Applied Soft Computing Journal 98 (2021) 106740

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Accelerator for crosswise computing reduct

Zehua Jiang [a], Keyu Liu [a], Jingjing Song [a,b], Xibei Yang [a,*], Jinhai Li [c], Yuhua Qian [d]

[a] School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212003, PR China
[b] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian, 363000, PR China
[c] School of Science, Kunming University of Science and Technology, Kunming, Yunnan, 650500, PR China
[d] Institute of Big Data Science and Industry, Shanxi University, Taiyuan, Shanxi, 030006, PR China

ARTICLE INFO

Article history:
Received 28 October 2019
Received in revised form 6 August 2020
Accepted 16 September 2020
Available online 21 September 2020

Keywords:
Accelerator
Attribute reduction
Cross computation
Rough set

ABSTRACT

Attribute reduction, as a technique for selecting qualified attributes which can satisfy the intended constraint related to considered measure, has been widely explored. Notably, one and only one reduct is derived through using one searching strategy in most cases. Nevertheless, only one reduct may be not enough for us to evaluate its effectiveness. To fill such gap, an approach of crosswise computing reduct is proposed for obtaining multiple reducts. The computation of reduct is realized through partitioning the whole data into several groups, and crosswise selecting some groups to form different subsets of data, then computing reducts over these different subsets of data. Moreover, to speed up the process of crosswise computing reduct, an acceleration strategy is designed. The main thinking of our acceleration strategy is to compute the reduct over different subsets of data on the basis of reduct over the whole data. The experimental results over 16 data sets show the following superiorities of our strategy: (1) our approach can decrease the elapsed time of crosswise computing reducts significantly; (2) our approach can not only provide reduct with higher stability, but also maintain the classification performance; (3) the attributes in reduct can provide more stable classification results.

1. Introduction

Attribute reduction [1–6] has gained a substantial attention with respect to different applications [7–17]. This can be attributed to the fact that attribute reduction helps us select attributes with various semantic explanations which are resulted from different constraints in the corresponding definitions [18, 19]. Generally, the aim of attribute reduction is to obtain a qualified attribute subset from the raw condition attributes which satisfies appointed constraint, and such subset is referred to as the reduct. It should be emphasized that the attributes in reduct should make contribution to preserving/increasing/decreasing the value of measure related to the appointed constraint, e.g., approximation quality (dependency degree) [20], classification accuracy [21], conditional entropy [22–24] and so on.

Through reviewing the previous researches carefully, lots of searching strategies [25–34] have been put forward to obtain reducts in terms of different constraints. Note that no matter which strategy is adopted, some essential issues related to attribute reduction should be seriously considered.

1. How to evaluate the the performance related to stability of reduct? For example, if one and only one reduct is derived over the whole data, then it is difficult for us to determine whether the reduct is stable or not. This is mainly because the reduct is obtained without considering the variation of data. From this point of view, computing multiple reducts is required.
2. How to evaluate the stability of searching strategy from the viewpoint of reduct? For instance, if one and only one reduct is derived over the whole data through using one searching strategy, then it is difficult for us to evaluate the performance of the searching strategy. This is mainly because such reduct cannot reveal the variation of reducts. From this point of view, computing multiple reducts is necessary.

To derive multiple reducts, a direct method is to iterate the process of computing reduct over different sets of data. According to this thinking, the method of crosswise computing reduct will be proposed. The main mechanism of crosswise computing reducts is that deriving multiple reducts through using multiple different subsets of data. Firstly, partition the whole data into K different groups with the same size; secondly, select K-1 groups to form the subset of data; finally, compute reducts over the subsets of data. Immediately, repeat the process and then K reducts will be obtained. The detailed process is shown in the following

* Corresponding author.
E-mail addresses: jzhjiangzehua@163.com (Z. Jiang), liukeyu@stu.just.edu.cn (K. Liu), songjingjing108@163.com (J. Song), jsjxy_yxb@just.edu.cn (X. Yang), jhlixjtu@163.com (J. Li), jinchengqyh@126.com (Y. Qian).

https://doi.org/10.1016/j.asoc.2020.106740
1568-4946/

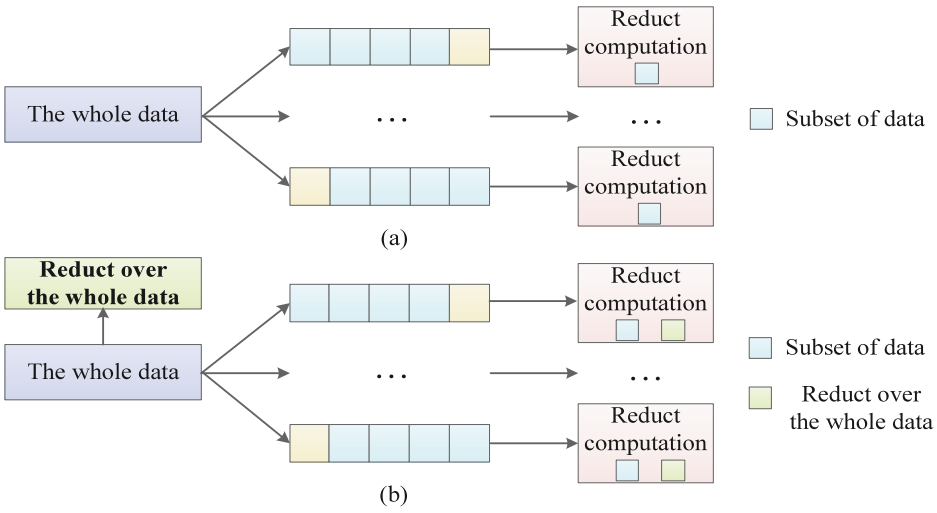

Fig. 1. The frameworks of traditional strategy and acceleration strategy.

Fig. 1(a). In Fig. 1(a), the whole data U is divided into 5 groups with the same size $\{U_1, U_2, \ldots, U_5\}$. For each round, 4 groups compose the subset of data for computing reduct. Specifically, in the first round, $\{U_1, U_2, U_3, U_4\}$ is used for computing reduct; in the second round, $\{U_1, U_2, U_3, U_5\}$ is used for computing reduct; ...; in the last round, $\{U_2, U_3, U_4, U_5\}$ is used for computing reduct. Eventually, multiple reducts over different subsets of data will be derived. It follows that the performances of reducts can be evaluated, and the variation of these reducts can be analyzed, then the performance of searching strategy can also be evaluated through considering the variation of reducts.

Though multiple reducts can be obtained through crosswise computing reduct, it is time-consuming, especially the number of K is great or the volume of data is large. Take the case that K is great as an example, given a data set with s condition attributes, in the worst case, to find satisfactory reduct, all of the condition attributes should be selected and added into reduct, then the number of times to evaluate the candidate attributes will be $\frac{s\cdot(s+1)}{2}$. Furthermore, if the data set is partitioned into K groups, and K reducts are required, then the number of times to repeat the process of deriving one reduct is K. In the worst case, to find K satisfactory reducts, all of the candidate attributes should be selected and added into reduct, then the number of times to evaluate attributes will be $K \cdot \frac{s\cdot(s+1)}{2}$. Obviously, time consumption of crosswise computing multiple reducts is higher. How to speed up the process of crosswise computing reduct deserves to be concerned.

To alleviate the problem, a novel acceleration approach will be proposed in this paper. The main mechanism of our acceleration strategy is that the reduct over the subset of data will be guided by reduct over the whole data. Firstly, obtain the reduct over the whole data; secondly, evaluate the condition attributes which are in the complement of such subset instead of raw condition attributes; finally, derive the reduct over the subset of data which satisfies the intended constraint. The detailed process is shown in the above Fig. 1(b). In Fig. 1(b), the reduct over the whole data is computed beforehand, then the whole data is partitioned into 5 groups with the same size. For each round, 4 groups compose the subset of data for computing reduct. Note that the reduct over subset of data is computed on the basis of the reduct over the whole data. Obviously, the space of searching attributes can be compressed. Therefore, it is highly possible that the time consumption of computing multiple reducts can be decreased.

Following the mentioned above, the main topics discussed in this paper can be concluded. In most cases, one and only one reduct is derived through using one searching strategy. Nevertheless, one reduct may involve some limitations: (1) one reduct may be powerless in evaluating both the performance related to stability of reduct and the stability of searching strategy; (2) the time efficiency of computing multiple reducts may not be satisfactory. To overcome these limitations, the method of crosswise computing reduct for deriving multiple reducts will be proposed. The cross computation of reduct is realized through computing reducts over different subsets of data. Additionally, an acceleration strategy is designed for accelerating the process of crosswise computing reduct. Our acceleration strategy is designed through considering that the computation of reducts over different subsets of data may be guided by the reduct over the whole data.

The rest of this paper is organized as follows. Basic notions of attribute reduction will be illustrated in Section 2. Crosswise computing reduct and our accelerator will be proposed in Section 3. In Section 4, comparative experimental results over 16 data sets will be shown, as well as the corresponding analyses. This paper will be ended with conclusions and future perspectives in Section 5.

2. Preliminaries

In the field of rough set theory [8,35–37], a decision system can be formally represented as $\mathrm{DS} = \langle U, AT, d\rangle$, in which U is a nonempty finite set of the samples such that $U = \{x_1, x_2, \ldots, x_n\}$, i.e., universe, AT is a set of the condition attributes, and d is the decision attribute. $\forall x_i \in U$, $a(x_i)$ denotes the value of x_i over condition attribute $a \in AT$, $d(x_i)$ denotes the value of x_i over decision attribute d.

