = 0$ for $\forall k \neq u$. □

Based on Lemma 3.1, we give the following theorem.
Theorem 3.1. Let $DT = (U, C \cup D)$ be a decision table, $B \subseteq C$. If $E^{U(C)}(D|C) = E^{U(C)}(D|C)$, then $\text{POS}^{U(C)}(D) = \text{POS}^{U(C)}(D)$ and $U'_B/B = U'_C/C$, where $U'_B = U - \text{POS}^{U(C)}(D)$ and $U'_C = U - \text{POS}^{U(C)}(D)$.

Proof. By the existing condition $B \subseteq C$, it is obvious that $U/B \geq U/C$. Without any loss of generalization, we suppose that $U/C = \{X_1, X_2, \ldots, X_m\}$, $U/B = \{X_1, X_2, \ldots, X_m, 1, X_{m+1,1}, \ldots, X_{m+1,2}, \ldots, X_n\}$. Then

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

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|X_i \cap Y_j|}{|U|} \cdot \frac{|Y_j| - |X_i|}{|U|}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|X_i \cap Y_j|}{|U|} \cdot \frac{|Y_j| - |X_i|}{|U|}$$

where $\mu_{ij} = \frac{|X_i \cap Y_j|}{|U|}$, $0 \leq \mu_{ij} \leq 1$.

Furthermore, because $E^{U(C)}(D|C) - E^{U(C)}(D|B) = 0$, we have that

$$\sum_{j=1}^{m} (\mu_{ij} + \mu_{ij} - 2 \mu_{ij} \times \mu_{ij}) = 0 \iff \sum_{j=1}^{m} \mu_{ij} + \sum_{j=1}^{m} \mu_{ij} = 2 \sum_{j=1}^{m} \mu_{ij} \times \mu_{ij} = 1.$$

According to Lemma 3.1, if $\sum_{j=1}^{m} \mu_{ij} \times \mu_{ij} = 1$, then $3 \leq n < m$ such that $\mu_{ij} = \mu_{ij} = 0$ for $j \neq n$, that is to say, the equivalent classes $X_n$ and $X_n$ belong to the same decision class, i.e., $X_n, X_n \in Y_n$. Thus, $X_n, X_n \in \text{POS}^{U(C)}(D)$ and $X_n, X_n \in \text{POS}^{U(C)}(D)$.

And because of $U/C = \{X_1, X_2, \ldots, X_m\}$ and $U/B = \{X_1, X_2, \ldots, X_n, 1, \ldots, X_m, 1, \ldots, X_{m+1,1}, \ldots, X_{m+1,2}, \ldots, X_n\}$, the objects in $\text{POS}^{U(C)}(D)$ is the same as the ones in $\text{POS}^{U(C)}(D)$, and the equivalence classes in $U/C$ are identical with the ones in $U/B$.

Therefore, if $U/B \supseteq U/C$ and $E^{U(C)}(D|B) = E^{U(C)}(D|C)$, then $\text{POS}^{U(C)}(D) = \text{POS}^{U(C)}(D)$ and $U'_B/B = U'_C/C$. □

Theorem 3.1 states, for two different decision tables, the equivalence classes that are not in the positive regions of them are identical with each other if the partition derived from the condition attribute set in one decision table is coarser than the one in the other and the values of complement conditional entropy of these two tables are equal.

Theorem 3.2 [29]. Let $DT = (U, C \cup D)$ be a decision table, $B \subseteq C$, then

1. $\text{Sig}_{\text{outer}}^{U(C)}(a, B, C, D, U) = \frac{|U|}{|U|} \text{Sig}_{\text{outer}}^{U(C)}(a, B, C, D, U)'$.

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

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

4. $\text{CE}^{U(C)}(D|B) = \frac{|U|^2}{|U|} \text{CE}^{U(C)}(D|B)'$.

By means of Theorem 3.2, the inherent relationships between the outer significance measures based on positive region, Shannon's conditional entropy, complement conditional entropy and combination conditional entropy in $(U, C \cup D)$ and in $(U'_B, C \cup D)$ were revealed.

