



Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation

J. Qian^{a,b,c}, D.Q. Miao^{a,c,*}, Z.H. Zhang^{a,c}, W. Li^{a,c}

^a Department of Computer Science and Technology, Tongji University, Caoan Road 4800, 201804 Shanghai, China

^b College of Computer Engineering, Jiangsu Teachers University of Technology, Zhongwu Road 1801, 213015 Changzhou, China

^c Key Laboratory of Embedded System and Service Computing, Ministry of Education of China, Tongji University, 201804 Shanghai, China

ARTICLE INFO

Article history:

Received 21 December 2009

Received in revised form 14 May 2010

Accepted 30 July 2010

Available online 18 August 2010

Keywords:

Attribute reduction

Positive region

Discernibility matrix

Information entropy

Hybrid attribute measure

Boundary region

ABSTRACT

Attribute reduction is one of the key issues in rough set theory. Many heuristic attribute reduction algorithms such as positive-region reduction, information entropy reduction and discernibility matrix reduction have been proposed. However, these methods are usually computationally time-consuming for large data. Moreover, a single attribute significance measure is not good for more attributes with the same greatest value. To overcome these shortcomings, we first introduce a counting sort algorithm with time complexity $O(|C| |U|)$ for dealing with redundant and inconsistent data in a decision table and computing positive regions and core attributes ($|C|$ and $|U|$ denote the cardinalities of condition attributes and objects set, respectively). Then, hybrid attribute measures are constructed which reflect the significance of an attribute in positive regions and boundary regions. Finally, hybrid approaches to attribute reduction based on indiscernibility and discernibility relation are proposed with time complexity no more than $\max(O(|C|^2 |U/C|), O(|C| |U|))$, in which $|U/C|$ denotes the cardinality of the equivalence classes set U/C . The experimental results show that these proposed hybrid algorithms are effective and feasible for large data.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

In some data mining and machine learning tasks, there are a number of attributes. It is observed that some of the attributes which may be irrelevant to the decision making will deteriorate the performance of learning algorithms and decrease the generalization power of the learned classifiers. Attribute reduction, also called feature selection, is often carried out as a preprocessing step to find a minimum subset of attributes that provides the same descriptive or classification ability as the entire set of attributes. In recent years, rough set theory [17,18] has been widely employed in attribute reduction. Different attribute reduction algorithms [2,4,6–8,12,17,18,20,23,31,32,34,35] have been developed for finding a single reduct or the set of all reducts. The main goal of any reduct construction method is to extract a minimal set of attributes such that these relevant attributes can maximize interclass differences and minimize intraclass differences. However, it is not surprising that attribute reduction methods still may not be applied to large-scale data with high dimensions. Therefore, developing an efficient and effective approach to attribute reduction is very desirable.

Among these existing methods [2,6,8,12,14,17,20,23,32,34,35], one group method focuses on the indiscernibility relation in a universe that captures the equivalence of objects, while the other group considers the discernibility relation that

* Corresponding author at: Department of Computer Science and Technology, Tongji University, Caoan Road 4800, 201804 Shanghai, China. Tel.: +86 21 69589867.

E-mail addresses: qjqjlyf@163.com (J. Qian), miaoduoqian@163.com (D.Q. Miao), zehua.zzh@gmail.com (Z.H. Zhang), jx_wenli@yahoo.com.cn (W. Li).

explores the differences of objects. For indiscernibility relation, one can employ it to induce a partition of the universe and thereby to construct positive regions whose objects can be undoubtedly classified into a certain class with respect to the selected attributes. Thus, attribute reduction algorithms based on positive region have been proposed. Among these algorithms in [15,28], some researchers studied the simplified decision table and some effective sorting algorithms which have solved to some extent the inefficient problem. Hence, its time complexity in [28] is $\max(O(|C||U|), O(|C|^2|U/C|))$ at best so far ($| \cdot |$ denotes the cardinality of a set). For discernibility relation, we have attribute reduction algorithms based on discernibility matrix and information entropy. Reduction methods based on discernibility matrix [20] have high cost of storage with space complexity $O(|C||U|^2)$ for a large decision table with $|U|$ objects and $|C|$ conditional attributes. Thus, storing and deleting the element cells in a discernibility matrix is a time-consuming process. Many researchers [25,30,32,33] studied discernibility matrix construction. Nguyen and Nguyen [15] and Korzeń and Jaroszewicz [10] implemented efficient heuristic algorithms for computing a reduct in $O(|C|^2|U|\log|U|)$ time. These methods based on discernibility matrix are not feasible for dealing with huge data. As well, attribute reduction algorithms based on information entropy [11,13,24,26] have been developed. To the best of our knowledge, their time complexities [11] are no less than $O(|C|^2|U|\log|U|)$. However, this group method is still computationally very expensive, which is intolerable for large data.

On the other hand, in heuristic search strategy among attribute reduction methods, we usually start with an empty set of attributes, and then add the selected feature into the reduct one by one. Some attribute significance measures such as dependency function [17,28], information gain [13,24,26], consistency [4], and other measures [3,5,9,16,19,21,22,27] are employed to select the most significant attribute in each round. When more attributes have the same significance, in such cases, we randomly select any one in general. However, a different subset of the selected attributes may make a great difference in classification accuracy. Thus, constructing a rational attribute significance measure is a fundamental and substantial research topic in attribute reduction.

Since an efficient attribute reduction method reflects in three aspects—computing equivalence classes, reducing storage and minimizing the search space, our focus is on how to improve the time efficiency of a heuristic attribute reduction algorithm and construct an effective attribute significance measure. In order to reduce time complexity and search space, we first employ the counting sort algorithm with time complexity $O(|C||U|)$ to compute the equivalence classes and core attributes. Then, we analyze the relationships among three classical algorithms based on indiscernibility and discernibility relation, and transform attribute reduction algorithms based on discernibility matrix and information entropy into the methods based on boundary region so that we can integrate the two group methods mentioned above into two hybrid approaches. Finally, we construct two hybrid attribute selection measures which contain the information from the positive and boundary regions and present two hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. Their time complexities are no more than $\max(O(|C||U|), O(|C|^2|U/C|))$. Numerical experiments show that our hybrid methods are more efficient and feasible for large data and the attribute significance measures are more rational, reliable and practical.

The rest of this paper is organized as follows. In Section 2, we review necessary concepts in rough set theory. Section 3 presents some efficient algorithms for calculating equivalence classes and core attributes. Section 4 provides hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. Section 5 gives some numerical experiments to validate the efficiency and effectiveness of the proposed algorithms. Finally, we conclude this paper and several issues for future work.

2. Basic notions

In this section, we review the basic notions of the Pawlak rough set model regarding classification and approximation. For classification tasks, we consider a special information table with a set of decision attributes. Such an information table is called a decision table.

Definition 1. A decision table is defined as: $S = (U, At = C \cup D, \{V_a | a \in At\}, \{I_a | a \in At\})$, where $U = \{x_1, x_2, \dots, x_n\}$ is a finite nonempty set of objects, At is a finite nonempty set of attributes, $C = \{c_1, c_2, \dots, c_m\}$ is a set of condition attributes describing the objects, and D is a set of decision attributes that indicates the classes of objects. V_a is a nonempty set of values of $a \in At$, and $I_a : U \rightarrow V_a$ is an information function that maps an object in U to exactly one value in V_a . $a(x)$ denotes the value of attribute a for object x .

For simplicity, we assume $D = \{d\}$ in this paper, where d is a decision attribute which describes the decision for each object, and $V_d = \{1, 2, \dots, k\}$. A table with multiple decision attributes can be easily transformed into a table with a single decision attribute by considering the Cartesian product of the original decision attributes.

Definition 2. A relative indiscernibility relation with respect to $A \subseteq C$ is defined as:

$$IND(A|D) = \{(x, y) \in U \times U | \forall a \in A [a(x) = a(y)] \wedge d(x) = d(y)\}. \quad (1)$$

A relative discernibility relation with respect to $A \subseteq C$ is defined as:

$$DIS(A|D) = \{(x, y) \in U \times U | \exists a \in A [a(x) \neq a(y)] \wedge d(x) \neq d(y)\}. \quad (2)$$

An indiscernibility relation is reflexive, symmetric and transitive, and thus is an equivalence relation. It determines a partition of U , denoted by $U/IND(A)$, U/A or π_A . The equivalence class of $U/IND(A)$ containing x is given by $[x]_{IND(A)} = [x]_A = \{y \in U | (x, y) \in IND(A)\}$, or $[x]$ if $IND(A)$ is understood. A relative discernibility relation is irreflexive, symmetric, but not transitive. The duality of indiscernibility and relative discernibility relation suggests that two objects are either indiscernible or discernible.

Consider a partition $\pi_D = \{D_1, D_2, \dots, D_k\}$ of the universe U with respect to the decision attribute D and another partition $\pi_A = \{A_1, A_2, \dots, A_r\}$ defined by a set of condition attributes A . The equivalence classes induced by the partition (i.e. $U/IND(A)$) are the basic blocks to construct the Pawlak rough set approximations.

Definition 3. For a decision class $D_i \in \pi_D$, the lower and upper approximations of D_i with respect to a partition π_A are defined by Pawlak [17]:

$$\begin{aligned} apr_A(D_i) &= \{x \in U | [x]_A \subseteq D_i\}; \\ \overline{apr}_A(D_i) &= \{x \in U | [x]_A \cap D_i \neq \emptyset\}. \end{aligned} \quad (3)$$

For the partition π_D , we can compute its lower and upper approximations in terms of k two-class problems. $POS_A(D)$ indicates the union of all the equivalence classes defined by π_A that each for sure can induce a certain decision. $BND_A(D)$ indicates that the union of all the equivalence classes defined by π_A that each can induce a partial decision.

