Neurocomputing 123 (2014) 185–196

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Analyzing rough set based attribute reductions by extension rule

Bing Li*, Tommy W.S. Chow, Peng Tang

Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 13 June 2012
Received in revised form
25 April 2013
Accepted 14 July 2013
Communicated by N.T. Nguyen
Available online 17 August 2013

Keywords:
Attribute reduction
Extension rule
Reduction distribution
Rough set

ABSTRACT

An improved discernibility function for rough set based attribute reduction is defined to keep discernibility ability and remove redundant attributes without the precondition of the Positive Region. On the basis of discernibility function, the solution of rough set based attribute reduction can be found by satisfiability methods. With extension rule theory, a satisfiability method, the distribution of solutions with different number of attributes is obtained without enumerating all attribute reductions. Then, it is easy to search the attribute reduction with the smallest number of attributes. In addition, the cost of space and time is analyzed to find factors playing role in the computation of the method.

1. Introduction

The size of datasets has been increasing dramatically, so it has been an important issue to reduce huge objects and large dimensionality in datasets. Attribute reduction, also called feature selection, finds a subset of attributes to reduce dimensionality. By reducing attributes, it can save the cost of computational time and memory; it is also useful to improve predicative ability as a result of removing redundant and irrelevant attributes [1,2].

There exist two major approaches in attribute reduction: individual evaluation and subset evaluation. Individual evaluation, also known as ranking, assigns each attribute a weight representing its degree of relevance [39,40]. This method is incapable of removing redundancy because of similar weights among redundant attributes. So it is always set as a principal or auxiliary section. According to different mechanisms of reduction, subset evaluation falls into three catalogs: wrapper method, embedded method and filter method. The wrapper method [3,4] utilizes a classifier of interest as a black box to score subsets of attributes according to their predictive power. It can provide a highly predictive subset; however, the bias of classifiers and the setup of experiments play role in the performance of the subset. In addition, large computational cost is also needed. An improved method, the embedded method, incorporates attribute reduction as the part of training process [41,42]. Comparing with the wrapper method, it does not split data into training and validation sets, and it finds solution faster by avoiding retraining the classifier. However, this method is also dependent on classifiers. The filter method selects a subset according to a selection measure. Selection criteria include: distance measure [5,6], dependency measure [7,8], consistency measure [9], and information measure [10–12]. The filter method gives a subset achieving the balance between predicative power and computational cost. Moreover, it is independent on classifiers. So it is more practical comparing with the wrapper and embedded methods.

Most of attribute reduction methods just evaluate the performance of a subset according to predictive accuracy. However, approximate original class distribution is also an important evaluation rule [43]. Rough set based attribute reduction, the filter method with dependency measure, supports both rules. It is proposed by Pawlak [13–17] as a mathematical theory of set approximation, which is used in machine learning [18,19]. Rough set based attribute reduction finds particular subsets of conditional attributes providing the same information for classification purpose with the original set. This selection mechanism keeps the same class distribution with the original set. And its performance of predictive accuracy has been verified to be better or comparable with other methods in large amount of works. Moreover, rough set method has its own advantages. First, it needs no parameters. For general methods, they need take large computational cost to find a super parameter. It is impossible to assess the performance about all values of parameters. Second, it has explicit stopping criterion. The advantages of rough set come from that it deals with data in human-like fashion [44].

The advantages of rough set are obvious; however, its problem is computational complexity, which must be considered. The core issue of rough set theory is "discernibility function" taking $O(n^2)$ time complexity and $O(n^2 \times m)$ memory complexity, where n is the number of objects, and m is the number of attributes. Minimal reduction problem is even NP-hard [21], where the number of attributes is smallest among all possible reductions. Knowledge based methods have been proposed in the area of rough set

* Corresponding author. Tel.: +852 34429963.
E-mail address: bingli5-c@my.cityu.edu.hk (B. Li).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.07.006

[7,8,22–29], and each of them aims at its own requirement. According to their mechanisms, each method just finds a subset of attributes providing the same classification information with the original set, but no one can give a fair evaluation among these methods.

Propositional satisfiability problem (SAT) is one of the most studied NP-complete problems because of its significance in both theoretical research and practical application. Several SAT solvers [30,31,45] are employed in rough set based attribute reduction. However, there are several problems remaining to be done, including building discernibility matrix without any precondition and large addition of computational cost; a complete description of all solutions; analysis of factors playing role in computational cost about space and time. An improved discernibility function reduces the redundant attributes causing by the samples in the both Positive Region and Boundary Region. It just takes $O(n)$ addition of cost to overcome the redundancy. Moreover, the reasons of redundancy are shown by the proof of discernibility function. Extension rule [32,33] is a suitable tool to find solution of rough set reductions, which checks the satisfiability by using inverse of resolution. An important advantage of extension rule is that the combinations of attributes in inverse of resolution are smaller than the reductions, which have been verified in the experiments. So it is useful to save computational cost. By the results of discernibility function extended, the distribution of attribute reductions with different size is found [46]. It provides a new view to analyze attribute reduction based on rough set. And, according to the distribution, it is easy to find the minimal reduction. In this process, computational cost is also analyzed.

The rest of paper is organized as follows. The basic knowledge about attribution reduction using rough set is given in Section 2. Its relationship with SAT is proved in Section 3. The experimental results are presented in Section 4. Finally, the conclusion is drawn in Section 5.

2. Background

In this section, the basic notions [13–17,21,45] related to information system and rough set are shown.

Definition 2.1. Let $IS(U,A)$be an information system, where U is a nonempty finite set of objects and A is a nonempty finite set of attributes so that $f: U \to V_f$ for every $f \in A$. $V_f$is the set of values that f takes. For any $B \subseteq A$, an indiscernible relation $IND(B)$ is

$$IND(B) = \{(x,y) \in U^2 | \forall f \in B, f(x) = f(y)\} \quad (1)$$

Dataset can be seen as an information system, where samples are the objects of U and attributes are the elements of A [34].

Definition 2.2. A partition of U generated by $\{a^*\}$ is defined as

$$U/IND(\{a^*\}) = \{\{x \in U | a^*(x) = i\},\ i \in V_{a^*}\}, \quad (2)$$

where a^* is the decisional attribute.

If$(x,y) \in IND(B)$, x and y are indiscernible according to the subset B. The equivalence class of x on the B-indiscernible relation is denoted by $[x]_B$. If x and y are indiscernible according to the subset B, $y \in [x]_B$. Construct the B-lower approximations and B-upper approximations of X as

$$\underline{BX} = \{x | [x]_B \subseteq X\}, \quad (3)$$

$$\overline{BX} = \{x | [x]_B \cap X \neq \varnothing\}, \quad (4)$$

where (3) is the B-lower approximations, and (4) is the B-upper approximations.

By the definition of the B-lower approximations and B-upper approximations, the objects in U can be partition into three regions which are the Positive Region, the Boundary Region, and the Negative Region.

Definition 2.3. B-Positive Region, B-Boundary Region and B-Negative Region are defined as

$$POS_B(\{a^*\}) = \bigcup_{X \in U/IND(\{a^*\})} \underline{BX}, \quad (5)$$

$$BND_B(\{a^*\}) = \bigcup_{X \in U/IND(\{a^*\})} \overline{BX} - \bigcup_{X \in U/IND(\{a^*\})} \underline{BX}, \quad (6)$$

$$NEG_B(\{a^*\}) = U - \bigcup_{X \in U/IND(\{a^*\})} \overline{BX} \quad (7)$$

The three regions are defined with respect to $\{a^*\}$ which is the set of decisional attribute.

Definition 2.4. In an information system $IS = (U,A)$, an $n \times n$ matrix (c_{ij}) called *discernibility matrix* of IS is defined as

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j), x_i, x_j \in U\} \quad \text{for} \quad i,j = 1,\ldots,n \quad (8)$$

The discernibility matrix is denoted as $M(IS)$. It is straightforward to find $M(IS)$is symmetric and $c_{ii} = \varnothing$.

Definition 2.5. *Discernibility function* f_{IS} for an information system $IS = (U,A)$ is a Boolean function of m variables $a_1,\ldots,a_m$, defined as

$$f_{IS}(a_1,\ldots,a_m) = \wedge \{\vee (c_{ij}) : 1 \le j < i \le n,\ c_{ij} \neq \varnothing\}, \quad (9)$$

where a_i denotes an attribute in A and $\vee (c_{ij})$ is the disjunction of the variables inc_{ij}.

