Information Sciences 271 (2014) 65–81

Contents lists available at ScienceDirect

Information Sciences

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

Quick attribute reduct algorithm for neighborhood rough set model

Liu Yong [a,b,*], Huang Wenliang [c], Jiang Yunliang [d,e], Zeng Zhiyong [a]

[a] Institute of Cyber Systems and Control, Zhejiang University, 310027 Hangzhou, China
[b] State Key Laboratory of Industrial Control and Technology, Zhejiang University, 310027 Hangzhou, China
[c] China United Network Communication Corporation, 100032 Beijing, China
[d] Huzhou Teachers College, 313000 Huzhou, China
[e] Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 25 June 2013
Received in revised form 7 January 2014
Accepted 11 February 2014
Available online 20 February 2014

Keywords:
Rough set
Neighborhood rough set
Attribute reduct
Bucket

ABSTRACT

In this paper, we propose an efficient quick attribute reduct algorithm based on neighborhood rough set model. In this algorithm we divide the objects (records) of the whole data set into a series of buckets based on their Euclidean distances, and then iterate each record by the sequence of buckets to calculate the positive region of neighborhood rough set model. We also prove that each record's θ-neighborhood elements can only be contained in its own bucket and its adjacent buckets, thus it can reduce the iterations greatly. Based on the division of buckets, we then present a new fast algorithm to calculate the positive region of neighborhood rough set model, which can achieve a complexity of $O(m|U|)$, m is the number of attributes, $|U|$ is the number of records containing in the data set. Furthermore, with the new fast positive region computation algorithm, we present a quick reduct algorithm for neighborhood rough set model, and our algorithm can achieve a complexity of $O(m^2|U|)$. At last, the efficiency of this quick reduct algorithm is proved by comparable experiments, and especially this algorithm is more suitable for the reduction of big data.

1. Introduction

Attribute reduct, originally presented by Pawlak's in rough set theory [1,2], is a quite useful data preprocessing technique. The core idea of attribute reduct is to obtain the sub attribute set (or feature set) which can keep the same discriminability comparing with the full attribute set. It is also mentioned as a semantic-preserving dimension reduction [3,4], thus the attribute reduct can be implemented in feature selection [5–11], data mining [12–14] and data compression [15], etc.

The classical attribute reduct algorithms are established on the equivalence approximate space and only compatible for discrete data set. They need to scatter the records when processing *continuous numerical data*, denoted by numerical data in the following paper, this will lead to losing of information (including the neighborhood structure information and order structure information in real spaces) [3,16], so the reduct of the numerical data set are strongly related with the methods of scatting. To overcome this drawback, many extensions of classical rough set theory and their corresponding definitions on attribute reduct have been presented, such as fuzzy rough set [4,17–19], tolerance approximate models [20], similarity rough approximate model [21], dominance approximation relation model [22], covering approximation model [23–26]

[*] Corresponding author at: Institute of Cyber Systems and Control, Zhejiang University, 310027 Hangzhou, China. Tel.: +86 138 0571 9977.
E-mail address: cckaffe@hotmail.com (L. Yong).

http://dx.doi.org/10.1016/j.ins.2014.02.093
0020-0255/© 2014 Elsevier Inc. All rights reserved.

and neighborhood granular model [27,28]. Among all the extensions, neighborhood rough set model [5,16,29,30] can be regarded as a specified implementation of the neighborhood granular model. The neighborhood rough set model can process both numerical and discrete data set via the θ-neighborhood set,[1] which will not break the neighborhood structure and order structure of data set in real spaces. Although the attribute reduct of neighborhood rough set model has been successfully implemented in many applications, e.g. feature selection [16], classifier [30], rule learning [31], etc., it still suffers from the low computation efficiency of attribute reduct on neighborhood rough set model, especially in computing the neighborhood of each record, which is a quite usual and inevitable operation in reduct algorithm of neighborhood rough set model.

In classical rough set model, it has been proved that finding the minimal attribute reduct is NP-hard [32], so most of the reduct solutions are aimed to find a reasonable short redut. Generally speaking, there are two main kinds of attribute reduct approaches, i.e. indiscernibility matrix based methods [32–35] and significant metric function based methods [3,10,36,37]. The indiscernibility matrix based methods need to build an indiscernibility matrix, whose elements in that matrix indicate the different attributes between every two records with different decisions (labels), and then combine all the elements in that matrix to obtain the reduct. However, these methods are hard to be extended to other approximation models, e.g. fuzzy rough set model, neighborhood rough set model, etc., furthermore, their temporal and spacial computation costs are also high; The significant metric based methods normally employ a monotonic significant metric, which is related with the positive region, to test whether current selected attribute set is a reduct. The significant metric functions used in reduct algorithms can be positive region (dependency) [3,38], inconsistency [37], and entropy [39,40], etc. The candidate attribute subsets constructing policies used in significant metric function based methods include greedy searching [38], heuristic searching [10], evolutionary computation based searching [41], etc. Among all the policies, the greedy searching is commonly used due to its high efficiency. Especially the forward greedy searching can prefer to generate shorter reduct which satisfies the “shortest bias” in classifier construction, thus it is widely used in machine learning cases. In those significant metric function based methods, sorting methods are also widely adopted to decrease the computation complexities of the attribute reduct algorithms, e.g., Nguyen proposed their attribute reduct algorithm [42] by sorting decision table, Liu proposed their attribute reduct algorithm [43] by sorting the partitioning universe, Hu and Wang proposed their attribute reduct algorithm [44] by quick sorting the two dimensional table. The significant metric function based methods are easy to be implemented in those extended approximation models. However, the time-consuming of attribute reduct algorithms on those extended models will increase significantly due to the promotion of computation complexity on differing the approximation relations between records and attributes under the extended relation models, e.g fuzzy equivalence relation model [45], θ-neighborhood relation model [16].

In this paper, we present a quick reduct algorithm on neighborhood rough set model. As we know, the neighborhood rough set model defines the neighborhood relationship between every two records with θ-neighborhood relation, which concerns the Euclidean distance between those two records less than θ ($\theta > 0$). It needs to iterate whole the sample set to obtain one record’s θ-neighborhood set, thus the computation cost to obtain the θ-neighborhood sets of every records will be approximated to the square of the record number. This complexity will become unacceptable when the number of records in data set increases to a large scale, which may be the core challenge in big data problems. Current attribute reduct algorithms on the data sets with huge records focus on reducing the number of records involving in the calculation of the positive region, such as the fast reduct algorithm (F2HARNRS) in [30]. Although this records reducing based approach can decrease the records involving in the calculation of the positive region, its efficiency is highly sensitive to the distribution of the data set.[2] Thus we present a new quick attribute reduct algorithm on neighborhood rough set model, which tries to optimize the computation of θ-neighborhood on neighborhood rough set model. To the best of our knowledge, there are no attentions are paid on reducing the computation complexity of reduction algorithm via increasing the efficiency of computing θ-neighborhood.

In our approach, we present a hash based method to divide those records into a series of sequenced buckets, and we also prove that each record’s θ-neighborhood elements can only exist in its own bucket and its adjacent buckets. Based on the division of buckets, we then present a new fast algorithm, F-*POS*, to calculate the positive region of neighborhood rough set model, which only needs to iterate the records in three buckets to obtain one record’s θ-neighborhood and can achieve a complexity of $O(m|U|)$, m and $|U|$ are the number of attributes and records containing in the data set. Using the F-*POS* algorithm to calculate the positive region of the neighborhood rough set model, we then present a quick neighborhood rough set reduct algorithm, which can achieve a computation complexity of $O(m^2|U|)$.

The rest of the paper is organized as follows. Section 2 presents related concepts and definitions of the neighborhood rough set model. Section 3 presents our quick attribute reduct algorithm on neighborhood rough set model. Experimental analysis is given in Section 4. Conclusions come in Section 5.

2. Neighborhood rough set model

The core concept of neighborhood rough set model is to extend the equivalent approximation of classical rough set model with neighborhood approximation, which enables it to support both numerical and discrete data types. This section will only

[1] A record x_i’s θ-neighborhood set includes all the records whose distances to x_i are less than θ.

[2] Some bad distribution will lead to few records being reduced in each iteration.

introduce several necessary concepts on neighborhood rough set model and its reduct, some further details can be found in reference [16,30].

In general, a decision system can be denoted as $\langle U, C\bigcup D\rangle$, where $U = \{x_1, x_2, \ldots, x_n\}$ is the set of records, C is the conditional attribute set, $C = \{a_1, a_2, \ldots, a_m\}$, and D is the set of decision attributes.

To illustrate the operations and concepts presented in neighborhood rough set model, an example data set (Table 1) will be used. Here, the table consists of four conditional attributes (a; b; c; d), one decision attribute (e) and four records.