Theorem 3.3. Let $DT = (U, C \cup D)$ be a decision table, $B \subseteq C$. If $\text{Sig}_{\text{outer}}^{U(C)}(a, B, C, D, U) = 0$ for $\forall a \in C - B$, then $\text{Sig}_{\text{outer}}^{U(C)}(a, B, C, D, U) = 0$, where $A = 1, 2, 3, 4$.

Proof. By the existing condition $\text{Sig}_{\text{outer}}^{U(C)}(a, B, C, D, U) = 0$, i.e., $E^{U(C)}(D|B \cup \{a\}) = E^{U(C)}(D|B)$ and Theorem 3.1, we have that $\text{POS}^{U(C)}(D) = \text{POS}^{U(C)}(D)$, and then $U'_B/B = U'_B/B$.

For convenience, suppose we have

$$U_B/B = \{X_1, X_2, \ldots, X_n\}, U'_B/B = \{X_1, X_2, \ldots, X_n\},$$

and in

$$\text{POS}^{U(C)}(D)/B = \{X_1, X_2, \ldots, X_n\},$$

$$\text{POS}^{U(C)}(D)/B = \{X_1, X_2, \ldots, X_n\}(l \geq m).$$

By means of different values of $A$, four cases will be considered in the following proof.

1. $A = 1$

Because of $B \supseteq B$, it is obvious that $U_B/B = U'_B/B$ and then $\text{POS}^{U(C)}(D) = \text{POS}^{U(C)}(D)$. Thus, we can obtain that

$$\text{POS}^{U(C)}(D) = \bigcup \{X_1, X_2, \ldots, X_n\} \subseteq \{X_1, X_2, \ldots, X_n\},$$

$$\text{POS}^{U(C)}(D) = \bigcup \{X_1, X_2, \ldots, X_n\} \subseteq \{X_1, X_2, \ldots, X_n\}.$$

Furthermore, we can obtain that

$$\text{Sig}_{\text{outer}}^{U(C)}(a, B', C, D, U) = \frac{1}{|U|} \left( \text{POS}^{U(C)}(D) - \text{POS}^{U(C)}(D) \right) = 0.$$

2. $A = 2$

By the condition $B \supseteq B'$, it is easy to obtain $U_B/B' = U'_B/B'$ and then $\text{POS}^{U(C)}(D) \subseteq \text{POS}^{U(C)}(D)$ and $\text{POS}^{U(C)}(D) \subseteq \text{POS}^{U(C)}(D)$. Therefore, we can obtain that

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

$$= \frac{m}{|U|} \sum_{j=1}^{m} \frac{|X_j \cap Y_j|}{|X_j|} \log \frac{|X_j \cap Y_j|}{|X_j|}.$$
Therefore, we have that

\[
\begin{align*}
\text{Corollary 3.1.} & \quad \text{Let } DT = (U, C \cup D) \text{ be a decision table, } B \subseteq B' \subseteq C. \text{ If } \\
B_1^+ & = \{a | \text{Sigouter}_{3}(a, B, C, D, U) = 0, a \in C - B' \}, \\
B_1^- & = \{a | \text{Sigouter}_{4}(a, B', C, D, U) = 0, a \in C - B' \}, \text{ then } \\
B_3' & \subseteq B_1^+,
\end{align*}
\]

where \( \Delta = 1, 2, 3, 4 \).

It is easy to prove this corollary by means of Theorem 3.3.

\[
\begin{align*}
\text{Theorem 3.4.} & \quad \text{Let } DT = (U, C \cup D) \text{ be a decision table, } B \subseteq B' \subseteq C. \text{ If } \\
b, c \in C - B - B_1^+ \text{ and } \text{Sigouter}_{3}(b, B', C, D, U) > \text{Sigouter}_{4}(c, B', C, D, U), \text{ then } \\
\text{Sigouter}_{3}(b, B', c, D, U_b) > \text{Sigouter}_{4}(c, B', C, D, U_b),
\end{align*}
\]

where \( C_0 = C - B_1, B_0 = \{a | \text{Sigouter}_{3}(a, B, C, D, U) = 0, a \in C - B' \}, U_b = U - \text{POS}_{B_0}^{(C,D)}, B_1'^+ = \{a | \text{Sigouter}_{4}(a, B', C, D, U) = 0, a \in C - B' \}, \Delta = 1, 2, 3, 4.