Given a decision system, each attribute subset determines a binary indiscernibility relation. For each $A \subseteq AT$, the binary indiscernibility relation determined by A can be given by $\mathrm{IND}_A = \{(x_i, x_j) \in U \times U : a(x_i) = a(x_j), \forall a \in A\}$. Notably, in real-world applications, continuous data and even mixed data are ubiquitous, and indiscernibility relation may fail to distinguish samples with continuous values of attributes. In view of this, the neighborhood relation is employed to determine whether samples can be distinguished. For a given decision system, δ is a radius, then the neighborhood relation is expressed as $\mathrm{N}_A = \{(x_i, x_j) \in U \times U : \mathrm{dis}_A(x_i, x_j) \leq \delta\}$, where $\mathrm{dis}_A(x_i, x_j)$ indicates the distance between samples x_i and x_j over A.

As one of the crucial topics in rough set theory [5,21,38–42], attribute reduction has been paid much attention to. With respect

to different requirements, various definitions of attribute reduction [43,44] have been proposed. To extract the commonness from those definitions, Yao et al. [45] have proposed the following general form of attribute reduction.

Definition 1. Given a decision system DS, $\forall A \subseteq AT$, ρ-constraint is a given constraint based on the measure ρ, then A is referred to as a ρ-reduct if and only if:

1. A satisfies ρ-constraint;
2. $\forall A' \subset A$, A' does not satisfy ρ-constraint.

Generally, different measures may lead to different forms of ρ-constraint. For example, if the higher value of measure is expected (e.g., approximation quality and classification accuracy), then ρ-constraint can be set as $\rho_A \geq \rho_{AT}$ (ρ_A implies the evaluation of attribute subset A through using measure ρ); if the lower value of measure is expected (e.g., conditional entropy and decision error rate), then ρ-constraint can be set as $\rho_A \leq \rho_{AT}$.

Obviously, the ρ-reduct can be regarded as the smallest attribute subset which satisfies the given ρ-constraint: the first condition guarantees the satisfaction of the ρ-constraint; the second condition guarantees that there is no redundant attribute in the reduct.

To obtain the reduct shown in Definition 1, searching strategy is an important factor. In recent years, many researchers have proposed numerous searching strategies [26,30,34,46,47] with regard to different purposes. It should be noticed that the greedy searching strategy [20] has been widely accepted because of its lower time complexity. In such searching strategy, the significance function plays a key role in selecting attributes. Two most commonly used forms which are closely related to ρ-constraint mentioned above are shown as follows.

Definition 2. Given a decision system DS, $\forall A \subseteq AT$, $\forall a \in AT - A$, two forms of the significance of the candidate attribute a is defined as follows:

$$Sig(a, A) = \rho_{A\cup\{a\}} - \rho_A; \tag{1}$$

$$Sig(a, A) = \rho_A - \rho_{A\cup\{a\}}. \tag{2}$$

If the higher value of measure is required, then Eq. (1) is suitable for computing the significance of candidate attribute a; if the lower value of measure is required, then Eq. (2) is suitable for computing the significance of candidate attribute a. Immediately, the following greedy searching based algorithm to obtain reduct is shown in the following.

Algorithm 1. Greedy Searching Strategy for Computing Reduct.

Input: a decision system DS and a given measure ρ.
Output: a ρ-reduct A.
1. $A = \emptyset$;
2. Do
(1) $\forall a \in AT - A$, compute $Sig(a, A)$;
(2) select the most significant attribute b where $b \in AT - A$;
(3) $A = A \cup \{b\}$;
Until ρ-constraint is satisfied;
3. Do
For each $c \in A$
(1) compute $\rho_{A-\{c\}}$;
(2) **If** ρ-constraint is satisfied
$A = A - \{c\}$;
End
Until A does not change or $|A| = 1$;
4. Return A.

For Algorithm 1, in the worst case, if no attribute is redundant, then the number of times to evaluate attributes is $\frac{|AT|\cdot(|AT|+1)}{2}$, in which $|AT|$ denotes the cardinality of AT. Therefore, the time complexity of Algorithm 1 is $O(|U|^2 \cdot |AT|^2)$.

3. Acceleration strategy for computing multiple reducts

3.1. Cross computation of reduct

Algorithm 1 shows us the computation of reduct over the whole data U in the decision system DS. Obviously, through using Algorithm 1, one and only one reduct is derived. Nevertheless, as discussed in Section 1, one reduct may involve some limitations: (1) the performances related to stability of reduct cannot be revealed through using one and only one reduct; (2) the stability of searching strategy cannot be evaluated through using only one reduct. From this point of view, a novel method for deriving multiple reducts is necessary. In the following, the method of crosswise computing reduct will be proposed. The detailed process of crosswise computing reducts are:

1. partition U into K different groups with the same size;
2. select K-1 groups to compose the subset of data;
3. compute the reduct over such subset of data by using Algorithm 1;
4. repeat the above steps 2 and 3 over K different groups.

Following the above steps, the algorithm of crosswise computing reduct is shown in Algorithm 2.

Algorithm 2. Cross Computation of Reduct.

Input: a decision system DS, a given measure ρ and the number of folds K.
Output: a set of ρ-reducts $\mathbb{A}$.
1. Partition universe U into K disjoint groups $U_1, U_2, \ldots, U_K$ with the same size;
2. For $g = 1 : K$
(1) $U' = U - U_g$;
(2) $DS' =< U', AT, d >$;
(3) compute the reduct A_g over DS′ by using Algorithm 1;
(4) add A_g into $\mathbb{A}$;
End
3. Return a set of reducts $\mathbb{A} = \{A_1, A_2, \ldots, A_K\}$.

For Algorithm 2, if $g = 1$ (i.e., $U - U_1$ is used to compute reduct), then the time complexity of computing A_1 is $O(|U - U_1|^2 \cdot |AT|^2)$; if $g = 2$ (i.e., $U - U_2$ is used to compute reduct), then the time complexity of computing A_2 is $O(|U - U_2|^2 \cdot |AT|^2)$; …; if $g = K$ (i.e., $U - U_K$ is used to compute reduct), then the time complexity of computing U_K is $O(|U - U_K|^2 \cdot |AT|^2)$. Therefore, the time complexity of Algorithm 2 is $O\big((|U - U_1|^2 + |U - U_2|^2 + \cdots + |U - U_K|^2) \cdot |AT|^2\big)$, i.e., $O\big((K-1) \cdot |U|^2 \cdot |AT|^2\big)$.

3.2. Acceleration strategy for crosswise computing reduct

Algorithm 2 shows us the detailed process of crosswise computing reduct. In Algorithm 2, for each iteration, the searching of attributes begins with an empty set and the candidate attributes are evaluated based on the significance shown in Def. 2. Obviously, the process of computing one reduct over one subset of data is the same with what has been presented in Algorithm 1. In other words, the computation of reducts over different subsets of data is actually realized based on the same searching process.

Accordingly, some crucial issues related to crosswise compute reduct should be addressed. Firstly, the time consumption of computing reduct through using Algorithm 2 is high. This is

mainly because the searching strategy which is used for computing different reducts over different subsets of data is the same. Consequently, if the number of folds is great, then the time complexity of Algorithm 2 will be high, it follows that huge elapsed time may be required, especially the volume of data is large. Secondly, it is frequent that the reducts over different subsets of data may be quite different. Such case will result in some undesirable results, e.g., the classification results over testing samples may be quite different, and then it is difficult to determine which labels can be regarded as the final outputs of the testing samples.

To fill these gaps, a novel searching strategy for speeding up the process of crosswise computing reduct will be proposed. Our searching strategy is designed based on the assumption that the reduct over the whole data may provide guidance for the computation of reduct over various subsets of data. Specifically, compute the reduct over the whole data, and then compute the reduct over different subsets of data on the basis of reduct over the whole data. Obviously, when computing reducts over different subsets of data, the searching space of attributes is the complement of the reduct over the whole data (i.e., A_{raw}). That is, to obtain the reduct over various subsets of data, in each iteration, we begin the addition of attributes with the searching of $AT - A_{raw}$ instead of raw attributes AT. The detailed process of our acceleration strategy are shown as follows:

1. compute the reduct A_{raw} over U by using Algorithm 1;
2. partition U into K different groups with the same size;
3. select K-1 groups to compose the subset of data;
4. for one subset of data, if A_{raw} satisfies the conditions shown in Def. 1, then A_{raw} can also be regarded as the reduct over such subset of data; otherwise, select some important attributes from $AT - A_{raw}$, and add them into A_{raw} until the conditions over such subset of data are satisfied;
5. repeat steps 3 and 4 over K different groups.

Following the main steps shown above, it is clear that our acceleration strategy is quite different from Algorithm 2 in computing multiple reducts. In the framework of our accelerator, for each iteration of computing reduct over one subset of data, only the attributes in $AT - A_{raw}$ should be checked. In view of this, the searching space of condition attributes can be reduced, it follows that the elapsed time of crosswise computing reduct may be decreased.

Additionally, there may be higher similarity among reducts which are derived over different subsets of data. This is mainly because the searchings of reducts over different subsets of data are guided by the reduct over the whole data. It follows that more same attributes may exist among different reducts. Consequently, comparing with Algorithm 2, our acceleration strategy may generate reducts with higher similarity. The detailed process of our accelerator is shown in the following Algorithm 3.

Algorithm 3. Acceleration Strategy for Crosswise Computing Reduct.

Input: a decision system DS, a given measure ρ and the number of fold K.

Output: a set of ρ-reducts $\mathbb{A}$.