2.1. θ-Neighborhood relation

The neighborhood rough set model is established on the θ-neighborhood relation which uses a distance metric function based neighborhood relation to replace the equivalence relation in classical rough set model.

Definition 1. To a nonempty set $U = \{x_1, x_2, \ldots, x_n\}$, existing a metric function f satisfying:

(1) **Non-negative**: $f(x_i, x_j) \geqslant 0$, if $x_i = x_j$, then $f(x_i, x_j) = 0$;
(2) **Symmetry**: $f(x_i, x_j) = f(x_j, x_i)$;
(3) **Triangle inequality**: $f(x_i, x_j) \leqslant f(x_i, x_k) + f(x_j, x_k)$.

We then note f as the metric function (or distance function) of U, and $\langle U, f\rangle$ is the tolerance space. Normally, distance function f can be implemented with p-norm, Frobenius norm, etc.

Definition 2. Assuming $\langle U, f\rangle$ is the tolerance space, $\forall x_i \in U,\ \theta \geqslant 0$, having

$$\theta(x_i) = \{x | f(x_i, x) \leqslant \theta, x \in U\}$$

We then note $\theta(x_i)$ as the θ-neighborhood set of x_i, and it also denoted as the hypersphere with center x_i and radius θ.

Example 1. Considering the decision system in Table 1, assuming 2-norm is used for the metric function $f(x, x_i)$, we have:

$$f(x_1, x_2) = 0.1175, \quad f(x_1, x_3) = 0.2423, \quad f(x_1, x_4) = 0.3813$$
$$f(x_2, x_3) = 0.2644, \quad f(x_2, x_4) = 0.3289, \quad f(x_3, x_4) = 0.2555$$

Assuming $\theta = 0.2$, then the θ-neighborhood set of x_i can be given as follow:

$$\theta(x_1) = \{x_1, x_2\}, \quad \theta(x_2) = \{x_1, x_2\}, \quad \theta(x_3) = \{x_3\}, \quad \theta(x_4) = \{x_4\}$$

Based on the above definitions, we can obtain the following characteristics:

(1) $\theta(x_i) \neq \emptyset$, for $x_i \in \theta(x_i)$;
(2) $x_j \in \theta(x_i) \Rightarrow x_i \in \theta(x_j)$;
(3) $\bigcup_{i=1}^{n} \theta(x_i) = U$.

From the set $\{\theta(x_i) |\ i = 1, 2, \ldots, n\}$, we can obtain the θ-neighborhood relationship N on U, which can be presented as a symmetrical matrix $M(N) = (r_{ij})_{n\times n}$, if $x_j \in \theta(x_i)$ then $r_{ij} = 1$ otherwise $r_{ij} = 0$.

The θ-neighborhood relation is a specific neighborhood relation, which satisfy the properties of reflexivity and symmetry. And it draws the objects together with similarity or indistinguishability in terms of distances and the samples in the same neighborhood granule are close to each other.

2.2. Neighborhood decision systems (NDS)

In the decision system $\langle U, C\bigcup D\rangle$, U is the set of records (object space), C is the conditional attribute set, D is the decision attribute set, a certain $\theta(\theta > 0)$ and C will generate a θ-neighborhood relationship N. We then call this system θ-neighbor-

Table 1
An example data set.

$x_i \in U$	a	b	c	d	$\Rightarrow e$
x_1	0.10	0.20	0.61	0.20	Y
x_2	0.13	0.22	0.56	0.10	Y
x_3	0.14	0.23	0.40	0.31	N
x_4	0.16	0.41	0.30	0.16	N

hood decision system (we will also call it neighborhood decision system (NDS) in the following sections of this article), denoted as $NDS = \langle U, C \bigcup D, \theta \rangle$. Then the lower and upper approximation for the neighborhood decision system can be presented as follow:

Definition 3. In a neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle$, B is a subset of $C(B \subseteq C)$, for arbitrary $X \subseteq U$, two sets of records, called lower and upper approximations of X in terms of relation N with respect to B, are defined as [30]:

$$\underline{N_B}X = \{x_i | \theta_B(x_i) \subseteq X, x_i \in U\}$$
$$\overline{N_B}X = \{x_i | \theta_B(x_i) \cap X \neq \emptyset, x_i \in U\}$$

Here $\theta_B(x_i)$ is calculated as follow:

$$\theta_B(x_i) = \{x | f(B(x_i), B(x)) \leqslant \theta, x \in U\}$$

And $B(x)$ is a function to extract the sub vector from x, only the dimensions whose attributes are contained in the attribute set B will be chosen. That is $B(x) = \{a(x) | \forall a \in B\}$.

And the lower and upper approximations of the partitions of decision attribute D can be defined as follow:

Definition 4. In a neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle$, $D_1, D_2, \ldots, D_n$ are the record subsets with decisions 1 to n, $\theta_B(x_i)$ is the neighborhood information granules including x_i and generated by attribute set B, $B \subseteq C$, Then the lower and upper approximations of the decision D with respect to attribute set B are defined as [30]:

$$\underline{N_B}D = \bigcup_{i=1}^{n} \underline{N_B}D_i = POS_B(D)$$
$$\overline{N_B}D = \bigcup_{i=1}^{n} \overline{N_B}D_i$$

where

$$\underline{N_B}D_i = \{x_i | \theta_B(x_i) \subseteq D_i, x_i \in U\}$$
$$\overline{N_B}D_i = \{x_i | \theta_B(x_i) \cap D_i \neq \emptyset, x_i \in U\}$$

Similar with the definition in rough set [1], the lower approximation of the decision is defined as the union of the lower approximation of each decision class. The lower approximation of the decision is also called the positive region of the decision, denoted by $POS_B(D)$, which is the subset of records whose neighborhoods consistently belong to one of the decision classes.

Then we can present the definition of dependency.

Definition 5. In a neighborhood decision system [30]:

$NDS = \langle U, C \bigcup D, \theta \rangle$, B is a subset of C, $B \subseteq C$, the dependency degree of D to B is defined as the ratio of consistent records:

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

Obviously, the dependency degree $\gamma_B(D) \in [0, 1]$.

Here, we will use Example 2 to demonstrate the above concepts in neighborhood rough set model.

Example 2. The decision attribute (e) in Table 1 will divide the whole universal into two partitions, that is $U/D = \{D_1, D_2\} = \{\{x_1, x_2\}, \{x_3, x_4\}\}$, assuming $\theta = 0.2$ and 2-norm metric function is used, then based on the θ-neighborhood results in Example 1, the lower and upper approximations of D_1 with respect to attribute set C is:

$$\underline{N_C}D_1 = \{x_1, x_2\}, \quad \overline{N_C}D_1 = \{x_1, x_2\}$$

and the lower and upper approximations of D_2 with respect to attribute set C is:

$$\underline{N_C}D_2 = \{x_3, x_4\}, \quad \overline{N_C}D_2 = \{x_3, x_4\}$$

Thus the positive region of decision can be obtained:

$$POS_C(D) = \underline{N_C}D = \underline{N_C}D_1 \cup \underline{N_C}D_2 = \{x_1, x_2, x_3, x_4\}$$

and then $\gamma_C(D) = 1$.

In the neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle$, $B \subseteq C$, $\forall a \in B$, if the dependency degree $\gamma_{B-a}(D) < \gamma_B(D)$, then the attribute a is indispensably to the set B, otherwise a is redundant to B, that it can be removed from B. Then we can present the definition of reduct in neighborhood decision system similar with classical rough set model.

Definition 6. In a neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle,\ B \subseteq C$, we say attribute subset B is a relative reduct if

(1) $\gamma_B(D) = \gamma_C(D)$;
(2) $\forall a \in B,\ \gamma_{B-a}(D) < \gamma_B(D)$.

From the above definitions, we can conclude two useful theorems, which can be used to decrease computation costs of reduct algorithms in neighborhood decision system.

Theorem 1. *In a neighborhood decision system* $NDS = \langle U, C \bigcup D, \theta \rangle,\ B_1,\ B_2 \subseteq C,\ B_1 \subseteq B_2$, *with the same metric function f and threshold* θ *in computing neighborhoods, we have*

(1) $\forall X \subseteq U,\ \underline{N_{B_1}}X \subseteq \underline{N_{B_2}}X$;

(2) $POS_{B_1}(D) \subseteq POS_{B_2}(D),\ \gamma_{B_1}(D) \leqslant \gamma_{B_2}(D)$.

The proof of Theorem 1 can be found in [16], and it shows adding a new attribute to the attribute subset at least does not decrease the dependency. This property can be used to guarantee the greedy searching policy in attribute selection (reduct) algorithm in [16]. When adding any new attribute into the existing subset in the greedy searching policy, it does not lead to a decrease of the significance of the new subset.