Example 2.1. A simple example represented in Table 1 is considered to show the discernibility matrix and discernibility function. For information system in Table 1, there are 5 objects and 5 attributes $\{a_1,a_2,a_3,a_4,a^*\}$. Table 2 shows the related discernibility matrix according to Definition 2.4. Then, the discernibility function can be found.

$$\begin{aligned} f_{IS} = {} & (a_1 \vee a_3) \wedge (a_1 \vee a_2 \vee a_3) \wedge (a_1 \vee a_2 \vee a_3 \vee a_4 \vee a^*) \wedge \\ & \times (a_1 \vee a_2 \vee a^*) \wedge a_2 \wedge (a_2 \vee a_4 \vee a^*) \\ & \times (a_2 \vee a_3 \vee a^*) \wedge (a_4 \vee a^*) \wedge (a_3 \vee a^*) \wedge (a_3 \vee a_4) \end{aligned}$$

3. Extension rule for attribute reductions

In this section, we prove that attribute reduction based on rough set can be solved by SAT with defining a^*-discernibility matrix. By employing the extension rule, the distribution of all

Table 1
Information system of Example 2.1.

U	a_1	a_2	a_3	a_4	a^*
1	0	1	1	1	1
2	1	1	0	1	1
3	1	0	0	1	1
4	1	0	0	0	0
5	1	0	1	1	0

Table 2
Discernibility matrix of Example 2.1.

Object	1	2	3	4	5
1		a_1,a_3	a_1,a_2,a_3	a_1,a_2,a_3,a_4,a^*	a_1,a_2,a^*
2	a_1,a_3		a_2	a_2,a_4,a^*	a_2,a_3,a^*
3	a_1,a_2,a_3	a_2		a_4,a^*	a_3,a^*
4	a_1,a_2,a_3,a_4,a^*	a_2,a_4,a^*	a_4,a^*		a_3,a_4
5	a_1,a_2,a^*	a_2,a_3,a^*	a_3,a^*	a_3,a_4	

possible solutions with different amount of attributes is found. Then, the attribute reduction with the smallest size is obtained.

Definition 3.1. In an information system $IS=(U,A)$, an $n\times(n+1)$ matrix (c^*_{ij}) called *a*-discernibility matrix* of IS is defined with two steps:

i.

$$c^*_{ij}=\begin{cases}\{a\in A-\{a^*\}: a(x_i)\neq a(x_j)\} & \text{if } a^*(x_i)\neq a^*(x_j)\\ A-\{a^*\} & \text{if } a^*(x_i)=a^*(x_j)\end{cases}\quad \text{for } i,j=1,\ldots,n \tag{10}$$

$$c^*_{i(n+1)}=\begin{cases}\varnothing & \text{if } \exists j\in\{1,\cdots,n\},\ c^*_{ij}=\varnothing\\ A-\{a^*\} & \text{otherwise}\end{cases}\quad \text{for } i=1,\ldots,n; \tag{11}$$

ii.

$$c^*_{ij}=A-\{a^*\}\quad \text{if } c^*_{i(n+1)}=c^*_{j(n+1)}=\varnothing\quad \text{for } i,j=1,\ldots,n \tag{12}$$

The *a**-discernibility matrix is denoted as$M^*(IS)$, and the $n+1$ column is called *state checking*. *a*-discernibility function* is defined as$f^*_{IS}(a_1,\ldots,a_{m-1})=\wedge\{\vee(c^*_{ij}):1\le i<j\le n+1, c^*_{ij}\neq\varnothing\}$. Table 3 shows the *a**-discernibility matrix of Example 2.1.

Definition 3.2. [37] An information system $IS=(U,A)$ is consistent, if all objects, which have the same values on $A-\{a^*\}$, have the same value of decisional attribute a^*; for inconsistent information system, the samples with the same values on $A-\{a^*\}$ may have different values of decisional attribute a^*.

Theorem 3.1. *In a consistent information system$IS=(U,A)$, there is not $\varnothing$ in its a*-discernibility matrix, so all objects are in the Positive Region of conditional attribute set.*

Proof. If $\exists c^*_{ij}=\varnothing$, $i,j\in\{1,\ldots,n\}$, there are two objects x_i,x_j making $a(x_i)=a(x_i)$for$\forall a\in A-\{a^*\}$ and $a^*(x_i)\neq a^*(x_j)$. Then, according to Definition 3.2, IS is an inconsistent information system. It is contradictory with the condition, so $\forall c^*_{ij}\neq\varnothing$ for $i,j\in\{1,\ldots,n\}$. We conclude $c^*_{i(n+1)}\neq\varnothing$ for $i\in\{1,\ldots,n\}$ according to Definition 3.1. As a result, there is not $\varnothing$ in its *a**-discernibility matrix.

Assume an object x is not in the Positive Region. Because of $x\notin POS_{A-\{a^*\}}(\{a^*\})$, $[x]_{A-\{a^*\}}\not\subset S'$ for $\forall S'\in U/IND(\{a^*\})$. So $\exists y\in[x]_{A-\{a^*\}}$, $a^*(y)\neq a^*(x)$. As a result, $c^*_{ij}=\varnothing$, where c^*_{ij} is the element of $(x,\ y)$ in *a**-discernibility matrix. It is contradictory with the above result. As a result, all objects are in the Positive Region of conditional set. □

Theorem 3.2. *In an inconsistent information system $IS=(U,A)$, only the object, whose value of state checking is $\varnothing$, is not in Positive Region of conditional attribute set.*

Proof. It is straightforward according to Theorem 3.1. □

Definition 3.3. In an information system $IS=(U,A)$, a set B of attributes is an attribute reduction, if $B\subseteq A-\{a^*\}$ and $POS_B(\{a^*\})=POS_{A-\{a^*\}}(\{a^*\})$.

Theorem 3.3. *B is an attribute reduction if $f^*_{IS}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$, where $V_B(\cdot):a\to\{0\ ,\ 1\}$ such that $a\in B$ if$V_B(a)=1$.*

Proof. Sufficiency: B is an attribute reduction if $f^*_{IS}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.

(1) For any object $x\in POS_{A-\{a^*\}}(\{a^*\})$
For any object $y\in U$,
i. $y\in[x]_{A-\{a^*\}}$
When $y\in[x]_{A-\{a^*\}}$, $a(x)=a(y)$ for $\forall a\in A-\{a^*\}$. $B$is a subset of $A-\{a^*\}$, so $a'(x)=a'(y)$ for $\forall a'\in B$. As a result, $y\in[x]_B$.
ii. $y\notin[x]_{A-\{a^*\}}$ and $a^*(x)\neq a^*(y)$
Since $y\notin[x]_{A-\{a^*\}}$, $\exists c'=\{a\in A-\{a^*\}:a(x)\neq a(y)\}$. And$a^*(x)\neq a^*(y)$, $c'=c^*_{ij}$ where c^*_{ij} is the element of (x,y)in *a**-discernibility matrix. We have $\vee c^*_{ij}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$ because of $f^*_{IS}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$. As a result, $c'\cap B\neq\varnothing\Rightarrow y\notin[x]_B$.
iii. $y\notin[x]_{A-\{a^*\}}$ and $a^*(x)=a^*(y)$
$a^*(x)=a^*(y)\Rightarrow c^*_{ij}=A-\{a^*\}$, according to Definition 3.1. And $\exists c'=\{a\in A-\{a^*\}:a(x)\neq a(y)\}$ because of $y\notin[x]_{A-\{a^*\}}$. Since c' is a subset of c^*_{ij}, $c'\cap B=\varnothing$ is possible. So$y\in[x]_B$, if $c'\cap B=\varnothing$; otherwise $y\notin[x]_B$.
According to the proof from i. to iii., we can conclude $[x]_{A-\{a^*\}}\subseteq[x]_B$.
Since $x\in POS_{A-\{a^*\}}(\{a^*\})$, $\exists S'\in U/IND(\{a^*\})$ $[x]_{A-\{a^*\}}\subseteq S'$. According to iii., $a^*(x)=a^*(z)$ for $\forall z\in[x]_B-[x]_{A-\{a^*\}}$, so $z\in S'$. Then, we obtain $[x]_B\subseteq S's$. As a result, $x\in POS_B(\{a^*\})$.
(2) For any object $x\notin POS_{A-\{a^*\}}(\{a^*\})$
$x\notin POS_{A-\{a^*\}}(\{a^*\})$, so there is an object $y\in[x]_{A-\{a^*\}}$making $a(x)=a(y)$ for $\forall a\in A-\{a^*\}$ and $a^*(x)\neq a^*(y)$. B is a subset of $A-\{a^*\}$, so $y\in[x]_B$. Then, $x\notin POS_B(\{a^*\})$.
According to (1) and (2), $POS_B(\{a^*\})=POS_{A-\{a^*\}}(\{a^*\})$ is proved. That is to say that B is an attribute reduction.