\[
\begin{align*}
\text{Proof.} & \quad \text{In terms of the different values of } \Delta, \text{ we will give the proof from the following four cases.} \\
(1) & \quad \Delta = 1. \text{ By the existing condition } \text{Sigouter}_{3}(b, B', C, D, U) > \text{Sigouter}_{4}(c, B', C, D, U) \text{ and Theorem 3.2, we have that } \\
\text{Sigouter}_{3}(b, B', c, D, U_b) > \text{Sigouter}_{4}(c, B', C, D, U_b).
\end{align*}
\]

From the existing condition \( b, c \in C - B - B_1^+ \) and Corollary 3.1, it is obvious that \( b, c \in C - B - B_1^+ \), \( B_1' = \{a | \text{Sigouter}_{3}(a, B', C, D, U) = 0, a \in C - B' \} \). Therefore, we have that \( B' \cap B_1' = \emptyset \) and

\[
\text{Sigouter}_{3}(b, B', c, D, U_b) = \frac{1}{|U_b|} \times \left( \text{POS}_{B_0}^{(C,D)}(D) - \text{POS}_{B_0}^{(C,D)}(D') \right)
\]

In similarity,

\[
\text{Sigouter}_{4}(c, B', C, D, U_b) = \text{Sigouter}_{4}(c, B', C, D, U_b).
\]

Therefore, one has

\[
\text{Sigouter}_{3}(b, B', c, D, U_b) > \text{Sigouter}_{4}(c, B', C, D, U_b).
\]

(2) \( \Delta = 2 \).

By the existing condition \( \text{Sigouter}_{3}(b, B', C, D, U) > \text{Sigouter}_{4}(c, B', C, D, U) \) and Theorem 3.2, we have that

\[
\text{Sigouter}_{3}(b, B', C, D, U_b) > \text{Sigouter}_{4}(c, B', C, D, U_b).
\]

From the existing condition \( b, c \in C - B - B_1^+ \) and Corollary 3.1, it is
easy to obtain that \( b, c \not\in B_3 \), where \( B_3 = \{ a; \text{Sig}_{outer}^\text{inner}(a, B, C, D, U) = 0, a \in C - B' \} \).

Thus, one has that \( B' \cap B_3 = \emptyset \) and

\[
\text{Sig}_{outer}(b, B', C_3, D, U_3) = \left( H_{\text{outer}}^{C_3}(D'B') - H_{\text{outer}}^{C_3}(D'B' \cup \{ b \}) \right)
\]

\[
= \left( H_{\text{outer}}^{C_3}(D'B') - H_{\text{outer}}^{C_3}(C_3)(D'B' \cup \{ b \}) \right)
\]

\[= \text{Sig}_{outer}(b, B', C_3, D, U_3).\]

In similarity,

\[\text{Sig}_{outer}(c, B', C_3, D, U_3) = \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

Therefore, one has

\[\text{Sig}_{outer}(b, B', C_3, D, U_3) > \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

\( \square \)

Furthermore, by means of \( b, c \in C - B' - B_3 \) and Corollary 3.1, it is easy to obtain that \( b, c \not\in B_3 \), where

\[B_3 = \{ a; \text{Sig}_{outer}^\text{inner}(a, B, C, D, U) = 0, a \in C - B' \}.\]

Therefore, we have that \( B' \cap B_3 = \emptyset \) and

\[\text{Sig}_{outer}(b, B', C_3, D, U_3)
\]

\[\text{Sig}_{outer}(c, B', C_3, D, U_3) = \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

\[\text{Sig}_{outer}(c, B', C_3, D, U_3) = \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

\[\text{Sig}_{outer}(c, B', C_3, D, U_3) = \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

Therefore, we have that

\[\text{Sig}_{outer}(b, B', C_3, D, U_3) > \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

\[\square \]

In similarity,

\[\text{Sig}_{outer}(c, B', C_3, D, U_3) = \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

Therefore, one has

\[\text{Sig}_{outer}(b, B', C_3, D, U_3) > \text{Sig}_{outer}(c, B', C_3, D, U_3).\]

\[\square \]

According to \( b, c \in C - B' - B_3 \) and Corollary 3.1, it is easy to obtain that \( b, c \not\in B_3 \), where \( B_3 = \{ a; \text{Sig}_{outer}^\text{inner}(a, B, C, D, U) = 0, a \in C - B' \}.\)

\[\text{Sig}_{outer}(b, B', C_3, D, U_3) = \text{Sig}_{outer}(b, B', C_3, D, U_3).\]

\[\square \]
efficiency of attribute reduction algorithm, a novel accelerator will be presented in this paper, which is based on the principle that the rank of attribute significant measures are preserved while the insignificant attributes to the process of attribute reduction are removed and the useless objects for computing reducts are simultaneously deleted within each iteration. In the following, the description of this accelerator is shown.