1. Compute A_{raw} over U by using Algorithm 1;

2. Partition universe U into K disjoint groups $U_1, U_2, \ldots, U_K$ with the same size;

3. For $g = 1 : K$

(1) $A_g = A_{raw}$;

(2) **If** A_g is the ρ-reduct over $U - U_g$;

go to (5);

Else

go to (3);

End

(3) **Do**

(i) $\forall a \in AT - A_g$, compute $\mathrm{Sig}(a, A_g)$ over $U - U_g$;

(ii) select the most significant attribute b where $b \in AT - A_g$;

(iii) $A_g = A_g \cup \{b\}$;

Until ρ-constraint is satisfied over $U - U_g$;

(4) **Do**

For each $c \in A_g$

(1) compute $\rho_{A_g - \{c\}}$;

(2) **If** ρ-constraint is satisfied

$A_g = A_g - \{c\}$;

End

Until A_g does not change or $|A_g| = 1$;

(5) add A_g into $\mathbb{A}$;

End

4. Return a set of reducts $\mathbb{A} = \{A_1, A_2, \ldots, A_K\}$.

The time complexity of Algorithm 3 is $O\left(|U|^2 \cdot |AT|^2 + (|U - U_1|^2 + |U - U_2|^2 + \cdots + |U - U_K|^2) \cdot |AT - A_{raw}|^2\right)$, i.e., $O\big(|U|^2 \cdot |AT|^2 + (K-1) \cdot |U|^2 \cdot |AT - A_{raw}|^2\big)$. The reasons are:

1. the time complexity of computing A_{raw} is $O(|U|^2 \cdot |AT|^2)$, because such process is the same to Algorithm 1;
2. if $g = 1$ (i.e., $U - U_1$ is used to compute reduct), in the worst case, A_{raw} does not satisfy the requirements over $U - U_1$, and all of the candidate attributes in $AT - A_{raw}$ should be evaluated and added into A_{raw}, then the time complexity of computing A_1 is $O(|U - U_1|^2 \cdot |AT - A_{raw}|^2)$; similarly, the time complexity of computing A_2 is $O(|U - U_2|^2 \cdot |AT - A_{raw}|^2)$; $\cdots$; the time complexity of computing A_K is $O(|U - U_K|^2 \cdot |AT - A_{raw}|^2)$.

Notably, for different subsets of data, if the reduct over universe U satisfies the conditions of attribute reduction over different subsets of data, then the attributes in $AT - A_{raw}$ will not be evaluated. In such case, the time complexity of Algorithm 3 is $O(|U|^2 \cdot |AT|^2)$. Obviously, the time complexity is lower than that of Algorithm 2.

To understand the process of Algorithm 3 clearly, an example will be shown as follows.

Table 1
An artificial data set.

	a_1	a_2	a_3	a_4	a_5	d
x_1	0.3188	0.5785	0.3955	0.6797	0.3342	1
x_2	0.4242	0.2373	0.3674	0.1366	0.6987	1
x_3	0.5079	0.4588	0.9880	0.7212	0.1978	1
x_4	0.0855	0.963	0.0377	0.1068	0.0305	1
x_5	0.2625	0.5468	0.8852	0.6538	0.7441	1
x_6	0.8010	0.5211	0.9133	0.4942	0.5000	1
x_7	0.0292	0.2316	0.7962	0.7791	0.4799	2
x_8	0.9289	0.4889	0.0987	0.7150	0.9047	2
x_9	0.7303	0.6241	0.2619	0.9037	0.6099	2
x_{10}	0.4886	0.6791	0.3354	0.8909	0.6177	2

Example 1. Given a decision system DS as shown in Table 1, in which $U = \{x_1, x_2, \ldots, x_{10}\}$, $AT = \{a_1, a_2, \ldots, a_5\}$, d is the decision attribute.

Supposing that 5 reducts with respect to approximation quality over radius 0.2 are required. The detailed process of computing reduct through using Algorithm 3 is shown in the following.

1. Obtain the reduct over U, $A_{raw} = \{a_4\}$.

Table 2
Data sets description.

ID	Data sets	Samples	Attributes	Decision classes	Sources
1	Breast Tissue	106	9	6	[48]
2	Brain Tumor	90	5920	5	[49]
3	Cardiotocography	2126	21	10	[48]
4	Crowdsourced Mapping	10845	28	6	[48]
5	GCM	198	11370	14	[50]
6	Gesture Phase Segmentation	9901	18	2	[48]
7	Ionosphere	351	34	2	[48]
8	Letter Recognition	20000	16	26	[48]
9	Madelon	2600	500	2	[48]
10	Musk (Version 1)	476	166	2	[48]
11	Optical Recognition of Handwritten Digits	5620	64	9	[48]
12	Statlog (Image Segmentation)	2310	18	7	[48]
13	Steel Plates Faults	1941	33	2	[48]
14	Urban Land Cover	675	148	9	[48]
15	Wall-Following Robot Navigation	5456	24	4	[48]
16	Wireless Indoor Localization	2000	7	4	[48]

2. Partition universe U into 5 disjoint groups: $U_1 = \{x_1, x_5\}$, $U_2 = \{x_6, x_{10}\}$, $U_3 = \{x_3, x_7\}$, $U_4 = \{x_2, x_8\}$, $U_5 = \{x_4, x_9\}$.
3. (1) For the subset of data $U - U_1$, $\rho_{AT} = 0.5000$, $\rho_{A_{raw}} = 0.5000$. Obviously, $\rho_{A_{raw}} = \rho_{AT}$, that is, A_{raw} satisfies constraint over $U - U_1$. Therefore, the reduct A_1 over $U - U_1$ is generated, $A_1 = A_{raw} = \{a_4\}$.
   (2) For the subset of data $U - U_2$, $\rho_{AT} = 0.3750$, $\rho_{A_{raw}} = 0.2500$. Obviously, $\rho_{A_{raw}} < \rho_{AT}$, that is, A_{raw} does not satisfy constraint over $U - U_2$. Let $A_2 = A_{raw}$, evaluate and select important attributes in $AT - A_2 = \{a_1, a_2, a_3, a_5\}$, until constraint over $U - U_2$ is satisfied. Therefore, the reduct A_2 over $U - U_2$ is generated, $A_2 = \{a_1, a_4\}$.
   (3) For the subset of data $U - U_3$, $\rho_{AT} = 0.6250$, $\rho_{A_{raw}} = 0.6250$. Obviously, $\rho_{A_{raw}} = \rho_{AT}$, that is, A_{raw} satisfies constraint over $U - U_3$. Therefore, the reduct A_3 over $U - U_3$ is generated, $A_3 = A_{raw} = \{a_4\}$.
   (4) For the subset of data $U - U_4$, $\rho_{AT} = 0.3750$, $\rho_{A_{raw}} = 0.2500$. Obviously, $\rho_{A_{raw}} < \rho_{AT}$, that is, A_{raw} does not satisfy constraint over $U - U_4$. Let $A_4 = A_{raw}$, evaluate and select important attributes in $AT - A_4 = \{a_1, a_2, a_3, a_5\}$, until constraint over $U - U_4$ is satisfied. Therefore, the reduct A_4 over $U - U_4$ is generated, $A_4 = \{a_1, a_4\}$.
   (5) For the subset of data $U - U_5$, $\rho_{AT} = 0.6250$, $\rho_{raw} = 0.6250$. Obviously, $\rho_{A_{raw}} = \rho_{AT}$, that is, A_{raw} satisfies constraint over $U - U_5$. Therefore, the reduct A_5 over $U - U_5$ is generated, $A_5 = A_{raw} = \{a_4\}$.
4. Derive the set of reducts $\{A_1, A_2, A_3, A_4, A_5\}$.

4. Experimental analyses

4.1. Data sets

To demonstrate the effectiveness of proposed acceleration strategy, i.e., Algorithm 3, 14 UCI and 2 gene data sets have been employed to conduct the experiments. The detailed description of these data sets is presented in Table 2. It should be emphasized that all of these data sets have been normalized by column in our experiments.

4.2. Measures used in experiments

In rough set theory, with respect to different requirements, various constraints have been constructed through using different measures with different explanations. In our experiments, four different measures have been employed. The details of these measures are displayed in the following.

- Approximation quality [20] is a measure to characterize the approximation ability of condition attributes in terms of decision attribute. Given a decision system DS and a radius δ, $\forall A \subseteq AT$, the approximation quality of A with respect to d is formulated as

$$\gamma_A = \frac{|\{x_i \in U : \mathrm{N}_A(x_i) \subseteq [x_i]_d\}|}{|U|}, \tag{3}$$

in which $\mathrm{N}_A(x_i) = \{x_j \in U : \mathrm{dis}_A(x_i, x_j) \leq \delta\}$, δ is a radius; $[x_i]_d$ indicates the set of samples which belong to the same decision class with x_i.

Approximation quality reflects the percentage of samples which certainly belong to one decision class. The higher the value of approximation quality, the higher the degree of such belongingness. From this point of view, the ρ-constraint can be set as $\gamma_A \geq \gamma_{AT}$.

- Neighborhood discrimination index [23] is a measure to characterize the discriminating ability of condition attributes in terms of decision attribute. Given a decision system DS and a radius δ, $\forall A \subseteq AT$, the neighborhood discrimination index of A with respect to d is formulated as

$$\mathrm{DI}_A = \log \frac{|\mathrm{N}_A|}{|\mathrm{N}_A \cap \mathrm{IND}_d|}, \tag{4}$$

in which $\mathrm{N}_A = \{(x_i, x_j) \in U \times U : dis_A(x_i, x_j) \leq \delta\}$, $\mathrm{IND}_d = \{(x_i, x_j) \in U \times U : d(x_i) = d(x_j)\}$.

The smaller the value of neighborhood discrimination index, the stronger the discriminating ability. From this point of view, the ρ-constraint can be set as $\mathrm{DI}_A \leq \mathrm{DI}_{AT}$.