Definition 4. For a decision table S , a positive region and boundary region of a partition π_D with respect to a partition π_A are defined as:

$$\begin{aligned} POS_A(D) &= \bigcup_{1 \leq i \leq k} apr_A(D_i); \\ BND_A(D) &= \bigcup_{1 \leq i \leq k} (\overline{apr}_A(D_i) - apr_A(D_i)). \end{aligned} \quad (4)$$

Definition 5. For a decision table S , all objects in $POS_C(D)$ are called consistent objects. The objects in $U - POS_C(D)$, denoted as D_{k+1} , are called inconsistent objects. If $POS_C(D) = U$, then the decision table is consistent, $D_{k+1} = \emptyset$; else it is inconsistent.

Theorem 1. For a decision table S , $A \subseteq C$, $POS_A(D) = POS_C(D) \iff apr_A(D_i) = apr_C(D_i) \ \forall i \in \{1, 2, \dots, k, k+1\}$.

Proof. The proof follows directly from Definitions 3–5. \square

Definition 6 [28]. For a decision table S , let $\pi_C = \{[x'_1]_C, [x'_2]_C, \dots, [x'_r]_C\}$, $U' = \{x'_1, x'_2, \dots, x'_r\}$, $U'_{POS} = \{x'_{i_1}, x'_{i_2}, \dots, x'_{i_t}\}$, all the objects in U'_{POS} are consistent, all the objects in $U' - U'_{POS}$, denoted as U'_{BND} , are inconsistent in original decision table S , then $S' = (U' = U'_{POS} \cup U'_{BND}, At = C \cup D, \{V_a | a \in At\}, \{I_a | a \in At\})$ is called a simplified decision table.

For computing efficiently, we denote the decision value of all objects in U'_{BND} by $k+1$ without affecting the attribute reduction result of an original decision table in the Pawlak rough set model. Thus, S' can be regarded as a consistent decision table. In what follows, we always discuss the consistent simplified decision table S' .

Definition 7. For a simplified decision table S' , $a \in C$, if $POS_{C-(a)}(D) \neq POS_C(D)$, then a is indispensable in C ; otherwise a is relatively dispensable in C . The set of all indispensable attributes is a core set, denoted as $Core(C)$.

3. Some key efficient algorithms in attribute reduction methods

In this section, we will introduce a counting sort algorithm for computing equivalence classes, positive regions and core attributes, which plays a pivotal role in attribute reduction methods.

3.1. A counting sort algorithm for calculating $U/IND(A)$

In Pawlak rough set model, classical attribute reduction algorithms mainly deal with categorical data. Thus, we can recode the symbol attributes with a set of consecutive nature numbers. For example, as to sex attribute, it takes values in set $\{M, F\}$ and can be recoded with 1 and 2. Therefore, we focus on discussing a decision table with a set of integral numbers in this paper. In order to calculate $U/IND(A)$ ($A \subseteq C$), we introduce an efficient counting sort algorithm, which time complexity is cut down to $O(|A||U|)$.

Algorithm 1. Computing $U/IND(A)$ algorithmInput: A decision table S , $U = \{x_1, x_2, \dots, x_n\}$, $A = \{c_1, c_2, \dots, c_s\}$ Output: $U/IND(A)$

Begin

Step 1. To every c_i ($i = 1, 2, \dots, s$) denote the maximum value on attribute c_i by M_i . Initialize every element in array Order with $1, \dots, n$, respectively, and all the elements in array TempOrder with 0.**Step 2.** for $i = 1$ to s do2.1 for $k = 1$ to M_i do

{Count[k] = 0;}

2.2 for $j = 1$ to n do{Count[$c_i(x_j)$]++; TempOrder[j] = Order[j];}//Count[$c_i(x_j)$] now contains the number of elements equal to $c_i(x_j)$ 2.3 for $k = 2$ to M_i do

{Count[k] = Count[k – 1] + Count[k];}

//Count[k] now contains the number of elements less than or equal to k 2.4 for $j = n$ to 1 do{Order[Count[$c_i(x_j)$]] = TempOrder[j]; Count[$c_i(x_j)$]––;}**Step 3.** Let the object sequence from Step 2 be x'_1, x'_2, \dots, x'_n ; $t = 1$; $B_t = x'_1$;for $j = 2$ to n do

{

if $c_i(x'_j) = c_i(x'_{j-1})$ for all $c_i \in A$ ($i = 1, 2, \dots, s$), then $B_t = B_t \cup \{x'_j\}$;else { $t = t + 1$; $B_t = \{x'_j\}$ };

}

End.

After simply analyzing and calculating, Step 1 needs time $O(|A||U|)$ for computing the maximum value of each attribute and Step 3 takes time $O(|A||U|)$. Step 2.1 takes time $O(|A|)$, Step 2.2 takes time $O(|M_i|)$, Step 2.3 takes time $O(|U|)$, and Step 2.4 takes time $O(|U|)$, thus the overall time of Step 2 in each loop is $O(|U| + M_i)$. The overall loop of Step 2 has $|A|$ times, so we can get the time complexity of Algorithm 1 $O(|A||U| + \sum_{1 \leq i \leq |A|} M_i)$. In most cases, especially to the large-scale decision table, there is often $\max(M_i) \leq |U|$ ($1 \leq i \leq |A|$), $|A||U| + \sum_{1 \leq i \leq |A|} M_i \leq |A||U| + |A||U|$, therefore the time complexity of Algorithm 1 is $O(|A||U|)$. It is easy to check that the space complexity of Algorithm 1 is $O(|U|)$.

Example 1. Table 1 provides a decision table S , where $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is the set of objects, $C = \{c_1, c_2, c_3, c_4, c_5\}$ is the set of condition attributes, and $D = \{d\}$ is the set of decision attribute. In Step 1, we calculate $M_1 = 3$, and initialize Order[1] = 1, Order[2] = 2, ..., Order[9] = 9. In Step 2.2, Count[1] = 1, Count[2] = 3, Count[3] = 5. In Step 2.3, Count[2] = 4 and Count[3] = 9, which means that these objects equal to 2 on c_1 are arranged from the second position to the fourth position and the objects with the attribute value equal to 3 from fifth position to ninth position, respectively. In Step 2.4, sorting on attribute c_1 , we can get the object sequence 1, 2, 6, 7, 3, 4, 5, 8, 9. Then, we sort the previous object sequence in terms of condition attributes c_2, c_3, c_4, c_5 , respectively. Finally, we can obtain the object sequence 6, 7, 8, 9, 3, 4, 2, 1, 5. In Step 3, we can calculate the equivalence classes $\{\{6, 7\}, \{8, 9\}, \{3\}, \{4\}, \{2\}, \{1\}, \{5\}\}$.

By Algorithm 1, we can acquire the equivalence classes of a decision table. But misusing a reduct definition that is only for a consistent table for an inconsistent table in the Pawlak rough set model will cause a problem in [14]. Therefore, we must transform an inconsistent decision table into a simplified decision table by Algorithm 2.

Table 1
A decision table.

U	c_1	c_2	c_3	c_4	c_5	d
1	1	1	2	1	3	3
2	2	1	1	1	3	4
3	3	1	2	2	2	1
4	3	3	3	3	2	2
5	3	3	3	3	3	2
6	2	1	1	2	1	2
7	2	1	1	2	1	1
8	3	1	3	1	2	2
9	3	1	3	1	2	3

Algorithm 2. Acquiring simplified decision table S' Input: A decision table S , $U = \{x_1, x_2, \dots, x_n\}$, π_C Output: A simplified decision table S'

Begin

Step 1. Denote the maximum value of decision attribute by M_d and let $U' = \emptyset$;**Step 2.** For any equivalence class C_i in π_C do

{
 if all objects from C_i have the same decision value, then append the first object in C_i into U' ;
 else denote the decision value of the first object in C_i by $M_d + 1$ and append it into U' ;
 }

End.

Obviously, the time complexity of Algorithm 2 is $O(|U|)$.

Remark 1. In U' , all objects whose decision value is $k + 1$ are inconsistent in the original decision table. It is easily verified that the simplified decision table S' is consistent and preserves positive regions in the original decision table.

Example 2. Considering an inconsistent decision table S represented in Table 1, π_C in Example 1, we can acquire the consistent simplified decision table S' illustrated in Table 2 by Algorithm 2. Those objects whose decision value is denoted by '*' are inconsistent in the original decision table.

3.2. A computing positive region algorithm

Let us assume that $D = \{1, 2, \dots, k, k + 1\}$ in S' , where the value equal to $k + 1$ denotes the decision value of all inconsistent objects in the original decision table. For an equivalence class A_p in π_A , denote the number of these objects equal to j on d by n_p^j . It is obvious that $n_p^1 + \dots + n_p^{k+1}$ equals n_p , the number of the objects in A_p . If there exist at least two different decision values in A_p , A_p does not belong to the positive regions. In other words, A_p for sure belongs to the positive regions if $n_p^i > 0$ and $n_p^j = 0$ where $i = 1, \dots, j - 1, j + 1, \dots, k + 1$.

Theorem 2. For any equivalence class A_p in a decision table, $A_p \subseteq POS_A(D)$ iff $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j = 0$.

Proof. If A_p for sure belongs to the positive regions $POS_A(D)$, then the decision value of all objects in A_p are the same. Suppose the decision value of those objects is j , then $n_p^j > 0$ and the others equal 0, so $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j = 0$.

Conversely, if $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j = 0$, assume that A_p does not belong to the positive regions. Obviously, we have at least two different decision values in A_p . Suppose that two decision values are i and j , then we have $n_p^i > 0$ and $n_p^j > 0$, so $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j > 0$. This is in contradiction with the condition that $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j = 0$. The theorem holds. \square

Property 1. For an equivalence class A_p in π_A , A_p belongs to the boundary regions iff $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j > 0$.

Proof. We prove this property directly according to Theorem 2. \square

Below we present a computing positive region algorithm. According to Theorem 2, we can judge whether A_p belongs to the positive regions. If $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j$ is not 0, the value can be used for attribute reduction algorithm based on boundary region in the next section.

Table 2
Simplified decision table from Table 1.