Theorem 2. *In a neighborhood decision system* $NDS = \langle U, C \bigcup D, \theta \rangle$, *if* $A_1,\ A_2 \subseteq C,\ A_1 \subseteq A_2$, *then* $\forall x \in U,\ x \in POS_{A_1}(D) \Rightarrow x \in POS_{A_2}(D)$.

The proof of Theorem 2 can be easily obtained from Theorem 1. It can guarantee that the records belongs to the positive region of A_1 will also belongs to the positive region of A_2, when $A_1 \subseteq A_2$. This characteristic may be quite useful when using forward greedy searching to calculate the reduct. Forward greedy reduct algorithms add one attribute with most significant into the reduct candidate set in each iteration. Then those forward greedy reduct algorithms can ignore those records belonging to positive region in the previous iteration with Theorem 2, for those records will always belong to the positive region regardless of which attribute is added to the reduct candidate set. Theorem 2 is also the theoretical base of the fast reduct algorithm (F2HARNRS) in [16].

3. Quick reduct algorithm for NDS

As the reduct defined in NDS is similar with the classical rough set theory, many classical quick reduct approaches can also be implemented into the reduct of NRS, such as greedy search based approaches [3] guaranteed by the monotonic Theorem 1 of the NDS [16]. Inspired by the Occam's razor theory, the forward greedy searching policy is adopted in Hu's NDS reduct algorithms [16], which will prefer to produce shorter reduct for NDS and may provide a better performance in generalized learning algorithms with those attributes after reduction. With the guarantee of Theorem 2, Hu et al. [16] presented their F2HARNRS algorithm, which constructs the reduct from empty set, adds attributes one by one into the reduct with forward greedy searching policy, and removes the records that are obtained in last iteration and already in positive region to reduce the computation cost. Their experiments also show that their approach can reduce the computation cost significantly comparing with the other approach which does not remove records that are belonged to positive region in the last attribute selection iteration.

Although the F2HARNRS algorithm has noticed to decrease the computation cost by reducing the number of object space ($|U|$), the temporal computation cost for the positive region of NDS is still heavy burden. To calculate the positive region of NDS, it needs to iterate each record in U (or U' in reduced object space), thus it will cost a complexity of $O(m|U|^2)$ to obtain the neighborhood set for each record.[3]

In this section, we will present a bucket based fast algorithm to calculate the positive region of NDS. The calculation for the positive region is the most frequent operation in NDS reduction. We then present our quick reduct algorithm for NDS which also implements both the forward greedy searching policy (based on Theorem 1) and record reducing policy (based on Theorem 2) to construct the reduct of NDS.

With our bucket model for NDS, the temporal complexity for the positive region of NDS can be reduce to $O(m|U|)$. And the corresponding temporal complexity of reduct algorithm for NDS can be reduced to $O(m^2|U|)$ at the worst condition.

3.1. Buckets model for NDS

Before we present our quick reduct algorithm for NDS, the buckets model used in our approach should be presented firstly.

[3] In NDS, the calculation for positive region with single attribute can be reduced to $O(|U|log|U|)$ with the sorting based approach, however, the calculation of positive region with multiple attributes will still be $O(m|U|^2)$, m is the number of attributes.

Definition 7. In a neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle$, x_0 is a special record constructed from U, where $\forall a \in C$, $a(x_0) = \min[a(x_i)]$, $x_i \in U$. Then the records in U can be divided into finite buckets, $B_0, \ldots, B_k$:

$$B_k = \{x_i | x_i \in U \text{ and } \lceil f(x_0, x_i)/\theta \rceil = k\}$$

The bucket can be regarded as a hash function mapping with the distance from x_i to x_0, and the distribution can be illustrated in Fig. 1. Here, if $x_0 \in U$, then $B_0 = \{x_0\}$, otherwise $B_0 = \emptyset$.

Example 3. we can construct the x_0 for the NDS date set in Table 1 as follow:

$$x_0 : a = 0.10, \quad b = 0.20, \quad c = 0.30, \quad d = 0.10$$

Then U in Table 1 can be divided into the following buckets (using 2-norm metric function and $\theta = 0.08$):

$$B_0 = B_1 = B_2 = \emptyset, \quad B_3 = \{x_3, x_4\}, \quad B_4 = \{x_2\}, \quad B_5 = \{x_1\}.$$

Based on the definition of buckets dividing in NDS, we can present the following theorem, which can reduce the searching space when calculating the positive region of NDS.

Theorem 3. *In a neighborhood decision system $NDS = \langle U, C \bigcup D, \theta \rangle$, $B_0, \ldots, B_k$ is the buckets, then $\forall x_i \in B_q$ $(q = 1, 2, 3.., k-1)$, the θ-neighborhood elements of x_i are only contained in B_{q-1}, B_q, B_{q+1}. If $x_i \in B0$, then the θ-neighborhood elements of x_i are only contained in B_0, B_1. If $x_i \in B_k$, then the θ-neighborhood elements of x_i are only contained in B_{k-1}, B_k.*

Proof. From Definition 7, $x_i \in B_q \rightarrow (q-1)\theta < f(x_i, x_0) \leqslant q\theta$. Assuming x' is a record lying in B_{q+2}, then $(q+1)\theta < f(x', x_0) \leqslant (q+2)\theta$. Combing the above two inequalities, we have $f(x', x_0) - f(x_i, x_0) > \theta$. Based on the *triangle inequality* in Definition 1, we have $f(x_i, x') > f(x', x_0) - f(x_i, x_0)$, thus $f(x_i, x') > \theta$, so x' is not the θ-neighborhood element of x_i. Similarly, we can also proof that the record $x' \in B_{q-2}$ is not the θ-neighborhood element of x_i, and the proofs for $x' \in B_{q \pm j}$, $j = 3, 4, 5\ldots$are also similar. The proofs for the conditions that $x_i \in B_0$ and $x_i \in B_k$ are also similar. □

Theorem 3 can also be easily understood based on Fig. 1. Each record in U are assigned into the quarter-circled region with its distance to x_0. The θ-neighborhood set of x_i can be viewed as a hypersphere, which uses x_i as the center and with the radius of θ. We can conclude from Fig. 1, that if $x_i \in B_q$, x_i's θ-neighborhood set elements can only exist in $B_{q-1} \cup B_q \cup B_{q+1}$. Furthermore, if all the elements in x_i's θ-neighborhood set have the same decision attribute, then x_i is an element of positive region.

3.2. Fast positive region computation algorithm

Based on Theorem 3, we can present our fast positive region computation algorithm (F-*POS*, Algorithm 1).

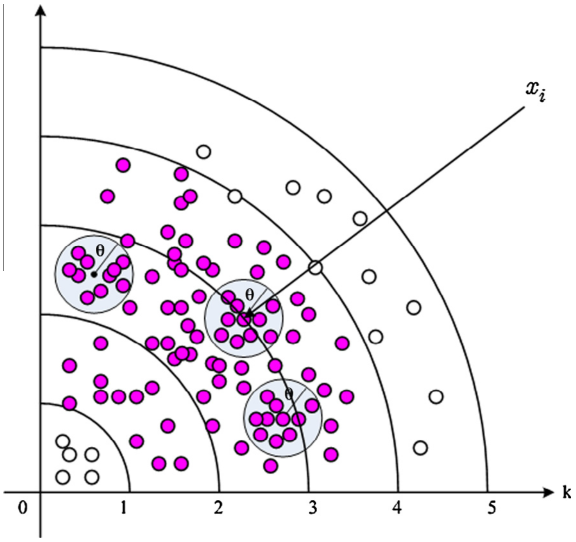


Fig. 1. The distribution of the buckets.

Algorithm 1. $\mathrm{F}-POS(U, P, D, \theta)$

```
Input: $U,\ P,\ D,\ \theta$
Output: $F = \{F_1, F_2, \ldots, F_{|U|}\}$
1 $F_i \leftarrow 0,\ i = 1, \ldots, |U|$;
2 for each $x_i \in U$ do
3    Hash $(P(x_i), B_k)$;
4 end
5 for each $x_i \in U(x_i \in B_k)$ do
6    $flag \leftarrow 0$;
7    for each $x_j \in B_{k-1} \cup B_k \cup B_{k+1}$ do
8      if $f(P(x_i), P(x_j)) \leqslant \theta$ and $D(x_j) \neq D(x_i)$ then
9        $flag \leftarrow 1$;//find a neighbor with different decision values, set flag
10         break;
11      end
12   end
13   if $flag \neq 1$ then
14      $F_i \leftarrow 1$; then
15   end
16 end
```

In our F-*POS* algorithm, the input parameters are $U,\ P(P \subseteq C),\ D$ and θ of current decision system, and the output is a boolean sequence, F, with the length of $|U|$, each F_i in F indicates whether the x_i in $|U|$ is an element of $POS_P(D)$. If F_i equals to 1, it represents x_i is an element of $POS_P(D)$, and when F_i equals 0, it represents x_i is not an element of $POS_P(D)$. Hash $(P(x_i), B_k)$ in step 3 is a hash function to map each x_i into its corresponding bucket B_k, and the mapping function is

$$B_k = \{x_i | \forall x_i \in U, \quad \lceil f(P(x_0), P(x_i))/\theta \rceil = k\}.$$

Here $P(x_i)$ is a function to obtain x_i's values of those attributes belonging to attribute set P. $D(x_i)$ is a function to obtain the decision values of x_i. Based on Theorem 3, we only need to search the items in the buckets $B_{q-1}, B_q, B_{q+1},\ (q = 1, 2, \ldots, k-1)$ when calculating the neighborhood of $x_i(x_i \in B_q)$. Assumed there are k buckets and the records are averagely divided into the buckets, then the computation complexity is $O(m\frac{|U|^2}{k})$, and mostly, k will be closed to $|U|$ and then the computation complexity of Algorithm 1 can be approximately approached to $O(m|U|)$.