- Conditional entropy [24] is another commonly used measure to characterize the discriminating ability of condition attributes in terms of decision attribute. Up to now, many different definitions of conditional entropy [22,39,51,52] have been proposed. A widely used form is shown as follows. Given a decision system DS and a radius δ, $\forall A \subseteq AT$, the conditional entropy of A with respect to d is formulated as

$$\mathrm{ENT}_A = -\frac{1}{|U|} \sum_{x_i \in U} |\mathrm{N}_A(x_i) \cap [x_i]_d| \log \frac{|\mathrm{N}_A(x_i) \cap [x_i]_d|}{|\mathrm{N}_A(x_i)|}. \tag{5}$$

The lower the value of conditional entropy, the stronger the discriminating ability. From this point of view, the ρ-constraint can be set as $\mathrm{ENT}_A \leq \mathrm{ENT}_{AT}$.

- Neighborhood decision error rate [21] is a measure to characterize the classification performance of condition attributes in terms of decision attribute. Given a decision

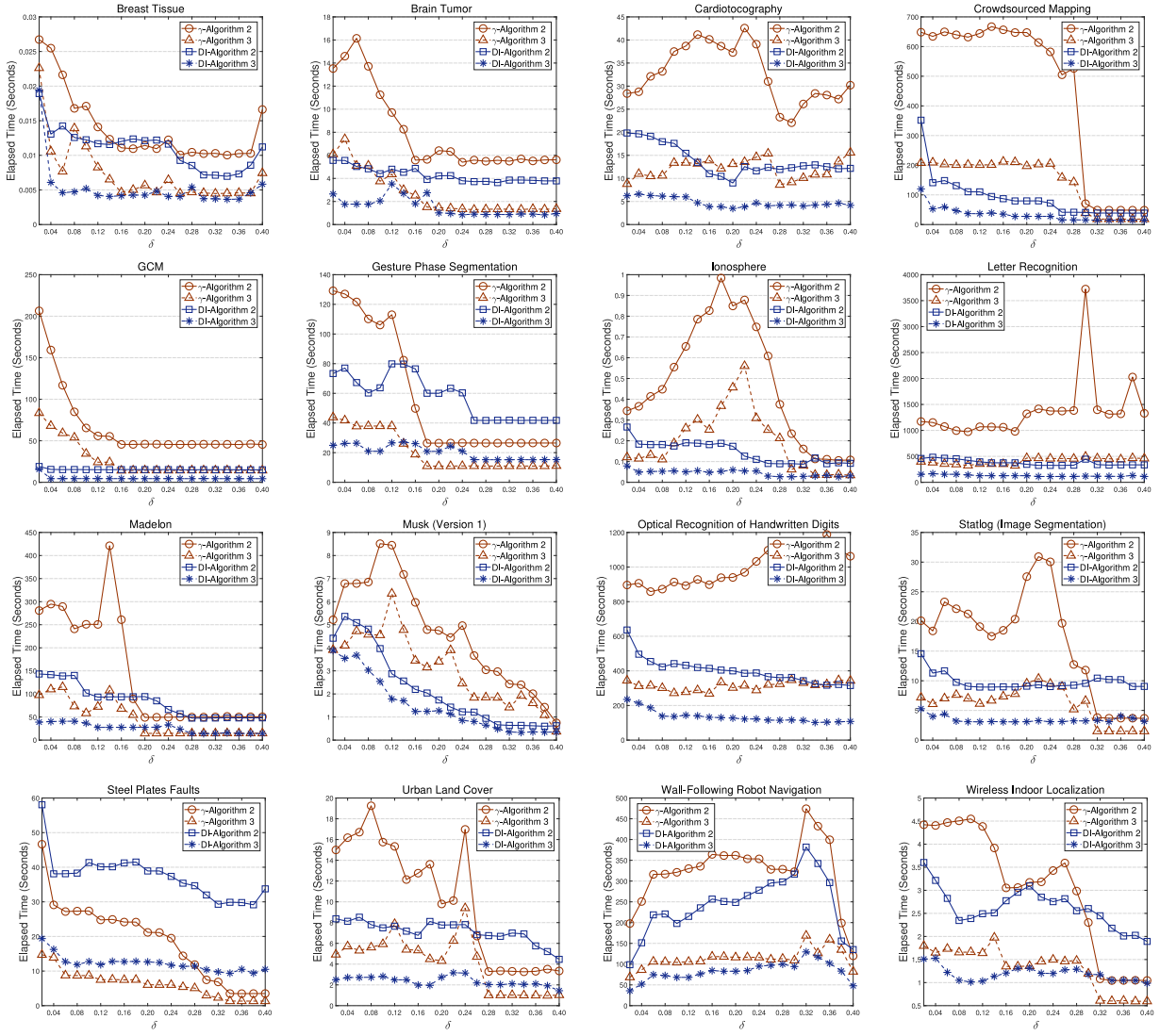

Fig. 2. Results of elapsed time w.r.t. approximation quality and neighborhood discrimination index.

system DS, $\forall A \subseteq AT$, the neighborhood decision error rate of A with respect to d is formulated as

$$\mathrm{NDER}_A = \frac{|\{x_i \in U : \mathrm{NPre}_A(x_i) \neq d(x_i)\}|}{|U|}, \tag{6}$$

in which $\mathrm{NPre}_A(x_i)$ indicates the predicted label of x_i by using neighborhood classifier [20].

The smaller the value of neighborhood decision error rate, the greater the classification performance. From this point of view, the ρ-constraint can be set as $\mathrm{NDER}_A \leq \mathrm{NDER}_{AT}$.

4.3. Experimental setup

In our experiments, neighborhood rough set [20,53] and four measures (i.e., approximation quality, neighborhood discrimination index, conditional entropy, neighborhood decision error rate) are employed. It should be noticed that the appointed radius is essential in the process of constructing neighborhood rough set. For example, if a very small value of radius used in neighborhood relation, then the neighborhoods of samples may only contain themselves, it follows that such neighborhood relation may be not suitable for distinguishing samples. To alleviate this problem, Hu et al. [20] have proposed the modified radius. Given a decision system and radius δ, $\forall x_i \in U$, $A \subseteq AT$, the modified radius with respect to x_i is computed as follows:

$$\delta'(x_i) = \min_{1\leq j\leq n, j\neq i}(\mathrm{dis}_A(x_i, x_j)) + \delta \times (\max_{1\leq j\leq n, j\neq i}(\mathrm{dis}_A(x_i, x_j)) - \min_{1\leq j\leq n, j\neq i}(\mathrm{dis}_A(x_i, x_j))), \tag{7}$$

in which $\min_{1\leq j\leq n, j\neq i}(\mathrm{dis}_A(x_i, x_j))$ implies the minimal value of distance between x_i and samples in $U - \{x_i\}$; $\max_{1\leq j\leq n, j\neq i}(\mathrm{dis}_A(x_i, x_j))$ implies the maximal value of distance between x_i and samples in $U - \{x_i\}$. Moreover, 20 different radii (i.e., 0.02, 0.04, . . ., 0.4) are used in our experiments.

Moreover, to estimate the performances related to reducts, each data set is partitioned into two parts randomly: 80% samples compose the training set for computing reduct, and the remaining 20% samples compose the testing set for evaluating the performances related to reduct. Furthermore, the training set is randomly partitioned into 5 groups with the same size. For each round, 4 groups compose the subset of training set, then 5 different subsets of training set can be derived for computing reducts. In the following, Algorithm 2 and Algorithm 3 are used to compute reducts, respectively. The time consumption of computing reducts is compared, then the detailed results and analyses will be shown in Section 4.4; the stabilities of reducts are

Table 3
Detailed explanations of different expressions.

Expression	Measure	Used algorithm
γ-Algorithm 2	Approximation quality	Algorithm 2
γ-Algorithm 3	Approximation quality	Algorithm 3
DI-Algorithm 2	Neighborhood discrimination index	Algorithm 2
DI-Algorithm 3	Neighborhood discrimination index	Algorithm 3
ENT-Algorithm 2	Conditional entropy	Algorithm 2
ENT-Algorithm 3	Conditional entropy	Algorithm 3
NDER-Algorithm 2	Neighborhood decision error rate	Algorithm 2
NDER-Algorithm 3	Neighborhood decision error rate	Algorithm 3

compared, then the detailed results and analyses will be shown in Section 4.5; classification performances related to reducts over CART and KNN classifiers are compared, then the detailed results and analyses will be shown in Section 4.6; the stabilities of classification results are compared, then the detailed results and analyses will be shown in Section 4.7.

To facilitate the understanding of the experimental results, some explanations of used expressions in Sections 4.4–4.7 are shown in Table 3.

In Table 3, the "Expression" denotes the results related to corresponding reduct, and such reduct is computed through using the "Used algorithm" based on the "Measure". For example, "γ-Algorithm 2" denotes the results related to reduct, and such reduct is computed through using Algorithm 2 based on the measure of approximation quality.

4.4. Comparisons of elapsed time

In this experiment, the time consumption of computing reducts over four measures through using Algorithm 2 and Algorithm 3 will be compared. When Algorithm 2 is used to compute reducts, the time consumption is the total elapsed time of computing reducts over 5 different subsets of training set; when Algorithm 3 is used to compute reducts, the time consumption is the sum of the elapsed time for computing reduct over the whole training set and that for computing reducts over 5 subsets of training set. The detailed results are shown in Figs. 2 and 3.

With a careful investigation of Fig. 2, it is not difficult to observe the following.