**Algorithm 3.** Accelerator for attribute reduction from the perspective of objects and attributes (ACC2)

\[
\text{Input: Decision table DT = (U,C ∪ D);} \\
\text{Output: One reduct.} \\
\text{Step 1: } \text{red} \rightarrow \emptyset; /\text{red is the pool to conserve the selected attributes} \\
\text{Step 2: Compute } \Sigma^{\text{inner}}(a_k,C,D,U), k \leq |C|; \\
\text{Step 3: Put } a_k \text{ into red, where } \Sigma^{\text{inner}}(a_k,C,D,U) > 0; /\text{These attributes form the core of the given decision table} \\
\text{Step 4: } i \leftarrow 1, U_i \leftarrow U, C_i \leftarrow C \text{ and } C_{\text{insig}} \leftarrow \emptyset; \\
\text{Step 5: While } EF(U_i,C_i)(\text{red}, D) \neq EF(U_i,C_i)(C, D), \text{ Do [Compute the positive region } POS(C_i)(D), U_{i+1} = U_i - \text{POS}(U_i,C_i)(D), \\
\text{red} \rightarrow \text{red} \cup \{a_o\}, \text{ where } \Sigma^{\text{outer}}(a_o, \text{red}, C_i, D, U_{i+1}) = \max \{ \Sigma^{\text{outer}}(a_o, \text{red}, C_i, D, U_{i+1}) | a_o \in C_i - \text{red}\}, \text{ compute } C_{\text{insig}}, \text{ where } \\
C_{\text{insig}} = \{ a | \Sigma^{\text{outer}}(a, \text{red}, C_i, D, U_{i+1}) = 0, a \in C_i \}, C_{i+1} = C_i - C_{\text{insig}}, i \leftarrow i + 1; \\
\text{Step 6: return red and end.}
\]

Table 1 describes the 10 UCI data sets.

**Table 2** shows comparison of ACC1-PR with ACC2-PR using the ten datasets in Table 1, in which the comparisons of running time and reducts of these two algorithms. From Table 2, we can see that the running time of ACC2-PR is less than ACC1-PR on nine of 10 datasets, and the same attribute subset can be selected running these two algorithms ACC1-PR and ACC2-PR, which is determined by the rank preservation of significance measures in Section 3. The results shows that ACC2-PR is significant.

Furthermore, we take the datasets Gisette and Waveform ± noise for examples to explain the reason why attribute reduction algorithms are accelerated using the proposed accelerator, as shown in Tables 3 and 4. The tables indicate the number of objects and attributes within each loop of ACC2-PR. From Table 3, we can see that the number of objects and the number of attributes are 1079 and 4839 in the second loop respectively, and the number of objects and attributes are 1024 and 4610 within the third loop respectively. It is obvious that a lot of insignificant attributes are deleted in these loops, while the size of universe is still large. Therefore, compared with ACC1-PR, the computational time in the two loops is significantly reduced, which results in the running time computing the reducts of ACC2-PR is less than ACC1-PR as the dataset Gisette. Nevertheless, ACC2-PR are not ever faster than ACC1-PR as all of the datasets. Table 4 shows this case. From Table