U	c_1	c_2	c_3	c_4	c_5	d
6	2	1	1	2	1	5*
8	3	1	3	1	2	5*
3	3	1	2	2	2	1
4	3	3	3	3	2	2
2	2	1	1	1	3	4
1	1	1	2	1	3	3
5	3	3	3	3	3	2

Algorithm 3. Computing positive region algorithm

Input: A simplified decision table S' , equivalence classes π_A
Output: $|POS_A(D)|$
Begin
Step 1. Let tempValue = 0;
Step 2. For any equivalence class A_p in π_A do
{
Step 3. Compute the frequencies $(n_p^1, n_p^2, \dots, n_p^{k+1})$ of decision value in A_p ;
Step 4. if $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j = 0$, then tempValue = tempValue + $|A_p|$;
}
Step 5. Output tempValue.
End.

Obviously, the time complexity of **Algorithm 3** is determined by Step 2. The time complexity of Steps 2–4 is $O(|U|/C||M_d + 1|)$. In general, M_d is small and can be regarded as the constant, so we will not take it into account in computing time complexity and the time complexity of this algorithm is $O(|U|/C)$.

3.3. Calculating core attribute algorithm

In order to improve the time efficiency of attribute reduction algorithm, many researchers [30,36] study core attributes. Since core attributes can be used to decrease search space or reduce the size of discernibility matrix, a fast algorithm for computing core attributes is desirable. If a condition attribute c_i is a core attribute, the difference between $POS_C(D)$ and $POS_{C-c_i}(D)$ does not equal 0. The time complexity of acquiring the positive regions is $O(|C||U|)$, so the time complexity of computing $POS_C(D)$ and $POS_{C-c_i}(D)$ are $O(|C||U|)$. However, most existing algorithms compute all $POS_{C-c_i}(D)$ ($c_i \in C, i = 1, 2, \dots, m$) for acquiring core attributes and thus the time complexity is $O(|C|^2|U|)$. This is not what we desire. During computing $POS_{C-c_i}(D)$, we may sort the objects in U in terms of $c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_m$, respectively. Meanwhile, in the process of computing $POS_{C-c_{i+1}}(D)$, we also may sort the objects in U in terms of $c_1, \dots, c_i, c_{i+2}, \dots, c_m$, respectively. One can check that those two processes have too many repeating sorting operations and only one sorting operation is different. Since an important property of counting sort is that it is **stable**, namely, numbers with the same value appear in the sorted array in the same order as they do in the original array, we can get the result of $POS_{C-c_{i+1}}(D)$ induced from $POS_{C-c_i}(D)$, and propose a fast algorithm with time complexity $O(|C||U|)$ for computing core attributes.

Let $Order(U, A)$ be the object sequence after sorting all objects in U on attributes A ($A \subseteq C$). According to the counting sort algorithm, we can obtain the following property.

Theorem 3. $\pi_{C-c_{i+1}}(Order(U, C - c_{i+1})) \iff \pi_{C-c_{i+1}}(Order(Order(U, C - c_i), c_i))$, where $c_i, c_{i+1} \in C$.

Proof. On one hand, $Order(U, C - c_{i+1})$ means the sorting result of the objects in terms of attributes $c_1, \dots, c_i, c_{i+2}, \dots, c_s$, respectively. $\pi_{C-c_{i+1}}(Order(U, C - c_{i+1})) = \cap \pi_{c_k} (k = 1, \dots, i, i+2, \dots, s)$.

On the other hand, $Order(Order(U, C - c_i), c_i)$ represents the sorting result according to attribute c_i based on $Order(U, C - c_i)$. That is, $Order(Order(U, C - c_i), c_i)$ is the sorting result in terms of $c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_s, c_i$, respectively. Therefore, $\pi_{C-c_{i+1}}(Order(Order(U, C - c_i), c_i))$ also equals $\cap \pi_{c_k} (k = 1, \dots, i-1, i+2, \dots, s, i)$, namely, $\cap \pi_{c_k} (k = 1, \dots, i, i+2, \dots, s)$. The theorem holds. \square

Property 2. $\pi_{C-c_1}(Order(U, C - c_1)) \iff \pi_{C-c_1}(Order(U, C))$.

Proof. We prove this property directly according to **Theorem 3**. \square

In the following, a fast core computing algorithm is given according to the above ideas.

Algorithm 4. Calculating core attributes algorithm

Input: A simplified decision table S'
Output: Core attributes $Core(C)$
Begin
Step 1. By **Algorithm 1**, we compute π_C and obtain $Order(U, C)$;
Step 2. Compute $|POS_C(D)|$ by **Algorithm 3**;
Step 3. Calculate $|POS_{C-c_1}(D)|$;
Step 4. If $|POS_{C-c_1}(D)| < |POS_C(D)|$, then $Core(C) = Core(C) \cup \{c_1\}$;

Step 5. For $i = 2$ to m do
 {Sort the objects on $Order(U, C - c_{i-1})$ in terms of c_{i-1} , and acquire $Order(U, C - c_i)$;
 If $|POS_{C-c_i}(D)| < |POS_C(D)|$, then $Core(C) = Core(C) \cup \{c_i\}$;
 }

End.

One can find the time complexity of this algorithm is determined by Step 5. The time complexity of the first four steps are $O(|C||U|)$, $O(|U/C|)$, $O(|U/C|)$ and $O(1)$, respectively, while that of the fifth step is $O(|C||U|)$. Therefore, the complexity of Algorithm 4 is $O(|C||U|)$. Obviously, the space complexity of Algorithm 4 is $O(|U|)$.

4. Hybrid attribute reduction algorithms based on indiscernibility and discernibility relation

Among attribute reduction algorithms, for any two objects x and y in a decision table, four cases in [29] are identified as follows:

1. $d(x) \neq d(y)$, $C(x) = C(y)$;
2. $d(x) = d(y)$, $C(x) = C(y)$;
3. $d(x) \neq d(y)$, $C(x) \neq C(y)$;
4. $d(x) = d(y)$, $C(x) \neq C(y)$.

Among the above four cases, since case 4 is implied in case 2, hence we discuss the first three cases. For a decision table, case 1 indicates that it is inconsistent while case 2 represents that it is consistent. For attribute reduction algorithms, case 1 can be used for computing information gain, and case 2 is used to construct equivalence classes, whereas case 3 is employed to generate discernibility matrix or object pairs. However, we can transform case 3 into case 1 through simple computation. Therefore, only considering case 1 and case 2 is efficient for those two group methods as mentioned in Section 1. Thus, all reduct construction methods in terms of case 1 can be induced into attribute reduction algorithms based on boundary region. The relationships among three classical algorithms are illustrated in Fig. 1.

In the following, we investigate the issues in detail. Attribute reduction algorithms based on indiscernibility and discernibility relation, mainly including three classical algorithms based on positive region, discernibility matrix and information entropy, are illustrated in the next three subsections.

4.1. Attribute reduction algorithm based on indiscernibility relation

In order to efficiently develop an attribute reduction algorithm, heuristics is needed to guide the attribute reduction process. An important issue in guiding the search is the monotonicity of attribute significance measure. In general, any monotonic measure can be used to evaluate positive region preservation.

Definition 8. For a consistent simplified decision S' , let $A \subseteq C$ and $a \in C - A$. Then the significance of attribute a is defined by:

$$sig_1(A, a, D) = \frac{|POS_{A \cup a}(D)|}{|U'|}. \quad (5)$$

As $0 \leq sig_1(A, a, D) \leq 1$ and this attribute measure is monotonic, we can employ it to construct an attribute reduction algorithm based on indiscernibility relation as follows.

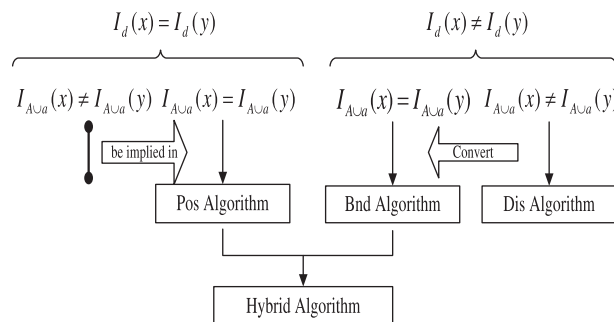


Fig. 1. Relationships among three classical algorithms.

Algorithm 5. Attribute reduction algorithm based on indiscernibility relationInput: a decision table S Output: a reduct Redu

Begin

Step 1. Compute a consistent simplified decision table S' using [Algorithm 2](#);**Step 2.** Let $\text{Redu} = \emptyset$;**Step 3.** Calculate core attributes $\text{Core}(C)$ by [Algorithm 4](#), $\text{Redu} = \text{Redu} \cup \text{Core}(C)$;**Step 4.** By [Algorithm 3](#), acquire the positive regions $\text{POS}_{\text{Redu}}(D)$ in U' ;**Step 5.** If $|\text{POS}_{\text{Redu}}(D)| = |U'|$, then turn to Step 9;**Step 6.** To each attribute $a \in C - \text{Redu}$, compute $\text{sig}_1(\text{Redu}, a, D)$ using [Algorithm 3](#);**Step 7.** Let $\text{sig}_1(\text{Redu}, a', D) = \max(\text{sig}_1(\text{Redu}, a, D))$ (if the attribute like that is not only one, select one attribute arbitrarily);**Step 8.** $\text{Redu} = \text{Redu} \cup a'$, turn to Step 4;**Step 9.** Output Redu .

End.

The time complexity of Step 1 is $O(|C||U|)$. That of Step 2 is $O(1)$. The time complexity of Step 3 is $O(|C||U|)$. The worst time complexity of Step 6 is $O(|C||U/C|)$. So, the worst time complexity of the fourth step to the eighth step is $O(|U/C||C|) + O(|U/C|(|C|-1)) + \dots + O(|U/C|) = O(|C|^2|U/C|)$. Therefore, the time complexity of this algorithm is no more than $\max(O(|C||U|), O(|C|^2|U/C|))$. The space complexity is no more than $O(|C||U|)$.

4.2. Attribute reduction method based on discernibility matrix

In the following two subsections, we mainly discuss two attribute reduction methods based on discernibility relation, which are based on discernibility matrix and information entropy, respectively.