1. For both approximation quality and neighborhood discrimination index, the time consumption of computing reducts by using Algorithm 2 is greater than that of computing reducts by using Algorithm 3. Take the results on data set "Cardiotocography (ID: 3)" as an example, the whole elapsed time of "γ-Algorithm 2" based on 20 radii is 653.9318 s while that of "γ-Algorithm 3" based on 20 radii is 242.5206 s; the whole elapsed time of "DI-Algorithm 2" is 276.9164 s while that of "DI-Algorithm 3" is 95.7227 s. To sum up, compared with Algorithm 2, Algorithm 3 can decrease the time consumption of finding reducts significantly. This is mainly because in Algorithm 3, when computing reducts over 5 subsets of training set, the searching of attributes begins with the selected attributes in reduct which is derived through using the whole training set, it follows that the searching space can be compressed, and then Algorithm 3 will decrease the time consumption of crosswise computing reducts.
2. No matter which algorithm is used to compute reducts, the time consumption of computing reducts related to approximation quality is higher than that of computing reducts related to neighborhood discrimination ability. Take the results on data set "Cardiotocography (ID: 3)" as an example, the whole elapsed time of "γ-Algorithm 2" based on 20 radii is 653.9318 s while that of "DI-Algorithm 2" based on 20 radii is 276.9164 s; the whole elapsed time of "γ-Algorithm 3" based on 20 radii is 242.5206 s while the total elapsed time of "DI-Algorithm 3" based on 20 radii is 95.7227 s.

With a deep investigation of Fig. 3, it is not difficult to observe the following. For both conditional entropy and neighborhood decision error rate, the elapsed time of computing reducts by using Algorithm 2 is greater than that of computing reducts by using Algorithm 3. Take the results on data set "Cardiotocography (ID: 3)" as an example, the total elapsed time of "ENT-Algorithm 2" is 317.1203 s while the total elapsed time of "ENT-Algorithm 3" is 105.4412 s; the total elapsed time of "NDER-Algorithm 2" is 469.0137 s while the total elapsed time of "NDER-Algorithm 3" is 184.6828 s. To sum up, compared with Algorithm 2, Algorithm 3 can decrease the time consumption of finding reducts significantly. This is mainly because in Algorithm 3, when computing reducts over 5 subsets of training set, the searching of attributes begins with the selected attributes in reduct which is derived through using the whole training set, it follows that the searching space can be compressed, and then Algorithm 3 can decrease the time consumption of crosswise computing reducts.

4.5. Comparisons of stabilities of reducts

The stability of reduct [54] has been paid much attention to. In this experiment, the stabilities of reducts derived by using different algorithms will be compared. Given a decision system DS, assuming that universe U is partitioned into K disjoint groups

Table 4
Results of stabilities of different reducts.

ID	γ-Algorithm 2	γ-Algorithm 3	DI-Algorithm 2	DI-Algorithm 3	ENT-Algorithm 2	ENT-Algorithm 3	NDER-Algorithm 2	NDER-Algorithm 3
1	0.7422	**0.9486**	0.6296	**0.9012**	0.5677	**0.7975**	0.4288	**0.8165**
2	0.1746	**0.9465**	0.1508	**0.9283**	0.0317	**0.9638**	0.0375	**0.8417**
3	0.4893	**0.9420**	0.9006	**0.9948**	0.9414	**0.9979**	0.6425	**0.9257**
4	0.7566	**0.9703**	0.7842	**0.9896**	0.4342	**0.9979**	0.7332	**0.9742**
5	0.0953	**0.9040**	0.4600	**0.9900**	0.4150	**0.9900**	0.0413	**0.7967**
6	0.9464	**1.0000**	0.9383	**0.9884**	0.9655	**1.0000**	0.8413	**0.9802**
7	0.5291	**0.8624**	0.8366	**0.9798**	0.2882	**0.9273**	0.9683	**1.0000**
8	0.9014	**0.9687**	0.9507	**1.0000**	0.7729	**0.9745**	0.8209	**0.9768**
9	0.8203	**0.9819**	0.7419	**0.9419**	0.4504	**0.9889**	0.4770	**0.9418**
10	0.2895	**0.9063**	0.3918	**0.9168**	0.3315	**0.9739**	0.4368	**0.8293**
11	0.5897	**0.9697**	0.7637	**0.9670**	0.8349	**0.9728**	0.1944	**1.0000**
12	0.6943	**0.9790**	0.8322	**0.9929**	0.9605	**0.9880**	0.7004	**0.9505**
13	0.9150	**0.9624**	0.8708	**0.9752**	0.8258	**0.9665**	0.8807	**0.9829**
14	0.2675	**0.8696**	0.5658	**0.9553**	0.2387	**0.9475**	0.1937	**0.9495**
15	0.6660	**0.9414**	0.7071	**0.9821**	0.8504	**0.9767**	0.7644	**0.9454**
16	0.8161	**0.9526**	0.7504	**0.9543**	0.7022	**0.9753**	0.8062	**0.9391**

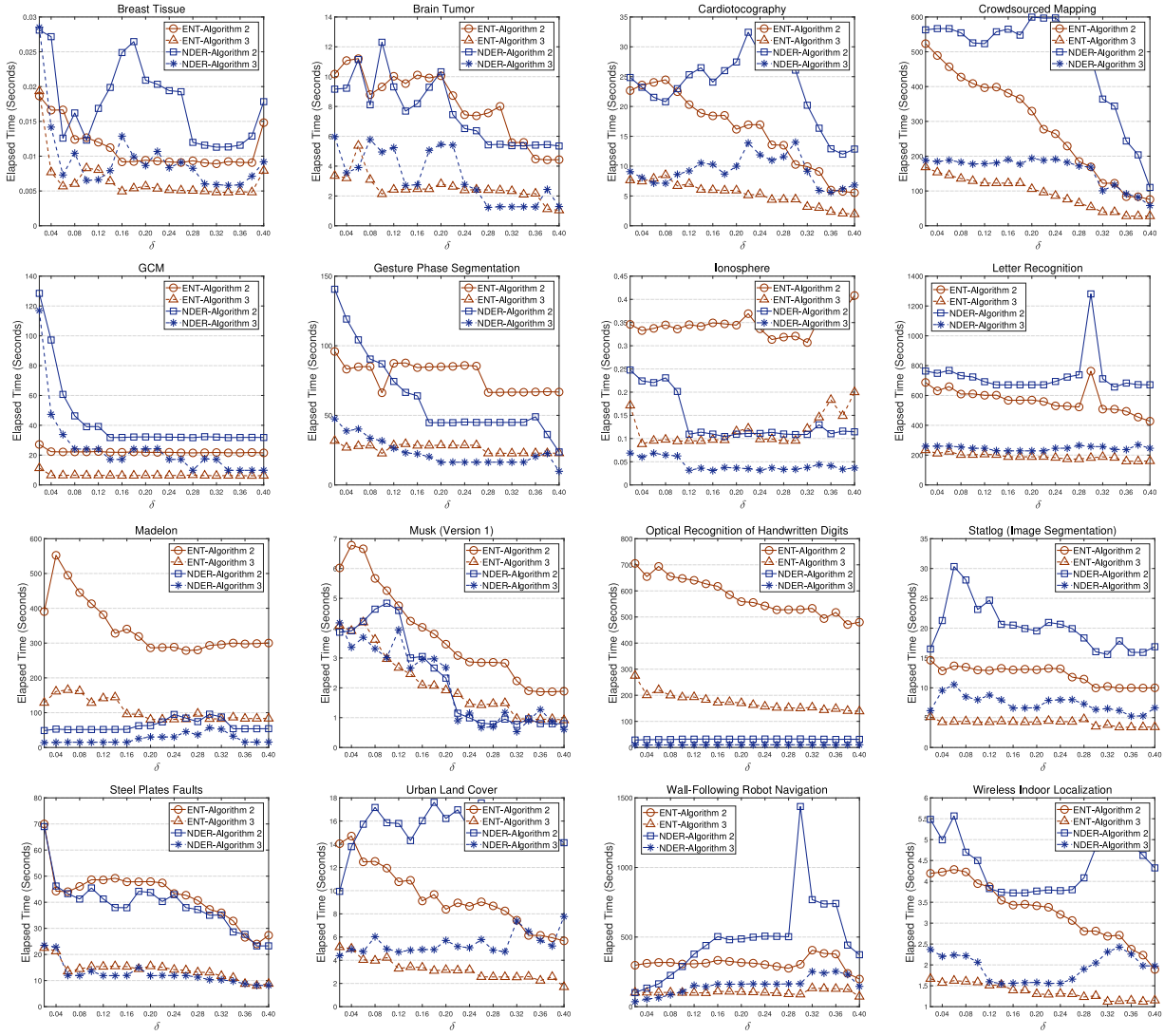

Fig. 3. Results of elapsed time w.r.t. conditional entropy and neighborhood decision error rate.

Table 5
p-values for stabilities of different reducts based on Algorithms 2 and 3.

ID	γ-Algorithm 2 & γ-Algorithm 3	DI-Algorithm 2 & DI-Algorithm 3	ENT-Algorithm 2 & ENT-Algorithm 3	NDER-Algorithm 2 & NDER-Algorithm 3
1	6.4485E−07	7.8213E−08	2.2901E−07	5.1888E−08
2	3.0253E−08	2.6924E−08	1.9255E−08	4.0500E−08
3	5.7186E−08	5.4889E−08	2.8009E−07	5.3578E−08
4	9.3626E−08	2.2680E−08	1.9544E−08	6.2771E−08
5	3.2448E−08	1.5926E−07	1.9881E−06	4.3589E−08
6	0.0045	0.0069	1.1052E−09	3.3124E−06
7	5.2960E−07	4.4242E−08	4.3982E−08	*0.0806*
8	9.8237E−08	6.3948E−09	3.9295E−08	2.9090E−08
9	0.0156	1.4057E−07	2.9550E−08	2.3050E−08
10	6.6063E−08	3.8587E−08	3.9954E−08	1.5480E−07
11	6.7478E−08	6.7288E−08	6.6909E−08	6.5039E−09
12	2.8636E−08	1.2086E−08	5.2024E−08	4.8004E−08
13	7.3740E−06	4.4443E−08	5.3422E−08	2.4153E−08
14	4.7371E−08	2.9227E−08	5.5799E−08	3.9893E−08
15	6.7193E−08	5.3578E−08	6.4490E−08	2.1186E−07
16	5.2094E−07	6.9673E−08	3.8998E−08	3.6586E−06

with the same size such that $U_1, U_2, \ldots, U_K$, the stability of reduct is formulated as

$$Sta_{reduct} = \frac{2}{K \cdot (K-1)} \sum_{g=1}^{K-1} \sum_{g'=g+1}^{K} \frac{|A_g \cap A_{g'}|}{|A_g \cup A_{g'}|}, \tag{8}$$

in which A_g is the reduct over $U - U_g$.