Necessity: B is an attribute reduction, so $f^*_{IS}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
B is an attribute reduction. That is to say $POS_B(\{a^*\})=POS_{A-\{a^*\}}(\{a^*\})$.
For any object pair (x,y), setc^*_{ij}as the element of (x,y) in *a**-discernibility matrix.

i. $x,y\in POS_{A-\{a^*\}}(\{a^*\})$ and $a^*(x)=a^*(y)$
Since $x,y\in POS_{A-\{a^*\}}(\{a^*\})$, the values of x and y in state checking column are not $\varnothing$ according to Theorem 3.1 and 3.2. So c^*_{ij} is not rewritten.
Because of $a^*(x)=a^*(y)$, we know$c^*_{ij}=A-\{a^*\}$. $B\subseteq A-\{a^*\}$, so $B\cap c^*_{ij}\neq\varnothing$. As a result, $\vee c^*_{ij}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
ii. $x,y\in POS_{A-\{a^*\}}(\{a^*\})$ and $a^*(x)\neq a^*(y)$
When $x,y\in POS_{A-\{a^*\}}(\{a^*\})$, x, y must be in $POS_B(\{a^*\})$. Because of $a^*(x)\neq a^*(y)$, there must be $a'\in B$ making $a'(x)\neq a'(y)$. $c^*_{ij}=\{a\in A-\{a^*\}:a(x)\neq a(y)\}$ according to Definition 3.1, so $a'\in c^*_{ij}$. Then, we have $\vee c^*_{ij}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
iii. One of x,y is in the Positive Region of conditional attribute set, and $a^*(x)\neq a^*(y)$
Assuming $x\in POS_{A-\{a^*\}}(\{a^*\})$ and $y\notin POS_{A-\{a^*\}}(\{a^*\})$, $x\in POS_B(\{a^*\})$ and $y\notin POS_B(\{a^*\})$ are concluded. With the additional condition $a^*(x)\neq a^*(y)$, there must an attribute a'in B making $a'(x)\neq a'(y)$. So$a'\in c^*_{ij}$. Then, $\vee c^*_{ij}(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
iv. One of x,y is in the Positive Region of conditional attribute set, and$a^*(x)=a^*(y)$
The process is the same with i.

Table 3
*a**-discernibility matrix of Example 2.1.

Object	1	2	3	4	5	State checking
1	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2	a_1,a_2,a_3,a_4
2	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_2,a_4	a_2,a_3	a_1,a_2,a_3,a_4
3	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_4	a_3	a_1,a_2,a_3,a_4
4	a_1,a_2,a_3,a_4	a_2,a_4	a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4
5	a_1,a_2	a_2,a_3	a_3	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4	a_1,a_2,a_3,a_4

v. $x,y\notin POS_{A-\{a^*\}}(\{a^*\})$
$x\notin POS_{A-\{a^*\}}(a^*)$, so there must be an object $z\in[x]_{A-\{a^*\}}$ and $a^*(x)\neq a^*(z)$. As a result, $\forall a\in A-\{a^*\}$, $a(x)=a(z)$. $c^*_{il}=\varnothing\Rightarrow c^*_{i(n+1)}=\varnothing$ according to Definition 3.1, where c^*_{il} is the element of (x,z) ina^*-discernibility matrix, and $c^*_{i(n+1)}$is the element of x in state checking column. For the same reason, $c^*_{j(n+1)}=\varnothing$, where $c^*_{j(n+1)}$is the element of $y$in state checking column. c^*_{ij} is rewritten as$A-\{a^*\}$, so $\vee c^*_{ij}(V_B(a_1),\cdots,V_B(a_{m-1}))=1$.

vi. Checking state column
When $c^*_{i(n+1)}=\varnothing,\vee(c^*_{i(n+1)})$ is not inf^*_{IS}; otherwise, $c^*_{i(n+1)}=A-\{a^*\}$, so $\vee c_{i(n+1)}{}^*(V_B(a_1),\cdots,V_B(a_{m-1}))=1$.
From i to vi, all kinds of elements in f^*_{IS} are contained, so$f^*_{IS}(V_B(a_1),\cdots,V_B(a_{m-1}))=1$.□

Assume an object $x\in POS_{A-\{a^*\}}(\{a^*\})$. In the sufficiency proof, $[x]_{A-\{a^*\}}\subseteq[x]_B$was found. That is to say$[x]_{A-\{a^*\}}\subset[x]_Bor[x]_{A-\{a^*\}}=[x]_B$. When

$$[x]_{A-\{a*\}}\subset[x]_B\Rightarrow\exists z,z\notin[x]_{A-\{a*\}},z\in[x]_B\Rightarrow\left\{\begin{array}{l}\forall a\in B,a(x)=a(z)\\ \exists c'=\{a'\in A-\{a*\}:a'(x)\neq a'(z)\}\\ a*(x)=a*(z)\end{array}\right\}\Rightarrow\left\{\begin{array}{l}x\notin[z]_{A-\{a*\}}\\ c'\cap B=\varnothing\end{array}\right\}$$
$$\left.\begin{array}{l}if\ \exists o\in[z]_{A-\{a*\}},\ a*(o)\neq a*(x)\\ x\in POS_{A-\{a*\}}(\{a*\})\\ f^*_{IS}(V_B(a_1),\cdots,V_B(a_{m-1}))=1\end{array}\right\}\Rightarrow\vee c^*_{ij}(V_B(a_1),\cdots,V_B(a_{m-1}))=1\Rightarrow\exists a'\in c'\cap B\Big\}\Rightarrow contradictory \quad (13)$$

where $c^*_{ij}$is the element of (x,o) in a^*-discernibility matrix. So there is not the object $o\in[z]_{A-\{a^*\}}$ with $a^*(o)\neq a^*(x)$. The analysis verifies that the attributes are reduced by the objects with the same decisional attribute.

The second step in Definition 3.1 is defined for an inconsistent information system. Assume an object pair (x,y),

$$\left.\begin{array}{l}\exists z\in U,\forall a\in A-\{a^*\},a(x)=a(z),a^*(x)\neq a^*(z)\\ \exists o\in U,\forall a\in A-\{a^*\},a(y)=a(o),a^*(y)\neq a^*(o)\end{array}\right\}$$
$$\Rightarrow\left\{\begin{array}{l}x,y\notin POS_{A-\{a^*\}}(\{a^*\})\\ x,y\notin POS_B(\{a^*\})\end{array}\right. \quad (14)$$

$f^*_{IS}(V_B(a_1),\cdots,V_B(a_{m-1}))=1$ assesses that B is an attribute reduction. If $c^*_{ij}\neq\varnothing$,$c^*_{ij}\cap B\neq\varnothing$, where c^*_{ij} is the element of (x,y) in a^*-discernibility matrix. According to the definition, c^*_{ij} s is rewritten as$A-\{a^*\}$, which makes sure $c^*_{ij}\cap B\neq\varnothing$ and does not add new attributes into B. Without the rewriting step, there may be redundant attributes inf^*_{IS}. This shows the second kind of reduced attributes.

Maximal consistent blocks can also favor the second reduction [35], but it needs additional computational time cost of $O(m\times n\times\log n)$ [36]. Comparing with this, our work just needs $O(n)$ computational cost of time.

Definition 3.4. A *a^*-improved discernibility function* f^{**}_{IS} for an information system $IS=(U,A)$ is a Boolean function of $m-1$ variables $a_1,\ldots,a_{m-1}$, which is defined as

$$f^{**}_{IS}(a_1,\ldots,a_m)=\wedge\{\vee(c^*_{ij}):1\le i<j\le n,a^*(x_i)\neq a^*(x_j),c^*_{ij}\neq\varnothing\}, \quad (15)$$

where a_i denotes an attribute in $A-\{a^*\}$.

Theorem 3.4. *a^*-improved discernibility function* $f^{**}_{IS}=f^*_{IS}$.

Proof. According to Definitions 3.1 and 3.4,

$$f^*_{IS}=f^{**}_{IS}\wedge\{\wedge\{\vee(c^*_{ij}):1\le i<j\le n,a^*(x_i)=a^*(x_j),c_{ij}\neq\varnothing\}\wedge\{\wedge\{\vee(c^*_{i(n+1)}):1\le i\le n,c^*_{i(n+1)}\neq\varnothing\}.$$

$$a^*(x_i)=a^*(x_j)\Rightarrow c^*_{ij}=A-\{a^*\}\Rightarrow\wedge\{\vee(c^*_{ij}):1\le i<j\le n,a^*(x_i)=a^*(x_j),c^*_{ij}\neq\varnothing\}=\vee(A-\{a^*\});$$

$$\text{if } c^*_{i(n+1)}\neq\varnothing,\ c^*_{i(n+1)}=A-\{a^*\}\ \text{ for }\ 1\le i\le n$$
$$\Rightarrow\wedge\{\vee(c^*_{i(n+1)}):1\le i\le n,c^*_{i(n+1)}\neq\varnothing\}=\vee(A-\{a^*\}).$$

So $f^*_{IS}=f^{**}_{IS}\wedge\{\vee(A-\{a^*\})\}=f^{**}_{IS}$.□

Definition 3.5. [20] An instance of SAT is a Boolean formula in conjunctive normal formula. Each disjunction formula is called a clause, and the instance is called a clause set.

An example of conjunctive normal formula is $(t\vee l)\wedge(\bar{t}\vee\bar{l})\wedge(t\vee m)$, where t, l and m are Boolean variables; overbar is logical operation denoting "not";$t\vee l$,$\bar{t}\vee\bar{l}$, and $t\vee m$ are clauses. Each Boolean variable can be assigned either true or false. According to the values assigned to the variables, the formula is evaluated to be true or false.