Here we will use an example to demonstrate the working process of F-*POS* algorithm and explain why it can be more efficient than the traditional approaches.

Example 4. Considering the example from the data set of Table 1. Assuming $P = \{a, b\},\ \theta = 0.08$, and the 2-norm metric function is used, thus $P(x_0) = \{a = 0.10, b = 0.20\}$. Firstly, we will initialize the flags as $F_1 = F_2 = F_3 = F_4 = 0$ (step1). Based on the definition of bucket model in NDS, we then can obtain the buckets as follows (steps 2–4), shown in Fig. 2:

$$B_0 = \{x_1\}, \quad B_1 = \{x_2, x_3\}, \quad B_2 = \emptyset, \quad B_3 = \{x_4\}$$

Then the algorithm will iterate each record in U, and test whether the records close to itself are also having the same decision values. Based on the definition of the lower approximation of NDS, if any of the records located within the hypersphere of x_i has different decision value, x_i will not be the element of $POS_P(D)$. So in step 8, our algorithm will test both the neighborhood relationship and decision relationship. And using step 13–15 to set flag for the record in positive region, whose elements in θ-neighborhood all have the same decisions.

The calculation for the θ-neighborhood of each record can be shown in Fig. 3. Based on Theorem 3, only the elements in neighbor buckets need to be concerned when computing the θ-neighborhood set. As B_3 only contain one record x_4, and its neighbor bucket B_2 is empty, Fig. 3 does not include the calculating processing of x_4. The Fig. 3(a) shows, both $P(x_2)$ and $P(x_3)$ are the θ-neighborhood of $P(x_1)$, however, their values of decision attribute are different, then $P(x_1)$ is not the element of POS_PD. Similarly, we can also obtain that $x_2,\ x_3$ are also not the elements of POS_PD, only x_4 is the element of POS_PD. Thus the output is $F_1 = F_2 = F_3 = 0,\ F_4 = 1$, that is $POS_P(D) = \{x_4\}$.

From Fig. 3, we also can find that the F-*POS* algorithm only needs to compute the metric function 7 times to obtain the positive region, while the traditional approaches need to compute the metric function 12 times (each record needs to compare with the other three records).

3.3. Fast hash attribute reduct algorithm for NDS

With the fast positive region computation algorithm, we then can present our Fast Hash Attribute Reduct Algorithm (FHARA, Algorithm 2). This algorithm constructs the reduct of NDS with greedy forward searching policy. The output variable

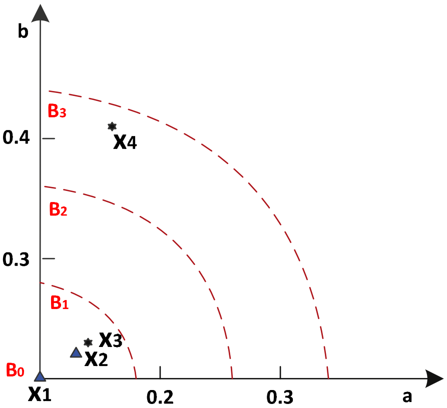
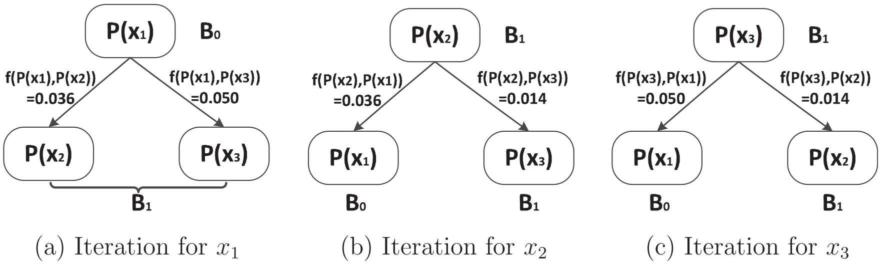


Fig. 2. Bucket distribution of Example 4. We use different color to represent different values of decision attributes.The horizonal coordinate is the value of attribute a, and the vertical coordinate is the value of attribute b, the origin of the coordinates is $x_0 = (0.1, 0.2)$, thus x_1 is located at origin and belongs to B_0. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Iteration for x_1 (b) Iteration for x_2 (c) Iteration for x_3

Fig. 3. Positive region computing processing for each record in U. As B_2 is empty, we do not draw it in (b and c).

reduct starts from empty set, in each iteration, the positive region of $reduct \cup \{a\}$ is calculated with Algorithm 1, only the attribute a with maximal positive region on $reduct \cup \{a\}$ can be added to the output set (step 13). Based on Theorem 2, those records that have belonged to the positive region of current reduct candidate set will also belong to the superset of current reduct candidate set, thus our algorithm will reduce the record set Q in step 14. There are two terminal conditions for Algorithm 2, the first one is that Q is reduced to empty (step 2), which means all the records have been discriminated (belonged to the positive region) with current selected attributes (those attributes construct a reduct set of NDS); the second terminal condition is that no more attribute can increase the positive region of output set *reduct* ($max\_Pos = 0$ in step 12). And here in step 6, we construct the positive region from the output of Algorithm 1, such as $F = \{1, 0, 1, 0\}$, we can obtain the positive region is $Pos = \{x_1, x_3\}$.

Example 5. Considering the NDS in Table 1, assuming $\theta = 0.2$ and 2-norm metric function is used. The example for the Algorithm 2 is shown in Fig. 4. In the beginning, the candidate set *reduct* is empty, and record space $Q = U = \{x_1, x_2, x_3, x_4\}$. In the first round of iteration, each positive region of $reduct \cup \{a\}(a \in C - reduct)$ is calculated, the results are $POS_{\{a\}} = POS_{\{b\}} = POS_{\{d\}} = \{\emptyset\}$, $POS_{\{c\}} = \{x_1, x_4\}$. Then the attribute c, which has maximal positive region ($max\_Pos = \{x_1, x_4\}$), is added to the *reduct*. The algorithm also reduce the record space with $Q \leftarrow Q - max\_Pos$, then $Q = \{x_2, x_3\}$. In the second round of iteration, the results of positive region in the new record space $Q = \{x_2, x_3\}$ are $POS_{\{a,c\}} = POS_{\{b,c\}} = \{\emptyset\}$, $POS_{\{d,c\}} = \{x_2, x_3\}$. Thus the attribute d is added to *reduct*. After reduce the record set Q, it become empty, the algorithm exits and output the reduct $\{c, d\}$.

Assumed that there are $|U|$ records in a decision table and k attributes are finally selected as the reduct among all the m attributes, and selecting an attribute averagely leads to $|U|/k$ samples added into the positive region, the computational times of reduct is

$$m|U| + (m-1)|U|\frac{k-1}{k} + \cdots + (m-k)|U|\frac{1}{k} < \frac{m|U|(1+,\ldots,+k)}{k} = \frac{m|U|(k+1)}{2}$$

And then the average computation complexity of Algorithm 2 is $O(m|U|k)$, in the worst case, the computational complexity of Algorithm 2 is $O(m^2|U|)$.