Obviously, $Sta_{reduct} \in [0,1]$ holds. Sta_{reduct} achieves the minimal value 0 if and only if $A_g \cap A_{g'} = \emptyset$, and it indicates that the obtained reduct is unstable completely; Sta_{reduct} achieves the maximal value 1 if and only if $A_g = A_{g'}$, and it indicates that

Table 6
Results of classification accuracies w.r.t. different reducts (CART classifier).

ID	γ- All	γ- Algorithm 2	γ- Algorithm 3	DI- All	DI- Algorithm 2	DI- Algorithm 3	ENT- All	ENT- Algorithm 2	ENT- Algorithm 3	NDER- All	NDER- Algorithm 2	NDER- Algorithm 3
1	0.4205 (9.25)	**0.5277 (11.61)**	0.5199 (11.35)	**0.5364 (11.80)**	0.5318 (11.70)	0.5295 (11.65)	0.2727 (6.00)	**0.3023 (6.65)**	0.2977 (6.55)	0.4977 (10.95)	**0.5386 (11.85)**	0.5153 (11.34)
2	0.5333 (9.60)	**0.5944 (10.70)**	0.5539 (9.97)	0.5139 (9.25)	**0.5494 (9.89)**	0.5150 (9.27)	0.5667 (10.20)	0.5722 (10.30)	**0.5922 (10.66)**	0.6139 (11.05)	**0.6256 (11.26)**	0.6106 (10.99)
3	**0.7683 (327.30)**	0.7346 (312.96)	0.7604 (323.91)	**0.7439 (316.90)**	0.7336 (312.50)	0.7385 (314.58)	**0.6362 (271.00)**	0.6253 (266.38)	0.6279 (267.50)	**0.7428 (316.45)**	0.7316 (311.65)	0.7389 (314.76)
4	**0.8650 (1876.15)**	0.8063 (1748.95)	0.8066 1749.61	**0.8657 (1877.65)**	0.7638 (1656.67)	0.7721 (1674.68)	**0.8614 (1868.40)**	0.8086 (1753.96)	0.8177 (1773.65)	0.7432 (1612.00)	0.8636 (1873.20)	**0.8677 (1881.95)**
5	0.1788 (7.15)	**0.2213 (8.85)**	0.2168 (8.67)	0.2075 (8.30)	**0.2185 (8.74)**	0.1925 (7.70)	0.1300 (5.20)	**0.1320 (5.28)**	0.1268 (5.07)	0.2638 (10.55)	0.2688 (10.75)	**0.2865 (11.46)**
6	**0.8742 (1731.70)**	0.5218 (1033.59)	0.5227 (1035.53)	**0.8533 (1690.45)**	0.8017 (1588.17)	0.8026 (1589.95)	**0.8763 (1736.00)**	0.7936 (1572.18)	0.7939 (1572.71)	**0.8556 (1694.95)**	0.7465 (1478.79)	0.7674 (1520.13)
7	**0.8697 (61.75)**	0.8694 (61.73)	0.8662 (61.50)	**0.8444 (59.95)**	0.8313 (59.02)	0.8380 (59.50)	0.8430 (59.85)	**0.8537 (60.61)**	0.8431 (59.86)	**0.8169 (58.00)**	0.8097 (57.49)	0.8120 (57.65)
8	**0.8577 (3430.80)**	0.8141 (3256.34)	0.8195 (3277.91)	**0.8570 (3427.90)**	0.7884 (3153.77)	0.7904 (3161.76)	**0.8566 (3426.30)**	0.7794 (3117.50)	0.7793 (3117.15)	0.6980 (2792.05)	0.8280 (3312.18)	**0.8316 (3326.41)**
9	**0.6677 (347.20)**	0.5268 (273.96)	0.5176 (269.16)	**0.7618 (396.15)**	0.5977 (310.79)	0.5974 (310.46)	**0.6879 (357.70)**	0.5225 (271.69)	0.5278 (274.48)	**0.5708 (296.80)**	0.5664 (294.55)	0.5685 (295.64)
10	**0.7495 (71.20)**	0.7171 (68.12)	0.7311 (69.45)	**0.7147 (67.90)**	0.7086 (67.32)	0.6978 (66.29)	**0.7379 (70.10)**	0.7197 (68.37)	0.7241 (68.79)	**0.7284 (69.20)**	0.6961 (66.13)	0.7140 (67.83)
11	0.7744 (870.40)	0.8721 (980.20)	**0.8721 (980.28)**	0.8294 (932.25)	0.8746 (983.02)	**0.8751 (983.60)**	0.7508 (843.95)	0.8670 (974.53)	**0.8700 (977.87)**	**0.1397 (157.00)**	0.1206 (135.50)	0.1370 (154.00)
12	0.8475 (391.55)	0.8424 (389.17)	**0.8494 (392.44)**	0.9009 (416.20)	0.9018 (416.62)	**0.9018 (416.64)**	**0.9153 (422.85)**	0.9010 (416.27)	0.9028 (417.09)	**0.9235 (426.65)**	0.9218 (425.85)	0.9234 (426.63)
13	0.8932 (346.55)	0.8971 (348.08)	**0.8972 (348.12)**	0.9726 (377.35)	0.9687 (375.85)	**0.9726 (377.37)**	0.9434 (366.05)	0.9460 (367.06)	**0.9475 (367.62)**	0.9724 (377.30)	**0.9748 (378.24)**	0.9744 (378.06)
14	0.5096 (68.80)	**0.5234 (70.66)**	0.5106 (68.93)	**0.7404 (99.95)**	0.7356 (99.30)	0.7314 (98.74)	0.4574 (61.75)	0.4520 (61.02)	**0.4634 (62.56)**	**0.7615 (102.80)**	0.7373 (99.54)	0.7555 (101.99)
15	0.9665 (1054.45)	0.9634 (1051.06)	**0.9667 (1054.64)**	**0.9575 (1044.65)**	0.9514 (1037.95)	0.9531 (1039.81)	**0.9633 (1050.95)**	0.9605 (1047.91)	0.9612 (1048.70)	**0.9737 (1062.35)**	0.9712 (1059.63)	0.9728 (1061.34)
16	**0.9165 (366.60)**	0.9153 (366.12)	0.9153 (366.11)	0.9600 (384.00)	**0.9614 (384.58)**	0.9608 (384.30)	**0.9394 (375.75)**	0.9355 (374.18)	0.9368 (374.73)	**0.9758 (390.30)**	0.9733 (389.33)	0.9750 (389.98)

Table 7
Results of classification accuracies w.r.t. different reducts (KNN classifier).