Theorem 3.5. *According to a^*-discernibility function, one instance of attribute reduction based on rough set is also an instance of SAT.*

Proof. According to Theorem 3.3, a subset of attributes is a reduction if the value of a^*-discernibility function is true. The attributes in a reduction can be seen as variables with true values. Otherwise, their values are false. Then, f^*_{IS} is an instance of SAT. □

Definition 3.3. About attribute reduction is different from the definitions in [13–17]. In the definitions in [13–17], there is not such attribute making $POS_B(\{a^*\})=POS_{B-\{a\}}(\{a^*\})$. In Definition 3.3, if $B-\{a\}$is a reduction, B is also a reduction. In this work, rough set based attribute reduction is solved by SAT. If an assignment for $B-\{a\}$ makes the instance of SAT satisfiable, the Boolean formula is also satisfiable with additional assignment for a. In order to keep consistent with SAT, attribute reduction is in the format of Definition 3.3.

Definition 3.6. [32] Give a clause C, $C'=(C\vee t)\wedge(C\vee\bar{t})$, where t is an attribute not in C. C' is called the result of extension rule on C.

Definition 3.7. [32] A clause is a maximum term if it contains all attributes in either positive form or negative form.

Theorem 3.6. [32] *For a clause set with $m-1$ attributes, if its clauses are all maximum terms, the clause set is unsatisfiable if it contains 2^{m-1} clauses.*

Theorem 3.7. [32] *Given a clause set $\sum=C_1\wedge C_2\wedge\cdots\wedge C_n$, let P_i be the set of all the maximum terms got from C_i by using extension rule, and set NM the number of distinct maximum terms obtained from $\sum$.*

$$NM=\sum_{i=1}^{n}|P_i|-\sum_{1\le i<j\le n}|P_i\cap P_j|+\sum_{1\le i<j<l\le n}|P_i\cap P_j\cap P_l|-\ldots+(-1)^{n+1}|P_1\cap P_2\cap\ldots\cap P_n|,$$
$$|P_i|=2^{m-1-|C_i|},$$
$$|P_i\cap P_j|=\begin{cases}0 & \text{there are complementary forms of the same attribute in } C_i \text{ and } C_j,\\ 2^{m-1-|C_i\cup C_j|} & \text{otherwise.}\end{cases} \quad (16)$$

By knowledge compilation using extension rule, each pair of clauses in the new clause set contains complementary forms of the same attribute [33]. So $NM=\sum_{i=1}^{n'}|P_i'|$, where P_i' is the set of all the maximum terms got from a clause after knowledge compilation. Algorithm 3.1 shows process of knowledge compilation [33].

Algorithm 3.1. Knowledge compilation using the extension rule.

Input: Let $\sum_1=C_1\wedge C_2\wedge,\cdots,\wedge C_n$ be a set of clauses; $\sum_2=\sum_3=\varnothing$.
While $\sum_1\neq\varnothing$
Loop
  Select a clause from $\sum_1$, say C_1, and add it into $\sum_2$
  While $i\prec$ the number of clauses in $\sum_1$
  Loop
    While $j\prec$ the number of clauses in $\sum_2$
    Loop
      If C_i and C_j contain complementary forms of the same attribute, skip;
      Else if C_i subsumes C_j, eliminate C_j from $\sum_2$;
      Else if C_j subsumes C_i, eliminate C_i from $\sum_1$;
      Else extend C_j on a variable using extension rule.
      $j=j+1$.
    End loop
    $i=i+1$.
  End loop
$\sum_3=\sum_3\cup\sum_2;\sum_2=\varnothing$
End loop
Output: $\sum_3$ is the result of the compilation process.

Theorem 3.8. *Set* $f_c=\vee(C_{ij}^*)$ *and* $f_c'=(\vee(C_{ij}^*)\vee t)\wedge(\vee(C_{ij}^*)\vee\bar{t})$, *where* t *is an attribute not included by* C_{ij}^*. $f_c(V_B(a_1),\cdots,V_B(a_{m-1}))=1$ *if* $f_c'(V_B(a_1),\cdots,V_B(a_{m-1}))=1$, *where* $V_B(\cdot):a\to\{0,1\}$ *such that* $a\in B$ *if* $V_B(a)=1$.

Proof. Set $f_{ct}'=\vee(C_{ij}^*)\vee t$ and $f_{c\bar{t}}'=\vee(C_{ij}^*)\vee\bar{t}$.

Necessary:

If $f_c(V_B(a_1),\ldots,V_B(a_{m-1}))=1$, it is direct to find $f_{ct}'(V_B(a_1),\ldots,V_B(a_{m-1}))=f_{c\bar{t}}'(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
So $f_c'(V_B(a_1),\cdots,V_B(a_{m-1}))=1$.

Sufficiency:

i. $V_B(t)=1$
$f_c'(V_B(a_1),\ldots,V_B(a_{m-1}))=1$ means $f_{c\bar{t}}'(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
And $V_B(t)=1\Rightarrow V_B(\bar{t})=0$, so $f_c(V_B(a_1),\ldots,V_B(a_{m-1}))=1$;
$V_B(t)=0$
$f_c'(V_B(a_1),\ldots,V_B(a_{m-1}))=1$ means $f_{ct}'(V_B(a_1),\ldots,V_B(a_{m-1}))=1$.
$V_B(t)=0$s, so $f_c(V_B(a_1),\cdots,V_B(a_{m-1}))=1$.□

The positive or negative form of an attribute is called the literature of the attribute. The literature set for one clause is composed by the literatures of its attributes.

Theorem 3.9. *If the literature set for a maximum term of conditional attributes is not a superset of any clause in* f_{IS}^{**}, *the attributes with negative form in the maximum term is a reduction; if the literature set for a maximum term of conditional attributes is not a superset of any clause in the extended result of* f_{IS}^{**}, *the attributes with negative form is a reduction.*

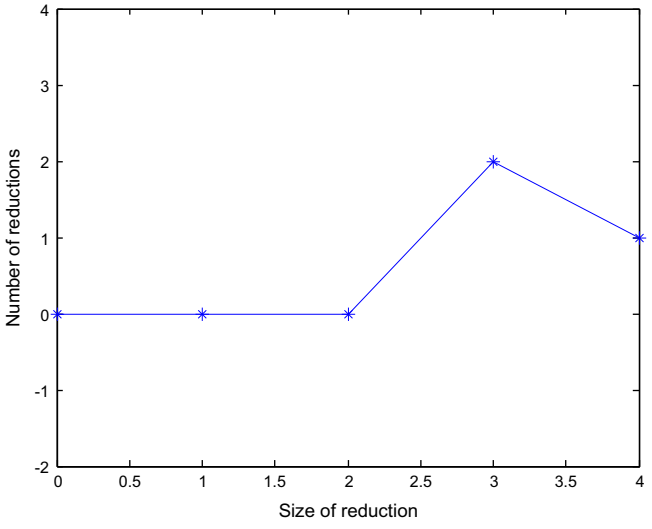

Fig. 1. The distribution of reductions with different size.

Table 4
Algorithm 3.2 on Example 2.1.

Clause	L	NL	j	$C(m-1-L,\ j-NL)$	Count set
a_3	1	0	–	–	{0, 4, 6, 4, 1}
			1	3	{0, 1, 6, 4, 1}
			2	3	{0, 1, 3, 4, 1}
			3	1	{0, 1, 3, 3, 1}
$a_4\vee\bar{a}_3$	2	1	–	–	{0, 0, 3, 3, 1}
			2	2	{0, 0, 1, 3, 1}
			3	1	{0, 0, 1, 2, 1}
$a_1\vee a_2\vee\bar{a}_3\vee\bar{a}_4$	4	2	–	–	{0, 0, 0, 2, 1}

Table 5
Dataset description.

	Dataset	Sample	Attribute	Class
1	Zoo	101	18	7
2	Soybean	47	36	4
3	Voting records (voting)	435	17	2
4	Connectionist bench (connectionist)	208	61	2
5	Pima Indians diabetes (pima)	768	9	2
6	LED display domain (LED)	1000	25	10
7	Hepatitis	155	20	2
8	Breast tumor diagnosis (breast)	699	10	2

Table 6
Number of clauses.

Dataset	Number with Definition 3.4	Exact number	Conditional attributes	Largest space in 3.4	Largest space of exact number	Reduced proportion (%)
Zoo	3873	956	17	65,841	16,252	75.32
Soybean	810	615	35	28,350	21,525	24.07
Voting	44,859	10,548	16	7,17,744	1,68,768	76.49
Connectionist	10,767	91	60	6,46,020	5460	99.15
Pima	1,34,000	82	8	10,72,000	656	99.94
LED	4,49,250	4,28,815	24	1,07,82,000	1,02,91,560	4.55
Hepatitis	5950	3097	19	1,13,050	58,843	47.95
Breast	1,10,378	291	9	9,93,402	2619	99.74

According the Theorem 3.8, it is easy to observe that f_{IS}^{**} is equal to the clause set extended by knowledge compilation. So we just give the proof of the extended set.