Table 1 Description of 10 UCI data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Number of objects</th>
<th>Number of attributes</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 KDDcup10per</td>
<td>494,021</td>
<td>42</td>
<td>13</td>
</tr>
<tr>
<td>2 Gisette</td>
<td>13,500</td>
<td>5000</td>
<td>5</td>
</tr>
<tr>
<td>3 Ticdate2000</td>
<td>3822</td>
<td>85</td>
<td>2</td>
</tr>
<tr>
<td>4 Satist</td>
<td>4435</td>
<td>35</td>
<td>6</td>
</tr>
<tr>
<td>5 Final-general</td>
<td>10,104</td>
<td>71</td>
<td>5</td>
</tr>
<tr>
<td>6 Arcene train</td>
<td>100</td>
<td>10,000</td>
<td>6</td>
</tr>
<tr>
<td>7 Mushroom</td>
<td>5644</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>8 Optdigits</td>
<td>3830</td>
<td>64</td>
<td>3</td>
</tr>
<tr>
<td>9 Waveform ± noise</td>
<td>5000</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>10 Connect</td>
<td>67,557</td>
<td>42</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2 The running time and reducts of Algorithms ACC1-PR and ACC2-PR.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>ACC1-PR Original attributes</th>
<th>ACC2-PR Original attributes</th>
<th>ACC1-PR Time (s)</th>
<th>ACC2-PR Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDDcup10per</td>
<td>42</td>
<td>24</td>
<td>736.743</td>
<td>24</td>
</tr>
<tr>
<td>Gisette</td>
<td>5000</td>
<td>13</td>
<td>2268.281</td>
<td>13</td>
</tr>
<tr>
<td>Ticdate2000</td>
<td>85</td>
<td>24</td>
<td>1.429</td>
<td>24</td>
</tr>
<tr>
<td>Satist</td>
<td>35</td>
<td>26</td>
<td>0.197</td>
<td>26</td>
</tr>
<tr>
<td>Final-general</td>
<td>71</td>
<td>21</td>
<td>9.961</td>
<td>21</td>
</tr>
<tr>
<td>Arcene train</td>
<td>10,000</td>
<td>4</td>
<td>1093.332</td>
<td>4</td>
</tr>
<tr>
<td>Mushroom</td>
<td>22</td>
<td>6</td>
<td>0.360</td>
<td>3</td>
</tr>
<tr>
<td>Optdigits</td>
<td>64</td>
<td>6</td>
<td>1.051</td>
<td>6</td>
</tr>
<tr>
<td>Waveform ± noise</td>
<td>24</td>
<td>14</td>
<td>3.251</td>
<td>14</td>
</tr>
<tr>
<td>Connect</td>
<td>42</td>
<td>34</td>
<td>128.649</td>
<td>34</td>
</tr>
</tbody>
</table>
681 form ± noise, ACC2-SCE is not better than ACC1-SCE. From Table 7, we can see that the number of insignificant attributes is zero within each loop. That is to say, for the dataset Waveform ± noise, ACC2-PR is not superior to ACC1-PR.

### 4.2.2. ACC1-SCE and ACC2-SCE

Table 5 shows the running time and reducts of ACC1-SCE and ACC2-SCE on the 10 datasets in Table 1. From Table 5, we can see that ACC2-SCE is faster than ACC1-SCE on nine of 10 datasets, and the attribute subset obtained by ACC2-SCE is the same as ACC1-SCE, which is determined by the rank preservation of significance measures in Section 3. The results show that ACC2-SCE is efficient.

Furthermore, we take the datasets Gisette and Waveform ± noise for examples to explain the reason why attribute reduction algorithms are accelerated using the proposed accelerator, as shown in Tables 6 and 7. From Table 6, we can see that the number of objects and the number of attributes are 1079 and 4839 within the second iteration of ACC2-SCE, respectively, and the number of objects and attributes are 1060 and 4610 within its third iteration. It is obvious that compared with ACC1-SCE, the running time of ACC2-SCE is evidently saved within these two loops. That is because that the number of objects is still very large, while a lot of insignificant attributes are deleted from the dataset Gisette in the process of reduction. Nevertheless, ACC2-SCE is not more efficient than ACC1-SCE as all of the datasets. Table 7 shows this case. From Table 7, we can see that the number of insignificant attributes is zero within each loop. That is to say, as dataset Waveform ± noise, ACC2-SCE is not better than ACC1-SCE.

### 4.2.3. ACC1-PCE and ACC2-PCE

Table 8 shows the running time and reducts of ACC1-PCE and ACC2-PCE on the 10 datasets in Table 1. From Table 8, we can see that ACC2-PCE is faster than ACC1-PCE on nine of 10 datasets, and the reducts obtained by ACC2-PCE is the same as ACC1-PCE, which is determined by the rank preservation of significance measures in Section 3. The results show that ACC2-PCE is significantly efficient.