ID	γ-All	γ-Algorithm 2	γ-Algorithm 3	DI-All	DI-Algorithm 2	DI-Algorithm 3	ENT-All	ENT-Algorithm 2	ENT-Algorithm 3	NDER-All	NDER-Algorithm 2	NDER-Algorithm 3
1	**0.5045 (9.30)**	0.4927 (10.84)	0.4818 (10.60)	0.5045 (11.10)	0.5073 (11.16)	**0.5114 (11.25)**	0.3136 (6.90)	**0.3618 (7.96)**	0.3536 (7.78)	0.4932 (10.85)	**0.5218 (11.48)**	0.5123 (11.27)
2	0.5167 (9.30)	**0.5572 (10.03)**	0.5172 (9.31)	0.5806 (10.45)	**0.6044 (10.88)**	0.5906 (10.63)	0.5972 (10.75)	0.5750 (10.35)	**0.6106 (10.99)**	0.6167 (11.10)	0.6133 (11.04)	**0.6211 (11.18)**
3	**0.7642 (325.55)**	0.7278 (310.04)	0.7570 (322.50)	**0.7383 (314.50)**	0.7261 (309.31)	0.7337 (312.56)	**0.6469 (275.60)**	0.6288 (267.88)	0.6322 (269.30)	0.7227 (307.85)	0.7226 (307.84)	**0.7319 (311.77)**
4	**0.9524 (2065.80)**	0.8695 (1885.89)	0.8680 (1882.74)	**0.9500 (2060.50)**	0.7834 (1699.16)	0.7894 (1712.16)	**0.9514 (2063.65)**	0.8605 (1866.41)	0.8702 (1887.49)	0.7570 (1642.00)	0.9368 (2031.99)	**0.9424 (2044.06)**
5	0.1800 (7.20)	**0.2140 (8.56)**	0.1955 (7.82)	**0.1313 (5.25)**	0.1380 (5.52)	0.1308 (5.23)	0.0550 (2.20)	**0.0735 (2.94)**	0.0690 (2.76)	0.1625 (6.50)	**0.2023 (8.09)**	0.1993 (7.97)
6	**0.8995 (1782.00)**	0.5333 (1056.44)	0.5337 (1057.31)	**0.8922 (1767.45)**	0.7353 (1456.72)	0.7362 (1458.36)	**0.9001 (1783.00)**	0.7906 (1566.11)	0.7905 (1565.96)	**0.8840 (1751.20)**	0.7015 (1389.74)	0.7092 (1404.94)
7	**0.8620 (61.20)**	0.8349 (59.28)	0.8293 (58.88)	0.8331 (59.15)	0.8268 (58.70)	**0.8349 (59.28)**	0.8430 (59.85)	0.8400 (59.64)	**0.8444 (59.95)**	**0.8028 (57.00)**	0.7954 (56.47)	0.7979 (56.65)
8	**0.9537 (3814.60)**	0.9133 (3653.21)	0.9194 (3677.45)	**0.9549 (3819.50)**	0.8400 (3360.04)	0.8427 (3370.67)	**0.9534 (3813.40)**	0.8567 (3426.72)	0.8562 (3424.72)	0.7401 (2960.30)	0.9194 (3677.69)	**0.9262 (3704.75)**
9	**0.6164 (320.55)**	0.5237 (272.30)	0.5166 (268.63)	**0.7338 (381.60)**	0.6123 (318.39)	0.6113 (317.90)	**0.6358 (330.60)**	0.5177 (269.19)	0.5174 (269.07)	0.5463 (284.05)	0.5579 (290.12)	**0.5641 (293.31)**
10	**0.7605 (72.25)**	0.7217 (68.56)	0.7400 (70.30)	0.6921 (65.75)	**0.7036 (66.84)**	0.7005 (66.55)	**0.7374 (70.05)**	0.7227 (68.66)	0.7196 (68.36)	0.6879 (65.35)	0.6816 (64.75)	**0.7087 (67.33)**
11	0.8448 (949.50)	0.9775 (1098.68)	**0.9792 (1100.63)**	0.8950 (1005.95)	0.9737 (1094.43)	**0.9740 (1094.76)**	0.8379 (941.80)	0.9720 (1092.48)	**0.9732 (1093.86)**	0.0996 (112.00)	0.1013 (113.81)	**0.1066 (119.80)**
12	**0.8634 (398.90)**	0.8489 (392.19)	0.8573 (396.09)	0.9222 (426.05)	**0.9250 (427.33)**	0.9230 (426.42)	**0.9060 (418.55)**	0.8984 (415.05)	0.9005 (416.02)	0.9399 (434.25)	0.9395 (434.04)	**0.9402 (434.36)**
13	0.8753 (339.60)	0.8814 (341.98)	**0.8815 (342.02)**	0.9661 (374.85)	0.9649 (374.37)	**0.9676 (375.43)**	0.9347 (362.65)	0.9384 (364.08)	**0.9396 (364.56)**	0.9445 (366.45)	**0.9502 (368.66)**	0.9486 (368.06)
14	0.4852 (65.50)	**0.5134 (69.31)**	0.4935 (66.62)	**0.7215 (97.40)**	0.7187 (97.02)	0.7201 (97.22)	**0.4537 (61.25)**	0.4476 (60.43)	0.4499 (60.74)	**0.7674 (103.60)**	0.7524 (101.57)	0.7639 (103.13)
15	**0.8649 (943.65)**	0.8589 (937.02)	0.8564 (934.29)	**0.8723 (951.70)**	0.8649 (943.66)	0.8633 (941.84)	**0.8575 (935.55)**	0.8497 (927.04)	0.8487 (925.90)	**0.8779 (957.80)**	0.8719 (951.28)	0.8683 (947.34)
16	0.8733 (349.30)	0.8763 (350.51)	**0.8769 (350.77)**	0.9630 (385.20)	**0.9643 (385.71)**	0.9641 (385.62)	0.9433 (377.30)	0.9408 (376.31)	**0.9438 (377.52)**	0.9788 (391.50)	0.9762 (390.50)	**0.9801 (392.06)**

Table 8
p-values for classification accuracies w.r.t. different reducts based on Algorithms 2 and 3 (CART classifier).

ID	γ-Algorithm 2 & γ-Algorithm 3	DI-Algorithm 2 & DI-Algorithm 3	ENT-Algorithm 2 & ENT-Algorithm 3	NDER-Algorithm 2 & NDER-Algorithm 3
1	0.0461	*0.9100*	*0.3799*	0.0073
2	7.4262E−04	*0.5225*	*0.4218*	*0.4005*
3	*0.2393*	*0.5397*	*0.8707*	*0.5426*
4	*0.5509*	*0.4468*	*0.7149*	*0.5699*
5	*0.5229*	0.0047	*0.4525*	*0.1253*
6	*0.8666*	*0.8998*	*0.9218*	*0.5030*
7	*0.7753*	*0.6217*	*0.0552*	*0.8025*
8	*0.4988*	*0.7227*	*0.8816*	*0.6348*
9	*0.5471*	*0.7738*	*0.5334*	*0.9026*
10	*0.1593*	*0.3609*	*0.5515*	*0.3503*
11	*0.3719*	*0.6357*	*0.2035*	6.8030E−09
12	*0.9348*	*0.6087*	*0.9883*	*0.7249*
13	*0.8680*	*0.6912*	*0.9774*	*0.9380*
14	*0.6847*	*0.4647*	*0.8603*	9.0868E−04
15	*0.9568*	*0.6553*	*0.8286*	*0.4320*
16	*0.8903*	*0.9460*	*0.5959*	*0.9888*

Table 9
p-values for classification accuracies w.r.t. different reducts based on Algorithms 2 and 3 (KNN classifier).

ID	γ-Algorithm 2 & γ-Algorithm 3	DI-Algorithm 2 & DI-Algorithm 3	ENT-Algorithm 2 & ENT-Algorithm 3	NDER-Algorithm 2 & NDER-Algorithm 3
1	*0.1314*	*1.0000*	*0.2946*	*0.1860*
2	0.0029	*0.7126*	*0.1931*	*1.0000*
3	*0.4249*	*0.6927*	*0.9136*	*0.6650*
4	*0.8814*	*0.5336*	*0.6949*	*0.2446*
5	*0.5665*	*0.8681*	*0.2597*	*0.5976*
6	*0.9513*	*0.9442*	*0.9888*	*0.9238*
7	*0.5136*	*0.4225*	*0.5239*	*0.8025*
8	*0.3368*	*0.6807*	*0.8496*	*0.6155*
9	*0.5194*	*0.6319*	*0.4903*	*0.4713*
10	*0.3103*	*0.8922*	*0.4816*	*0.3368*
11	*0.7149*	*0.8710*	*0.6749*	6.8030E−09
12	*0.4614*	*0.5487*	*0.9883*	*0.9353*
13	*0.8664*	*0.7548*	*0.7232*	*0.9380*
14	*0.8709*	*0.8496*	*0.9245*	*0.0809*
15	*0.3506*	*0.5700*	*0.7970*	*0.2557*
16	*0.8907*	*0.6446*	*0.9674*	*0.4012*

Table 10
Joint distribution of classification results.

	$\mathrm{Pre}_{A_g}(x_i) = d(x_i)$	$\mathrm{Pre}_{A_g}(x_i) \neq d(x_i)$
$\mathrm{Pre}_{A_{g'}}(x_i) = d(x_i)$	e	f
$\mathrm{Pre}_{A_{g'}}(x_i) \neq d(x_i)$	m	n

the obtained reduct is stable completely. Following Eq. (8), one stability of reduct can be derived based on one radius. For 20 radii, 20 stabilities of reducts will be derived, then the average value of them is shown in Table 4.

In Table 4, for any two compared values, the greater value is in bold. Through observing Table 4, it is not difficult to know that for four measures, the average values of stabilities of reducts derived by using Algorithm 3 are greater than that of reducts derived by using Algorithm 2. Take the results on data set "Cardiotocography (ID: 3)" as an example, the value of "γ-Algorithm 2" is 0.4893 while the value of "γ-Algorithm 3" is 0.9420; the value of "DI-Algorithm 2" is 0.9006 while the value of "DI-Algorithm 3" is 0.9948; the value of "ENT-Algorithm 2" is 0.9414 while the value of "ENT-Algorithm 3" is 0.9979; the value of "NDER-Algorithm 2" is 0.6425 while the value of "NDER-Algorithm 3" is 0.9257. Immediately, some conclusions can be derived. (1) Our Algorithm 3 can generate reducts with higher stabilities. This is mainly because in Algorithm 3, the searching of attributes does not begin with an empty set, but begins with the selected attributes in reduct which is derived by using the whole training samples. Following the mechanism, some same attributes will exist among different reducts over different subsets of training samples, and then the stabilities of reducts derived by using Algorithm 3 can be improved. (2) Our Algorithm 3 is more suitable for obtaining reduct. Such conclusion can be attributed to the fact that our Algorithm 3 can generate reduct with better adaptability. The reduct with higher stability is universal, and then the reduct may be still the reduct for a little finer and coarser level of information granulation which may be caused by slight variation of data. (3) The stable reduct may provide more stable classification results, then it is helpful for us to determine which labels can be regarded as the final outputs of the testing samples.

In addition, to further analyze the results of stabilities of different reducts from the viewpoint of statistics, the Wilcoxon signed rank test [55] is employed. The significance level is set as 0.05. For each algorithm, one stability of reduct can be derived based on one radius through using Eq. (8). For two compared algorithms, 20 stabilities of reducts can be derived respectively, because 20 radii are employed in our experiments. Then compare the difference between these two sets of stabilities of reducts through using Wilcoxon signed rank test to derive *p*-value. If the returned *p*-value is lower than 0.05, then the two algorithms perform significantly different from the perspective of stabilities of reducts; otherwise, they perform equally well. The detailed results are shown in Table 5.

In Table 5, the *p*-values greater than 0.05 are in italic. Through observing Table 5, it is obvious that for four measures, most of the *p*-values are lower than 0.05. Such result implies that Algorithm 2 and Algorithm 3 perform significantly different. In other words, Algorithm 2 and Algorithm 3 do not perform equally well from the perspective of stabilities of reducts.