Proof. Set $f' = C_1' \wedge C_2' \wedge \cdots \wedge C_{n'}'$ as the extended clause set of f_{IS}^{**}. $mt$is a maximum term of conditional attributes, where $LS_{C_i'} \not\subset LS_{mt}$s and $LS_{C_i'} \neq LS_{mt}$ for $i = 1, 2, \ldots, n'$.$LS_{C_i'}$ is the literature set of C_i';LS_{mt} is the literature set of mt. When$LS_{C_i'} \not\subset LS_{mt}$, mt cannot be extended by C_i'. $S_C = \{mt' : LS_{C_i'} \subseteq LS_{mt'}\}$, where mt' is a maximum term of conditional attributes. For a set B of attributes with negative form in mt, $mt(V_B(a_1), \ldots, V_B(a_{m-1})) = 0$, so$mt'(V_B(a_1), \ldots, V_B(a_{m-1})) = 1$. Set $\tilde{S}_C = \{\vee(LS_{mt'} - LS_{C_i'}) : mt' \in S_C\}$s. For a set of the conditional attributes not in C_i', $\tilde{S}_C$includes all its maximum terms, so $\wedge\{\tilde{S}_C\} = 0$for any assignment. $\wedge\{S_c(V_B(a_1), \ldots, V_B(a_{m-1}))\} = C_i'(V_B(a_1), \ldots, V_B(a_{m-1})) \vee \{\wedge\{\tilde{S}_c(V_B(a_1), \ldots, V_B(a_{m-1}))\}\} = 1$, so$C_i'(V_B(a_1), \cdots, V_B(a_{m-1})) = 1$. As a result,$f'(V_B(a_1), \cdots, V_B(a_{m-1})) = 1$. $B$is a solution of attribute reduction. □

Theorem 3.10. *The number of sets as attribute reduction is* $2^{m-1} - NM$, *where NM is the number of distinct maximum terms obtained from the extended* f_{IS}^{**}, *and* $m-1$ *is the number of conditional attributes.*

Proof. It is straightforward according to Theorems 3.7 and 3.9. □

For the Example 2.1, $f_{IS}^{**} = (a_1 \vee a_2 \vee a_3 \vee a_4) \wedge (a_1 \vee a_2) \wedge (a_2 \vee a_4) \wedge (a_2 \vee a_3) \wedge a_4 \wedge a_3$ according to Table 3. The result of knowledge compilation is $\sum = a_3 \wedge (a_4 \vee \bar{a}_3) \wedge (a_1 \vee a_2 \vee \bar{a}_3 \vee \bar{a}_4)$. The number of attribute reductions is $2^4 - (2^{4-1} + 2^{4-2} + 2^{4-4}) = 3$.

After complication knowledge, every clause has complementary literatures with all other clauses. That is to say that the maximum terms extended by each clause are totally different. The supplementary of maximum terms extended by the clauses is the set of all solutions. It is not generally necessary to obtain all solutions, because it will take memory cost of 2^{m-1}. In this work, the distribution of solution is found without extending maximum terms, which is shown in Algorithm 3.2. The distribution is useful to analyze the characteristic of solutions, which has been verified in experiments in Section 4.

Algorithm 3.2. Distribution of attribution reduction

Input: Clause set after knowledge compilation
$f' = C_1' \wedge C_2' \wedge \ldots \wedge C_{n'}'$; a count set includes m elements, where $m-1$ is the amount of conditional attributes.
Set$i = 0$.
While $i \leq m-1$
Loop
Value of the$(i+1)$ *th*element in count set is $C(m-1, i) = (m-1)!/i!(m-1-i)!$.
End loop
Set$i = 1$.
While $i \leq n'$
Loop
SelectC_i'fromf'.
Denote that the number of attributes in$C_i'$is L, and the number with negative form is NL.
The value of $(NL+1)$ *th*element in count set is decreased 1.
Set$j = NL+1$.
While$j \leq m-1-L+NL$
Loop
The$(j+1)$ *th*element in count set is decreased$C(m-1-L, j-NL) = (m-1-L)!/((j-NL)!(m-1-j-L+NL)!)$
$j = j+1$
End loop
$i = i+1$.
End loop
Output: The distribution of attribute reduction is in the count set.

For the Example 2.1, the result of knowledge compilation is $\sum = a_3 \wedge (a_4 \vee \bar{a}_3) \wedge (a_1 \vee a_2 \vee \bar{a}_3 \vee \bar{a}_4)$. The count set after initialization is {1, 4, 6, 4, 1}. Table 4 shows the process of Algorithm 3.2 on Example 2.1. Fig. 1 illustrates the distribution of reductions with different size.

Definition 3.8. *Minimal reduction* has the smallest number of attributes in all possible solutions of attribute reduction.

The number of attributes in minimal reduction can be obtained according to the distribution. With this prior information, the minimal reduction can be found by a simple forward search process.

Algorithm 3.3. Forward search with prior information

Input: The amount of attributes in minimal reductionn_1, and an empty setMR.
Find $\vee(c_{ij}^*)$in a^*-improved discernibility function has only one attributes; then the attribute is inserted intoMR.
Set$CS = A - \{a^*\} - MR$,$PS = TS = \varnothing$, and $n_1 = n_1 - |MR|$ where $|\cdot|$is the size of the object.
While $|POS_{MR \cup TS}(\{a^*\})| \neq |POS_{A-\{a^*\}}(\{a^*\})|$
Loop
Pick n_1 attributes of CS to composeTS.
If$TS \notin PS$,$PS = \{TS\} \cup PS$.
End loop
$MR = MR \cup TS$.
Output:MR.

According to Fig. 1, the amount of attributes in minimal reductions for Example 2.1 is 3 ($n_1 = 3$). $\vee(c_{ij}^*)$ for sample 3 and

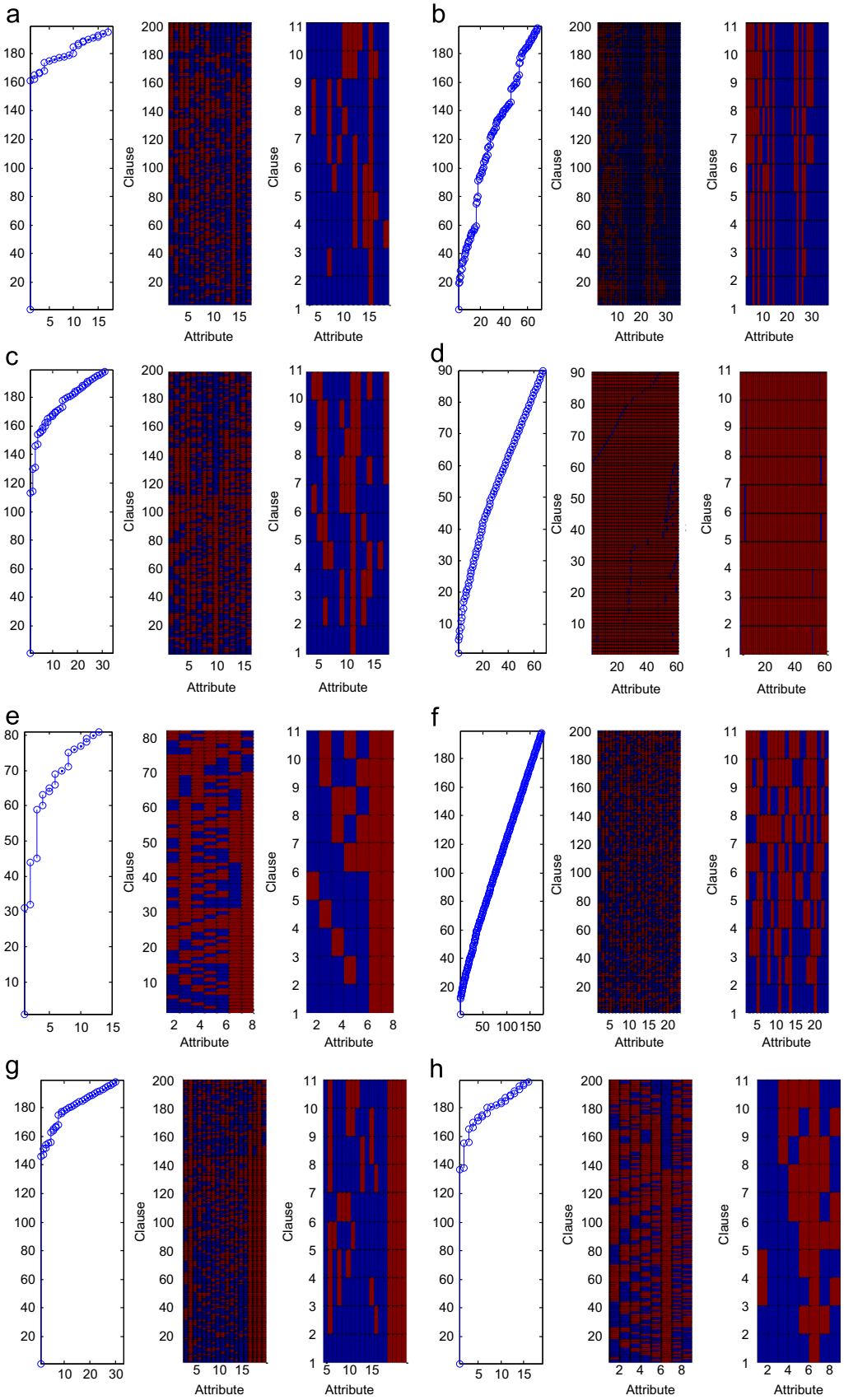

Fig. 2. Comparison of attribute values among different classes. a: Zoo; b: Soybean; c: Voting; d: Connectionist; e: Pima; f: LED; g: Hepatitis; h: Breast. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