Skowron and Rauszer [20] suggested a matrix representation for storing the set of attributes that discern pairs of objects, called a discernibility matrix. Hu and Cercone [6] improved the discernibility matrix in relation databases. However, when a decision table S is inconsistent, attribute reduction method based on discernibility matrix may give rise to the wrong result. Ye and Chen [33], Yang [30], and Pawlak [18] proposed the improved discernibility matrix. To judge whether a minimal attribute subset is a reduct without considering the property it preserves is not meaningful, and may lead to a wrong judgement. We propose the new discernibility matrix for a consistent simplified decision table.

Definition 9. Given a consistent simplified decision table S' and $\pi_D = \{D_1, D_2, \dots, D_{k+1}\}$. The discernibility matrix $M' = \{m(x, y)\}$ is defined as:

$$m(x, y) = \{a \in C | a(x) \neq a(y), x \in D_i, y \in D_j\}, \quad (6)$$

where $D_i \in \pi_D, D_j \in \pi_D, 1 \leq i < j \leq k+1$.

Remark 2. We propose the improved discernibility matrix that also holds the positive region preservation identified in [18,20]. In fact, all improved discernibility matrices consider the difference of an object pair (x, y) from two different decision equivalence classes of the positive regions or from such two objects which come from the positive regions and boundary regions respectively, and do not consider the differences of an object pair (x, y) among the objects from $U - \text{POS}_C(D)$. According to [Definition 9](#), we do not consider the difference of object pair (x, y) in D_{k+1} as well, because these objects are inconsistent in the original decision table.

In classical attribute reduction algorithms based on discernibility matrix, the main idea is to find such an attribute discerning the largest number of pairs of objects, namely, an attribute most often occurring in entries $m(x, y)$ of the discernibility matrix. After an attribute is added, cells containing that attribute are removed from the discernibility matrix. The process is repeated until the matrix contains only empty entries. However, this algorithm can create serious problem when data tables consist of more than 10,000 objects. Nguyen and Nguyen [15] and Korzeń and Jaroszewicz [10] implemented efficient heuristics for computing the number of these object pairs from the distribution of each attribute in $O(|C|^2|U|\log|U|)$ time, but they still are not feasible for large data. Below, we present a technique for computing of the number of pairs of objects discerned by a given attribute without constructing the discernibility matrix.

Consider a partition $\pi_D = \{D_1, D_2, \dots, D_k, D_{k+1}\}$ of the universe with respect to the decision attribute d . Suppose equivalence classes $\pi_A = \{A_1, A_2, \dots, A_r\}$. In A_p , there exists n_p^i objects whose decision value is i . Obviously, all objects whose decision value is i in any equivalence class forms D_i . In other words, $n_1^i + \dots + n_r^i$ equals n^i , the number of the objects in D_i . Similarly, we can check that $n_1^1 + \dots + n_r^1 + \dots + n_1^{k+1} + \dots + n_r^{k+1}$ equals n' , the number of the objects in U . For any two objects, if the decision values are different and the combinational values on condition attributes A are also different, then A can discern that two objects.

Definition 10. For a consistent simplified decision table S' , let $A \subseteq C$ and $\pi_A = \{A_1, A_2, \dots, A_r\}$, we get the number of object pairs that condition attributes A can discern

$$DIS_A^D = \sum_{1 \leq i < j \leq k+1} \sum_{1 \leq p < q \leq r} n_p^i n_q^j. \quad (7)$$

According to the above equation, computing DIS_A^D is very complex, therefore we calculate the number of those object pairs that A cannot discern in turn. The main idea is that for any two objects, if the decision values are different and the combination attribute values in terms of condition attributes A are same, then A cannot discern that two objects.

Definition 11. For a consistent simplified decision table S' , let $A \subseteq C$ and $\pi_A = \{A_1, A_2, \dots, A_r\}$, we get the number of pairs of objects that condition attributes A cannot discern

$$\widetilde{DIS}_A^D = \sum_{1 \leq p \leq r} \sum_{1 \leq i < j \leq k+1} n_p^i n_p^j. \quad (8)$$

According to Definition 11, the indiscernibility object pairs are generated from the boundary regions with respect to condition attributes from A . In particular, if A is \emptyset , then $\widetilde{DIS}_\emptyset^D = \sum_{1 \leq i < j \leq k+1} n^i n^j$. When \widetilde{DIS}_A^D does not equal 0, we must add some other attribute to discern those remaining indiscernibility object pairs. Assume that the attribute added into a reduct is an attribute a , then it can discern some of those object pairs and can not discern the rest.

Theorem 4. Given a consistent simplified decision table S' for $a \in C - A$, $DIS_{A \cup a}^D + \widetilde{DIS}_{A \cup a}^D = \widetilde{DIS}_A^D$.

Proof. It is obviously proven according to Definitions 10 and 11. \square

Property 3. Given a consistent simplified decision table S' for $a \in C$, $DIS_a^D + \widetilde{DIS}_a^D = \sum_{1 \leq i < j \leq k+1} n^i n^j$.

Proof. It is easily proven according to Theorem 4. \square

Definition 12. For a consistent simplified decision table S' , let $A \subseteq C$ and $a \in C - A$, then the significance of attribute a is defined by:

$$sig_2(A, a, D) = \frac{\widetilde{DIS}_{A \cup a}^D}{\widetilde{DIS}_A^D}. \quad (9)$$

As $\widetilde{DIS}_A^D \geq \widetilde{DIS}_{A \cup a}^D \geq 0$, we have $0 \leq sig_2(A, a, D) \leq 1$ and this attribute measure is monotonic. Therefore, we can employ it to construct discernibility matrix based attribute reduction algorithm as follows. $\widetilde{DIS}_{A \cup a}^D$ is computed by modifying Step 4 in Algorithm 3 as “Step 4 if $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j \neq 0$, then tempValue = tempValue + $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j$ ”.

Algorithm 6. Attribute reduction algorithm based on discernibility matrix

Input: A consistent decision table S' .

Output: A reduct Redu.

Begin

Step 1. Compute a consistent simplified decision table S' using Algorithm 2;

Step 2. Let Redu = \emptyset ;

Step 3. Calculate core attributes Core(C) by Algorithm 4, Redu = Redu \cup Core(C);

Step 4. By Algorithm 3, acquire \widetilde{DIS}_{Redu}^D in U' ;

Step 5. If $\widetilde{DIS}_{Redu}^D = 0$, then turn to Step 9;

Step 6. To each attribute $a \in C - Redu$, compute $\widetilde{DIS}_{Redu \cup a}^D$;

Step 7. Let $sig_2(Redu, a', D) = \min(sig_2(Redu, a, D))$ (if the attribute like that is not only one, select one attribute arbitrarily);

Step 8. Redu = Redu \cup a' , turn to Step 4;

Step 9. Output Redu.

End.

Comparing Algorithm 6 with Algorithm 5, one can verify that they are very similar except that attribute selection measure criteria and stop condition are different. Therefore, the complexity of this algorithm is no more than $\max(O(|C||U|), O(|C|^2|U|/|C|))$ as well.

4.3. Attribute reduction method based on information entropy

In traditional rough set theory, Miao and Hu [13] introduced mutual information entropy into attribute reduction to find a reduct of a decision table, and proposed a heuristic algorithm MIBARK. Wang et al. [26] and Liu et al. [11] developed attribute reduction algorithms based on condition information entropy. Entropy measures the average information provided by the information source. Thus, information gain can be used to guide the attribute reduction process. If an equivalence class belongs to the positive regions, its entropy is 0. Therefore, information gain is generated from the boundary regions.

For a consistent simplified decision table S' , let $\pi_D = \{D_1, D_2, \dots, D_k, D_{k+1}\}$, then information entropy of D is given by:

$$Info(D) = - \sum_{j=1}^{k+1} \frac{n^j}{n'} \log_2 \frac{n^j}{n'}. \quad (10)$$

For a classification problem, we need to evaluate the entropy of equivalence classes induced from the selected attribute(s). Let $A \subseteq C$, $\pi_A = \{A_1, A_2, \dots, A_r\}$, then the entropy $Info(A, D)$, condition entropy of A conditioned on D , is given by

$$Info(A, D) = - \sum_{p=1}^r \frac{n_p}{n'} \sum_{j=1}^{k+1} \frac{n_p^j}{n_p} \log_2 \frac{n_p^j}{n_p}. \quad (11)$$

Clearly, the value of information entropy indicates the impurity of an object set and achieves the minimum (usually 0) for a pure set, and maximum for a maximally impure set, respectively. As such, it can be computed from the boundary regions.

In the rest of this subsection, we give a attribute significance measure and attribute reduction algorithm based on information entropy.

Definition 13. For a consistent simplified decision table S' , let $A \subseteq C$ and $a \in C - A$, then the significance of attribute a is defined by:

$$sig_3(A, a, D) = \frac{Info(A \cup a, D)}{Info(A, D)}. \quad (12)$$

Information entropy can be seen as a kind of measurement of attribute importance, and attribute a that makes $Info(A \cup a, D)$ minimal is the most important. $Info(A, D)$ is computed by modifying Step 4 as “Step 4 if $\sum_{1 \leq i < j \leq k+1} n_p^i n_p^j \neq 0$, then $tempValue = tempValue - \frac{n_p}{n'} \sum_{j=1}^{k+1} \frac{n_p^j}{n_p} \log_2 \frac{n_p^j}{n_p}$ ”.

Algorithm 7. Attribute reduction algorithm based on information entropy

Input: A consistent decision table S' .

Output: A reduct Redu.

Begin

Step 1. Compute a consistent simplified decision table S' using Algorithm 2;

Step 2. Let Redu = \emptyset ;

Step 3. Calculate core attributes Core(C) by Algorithm 4, Redu = Redu \cup Core(C);

Step 4. Acquire $Info(\text{Redu}; D)$;

Step 5. If $Info(\text{Redu}; D) = 0$, then turn to Step 9;

Step 6. To each attribute $a \in C - \text{Redu}$, compute $sig_3(\text{Redu}, a, D)$;

Step 7. Let $sig_3(\text{Redu}, a', D) = \min(sig_3(\text{Redu}, a, D))$ (if the attribute like that is not only one, select one attribute arbitrarily);

Step 8. Redu = Redu \cup a' , turn to Step 4;

Step 7. Output Redu.