Furthermore, we take the datasets Gisette and Waveform ± noise for examples to explain the reason why attribute reduction algorithms are accelerated using the proposed accelerator, as shown in Tables 9 and 10. From Table 9, we can find that the number of objects and attributes are 2283 and 4953 within the second iteration, and number of objects and attributes are 1060 and 4834 within the third iteration respectively. Because the number of objects is still very large, while a lot of insignificant attributes are deleted from the dataset Gisette in the process of reduction. Nevertheless, ACC2-PCE is not more efficient than ACC1-PCE as all of the datasets. Table 10 shows this case. From Table 10, we can see that the number of insignificant attributes is zero within each loop.
ber of objects is still very large and a lot of insignificant attributes are deleted from these datasets, the running time of ACC2-PCE in these two iterations is much less than ACC1-PCE. However, ACC2-SCE is not ever efficient for all datasets. Table 10 shows this case. From Table 10, we can see that the number of insignificant attributes is zero within each iteration. That is to say, as dataset Waveform ± noise, ACC2-SCE is not better than ACC1-SCE.

Furthermore, we take the datasets Gisette and Waveform ± noise for examples to explain the reason why attribute reduction algorithms ACC2-CCE are accelerated using the proposed accelerator, as shown in Tables 12 and 13. From Table 12, we can see that ACC2-CCE is more timesaving than ACC1-CCE on nine of ten datasets, and the reducts obtained by ACC2-CCE is the same as ACC1-CCE, which is determined by the rank preservation of significance measures in Section 3. The results shows that ACC2-CCE is significantly efficient.

4.2.4. ACC1-CCE and ACC2-CCE

Table 11 shows the running time and reducts of running Algorithms ACC1-CCE and ACC2-CCE on the 10 datasets in Table 1. From Table 11, we can see that ACC2-CCE is more timesaving than ACC1-CCE on nine of ten datasets, and the reducts obtained by ACC2-CCE is the same as ACC1-CCE, which is determined by the rank preservation of significance measures in Section 3. The results shows that ACC2-CCE is significantly efficient.

In conclusion, based on the experimental analysis, it should be stressed that the new accelerating attribute reduction algorithms (ACC2-PR, ACC2-SCE, ACC2-PCE and ACC2-CCE) are all more efficient than the original accelerating algorithms in most of datasets, except for the datasets in which there are few insignificant attributes.

5. Conclusions

A new accelerator for attribute reduction has been proposed in this paper. We first find that there exist some insignificant attrib-
butes in the process of computing reducts, and proof that the significance of each attribute remain the same after deleting these insignificant attributes. We present a general accelerator based on perspective of objects and attributes. Comparison with the existing accelerator, the new one can simultaneously decrease the size of universe and the number of attributes within each iteration of the process of attribute reduction, which is the key point of further accelerating attribute reduction. Finally, we introduce four representative heuristic algorithms embedded the new accelerator based on the positive region, Shannon’s entropy and complement entropy. Experimental results show that the heuristic algorithms embedded the proposed accelerator can significantly reduce the computational time of attribute reduction.

Some future works are planned along the following directions: First, it would be interesting to investigate how our method can be extended to obtain attribute reducts from data with missing values and hybrid data. Second, since our current method requires continuous values of attribute be discretized, which motivate us to investigate how different discretization methods affect the performance of the proposed accelerator. Another direction is to extend our accelerator to the algorithms that deal with regression problems in which the class label is continuous values. Moreover, we will make effort to experiment our accelerated algorithms on genomic microarray data for effectively obtaining informative gene.

6. Uncited reference

Q2 [25].

Acknowledgements

The research was supported by the National Natural Science Foundation of China (No. 71031006), Science and Technology Basic Condition Platform Construction Project of Shanxi Province (Intelligent Data Analysis and Application), Key Project of Science and Technology of Shanxi Province (No. 20110321027-01), the Special Prapsahce Project for the National Key Basic Research and Development Program of China (973) (No. 2011CB311805), and the National Science Foundation of Shanxi Province (Nos. 2009020107-1, 2010021017-3).

References