sample 4 isa_4 and $\vee(c_{ij}^*)$ for sample 3 and sample 5 is a_3, illustrated in Table 3. So $MR=\{a_3, a_4\}$, and $n_1=1$. Assume $TS=\{a_1\}$; then $|POS_{MR\cup TS}(\{a^*\})|=|POS_{A-\{a^*\}}(a^*)|$. So a minimal reduction is$\{a_1, a_3, a_4\}$.

4. Experiments

In this section, several datasets are employed to verify the work. The number of clauses is discussed to analyze the factors playing role in computational cost. Table 5 shows the details of the used UCI datasets.

4.1. Analysis of a*-improved discernibility function

An important aspect of attribute reduction based on rough set is computational cost. According to Definition 3.4, the number of clauses ina^*-improved discernibility function is

$$\frac{n(n-1)}{2}-\frac{l_1(l_1-1)}{2}-\frac{l_2(l_2-1)}{2}-\cdots-\frac{l_w(l_w-1)}{2}=\frac{n^2-(l_1^2+l_2^2+\cdots+l_w^2)}{2}, \tag{17}$$

where n is the total number of samples and l_i is the number of samples in one class. But its exact number of clauses is smaller because of repeated clauses. Table 6 gives the number in Definition 3.4 and exact number of clauses. It is observed that the exact numbers of Connectionist, Pima, and Breast are much smaller than their number of Definition 3.4, since their proportions of repeated clauses are larger than 99%. The proportion of repeated clauses is decided by the value distribution of attributes. Similar values of samples in the same class and large distinction among different classes support repeated clauses. Another factor playing role in space cost is the size of clauses. In Table 6, the largest space cost is shown, by assuming every clause includes all conditional attributes.

4.2. The number of clauses after knowledge compilation

During the process of knowledge compilation using extension rule, one clause is deleted from the clause set, when it is subsumed by another clause [33]. So subsuming cuts down the computational cost of space and time in knowledge compilation. Since the amount of clauses in a^*-improved discernibility function is large, audiences cannot find anything in the figures, when all the clauses are shown. Hence, 200 clauses are randomly picked in Fig. 2 which illustrates the concept of subsuming. Every figure in Fig. 2 includes three subfigures. The middle one describes the elements of a^*-improved-discernibility function. One row represents one clause. Every column expresses one attribute. It must be noted that all attributes in discernibility function are with positive form. Hence, we do not underline literatures in this section to reduce the burden of description. Red denotes the attribute in a clause; blue means the attribute not in a clause. For a clause, its red attributes contains all red features of another clause. Then, the clause is directly deleted. The third subfigure shows only the first 10 clauses in order to explain subsuming clearly. For example, in the third subfigure of Fig. 2a, the red attributes of the first clause are included by other 9 clause. So the 9 clauses are subsumed by the first and deleted in the process of knowledge compilation. The first subfigure shows the sets of subsuming in the second subfigure. The clauses in every two circles are in a subsuming set where the attributes in every clause contain the attributes of the first clause. That is to say that only the first clause in a subsuming set is kept in knowledge compilation. So the clauses of Zoo, Voting, Hepatitis and Breast are deleted dynamically, shown in Fig. 2a, g, and h. The details of final results are illustrated in Table 7.

4.3. Distribution of solutions

According to Theorem 3.10, the amount of all possible solutions can directly be calculated according to distinct maximum terms. On the basis of clause set after knowledge compilation, the number of distinct maximum terms is found with Theorem 3.7. In Table 8, it can be observed that the proportion of clause to distinct maximum term (PCDMT) is small except Connectionist. Especially, for Soybean, the proportion is approximate to 0. This verifies that extension rule really saves computation cost for rough set based attribute reduction. Fig. 3 illustrates the distribution of solutions. For Zoo, Soybean, Connectionist, Hepatitis, the number of solutions taking $\lceil(m-1)/2\rceil$ conditional attributes is largest. In Table 8, we can find that the number of solutions in these four datasets is large. LED also has enough solutions; however, its 17 attributes are based on the first 7 attributes [38]. Especially,

Table 8
Description of maximum terms.

Dataset	Clauses in extended set	Distinct maximum terms	PCDMT (%)	Number of solutions
Zoo	55	56,624	0.1	74,448
Soybean	2826	3.092×10^9	≈0	3.1268×10^{10}
Voting	30	65,372	0.05	164
Connectionist	79	92	85.87	1.1529×10^{18}
Pima	24	98	24.49	158
LED	18,315	1,62,24,043	0.11	5,53,173
Hepatitis	1100	1,72,558	0.64	3,51,730
Breast	35	373	9.38	139

Table 7
Variation of clause number.

Dataset	Exact clauses	Subsuming sets	Clauses deleted	Proportion deleted (%)	Addition of clauses	Clauses in extended set
Zoo	956	14	942	98.54	41	55
Soybean	615	99	516	83.90	2727	2826
Voting	10,548	15	10,533	99.86	15	30
Connectionist	91	69	22	24.18	10	79
Pima	82	5	67	81.71	19	24
LED	4,28,815	1091	4,27,724	99.75	17,224	18,315
Hepatitis	3097	63	3034	97.97	1037	1100
Breast	291	20	271	93.13	15	35

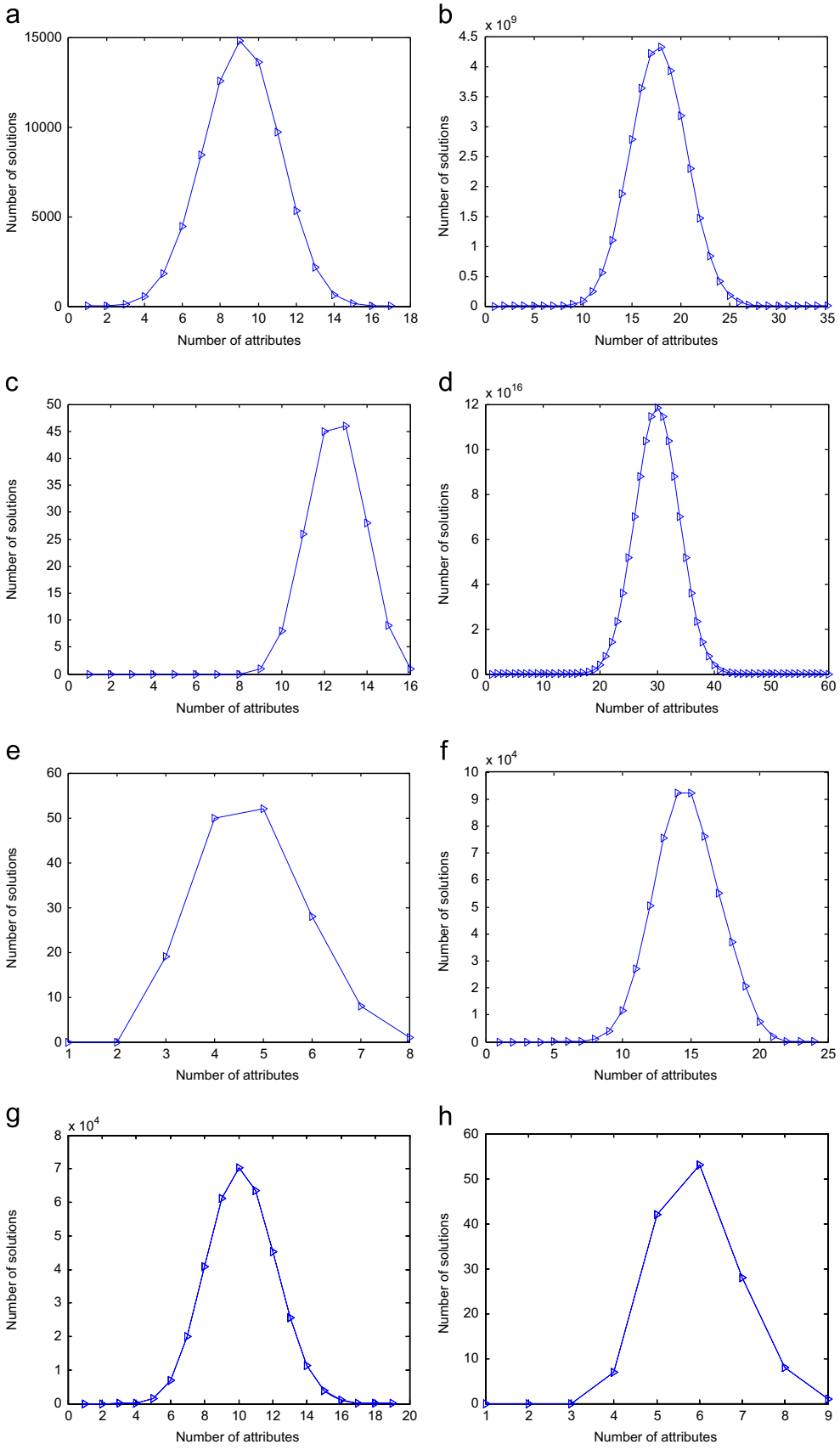

Fig. 3. Distribution of reductions. a: Zoo; b: Soybean; c: Voting; d: Connectionist; e: Pima; f: LED; g: Hepatitis; h: Breast.