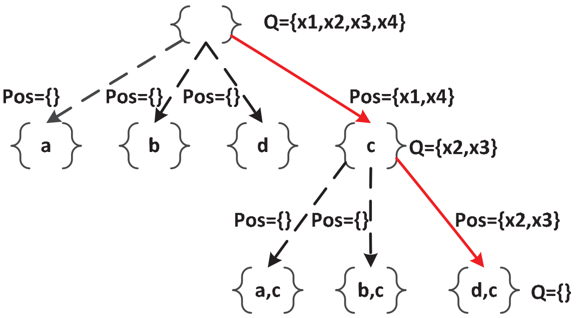


Fig. 4. Executing process for Algorithm 2 in the sample data set of Table 1, $\theta = 0.2$ and 2-norm metric function is used. The red arrow represents that the path is taken in current iteration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2. QuickReduct (Fast Hash Attribute Reduct Algorithm, FHARA)

```
Input: $U,\ C,\ D,\ \theta$
Output: $reduct$
$reduct \leftarrow \phi, Q \leftarrow U$
while $Q \neq \phi$ do
  $max\_Pos \leftarrow \phi, max \leftarrow \phi$; // set $max\_Pos$ and $max$ as empty in each beginning of the 'while' iteration
for each $a \in C - reduct$ do
    $F = \mathrm{F} - POS(Q, reduct \cup \{a\}, D, \theta)$;
    $Pos = \{x_i | F_i = 1\}$; // construct the positive region set based on the output of Algorithm 1.
    if $|max\_Pos| < |Pos|$ then
      $max\_Pos \leftarrow Pos$; // keep the current maximal positive region set.
      $max \leftarrow a$; // keep the candidate attribute with maximal positive region set.
    end
  end
  if $|max\_Pos| > 0$ then
    $reduct \leftarrow reduct \cup \{max\}$;// add attribute with maximal positive region into reduct.
    $Q \leftarrow Q - max\_Pos$; // reduce the record set.
  end
  else
    break; // no more attribute can increase the positive region of $reduct$, exit the iteration (while).
  end
end
```

4. Experiments

In this section, we empirically evaluate our approach with other current state-of-art methods. We carry out the comparable experiments to evaluate the efficiency and correctness of our quick reduct algorithm on NDS. We use 16 data sets from UCI,[4] the details of the data set are shown in Table 2.

In our experiments, we compare with the F2HARNRS (Fast forward heterogeneous attribute reduct based on neighborhood rough sets) and NFARNRS (Naive Forward Attribute Reduct Based on Neighborhood Rough Set model), which were proposed by Hu et al. [16]. All the three algorithms employed in our experiments are using the 2-norm metric functions.

All the algorithms are running on the same software and hardware platforms (CPU:Intel (R) core (TM)2 Quad Q8300 @ 2.50 Hz; RAM: 3.00 GB; Windows XP SP3; Python 2.6.2), and we execute the same algorithm on the same input data set over 200 times and calculate the average executing time.[5]

4.1. Correctness evaluation experiment

In the first experiment, we will evaluate the correctness of our algorithm comparing with F2HARNRS and NFARNRS. And we will compare the reduct results of our FHARA and F2HARNRS with the same θ on the same data set. For each data set, the

[4] http://archive.ics.uci.edu/ml/datasets.html.

[5] We execute 10 times for the shuttle data set and execute 5 times for the data set of skin, as the NFARNRS executing on full skin data set will cost about 50 h each time, and on full shuttle data set will cost about 5 h each time.

Table 2
Data sets used in experiments.

	Data set	Record number	Attribute number	Classies
1	Mushroom	8124	22	2
2	Letter	20,000	16	26
3	Pendigits	10,992	17	10
4	Abalone	4177	7	29
5	Libras movement	360	90	15
6	Voting	435	16	2
7	Wpbc	198	33	2
8	Wdbc	569	31	2
9	Soybean (large)	307	35	19
10	Page blocks	5473	10	54
11	German	1000	20	2
12	Australian	690	14	2
13	Cmc	1473	9	3
14	Segmentation	2310	19	7
15	Shuttle (training set)	43,500	9	5
16	Skin	245,057	3	2

θ is randomly chosen, and the results are shown in Table 3. According to Hu's work [16], the reduct results of NFARNRS and F2HARNRS will be exactly same, so we only compare our FHARAR with F2HARNRS in this experiment. The results show that our algorithm will output exactly the same reduct results as the F2HARNRS.

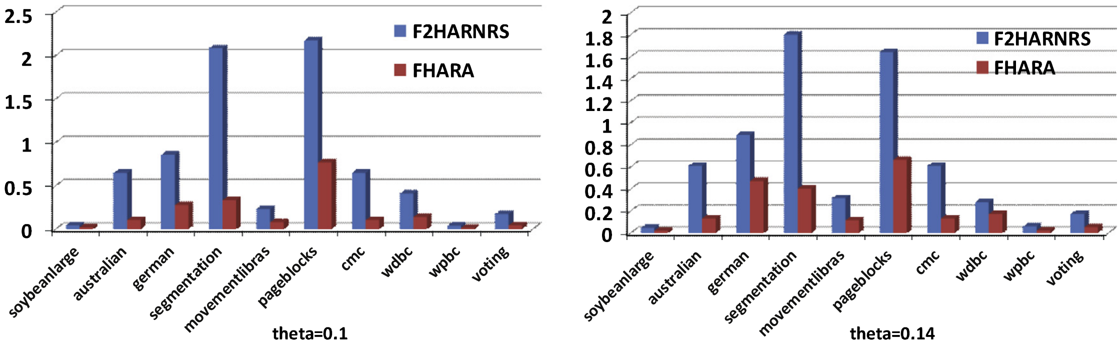
4.2. Computation efficiency evaluation experiments

In the efficiency evaluation experiments, we will use comparable experiments on those 16 UCI data sets to evaluate the computation performance on our approach, NFARNRS, and F2HARNRS. In our experiments, we will set the θ within the interval of $(0, 0.3]$, which is suggested in [16]. For the θ will control the granularity levels of NDS, which will lead to selecting different attributes as reduct, thus vary the classification performance. In Hu et al.'s work [16], they have experimental proved that [0.1, 0.3] is an optimal candidate interval for θ, where most of the classifiers can get good classification performance.

In the first efficiency evaluation experiment, we will evaluate the NDS positive region computation time on different algorithms (our Algorithm 1 and positive region calculation method in F2HARNRS and NFARNRS). As NFARNRS and F2HARNRS use the same method to compute the positive region, we will only compare our approach with the F2HARNRS on the executing time of computing NDS positive region in this experiment. We choose 12 data sets from Table 2, calculate their average executing time on computing NDS positive region, the results are shown in Figs. 5 and 6. The results show our algorithm (Algorithm 1) will achieve better performance on all the 12 data sets concerning the average calculation time of computing positive region. The results also suggest that our Algorithm 1 will save much more computation time in larger data set, such as the results on the data set of page blocks, segmentation, shuttle, and skin. In Fig. 6, we also compare the computation time under different setting of θ on the data sets of shuttle and skin, it also shows that our approach can always achieve better performance than the method used in F2HARNRS and NFARNRS under varied setting of θ. The results show in Fig. 6 also indicate that the computation on reduct may be sensitive to the setting of θ, we will carry out experiments in the later section for a further discussion.

Table 3
Attributes selected by two reduct algorithms, the attribute sequence number is based on the UCI data descriptions.

Data set	θ	F2HARNRS	FHARA
Mushroom	$\theta = 0.17$	5, 10, 11, 20, 2, 19	5, 10, 11, 20, 2, 19
Letter	$\theta = 0.17$	9, 10, 11, 8, 7, 4, 15, 12, 13, 3, 16, 5, 14, 1, 6, 2	9, 10, 11, 8, 7, 4, 15, 12, 13, 3, 16, 5, 14, 1, 6, 2
Pendigits	$\theta = 0.01$	4, 10, 3, 9, 1	4, 10, 3, 9, 1
Abalone	$\theta = 0.1$	4, 3, 5, 1, 6, 8, 7, 2	4, 3, 5, 1, 6, 8, 7, 2
Libras movement	$\theta = 0.11$	56, 18, 1	56, 18, 1
Voting	$\theta = 0.11$	4, 7, 2, 3, 15, 1	4, 7, 2, 3, 15, 1
Wpbc	$\theta = 0.18$	1, 23, 3	1, 23, 3
Wdbc	$\theta = 0.05$	28, 2	28, 2
Soybean (large)	$\theta = 0.12$	26, 28, 15, 22, 23, 31, 1, 3, 6, 10, 7, 4, 16	26, 28, 15, 22, 23, 31, 1, 3, 6, 10, 7, 4, 16
German	$\theta = 0.05$	4, 10, 2, 7, 16	4, 10, 2, 7, 16
Australian	$\theta = 0.3$	14, 13, 5, 9, 8, 1, 2, 11, 6, 4, 12, 3, 10, 7	14, 13, 5, 9, 8, 1, 2, 11, 6, 4, 12, 3, 10, 7
Cmc	$\theta = 0.06$	4, 3, 1, 8, 2, 7, 6, 5, 9	4, 3, 1, 8, 2, 7, 6, 5, 9
Segmentation	$\theta = 0.01$	18, 1, 2	18, 1, 2
Shuttle	$\theta = 0.14$	9, 1, 3, 8, 2	9, 1, 3, 8, 2
Skin	$\theta = 0.2$	3, 1	3, 1



(a) θ=0.1 (b) θ=0.14

Fig. 5. Average positive region calculation time of FHARA and F2HARNRS on ten UCI data sets. The axis of *Y* is computation time (S).