End.

Clearly, comparing Algorithm 7 with Algorithm 6, one can check that they are very similar except that the formula of attribute selection measure is different. Therefore, the complexity of this algorithm is no more than $\max(O(|C||U|), O(|C|^2|U/C|))$ as well.

4.4. Relationships among three classical attribute selection measures

In this subsection, the following theorem states the relationships among three classical attribute measures.

Theorem 5. For a consistent simplified decision table S' , let $P \subseteq C$ and $Q \subseteq C$, if $\pi_P \prec \pi_Q$, then:

- (1) $POS_P(D) \geq POS_Q(D)$.
- (2) $\widetilde{DIS}_P^D \leq \widetilde{DIS}_Q^D$.
- (3) $Info(P, D) \leq Info(Q, D)$.

Proof. Assume that $\pi_P = \{P_1, \dots, P_n\}$, $\pi_Q = \{Q_1, \dots, Q_m\}$. As $\pi_P \prec \pi_Q$, there exists a subset E_i of a set $\{1, 2, \dots, n\}$ such that $E_i \cap E_j = \emptyset$ where $i \neq j$, $i, j = 1, 2, \dots, m$. For any equivalence class, $Q_i = \cup_{j \in E_i} P_j$, where $i = 1, 2, \dots, m$.

- (1) If $\pi_P \prec \pi_Q$, we can have $POS_P(D) \geq POS_Q(D)$.
- (2) $\widetilde{DIS}_P^D = \sum_{1 \leq p \leq n} \sum_{1 \leq i1 < i2 \leq k+1} n_p^{i1} n_p^{i2} \leq \sum_{1 \leq q \leq m} \sum_{1 \leq i1 < i2 \leq k+1} \left(\sum_{j \in E_i} n_j^{i1} \right) \left(\sum_{j \in E_i} n_j^{i2} \right) \leq \sum_{1 \leq q \leq m} \sum_{1 \leq i1 < i2 \leq k+1} n_q^{i1} n_q^{i2} = \widetilde{DIS}_Q^D$.
- (3) $p(Q_i) = \sum_{j \in E_i} p(P_j)$, where $i = 1, 2, \dots, m$. As $\sum_{j=1}^n p(P_j) = \sum_{i=1}^m \sum_{j \in E_i} p(P_j)$

$$\begin{aligned} Info(P, D) &= - \sum_{j=1}^n p(P_j) \sum_{l=1}^{k+1} p(D_l | P_j) \log_2 p(D_l | P_j) \leq - \sum_{i=1}^m \sum_{j \in E_i} p(P_j) \sum_{l=1}^{k+1} p(D_l | P_j) \log_2 p(D_l | P_j) \\ &\leq - \sum_{i=1}^m p(Q_i) \sum_{l=1}^{k+1} p(D_l | Q_i) \log_2 p(D_l | Q_i) = Info(Q, D). \quad \square \end{aligned}$$

According to the above theorem, if $\pi_P \prec \pi_Q$, attributes P is more important than Q . Unfortunately, the condition $\pi_P \prec \pi_Q$ does not hold in general, consequently we cannot conclude that which one between attributes P and Q is more important. This can be verified by Example 3. Thus, three different attribute measures which derive from algebra and information views are not always identical.

4.5. Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation

In terms of Algorithms 5–7, these three algorithms are very similar. The subtle degree of differences lie in the attribute significance measure and stop condition. Algorithm 5 focuses on such an attribute with the maximal positive regions, and Algorithm 6 concerns about selecting the attribute of the minimal indiscernibility in boundary regions, whereas Algorithm 7 prefers an attribute with minimal information entropy. In fact, we can derive that Algorithm 5 always maximizes interclass differences, while Algorithms 6 and 7 minimize intra-class differences. Since three algorithms have the same time complexities and these significance attributes are different, thus developing some efficient and effective hybrid attribute reduction algorithms are desirable.

Let α and β be two parameters. Two hybrid attribute measures which reflect the significance of an attribute in the positive regions and boundary regions are constructed as follows.

Definition 14. For a consistent simplified decision table S' , let $A \subseteq C$ and $a \in C - A$, then the significance measure of attribute a is defined by:

$$sig_4(A, a, D) = \alpha sig_1(A, a, D) + \beta(1 - sig_2(A, a, D)). \quad (13)$$

Definition 15. For a consistent simplified decision table S' , let $A \subseteq C$ and $a \in C - A$, then the significance measure of attribute a is defined by:

$$sig_5(A, a, D) = \alpha sig_1(A, a, D) + \beta(1 - sig_3(A, a, D)). \quad (14)$$

Obviously, $sig_4(A, a, D)$ and $sig_5(A, a, D)$ reflect not only the size of positive regions but also the discrimination power of the boundary regions.

Theorem 6. For a consistent simplified decision table S' , let $A_1 \subseteq A_2 \subseteq C - A$, then

- (1) $sig_4(A, A_1, D) \leq sig_4(A, A_2, D)$.
- (2) $sig_5(A, A_1, D) \leq sig_5(A, A_2, D)$.

Proof

- (1) It is obvious that if $A_1 \subseteq A_2$, then $\pi_{A_2} \prec \pi_{A_1}$. According to Theorem 5, $POS_{A \cup A_1}(D) \leq POS_{A \cup A_2}(D)$ and $\widetilde{DIS}_{A \cup A_1}^D \geq \widetilde{DIS}_{A \cup A_2}^D$. Thus, $sig_1(A, A_1, D) \leq sig_1(A, A_2, D)$ and $sig_2(A, A_1, D) \geq sig_2(A, A_2, D)$. Therefore, $sig_4(A, A_1, D) \leq sig_4(A, A_2, D)$ holds.
- (2) If $A_1 \subseteq A_2$, then $\pi_{A_2} \prec \pi_{A_1}$. According to Theorem 5, $POS_{A \cup A_1}(D) \leq POS_{A \cup A_2}(D)$ and $Info(A \cup A_1, D) \geq Info(A \cup A_2, D)$. Therefore, it holds true that $sig_5(A, A_1, D) \leq sig_5(A, A_2, D)$. \square

In the following, we construct hybrid attribute reduction algorithms employing hybrid significance measures $sig_4(A, a, D)$ and $sig_5(A, a, D)$.

Table 3
Comparison of five attribute significance measures.

Attribute selection criteria	c_1	c_2	c_3	c_4	c_5
$sig_1(\phi, a, D)$	0.143	0.286	0	0.286	0.143
$sig_2(\phi, a, D)$	0.316	0.474	0.211	0.211	0.316
$sig_3(\phi, a, D)$	0.511	0.614	0.432	0.432	0.608
$sig_4(\phi, a, D)$	0.827	0.812	0.789	1.075	0.827
$sig_5(\phi, a, D)$	0.632	0.672	0.568	0.854	0.535

Algorithm 8. Hybrid attribute reduction algorithms based on indiscernibility and discernibility relation (CHybrids I and II)

Input: a decision table S

Output: a reduct $Redu$

Begin

Step 1. Compute a consistent simplified decision table S' by Algorithm 2;

Step 2. Let $Redu = \emptyset$;

Step 3. Calculate core attributes $Core(C)$ using Algorithm 4, $Redu = Redu \cup Core(C)$;

Step 4. By Algorithm 3, acquire $POS_{Redu}(D)$ in U' ;

Step 5. If $|POS_{Redu}(D)| = |U'|$, then turn to Step 9;

Step 6. To each attribute $a \in C - Redu$, compute $sig_4(Redu, a, D)(sig_5(Redu, a, D))$ using Algorithm 3;

Step 7. Let $sig_4(Redu, a', D) = \max(sig_4(Redu, a, D))(sig_5(Redu, a', D) = \max(sig_5(Redu, a, D)))$ (if the attribute like that is not only one, select one attribute arbitrarily);

Step 8. $Redu = Redu \cup a'$, turn to Step 4;

Step 9. Output $Redu$.

End.

Clearly, as Algorithm 8 is similar to those algorithms described above, the time complexities of our hybrid algorithms are no more than $\max(O(|C||U|), O(|C|^2|U/C|))$ as well.

Note that for CHybrid I with the attribute significance measure $sig_4(A, a, D)$, when $\alpha = 1$ and $\beta = 0$, it will degenerate to Algorithm 5; when $\alpha = 0$ and $\beta = 1$, it will degrade to Algorithm 6. Meanwhile, for CHybrid II with attribute measure $sig_5(A, a, D)$, when $\alpha = 0$ and $\beta = 1$, it will become Algorithm 7.

Example 3. Considering a decision table represented in Table 2, we can find there has no core attribute by Algorithm 4. In the following, we compute five reducts by the algorithms proposed above, respectively.

From Table 3, one can check the condition $\pi_P < \pi_Q$ in Theorem 5 does not hold in general. The reducts generated by Algorithms 5–7, CHybrid I, and CHybrid II are $\{c_2, c_1, c_4\}$, $\{c_3, c_4\}$, $\{c_3, c_4\}$, $\{c_4, c_1\}$ and $\{c_4, c_1\}$, respectively.

We generate decision rules with the reducts of five algorithms. The number of certain rules is 6 and the length of these rules is 15 by Algorithm 5, while Algorithms 6 and 7 generate six certain rules and the length of these rules is 12. However, our hybrid algorithms have six certain rules and the length of these rules is only 11.