Table 9
Time cost of discernibility function and knowledge compilation.

Dataset	Discernibility function		Knowledge compilation			
	Number in Definition 3.4	Time cost	Exact clauses	Clauses in extended set	Difference	Time cost
Zoo	3873	266.3	956	55	901	75,200
Soybean	810	94.7	615	2826	−2211	381.2
Voting	44,859	466.4	10,548	30	10,518	7168.8
Connectionist	10,767	273.8	91	79	12	3.2
Pima	1,34,000	1668.8	82	24	58	0
LED	4,49,250	35177.3	4,28,815	18,315	4,10,500	13,32,9542
Hepatitis	5950	235.8	3097	1100	1997	968.8
Breast	1,10,378	852.4	291	35	256	12.6

Table 10
Time cost of finding minimal reduction.

Dataset	Solution distribution	Find minimal reduction (MR)					
		Attributes in MR	Necessary attributes	Number of MR	Combination	Proportion of MR (%)	Time cost
Zoo	6.4	1	0	1	17	5.9	9.3
Soybean	695.1	2	0	4	595	6.7	113.6
Voting	3.7	9	7	1	36	2.8	49.4
Connectionist	91.8	1	0	8	60	13.3	13.3
Pima	2.4	3	0	19	56	33.9	34.5
LED	966.4	5	0	1	42504	2.4×10^{-7}	2452465.3
Hepatitis	77.2	3	0	15	969	1.5	97.4
Breast	11.1	4	1	7	56	12.5	33.6

$|POS_B(\{a^*\})| = |U|$ where B only has the first 6 attributes. So the amount of solutions having $\lceil (m-1-6)/2 \rceil + 6$ conditional attributes is largest.

4.4. Analysis of computational cost

The number of clauses in the extended set in Table 8 shows memory cost. In this section, the efficiency of time cost is discussed. Table 9 shows the cost of time to build a^*-improved discernibility function and knowledge compilation, where the unit is micsecond and the value is the average of 10 times experiments. It can be observed that the time cost for a^*-improved discernibility function is decided by the amount of clauses defined by 3.4. In the process of knowledge compilation, the difference between a^*-improved discernbility function and the extended set plays role in the time cost. However, the similar rule with a^*-improved discernibility function cannot be found. That is because that some clauses are directly deleted because of subsuming.

Table 10 gives the time cost of solution distribution and minimal reduction. By solution distribution, the number of attributes in minimal reduction is known. An attribute is necessary, when it is the unique element in one $\vee(c_{ij}^*)$. The necessary attributes must be contained by the minimal reduction, so they are inserted into the selected set before search process. Assume the number of necessary attributes is n_n, the size of minimal reduction is n_{MR}, $m-1$ is the number of attributes in conditional set. The difficult degree of finding a minimal reduction can be evaluated by the combinations of attributes with the size of $n_{MR}-n_n$. The number of combinations is

$$\frac{(m-n_n-1)\times(m-n_n-2)\times\ldots\times(m-n_{MR})}{(n_{MR}-n_n)\times(n_{MR}-n_n+1)\times\ldots\times 1} \tag{18}$$

Proportion of minimal reduction (MR) is

$$\frac{(n_{MR}-n_n)\times(n_{MR}-n_n+1)\times\ldots\times 1}{(m-n_n-1)\times(m-n_n-2)\times\ldots\times(m-n_{MR})} \times \text{the number of } MR \tag{19}$$

Larger value of formula (19) means less difficulty to find a minimal reduction. Besides the proportion of MR, the amount of samples also plays role in the time cost of minimal reduction, since the subset of selected attributes and necessary attributes must be verified to be a reduction. Fig. 4 illustrates the time cost of activities in Table 9 and Table 10. For Zoo, Voting, LED, and Hepatitis, the time cost in knowledge compilation is larger than other activities. Soybean is the only dataset, where the number of clauses after knowledge compilation increases nearly 4 times of the exact clauses, so the time cost of solution distribution is larger than other. The clauses in Definition 3.4 of Connectionist and Prima are 10767 and 134000, respectively. So building discernibility function needed largest time cost.

5. Conclusions

In this work, we import extension rule to rough set based attribute reduction. Extension rule, a propositional satisfiability problem (SAT) method, checks the satisfiability by using inverse of resolution. By employing extension rule on discernibility function, the distribution of attribute reductions with different size is found. It provides a new method to describe the solutions of attribute reduction based on rough set. After obtaining the distribution, it is easy to find a minimal attribute reduction. In addition, we define a new discernibility matrix for both inconsistent and consistent information system.

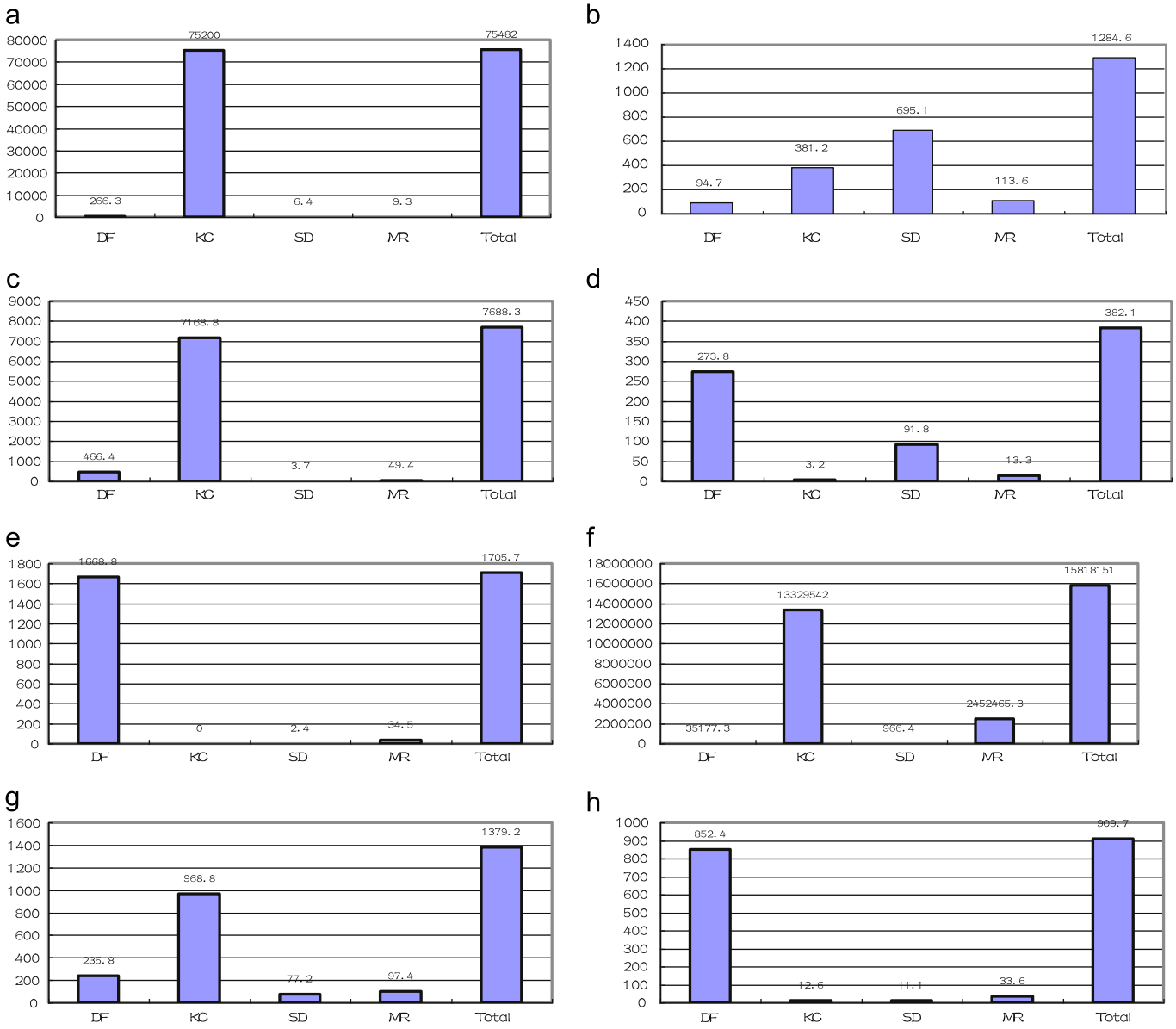

Fig. 4. Cost of time. DF expresses the time cost of building discernibility function; KC is the time cost of knowledge compilation; SD is the time cost of solution distribution; MR is to find the minimal reduction. a: Zoo; b: Soybean; c: Voting; d: Connectionist; e: Pima; f: LED; g: Hepatitis; h: Breast.