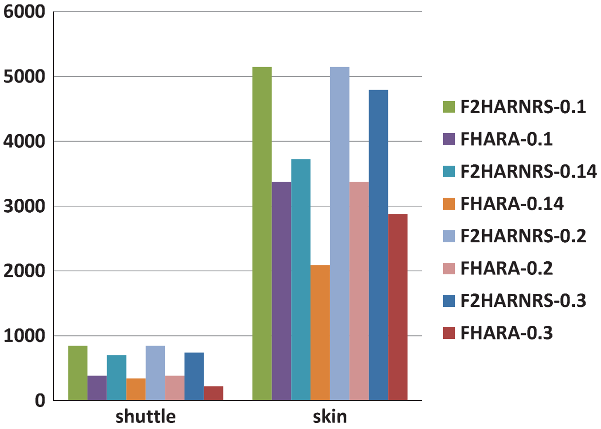


Fig. 6. Average positive region calculation time of FHARA and F2HARNRS on shuttle and skin data sets. The axis of *Y* is computation time (S). As the computation time scales of these two data sets are much larger than the other ten data sets, we use separated figure to display their computation time, and here 'FHARA-XX' means the average positive region computation time of the FHARA with $\theta = XX$.

In the second efficiency evaluation experiment, we will evaluate the average reduct time on these three algorithms. We also choose 12 data sets and increase the number of records used in the reduction and compare the variation of computational time with different reduction algorithms. The results are shown in Fig. 7. According to the results in Fig. 7, the computation time of our FHARA algorithm will always perform better than the other two algorithms' under all the 12 data sets, in particular, FHARA performs significant efficiently than the other two algorithms in the data sets of shuttle, skin, cmc, and pageblocks, etc. In this experiment, the results of FHARA and F2HARNRS on several data sets, such as segmentation, soybean, wpbc, voting and libras movement, look like indistinguishable, this is because that the NFARNRS costs much more computation time and causes the scale of the coordinates to hide the performance difference between FHARA and F2HARNRS when these three algorithms are shown in the same coordinates. So we present the detailed average computation time of the three algorithms on the data sets of libras movement (Table 4), soybean (Table 5), wpbc (Table 6), voting (Table 7) and segmentation (Table 8). Those results on Tables 4–8 show that our algorithm will always performance better than the other two algorithms and the time cost of our FHARA is much less than the other two algorithms, e.g. in the data set of segmentation with 2310 records, the time cost of our FHARA is only 1/400 of the NFARNRS's, and 1/20 of the F2HARNRS's, in the data set of voting with 435 records, the time cost of our FHARA is only 1/604 of the NFARNRS's, and 1/32 of the F2HARNRS's. So the experimental results on Fig. 7 and Tables 4–8 also show that, with the increment of data number, the computation time of F2HARNRS and NFARNRS will increase significantly, while the reduction computation time of our FHARA algorithm increases extremely slowly.

The previous two efficiency evaluation experiments are only sampling θ sparsely, which concerns the algorithms' computation efficiencies under varied record numbers with several sparse sampling of θ. The experiments also show that the reduct algorithms may be sensitive to the setting of θ, so we design the third efficiency evaluation experiment, which will concern the comparison results of different algorithms' reduction time under varied record numbers with dense sampling of θ.

In the third efficiency evaluation experiment, we will evaluate the reduction computation performances of FHARA and F2HARNRS on dense sampling of θ under varied record numbers. We first choose four data sets (mushroom, letter, pendigits,

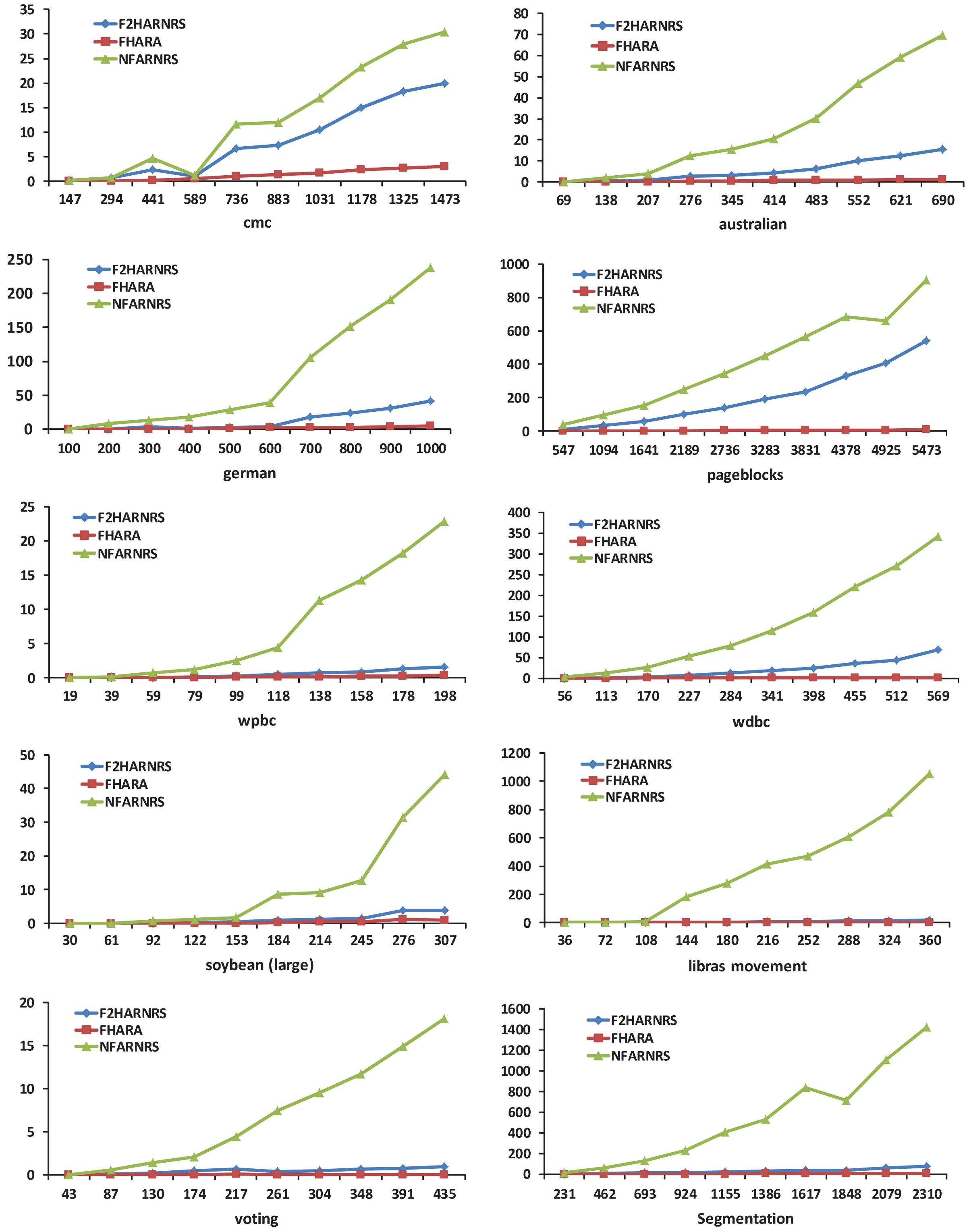


Fig. 7. Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on 12 UCI data sets. The axis of *Y* is computation time (S), the axis of *X* is the number of the data records.

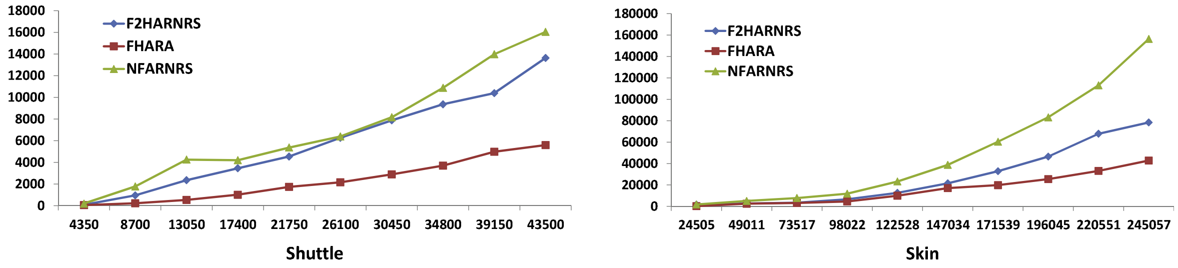


Fig. 7 (*continued*)

Table 4
Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on the data set of libras movement. The time unit of each computation algorithm is second in this table.