5. Experimental analysis

5.1. Efficiency evaluation

In order to evaluate the time efficiencies of our hybrid methods, we implement some experiments on a personal computer with Windows XP, 1.6 GHZ CPU and 1.0 GB memory. The software is Visual C# 2005. The objective of the following experiments is to show the time efficiencies of attribute reduction algorithms based on indiscernibility and discernibility relation. Since our hybrid approaches and traditional attribute reduction algorithms only deal with discrete attributes, we employ Rosetta software (<http://www.lcb.uu.se/tools/rosetta/>) to fill in some missing values and transform the numerical data into discrete ones. We perform the experiments on 10 datasets, which are eight publicly available datasets from UCI Repository of machine learning databases in [1] and two synthetic datasets. Each dataset has only one decision attribute. For two synthetic datasets, the values of their condition attributes and decision attribute are generated randomly from 0 to 9. The characteristics of 10 datasets are summarized in Table 4.

In what follows, we compare our hybrid algorithms (CHybrids I and II) with some classical algorithms. For convenience, we refer to Algorithms 5–7 as CPos, CDis and CInfo respectively. If these methods employ quick sort technique for computing equivalence classes, we denote them by QPos, QDis and QInfo. Note that CDis and QDis find a reduct without building discernibility matrix, while the algorithm in [20] (Dis) must construct discernibility matrix. We first compare the time consumption of counting sort algorithm with that of quick sort one on datasets 1 and 2 in computing equivalence classes

because counting sort technique plays a great important role in our approaches. As shown in Figs. 2 and 3, one can observe that counting sort algorithm is much more faster than quick sort one for large data. Moreover, it is easy to see that the curve corresponding to counting sort algorithm is almost linear.

Table 5 presents the number of the selected attributes. It is easy to see that the number of the selected attributes in our approaches is no more than that of other methods.

Each dataset is first loaded into the memory from the hard disk and the processing times of six algorithms are measured. The running times of these algorithms, expressed in seconds, are compared in Table 6. The time denotes ‘–’ in Table 6 if the running time is too long. From this table, our hybrid algorithms have demonstrated their advantages over three classical algorithms (QPos, QDis and QInfo) in time efficiency.

Table 6 shows that although both running times of three classical algorithms and our hybrid methods increase with size of datasets, the former increases much more rapidly than the latter. This difference can be illustrated by plotting the ratios of their running times as shown in Fig. 4.

We further perform the experiments on artificial data Dataset 1 for 5000 objects and number of condition attributes varying from 1000 to 10,000 (see Fig. 5). We also conduct the experiments on artificial data Dataset 2 for 50 condition attributes and for number of records varying from 200,000 to 1,400,000 (see Fig. 6). Since three classical algorithms (QPos, QDis and

Table 4

Description of the datasets.

No.	Dataset	Abbreviation	Objects	Attributes	Classes
1	HSV	HSV	122	11	4
2	Breast cancer	BC	286	9	2
3	Australian Credit Approval	Crd	690	14	2
4	Tic-tac-toe endgame	TTT	958	9	2
5	German	Ger	1000	20	2
6	Car	Car	1728	6	4
7	Chess-kr-vs-kp	Chess	3196	36	2
8	Connect	Con	67,557	42	3
9	Dataset1	DS1	5000	10,000	10
10	Dataset2	DS2	2,000,000	50	10

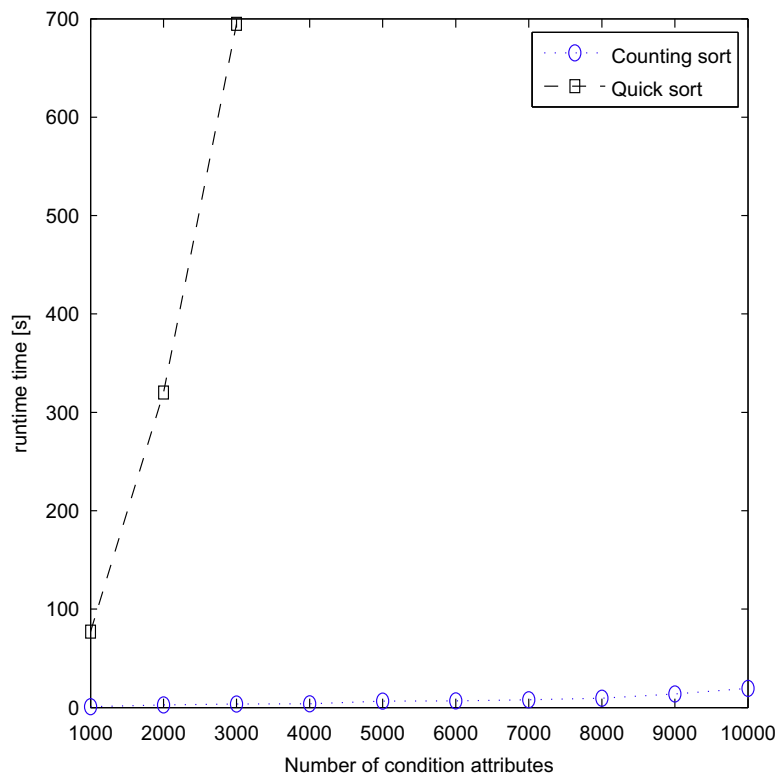


Fig. 2. Comparison with two sort algorithms on Dataset 1.

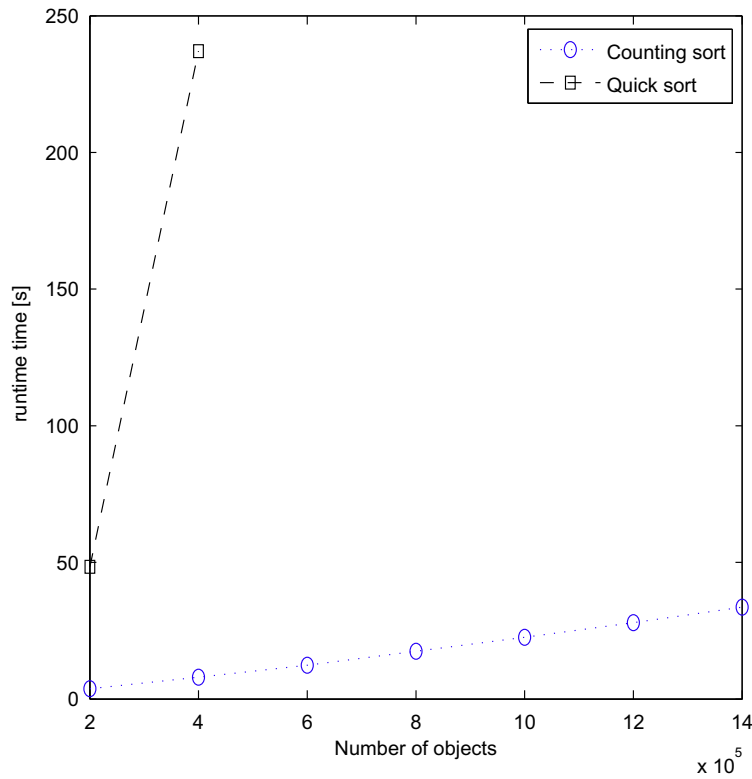


Fig. 3. Comparison with two sort algorithms on Dataset 2.

Table 5

Number of the selected attributes found by five methods.

No.	Data set	Number of selected attributes				
		QPos	QDis	QInfo	CHybrid I	CHybrid II
1	HSV	9	10	9	10	9
2	BC	9	9	9	9	9
3	Crd	11	11	11	11	11
4	TTT	8	8	8	8	8
5	Ger	9	9	9	9	9
6	Car	6	6	6	6	6
7	Chess	29	29	29	29	29
8	Con	34	34	34	34	34
9	DS1	–	–	–	9	9
10	DS2	–	–	–	12	12

Table 6

Running time of six methods.

No.	Data set	Running times				
		Dis	QPos	QDis	QInfo	CHybrid I/II
1	HSV	0.047	0.016	0.016	0.016	0.016/0.016
2	BC	0.094	0.016	0.016	0.016	0.016/0.016
3	Crd	0.687	0.031	0.031	0.047	0.016/0.016
4	TTT	1.031	0.031	0.031	0.047	0.016/0.016
5	Ger	1.828	0.047	0.047	0.047	0.016/0.016
6	Car	2.219	0.063	0.063	0.063	0.016/0.016
7	Chess	43.563	0.547	0.563	0.578	0.172/0.172
8	Con	–	239.578	292.641	274.953	17.812/19.219

QInfo) are not feasible for large data, here we only illustrate time efficiencies of our methods. The results shown in Figs. 5 and 6 display more detailed change trend of our algorithms in time efficiency with size of data set becoming increasing. It is easy to see that the curves corresponding to our approaches are almost linear. The experiments verify that our approaches with time complexities $\max(O(|C||U|), O(|C|^2|U/C|))$ are much more efficient, especially for larger datasets. Note that when the number of

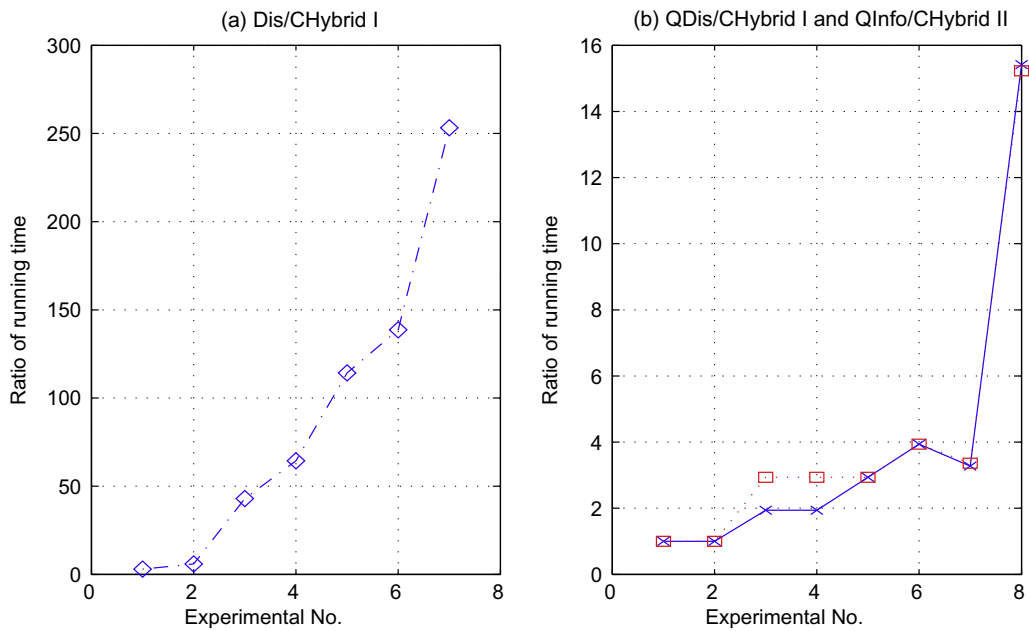


Fig. 4. Ratios of the running times in each sub-experiment.