References

[1] Y. Liu, Dimensionality reduction and main component extraction of mass spectrometry cancer data, Knowledge-Based Systems 26 (2012) 207–215.

[2] J. Lu, T. Zhao, Y. Zhang, Feature selection based-on genetic algorithm for image annotation, Knowledge-Based Systems 21 (8) (2008) 887–891.

[3] K. Ron, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.

[4] P. Bermejo, L. Ossa, J. Gámez, J. Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowledge-Based Systems 25 (1) (2012) 35–44.

[5] Z. Xu, I. King, M. Lyu, J. Rong, Discriminative semi-supervised feature selection via manifold regularization, IEEE Transactions on Neural Networks 21 (7) (2010) 1033–1047.

[6] Q. Hu, X. Che, L. Zhang, D. Yu, Feature evaluation and selection based on neighborhood soft margin, Neurocomputing 73 (10-12) (2010) 2114–2124.

[7] Y. Chen, D. Miao, R. Wang, K. Wu, A rough set approach to feature selection based on power set tree, Knowledge-Based Systems 24 (2) (2011) 275–281.

[8] N. Parthaláin, Q. Shen, R. Jensen, A Distance measure approach to exploring the rough set boundary region for attribute reduction, IEEE Transactions on Knowledge and Data Engineering 22 (3) (2010) 305–317.

[9] M. Dash, H. Liu, Consistency-based search in feature selection, Artificial Intelligence 151 (1-2) (2003) 155–176.

[10] D. Huang, T. Chow, Effective feature selection scheme using mutual information, Neurocomputing 63 (2005) 325–343.

[11] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[12] J. Sotoca, F. Pla, Supervised feature selection by clustering using conditional mutual information-based distances, Pattern Recognition 43 (6) (2010) 2068–2081.

[13] Z. Pawlak, Rough sets, Parallel Programming 11 (5) (1982) 341–356.

[14] Z. Pawlak, Rough Sets—Theoretical Aspects of Reasoning about Data, The Netherlands: Kluwer Academic Publishers, Dordrecht, 1991.

[15] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences. 177 (1) (2007) 3–27.

[16] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (1) (2007) 28–40.

[17] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (1) (2007) 41–73.

[18] W. Skarbek, Rough sets for enhancements of local subspace classifier, Neurocomputing 36 (2001) 67–83.

[19] R.W. Swiniarski, L. Hargis, Rough sets as a front end of neural-networks texture classifiers, Neurocomputing 36 (2001) 85–102.

[20] S. Kirkpatrick, B. Selman, Critical behavior in the satisfiability of random boolean expressions, Science 264 (5163) (1994) 1297–1301.

[21] A. Skowron, C. Rauszer, "The discernibility matrices and functions in information systems", in: R. Słowiński (Ed.), Intelligent Decision Support Handbook of Application and Advances of the Rough Sets Theory, Kluwer academic publishers, London, 1992, pp. 331–362. (Chapter 2).

[22] Y. Chen, Classifying credit ratings for Asian banks using integrating feature selection and the CPDA-based rough sets approach, Knowledge-Based Systems 26 (2012) 259–270.
[23] D. ŚLęzak, Degrees of conditional (in) dependence: a framework for approximate Bayesian networks and examples related to the rough set-based feature selection, Information Sciences 179 (2009) 197–209.
[24] N. Parthaláin, Q. Shen, Exploring the boundary region of tolerance rough sets for feature selection, Pattern Recognition 42 (5) (2009) 655–667.
[25] N. Zhong, J. Dong, S. Ohsuga, Using rough sets with heuristics for feature selection, Journal of Intelligent Information Systems 16 (2001) 199–214.
[26] C. Bae, W. Yeh, Y. Chun, S. Liu, Feature selection with intelligent dynamic swarm and rough set, Expert Systems with Applications 37 (10) (2010) 7026–7032.
[27] Y. Qian, J. Liang, W. Pedrycz, C. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, Pattern Recognition 44 (8) (2011) 1658–1670.
[28] M. Yang, P. Yang, A novel condensing tree structure for rough set feature selection, Neurocomputing 71 (4-6) (2008) 1092–1100.
[29] Q. Hu, Z. Xie, D. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (12) (2007) 3509–3521.
[30] R. Jensen, Q. Shen, A. Tuson, Finding rough set reducts with SAT, in: proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, August 31–September 3, Regina, Canada, 2005, pp. 194–203.
[31] J. Wang, G. Meng, X. Zheng, The attribute reduce based on rough sets and SAT algorithm, in: Proceedings of the 2008 International Symposium on Intelligent Information Technology Application, 2008, pp. 98–102.
[32] H. Lin, J. Sun, Y. Zhang, Theorem proving based on the extension rule, Journal of Automated Reasoning 31 (2003) 11–21.
[33] H. Lin, J. Sun, Knowledge compilation using the extension rule, Journal of Automated Reasoning 32 (2004) 93–102.
[34] J.W. Grzymala-Busse, Managing Uncertainty in Expert Systems, Kluwer Academic, Dordrecht (1991) 1–13.
[35] Y. Qian, J. Liang, Deyu Li, Feng Wang, Nannan Ma, Approximation reduction in inconsistent incomplete decision tables, Knowledge-Based Systems 23 (5) (2010) 427–433.
[36] S.H. Nguyen, H.S. Nguyen, Some efficient algorithms for rough set methods, in: Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty on Knowledge Based Systems, Granada, Spain, 1996, pp. 1451–1456.
[37] D.Q. Miao, Y. Zhao, Y.Y. Yao, H.X. Li, F.F. Xu, Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model, Information Sciences. 179 (24) (2009) 4140–4150.
[38] D. Aha, Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, International Journal of Man-Machine Studies 36 (2) (1992) 267–287.
[39] R. Bekkerman, R. El-Yaniv, N. Tishby, Y. Winter, Distributional word clusters vs. words for text categorization, JMLR 3 (2003) 1183–1208.
[40] G. Forman, An extensive empirical study of feature selection metrics for text classification, JMLR 3 (2003) 1289–1306.
[41] L. Breiman, J. Friedman, C. Stoneand, R. Olshen, Classification and Regression Trees, CRC Press, Boca Raton, 1984.
[42] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, 1993.
[43] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis. 1 (1997) 131–156.
[44] R. Jensen, C. Cornelis, Fuzzy-rough nearest neighbour classification and prediction, Theoretical Computer Science 412 (42) (2011) 5871–5884.
[45] H.S. Nguyen, Approximate Boolean Reasoning: Foundations and Applications in Data Mining, Transactions on Rough Sets V, LNCS 4100, Springer-Verlag, Berlin, Heidelberg (2006) 334–506.
[46] B. Li, P. Tang, T. Chow, Quantization of rough set based attribute reduction, A Journal of Software Engineering and Applications 5 (2012) 117–123.

Bing Li is currently pursuing the Ph.D. degree at the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. She received her B.Eng. degree and master degree from the College of Computer Science, Northeast Normal University, Jinlin, PR China in 2006 and 2009, respectively. Her current interests include data mining, machine learning, pattern recognition, and their applications.

Tommy W. S. Chow (IEEE M'93–SM'03) received the B.Sc (First Hons.) and Ph.D. degrees from the University of Sunderland, Sunderland, UK. He joined the City University of Hong Kong, Hong Kong, as a lecturer in 1988. He is currently a Professor in the Electronic Engineering Department. His research interests are in the area of Machine learning including supervised and unsupervised learning, data mining, pattern recognition and fault diagnostic. He worked for NEI Reyrolle Technology at Hebburn, England developing digital simulator for transient network analyser. He then worked on a research project involving high current density current collection system for superconducting direct current machines, in collaboration with the Ministry of Defense (Navy) at Bath, England and the International Research and Development at Newcastle upon Tyne. He has authored or coauthored of over 120 technical papers in international journals, 5 book chapters, and over 60 technical papers in international conference proceedings.

Peng Tang is currently pursuing the Ph.D. degree at the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His current interests include data mining, machine learning, pattern recognition, and their applications.