Instance numbers	NFARNRS (S)	F2HARNRS (S)	FHARA (S)
36	0.579698544	0.11735334	0.042481741
72	4.842059888	0.523103809	0.123771512
108	7.373427858	0.9355347	0.488945968
144	185.3370565	2.831223393	0.620579746
180	280.5722476	3.974187676	1.055898955
216	412.7207691	5.906941853	1.782873367
252	474.0606134	8.291753344	2.330809774
288	607.4953318	12.00184737	2.967219152
324	780.8056254	15.35826235	3.687163775
360	1051.08904	19.00440763	4.076125301

Table 5
Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on the data set of soybean. The time unit of each computation algorithm is second in this table.

Instance numbers	NFARNRS (S)	F2HARNRS (S)	FHARA (S)
30	0.012783284	0.015936726	0.006542877
61	0.129496368	0.049984568	0.014181613
92	0.670683145	0.11323452	0.039792752
122	1.258890871	0.207719985	0.082658317
153	1.707858241	0.374452716	0.124582039
184	8.667314465	0.936840435	0.233639406
214	9.189168811	1.142132898	0.500788084
245	12.73234688	1.491223766	0.594867788
276	31.36246836	3.773532087	1.129698088
307	44.26507174	3.773137466	1.021072137

Table 6
Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on the data set of Wpbc. The time unit of each computation algorithm is second in this table.

Instance numbers	NFARNRS (S)	F2HARNRS (S)	FHARA (S)
19	0.048943764	0.011553621	0.009396741
39	0.20043664	0.03996785	0.019678915
59	0.747016297	0.096125821	0.051711655
79	1.226693457	0.160700166	0.061534724
99	2.486746936	0.309933309	0.125223952
118	4.401155741	0.490474933	0.158511302
138	11.31393973	0.807055187	0.194374218
158	14.2926557	0.93857374	0.27093441
178	18.27321524	1.333048243	0.318311395
198	22.89722531	1.586228854	0.431918194

and abalone) from Table 2, and then vary their record numbers and the θ is also varied in our experiments from 0 to 0.3. We executed both algorithms (our approach and F2HARNRS) with the same data sets on the same software and hardware platform. Each algorithm in each number of data is executed 200 times and the average computation time, T_{FHARA} and $T_{F2HARNRS}$, are obtained. We then calculate the value $delta - T$ as:

Table 7
Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on the data set of voting. The time unit of each computation algorithm is second in this table.

Instance numbers	NFARNRS (S)	F2HARNRS (S)	FHARA (S)
43	0.045459162	0.02315404	0.005708196
87	0.531415689	0.122049632	0.026565082
130	1.424999305	0.228907869	0.034180377
174	2.079485583	0.46908221	0.041541996
217	4.393870453	0.650496944	0.075839773
261	7.4129489	0.358427023	0.018202335
304	9.520492514	0.485062363	0.021292655
348	11.72057474	0.639863824	0.029692088
391	14.90349265	0.777922696	0.030347262
435	18.11321985	0.956872458	0.031455327

Table 8
Average reduct calculation time of FHARA, NFARNRS and F2HARNRS on the data set of segmentation. The time unit of each computation algorithm is second in this table.

Instance numbers	NFARNRS (S)	F2HARNRS (S)	FHARA (S)
231	12.5507437	0.777636476	0.145922016
462	59.62445105	3.074984451	0.328288336
693	129.7941908	12.68498754	0.762090387
924	229.2793388	14.70641508	1.108778339
1155	406.6787214	18.5959915	1.230106095
1386	532.1880954	28.04397641	1.849015551
1617	838.7946082	38.82496652	2.807765934
1848	713.4534404	38.70158985	2.402179153
2079	1109.017058	60.65118051	4.86389887
2310	1418.995865	72.97050759	3.583880209

$$delta - T = T_{F2HARNRS} - T_{FHARA}$$

Then the number of data, θ, and $delta - T$ can represent a performance curved surface on those two algorithms. A larger $delta - T$ will represent a better performance on FHARA algorithm. The visualized results are shown in Figs. 8–11. In these figures, if the curve surface locates in the plane of (θ, number), it will represent that the temporal performance of FHARA is equal to the performance of F2HARNRS. Then if FHARA performances better than F2HARNRS, the curve surface will be far away upon the plane of (θ, number). The results on Figs. 8–11 show almost all the areas of the curve surfaces are far away upon the (θ, number) plane, thus FHARA performance better than F2HARNRS in almost all the sampling space. In these figures, we also use different colors to represent the values of $delta - T$. The colorful results also show that the proposed method can be more efficient than F2HARNRS under varied θ and record numbers.

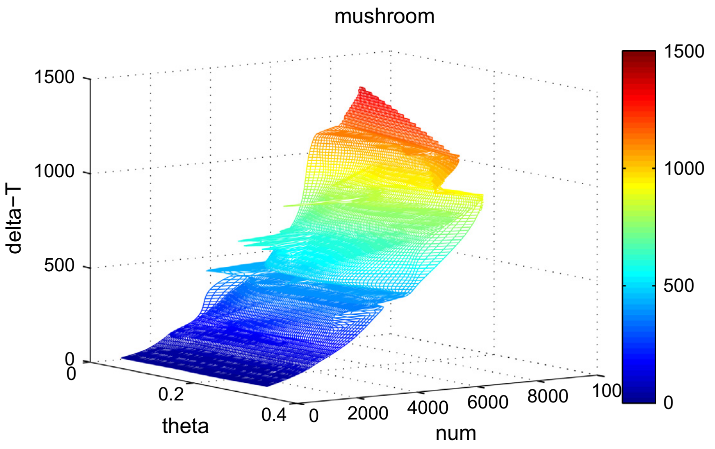


Fig. 8. Comparable experimental results on mushroom data set, $\theta \in (0, 0.3]$. The unit of vertical coordinate is second (S).

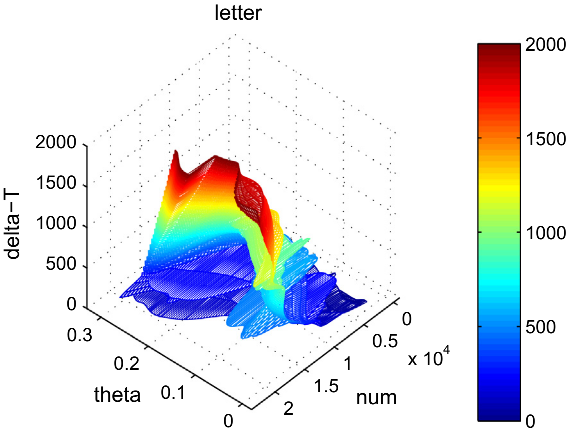


Fig. 9. Comparable experimental results on letter data set, $\theta \in (0, 0.3]$. The unit of vertical coordinate is second (S).

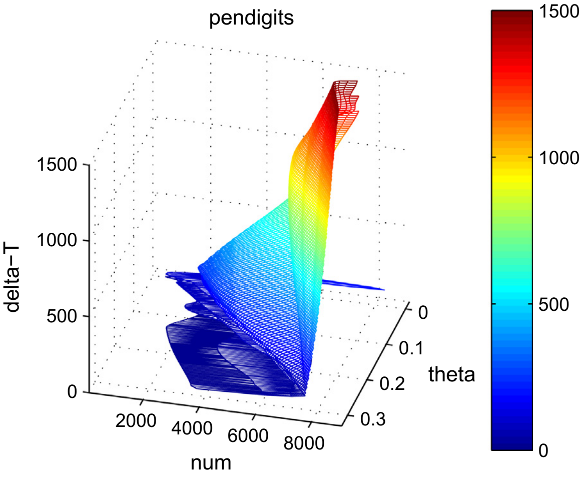


Fig. 10. Comparable experimental results on pendigits data set, $\theta \in (0, 0.3]$. The unit of vertical coordinate is second (S).

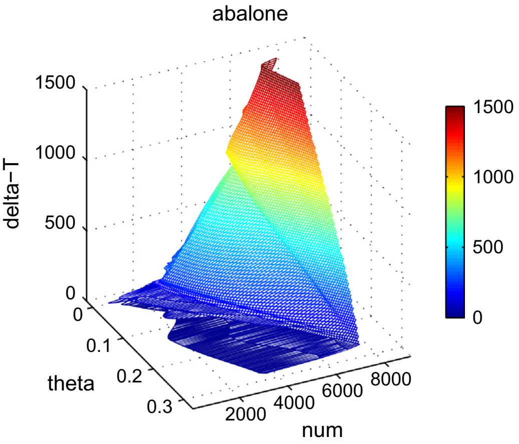


Fig. 11. Comparable experimental results on abalone data set, $\theta \in (0, 0.3]$. The unit of vertical coordinate is second (S).