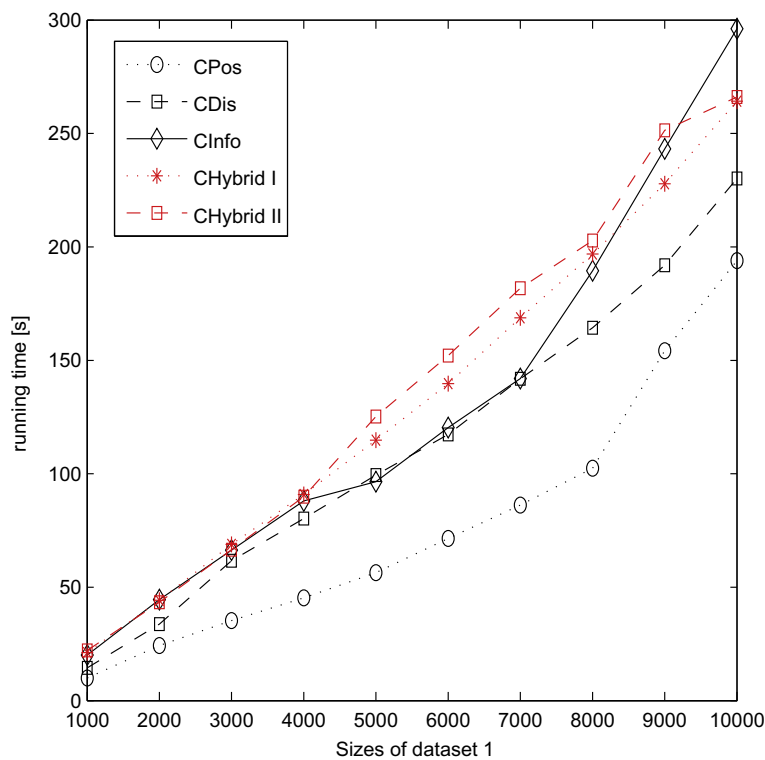


Fig. 5. Comparison with classical algorithms on artificial data Dataset 1.

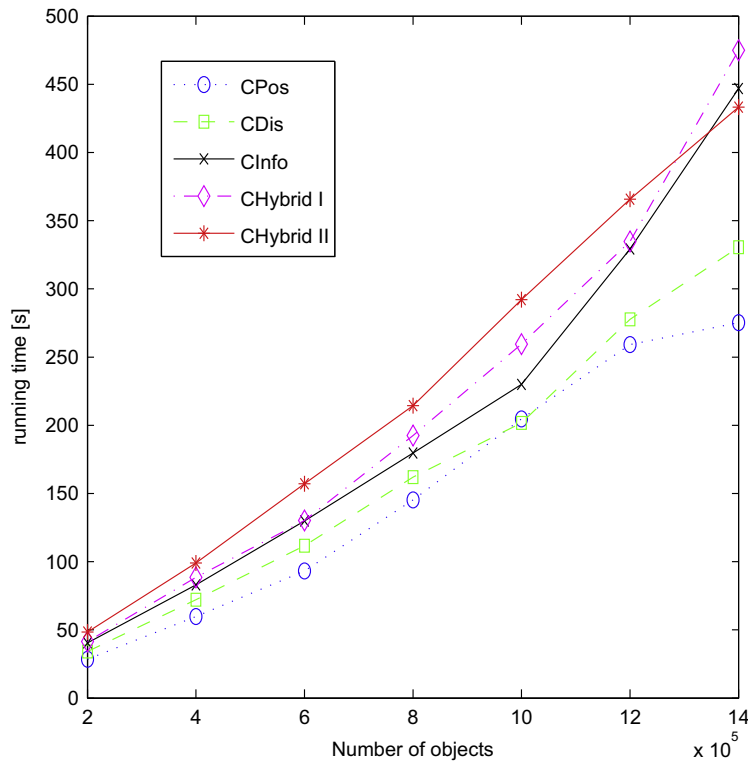


Fig. 6. Comparison with classical algorithms on artificial data Dataset 2

objects exceeds 1,400,000, our approaches cannot load the data into the memory because the processing data is too large. Therefore, how to efficiently improve attribute reduction algorithms for larger dataset is an interesting issue for future work.

5.2. Comparison of classification accuracy

This section presents some results of experimental studies on eight UCI datasets. The comparison is made in terms of the number of rules, the average length of rules, and classification accuracy.

From Table 7, one can find the number of rules and the size of rule length by CPos on tic-tac-toe Endgame are both less than those of algorithm CHybrid I. We perform the experiments on tic-tac-toe further on. Fig. 7 lists the different reducts generated from different attribute reduction algorithms. The number of rules and the rule lengths of our CHybrid algorithms are less than the mean value of those different reducts.

Table 8 provides the average length of rules on eight datasets. One can observe that our hybrid algorithms have minimal average length of rules in most cases. Table 9 presents the average classification accuracy by five methods described previously. The recorded values are acquired using 10-fold cross-validation. Classification accuracies are performed on the reduced data generated by attribute reduction algorithms, respectively. From Tables 8 and 9, the results show that our attribute significance measures in our hybrid approaches are more rational and effective.

Table 7
Number and length of decision rules.

No.	Data set	Number and length of rules				
		CPos	CDis	CIE	CHbrid I	CHybrid II
1	HSV	82/470	85/465	81/425	84/456	81/425
2	BC	183/764	183/764	185/798	183/764	185/798
3	Crđ	299/1756	300/1608	267/1439	288/1546	287/1533
4	TTT	359/2211	476/3099	320/2017	360/2215	320/2017
5	Ger	638/3005	638/2998	638/3005	642/2969	638/3005
6	Car	301/1673	328/1769	301/1673	301/1673	301/1673
7	Chess	140/2614	342/5355	129/1968	160/2555	139/2215
8	Con	34,225/585,407	36,050/576,609	34,119/551,937	34,284/552,506	34,476/556,234

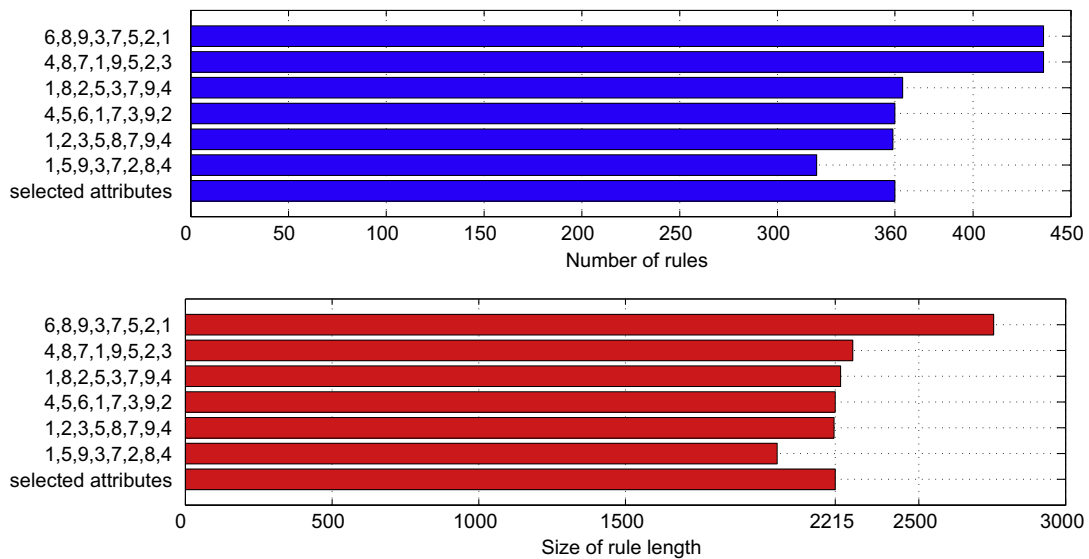


Fig. 7. Rules and rule length of different attribute reduction.

Table 8

Average length of rules.

No.	Data set	Average length of rules				
		QPos	QDis	QInfo	CHybrid I	CHybrid II
1	HSV	5.732	5.471	5.247	5.429	5.247
2	BC	4.175	4.175	4.314	4.175	4.175
3	Crd	5.873	5.360	5.40	5.368	5.341
4	TTT	6.159	6.511	6.303	6.153	6.303
5	Ger	4.71	4.699	4.71	4.625	4.71
6	Car	5.558	5.393	5.558	5.558	5.558
7	Chess	18.671	15.658	15.256	15.969	15.935
8	Con	17.105	15.995	16.177	16.116	16.134

The bold values denote the average length of rules of our hybrid algorithms are minimal among three algorithms — QPos, QDis and CHybrid I, or QPos, QInfo, and CHybrid II.

Table 9

Classification accuracy on different datasets.

No.	Data set	Classification accuracy				
		CPos	CDis	CInfo	CHybrid I	CHybrid II
1	HSV	0.428	0.477	0.460	0.478	0.478
2	BC	0.480	0.497	0.497	0.493	0.493
3	Crd	0.694	0.688	0.719	0.703	0.693
4	TTT	0.786	0.741	0.787	0.772	0.786
5	Ger	0.524	0.531	0.522	0.511	0.528
6	Car	0.892	0.892	0.892	0.892	0.892
7	Chess	0.921	0.855	0.930	0.916	0.929
8	Con	0.444	0.430	0.459	0.462	0.461
Average	–	0.646	0.639	0.656	0.655	0.658

5.3. Comparison of CHybrid I and II with other algorithms

Consistency-based attribute reduction algorithm (Consistency) in [4] employs consistency measure, which reflects not only the size of positive regions, but also the distribution of boundary samples. Therefore, even if the positive region is empty, we can still compare the distinguishing power of the attributes in terms of the sample distribution in boundary regions. It is very similar to our hybrid algorithms. The QuickReduct algorithm in [16] finds a minimal subset without exhaustively generating all possible subsets. The search begins with an empty subset, then add an attribute which results in the greatest increase in dependency value iteratively. Then the algorithm stops. In what follows, we compare our approaches with two algorithms described above in classification accuracy. Table 10 provides some classification accuracies using the

Table 10
Classification accuracy of five algorithms.

No.	Data set	Classification accuracy				
		CHybrid I	CHybrid II	Consistency	QuickReduct	Unreduce
1	HSV	0.478	0.478	0.467	0	0.459
2	BC	0.493	0.493	0.497	0.48	0.479
3	Crđ	0.703	0.693	0.655	0.694	0.702
4	TTT	0.772	0.786	0.777	0	0.689
5	Ger	0.511	0.528	0.507	0.114	0.52
6	Car	0.892	0.892	0.675	0.556	0.503
7	Chess	0.916	0.929	0.91	0.42	0.521
8	Con	0.462	0.461	0.44	0	0.358
Average	–	0.655	0.658	0.616	0.283	0.529

classifiers learned by our hybrid learner approaches described previously and performing on the unreduced data directly. Especially as to data HSV and tic-tac-toe, there are no attributes selected as each single attribute gets dependency zero, thus no attribute can be added into a reduct in the first loop and the algorithm stops here. For Quickreduct algorithm, the selected features are not enough for classification learning, therefore its classification accuracy is lower than that of other algorithms in most cases.

From Table 10, classification accuracies of our algorithms on eight datasets have a clear advantage over that of consistency-based algorithm in most cases. It must be remembered, however, that the figure demonstrates mean values, and that consistency-based algorithm also outperforms our algorithms in several cases.

5.4. Related discussion

In this subsection, we summarize the advantages of our hybrid approaches to attribute reduction and offer some comments.

Nguyen and Nguyen in [15] proposed the algorithm for computing of positive regions in $O(|C||U|\log|U|)$ time using $O(|U|)$ space. This algorithm used the lexicographical order and sorted objects in $O(|C||U|\log|U|)$ time. However, we introduce a counting sort algorithm with time complexity $O(|C||U|)$ to sort objects for computing positive regions and core attributes. In addition, Nguyen implemented Johnson strategy for computing of short reducts in $O(|C|^2|U|\log|U|)$, which is similar to Algorithm 6 employing quick sort technique. But our Algorithm 6 takes time in $\max(O(|C||U|), O(|C|^2|U|/C))$ without building the discernibility matrix as well. The main improvement in our hybrid algorithms lies in the sort technique and equivalence classes. From the experimental analysis, we can conclude that our hybrid approaches are faster than the corresponding algorithms by Nguyen in general. Furthermore the differences are profoundly larger when the size of the dataset increases.

On the other hand, when there are more attributes with the same significance in the positive regions, we can choose one with the greatest significance in the boundary regions and vice versa. Therefore, our hybrid significance measures are rational since they contain the information from the positive and boundary regions.

6. Conclusions

In this paper, we introduce a counting sort algorithm to compute equivalence classes and to deal with redundant and inconsistent data of an original decision table, and acquire a consistent simplified decision table. Then, we analyze attribute reduction algorithms based on discernibility matrix and information entropy and transform them into boundary region based attribute reduction methods. Finally, two hybrid algorithms are proposed with time complexity no more than $\max(O(|C||U|), O(|C|^2|U|/C))$. Meanwhile, two attribute significance measures are constructed which reflect not only the size of positive regions, but also the objects distribution in boundary regions. A series of experiments are conducted on 10 UCI data sets for evaluating the proposed methods. The results show that our methods are effective and can efficiently obtain an attribute reduct.

Future work includes improving the proposed algorithm further for larger data and numerical attributes without discretization.

Acknowledgments

The authors thank both the anonymous referees and the editor for their valuable comments and suggestions. This work is partially supported by the National Natural Science Foundation of China under Grant Nos.: 60775036, 60970061, the Research Fund for the Doctoral Program of Higher Education of China under Grant No.: 20060247039, and the Natural Science Research Fund of Higher Education of Jiangsu Province under Grant No.: 09KJD520004.

References

- [1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, Department of Information and Computer Science, Irvine, CA, 2007. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [2] J.G. Bazan, H.S. Nguyen, S.H. Nguyen, et al., Rough set algorithms in classification problem, in: *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, 2000, pp. 49–88.
- [3] T. Beaubouef, F.E. Petry, G. Arora, Information-theoretic measures of uncertainty for rough sets and rough relational databases, *Information Sciences* 109 (1998) 185–195.
- [4] M. Dash, H. Liu, Consistency-based search in feature selection, *Artificial Intelligence* 151 (2003) 155–176.
- [5] I. Düntsch, G. Gediga, Uncertainty measures of rough set prediction, *Artificial Intelligence* 106 (1998) 77–107.
- [6] X.H. Hu, N. Cercone, Learning in relational databases: a rough set approach, *Computational Intelligence* 11 (1995) 323–338.
- [7] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, *Pattern Recognition Letters* 40 (2007) 3509–3521.
- [8] Q.H. Hu, W. Pedrycz, D.R. Yu, et al, Selecting discrete and continuous features based on neighborhood decision error minimization, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 40 (2010) 137–150.
- [9] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, *IEEE Transactions on Knowledge and Data Engineering* 16 (2004) 1457–1471.
- [10] M. Korzeń, S. Jaroszewicz, Finding reducts without building the discernibility matrix, in: *Proceedings of 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005.
- [11] Q.H. Liu, F. Li, F. Min, M. Ye, et al, An efficient knowledge reduction algorithm based on new conditional information entropy, *Control and Decision* 20 (2005) 878–882 (in Chinese).
- [12] D.Q. Miao, L. Hou, A comparison of rough set methods and representative inductive learning algorithms, *Fundamenta Informaticae* 59 (2004) 203–219.
- [13] D.Q. Miao, G.R. Hu, A heuristic algorithm for reduction of knowledge, *Journal of Computer Research and Development* 36 (1999) 681–684 (in Chinese).
- [14] D.Q. Miao, Y. Zhao, Y.Y. Yao, et al, Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model, *Information Sciences* 179 (2009) 4140–4150.
- [15] S.H. Nguyen, H.S. Nguyen, Some efficient algorithms for rough set methods, in: *Proceedings of the International Conference on Information Processing and Management of Uncertainty on Knowledge Based Systems*, 1996, pp. 1451–1456.
- [16] N.M. Partháin, Q. Shen, Exploring the boundary region of tolerance rough sets for feature selection, *Pattern Recognition* 42 (2009) 55–667.
- [17] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences* 11 (1982) 341–356.
- [18] Z. Pawlak, *Rough Sets Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Boston, 1991.
- [19] Y.H. Qian, J.Y. Liang, C.Y. Dang, Knowledge structure, knowledge granulation and knowledge distance in a knowledge base, *International Journal of Approximate Reasoning* 50 (2009) 174–188.
- [20] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowiński (Ed.), *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, 1992.
- [21] D. Slezak, Normalized decision functions and measures for inconsistent decision tables analysis, *Fundamenta Informaticae* 44 (2000) 291–319.
- [22] D. Slezak, Various approaches to reasoning with frequency based decision reducts: a survey, in: *Rough Set Methods and Applications*, Physica-Verlag, Heidelberg, 2000, pp. 235–285.
- [23] R. Susmaga, Reducts and constructs in attribute reduction, *Fundamenta Informaticae* 61 (2) (2004) 159–181.
- [24] J. Wang, D.Q. Miao, Analysis on attribute reduction strategies of rough set, *Chinese Journal of Computer Science and Technology* 13 (1998) 189–192.
- [25] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix; the ordered attributes method, *Journal of Computer Science and Technology* 16 (2001) 489–504.
- [26] G.Y. Wang, H. Yu, D.C. Yang, Decision table reduction based on conditional information entropy, *Chinese Journal of Computers* 25 (2002) 760–766 (in Chinese).
- [27] M.J. Wierman, Measuring uncertainty in rough set theory, *International Journal of General Systems* 28 (1999) 283–297.
- [28] Z.Y. Xu, Z.P. Liu, B.R. Yang, et al, A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U|C))$, *Chinese Journal of Computers* 29 (3) (2006) 611–615 (in Chinese).
- [29] D. Yamaguchi, Attribute dependency functions considering data efficiency, *International Journal of Approximate Reasoning* (2009) 89–98.
- [30] M. Yang, An incremental updating algorithm of the computation of a core based on the improved discernibility matrix, *Chinese Journal of Computers* 29 (3) (2006) 407–413 (in Chinese).
- [31] Y.Y. Yao, Probabilistic rough set approximations, *International Journal of Approximate Reasoning* 49 (2008) 255–271.
- [32] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, *Information Sciences* 179 (7) (2009) 867–882.
- [33] D.Y. Ye, Z.J. Chen, An improved discernibility matrix for computing all reducts of an inconsistent decision table, in: *The Proceedings of the Fifth IEEE International Conference on Cognitive Informatics*, 2006, pp. 305–308.
- [34] Y. Zhao, Y.Y. Yao, F. Luo, Data analysis based on discernibility and indiscernibility, *Information Sciences* 177 (2007) 4959–4976.
- [35] N. Zhong, J.Z. Dong, Using rough sets with heuristics for feature selection, *Journal of Intelligent Information Systems* 16 (2001) 199–214.
- [36] J.W. Zhou, B.Q. Feng, Y. Liu, New algorithm for quick computing core, *Journal of Xi'an Jiaotong University* 41 (6) (2007) 681–688 (in Chinese).