5. Conclusion

The neighborhood rough set model and its reduct have been proved useful in many machine learning, data mining applications. As the big data processing has became the popular trend in machine learning and data mining, an efficient attribute reduct algorithm for neighborhood rough set model is desired. In this paper, we propose a quick attribute reduct algorithm for neighborhood rough set model. We present a new approach to divide all the records into a sequent bucket set, and we also prove that each record's θ-neighborhood elements can only exist in its own bucket and its adjacent buckets. Based on the division of buckets, we then present a new fast algorithm to calculate the positive region of neighborhood rough set model, which can achieve a complexity of $O(m|U|)$. With that new fast positive region computation algorithm, we present a quick reduct algorithm for neighborhood rough set model, and our algorithm can achieve a computation complexity of $O(m^2|U|)$ at the worst condition. The comparable experimental results show the better efficiency of our approach.

As the reduct algorithms on extended rough set models are much more complex than classical rough set models due to the promotion on the computation of new relation models, many of the traditional approaches on classical rough set reduct cannot be applied into the new models directly. In the future work, we will try to implement our bucket based quick reduct algorithm into other rough set relation models, and try to present a unified attribute reduct solution for multiple-relation models.

Acknowledgements

This research is based upon work supported in part by National Natural Science Foundation of China (61173123, 61370173) and Natural Science Foundation of Zhejiang Province (LR13F030003). This paper is also supported by the China Scholarship Council.

References

[1] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inform. Sci. 177 (1) (2007) 3–27.
[2] Z. Pawlak, J.W. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets, Commun. ACM 38 (11) (1995) 88–95.
[3] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, IEEE Trans. Knowl. Data Eng. 16 (12) (2004) 1457–1471.
[4] Q. Hu, D. Yu, Z. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, Pattern Recogn. Lett. 27 (5) (2006) 414–423.
[5] Q. Hu, J. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowl.-Based Syst. 21 (4) (2008) 294–304.
[6] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1–2) (2003) 155–176.
[7] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29–July 2, 2000, pp. 359–366.
[8] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.
[9] Z. Zhu, Y.-S. Ong, M. Dash, Wrapper–Filter feature selection algorithm using a memetic framework, IEEE Trans. Syst., Man, Cybern., Part B 37 (1) (2007) 70–76.
[10] N. Zhong, J. Dong, S. Ohsuga, Using rough sets with heuristics for feature selection, J. Intell. Inform. Syst. 16 (3) (2001) 199–214.
[11] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recogn. Lett. 24 (6) (2003) 833–849.
[12] S. Tsumoto, A new framework for incremental rule induction based on rough sets, in: GrC, 2011, pp. 681–686.
[13] Y.-P.O. Yang, H.-M. Shieh, G.-H. Tzeng, L. Yen, C.-C. Chan, Combined rough sets with flow graph and formal concept analysis for business aviation decision-making, J. Intell. Inform. Syst. 36 (3) (2011) 347–366.
[14] Y. Liu, C. Xu, Q. Zhang, Y. Pan, Rough rule extracting from various conditions: incremental and approximate approaches for inconsistent data, Fundam. Inform. 84 (3–4) (2008) 403–427.
[15] D. Slezak, P. Synak, J. Wroblewski, G. Toppin, Infobright Analytic database engine using rough sets and granular computing, in: GrC, 2010, pp. 432–437.
[16] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inform. Sci. 178 (18) (2008) 3577–3594.
[17] Y.Y. Yao, A comparative study of fuzzy sets and rough sets, Inform. Sci. 109 (1998) 21–47.
[18] Q. Hu, D. Yu, Z. Xie, J. Liu, Fuzzy probabilistic approximation spaces and their information measures, IEEE Trans. Fuzzy Syst. 14 (2) (2006) 191–201.
[19] D.S. Yeung, D. Chen, E.C.C. Tsang, J.W.T. Lee, X. Wang, On the generalization of fuzzy rough sets, IEEE Trans. Fuzzy Syst. 13 (3) (2005) 343–361.
[20] J.F. Peters, P. Wasilewski, Tolerance spaces: origins, theoretical aspects and applications, Inform. Sci. 195 (2012) 211–225.
[21] R. Slowinski, D. Vanderpooten, A generalized definition of rough approximations based on similarity, IEEE Trans. Knowl. Data Eng. 12 (2) (2000) 331–336.
[22] S. Greco, B. Matarazzo, R. Slowinski, Rough approximation by dominance relations, Int. J. Intell. Syst. 17 (2) (2002) 153–171.
[23] Y. Yao, B. Yao, Covering based rough set approximations, Inform. Sci. 200 (2012) 91–107.
[24] W. Zhu, F.-Y. Wang, The fourth type of covering-based rough sets, Inform. Sci. 201 (2012) 80–92.
[25] W. Zhu, Relationship among basic concepts in covering-based rough sets, Inform. Sci. 179 (14) (2009) 2478–2486.
[26] W. Zhu, F.-Y. Wang, On three types of covering-based rough sets, IEEE Trans. Knowl. Data Eng. 19 (8) (2007) 1131–1144.
[27] T.Y. Lin, Granular Computing, Chap. Granulation and Nearest Neighborhoods: Rough Set Approach, Physica-Verlag GmbH, Heidelberg, Germany, Germany, 2001, pp. 125–142. ISBN 3-7908-1387-7. <http://dl.acm.org/citation.cfm?id=761984.761990>.
[28] X. Yang, X. Li, T.Y. Lin, First GrC model – neighborhood systems the most general rough set models, in: GrC, 2009, pp. 691–695.
[29] Q. Hu, W. Pedrycz, D. Yu, J. Lang, Selecting discrete and continuous features based on neighborhood decision error minimization, IEEE Trans. Syst., Man, Cybern., Part B 40 (1) (2010) 137–150.
[30] Q. Hu, D. Yu, Z. Xie, Neighborhood classifiers, Expert Syst. Appl. 34 (2) (2008) 866–876.
[31] Y. Du, Q. Hu, P. Zhu, P. Ma, Rule learning for classification based on neighborhood covering reduction, Inform. Sci. 181 (24) (2011) 5457–5467.
[32] Z. Pawlak, Rough Sets, Theoretical Aspects of Reasoning about Data, Springer, Formerly Kluwer Academic Publishers, Boston, Dordrecht, London, 1991.
[33] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, Intell. Decis. Support (1992) 331–362.
[34] W.-X. Zhang, J.-S. Mi, W.-Z. Wu, Approaches to knowledge reductions in inconsistent systems, Int. J. Intell. Syst. 18 (9) (2003) 989–1000.
[35] J. Qian, D.Q. Miao, Z.H. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, Int. J. Approx. Reason. 52 (2) (2011) 212–230. ISSN 0888-613X, doi:10.1016/j.ijar.2010.07.011. <http://dx.doi.org/10.1016/j.ijar.2010.07.011>.
[36] Y. Liu, R. Xiong, J. Chu, Quick attribute reduction algorithm with hash, Chinese J. Comput. 32 (8) (2009) 1493–1499.
[37] Y. Liu, Y. Jiang, J. Yang, Feature reduction with inconsistency, Int. J. Cogn. Inform. Nat. Intell. 4 (2) (2010) 77–87, http://dx.doi.org/10.4018/jcini.2010040106. ISSN 1557-395. <http://dx.doi.org/10.4018/jcini.2010040106>.

[38] A. Chouchoulas, Q. Shen, Rough set-aided keyword reduction for text categorization, Appl. Artif. Intell. 15 (9) (2001) 843–873.

[39] G. Xin, G. Qiang, Z. Jing, Z. Zheng-chao, An attribute reduction algorithm based on rough set, information entropy and ant colony optimization, in: 2010 IEEE 10th International Conference on Signal Processing (ICSP), 2010, pp. 1313–1317.

[40] G. Wang, H. Yu, D. Yang, Decision table reduction based on conditional information entropy, Chinese J. Comput. 25 (7) (2002) 759–766.

[41] Y.-J. Lv, N.-X. Liu, Application of quantum genetic algorithm on finding minimal reduct, in: 2012 IEEE International Conference on Granular Computing vol. 0, 2007, p. 728. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/GrC.2007.87>.

[42] N.S. Hoa, Some efficient algorithms for rough set methods, in: Proceedings IPMU'96 Granada, Spain, 1996, pp. 1541–1457.

[43] S. Liu, Q. Chen, Z. Shi, A new method for fast computing positive region, Chinese J. Comput. Res. Dev. 40 (5) (2003) 637–642.

[44] G. Wang, F. Hu, Quick knowledge reduction based on divide and conquer method in huge data sets, in: Proceedings of the 2nd International Conference on Pattern Recognition and Machine Intelligence, PReMI'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 312–315. ISBN 3-540-77045-3, 978-3-540-77045-9. <http://dl.acm.org/citation.cfm?id=1781034.1781077>.

[45] R.B. Bhatt, M. Gopal, On fuzzy-rough sets approach to feature selection, Pattern Recogn. Lett. 26 (7) (2005) 965–975.