Table 11
Results of stabilities over classification results w.r.t. different reducts (CART classifier).

ID	γ-Algorithm 2	γ-Algorithm 3	DI-Algorithm 2	DI-Algorithm 3	ENT-Algorithm 2	ENT-Algorithm 3	NDER-Algorithm 2	NDER-Algorithm 3
1	0.7977	**0.8527**	0.7032	**0.7159**	0.6927	**0.7277**	0.7064	**0.7559**
2	0.7606	**0.8844**	0.6906	**0.8156**	0.7056	**0.8367**	0.7350	**0.8394**
3	0.7548	**0.8303**	0.8212	**0.8377**	0.7695	**0.7826**	0.8082	**0.8394**
4	0.8566	**0.8689**	0.8497	**0.8718**	0.8125	**0.8646**	0.8765	**0.8863**
5	0.8458	**0.8870**	0.8390	**0.8988**	0.8508	**0.8963**	0.7833	**0.8425**
6	0.7895	**0.7972**	0.8303	**0.8408**	0.8242	**0.8261**	0.7961	**0.8272**
7	0.8956	**0.9172**	0.9137	**0.9373**	0.8501	**0.8713**	0.9448	**0.9549**
8	0.8471	**0.8660**	0.8676	**0.8800**	0.8156	**0.8511**	0.8561	**0.8687**
9	0.6057	**0.6497**	0.7192	**0.7498**	0.5215	**0.6025**	0.6715	**0.7451**
10	0.6824	**0.7291**	0.7341	**0.7988**	0.7155	**0.7797**	0.7262	**0.7654**
11	0.8613	**0.8783**	0.8862	**0.8997**	0.8651	**0.8861**	0.8338	**0.8772**
12	0.8981	**0.9143**	0.9071	**0.9184**	0.9151	**0.9192**	0.9254	**0.9343**
13	0.9918	**0.9919**	0.9636	**0.9732**	0.9467	**0.9565**	0.9906	**0.9950**
14	0.7613	**0.8359**	0.8016	**0.8308**	0.6699	**0.7756**	0.7860	**0.8561**
15	0.9557	**0.9690**	0.9492	**0.9597**	0.9593	**0.9680**	0.9665	**0.9789**
16	0.9649	**0.9678**	0.9686	**0.9745**	0.9420	**0.9581**	0.9778	**0.9820**

Table 12
Results of stabilities over classification results w.r.t. different reducts (KNN classifier).

ID	γ-Algorithm 2	γ-Algorithm 3	DI-Algorithm 2	DI-Algorithm 3	ENT-Algorithm 2	ENT-Algorithm 3	NDER-Algorithm 2	NDER-Algorithm 3
1	0.7914	**0.8455**	0.7423	**0.7677**	0.6282	**0.6886**	0.7086	**0.7700**
2	0.7517	**0.9400**	0.7683	**0.8683**	0.7544	**0.8917**	0.7283	**0.9028**
3	0.7633	**0.8795**	0.8496	**0.8888**	0.8373	**0.8579**	0.8191	**0.8781**
4	0.9401	**0.9515**	0.8957	**0.9269**	0.8811	**0.9551**	0.9541	**0.9758**
5	0.8358	**0.9320**	0.8673	**0.9403**	0.8915	**0.9430**	0.7733	**0.8425**
6	0.8591	**0.8712**	0.8842	**0.8992**	0.9012	**0.9079**	0.8572	**0.8883**
7	0.8066	**0.8565**	0.8232	**0.8621**	0.8477	0.9404	0.7769	**0.7908**
8	0.9326	**0.9503**	0.9240	**0.9432**	0.8868	**0.9269**	0.9392	**0.9599**
9	0.6861	**0.7820**	0.7560	**0.8115**	0.5404	**0.7638**	0.6398	**0.7777**
10	0.7207	**0.8362**	0.7263	**0.8335**	0.7463	**0.8297**	0.7324	**0.8216**
11	0.9770	**0.9876**	0.9791	**0.9858**	0.9799	**0.9866**	0.9116	**0.9093**
12	0.9231	**0.9439**	0.9619	**0.9700**	0.9522	**0.9574**	0.9635	**0.9723**
13	0.9874	**0.9875**	0.9712	**0.9816**	0.9543	**0.9691**	0.9786	**0.9885**
14	0.7758	**0.8798**	0.8576	**0.9181**	0.6883	**0.8524**	0.8187	**0.9130**
15	0.8809	**0.9208**	0.8936	**0.9366**	0.9017	**0.9232**	0.8992	**0.9256**
16	0.9732	**0.9808**	0.9744	**0.9845**	0.9467	**0.9750**	0.9831	**0.9906**

4.6. Comparisons of classification performances

In this experiment, the classification performances related to reducts will be compared, then CART and KNN classifiers are employed to classify the testing samples through using corresponding reducts. Additionally, the parameter used in CART classifier is default in Matlab; the parameter used in KNN classifier is 3, that is, 3NN classifier is employed. In the following, for each radius, one classification accuracy can be derived through using one reduct over one classifier, and 5 classification accuracies can be derived through using 5 reducts over one classifier, then compute the average value of them. It follows that 20 average values of classification accuracies can be derived, because 20 radii are employed to compute reducts. Then compute the average value of these 20 average values of classification accuracies. The derived result is shown in Tables 6 and 7. It should be emphasized that the number of testing samples which are classified correctly is computed by using the same way. Whichever algorithm is used to compute reduct, the computation of classification accuracy and the number of correctly classified samples are the same. The detailed results are shown in Tables 6 and 7. Notably, "γ-All", "DI-All", "ENT-All" and "NDER-All" denote the classification accuracies related to reducts over the whole training set which are derived through using approximation quality, neighborhood discrimination index, conditional entropy and neighborhood decision error rate respectively.

In Tables 6 and 7, the value in parentheses is the number of testing samples which are classified correctly, and for the compared values, the greater value is in bold. With a deep investigation of Tables 6 and 7, it is not difficult to derive the following results: (1) in most cases, the classification accuracies related to reducts over different subsets of training samples are similar with that related to reducts over all of the training samples; (2) for four measures, the classification accuracies related to reducts derived by using Algorithm 3 are similar with that related to reducts derived by using Algorithm 2 in most cases. Take the results on data set "Breast Tissue (ID: 1)" as an example, for CART classifier, the values over "γ-All", "γ-Algorithm 2" and "γ-Algorithm 3" are 0.4205, 0.5277 and 0.5199, respectively; the values over "DI-All", "DI-Algorithm 2" and "DI-Algorithm 3" are 0.5364, 0.5318 and 0.5295, respectively; the values over "ENT-All", "ENT-Algorithm 2" and "ENT-Algorithm 3" are 0.2727, 0.3023 and 0.2977, respectively; the values over "NDER-All", "NDER-Algorithm 2" and "NDER-Algorithm 3" are 0.4977, 0.5386 and 0.5153, respectively.

Similar to Section 4.5, to further analyze the results of classification accuracies related to different reducts from the viewpoint of statistics, the Wilcoxon signed rank test is employed. The significance level is set as 0.05. For two compared algorithms, 20 average values of classification accuracies will be derived respectively, because 20 radii are used to derive reducts. Then compare the difference between these two sets of classification accuracies through using Wilcoxon signed rank test to derive p-value. If the p-value is lower than 0.05, then the two algorithms perform significantly different from the perspective of classification performance related to reduct; otherwise, they perform equally well. The detailed results are shown in Tables 8 and 9.

In Tables 8 and 9, the p-values greater than 0.05 are in italic. With a careful investigation of Tables 8 and 9, it is not difficult to conclude that for four measures, most of the returned p-values are higher than 0.05. Such result implies that Algorithm

Table 13
p-values for stabilities over classification results w.r.t. different reducts based on Algorithms 2 and 3 (CART classifier).

ID	γ-Algorithm 2 & γ-Algorithm 3	DI-Algorithm 2 & DI-Algorithm 3	ENT-Algorithm 2 & ENT-Algorithm 3	NDER-Algorithm 2 & NDER-Algorithm 3
1	0.0341	*0.7526*	*0.1892*	0.0137
2	3.8342E−05	2.8842E−07	4.7584E−07	1.5548E−05
3	1.9916E−04	0.0423	*0.2433*	0.0051
4	0.0228	0.0123	7.5402E−06	0.0026
5	0.0050	1.3501E−06	0.0006	2.7089E−04
6	0.0133	*0.2571*	*0.8334*	0.0475
7	*0.0845*	*0.0811*	0.0029	*0.8025*
8	0.0467	0.0192	0.0033	3.0554E−04
9	*0.2686*	*0.0503*	4.5334E−07	0.0032
10	0.0015	1.8430E−04	5.4754E−06	0.0090
11	0.0035	1.2435E−05	3.3711E−04	6.8030E−09
12	*0.2303*	1.1763E−03	*0.4539*	*0.1015*
13	*0.7438*	*0.4607*	*0.7016*	*0.3184*
14	0.0001	2.1979E−04	7.7797E−07	1.9001E−07
15	*0.3718*	0.0041	*0.1365*	0.0370
16	*0.6592*	*0.2275*	*0.1458*	*0.7371*

Table 14
p-values for stabilities over classification results w.r.t. different reducts based on Algorithms 2 and 3 (KNN classifier).

ID	γ-Algorithm 2 & γ-Algorithm 3	DI-Algorithm 2 & DI-Algorithm 3	ENT-Algorithm 2 & ENT-Algorithm 3	NDER-Algorithm 2 & NDER-Algorithm 3
1	0.0378	*0.3616*	0.0091	0.0031
2	6.4368E−08	3.7874E−05	6.6634E−07	7.5413E−08
3	1.2493E−05	0.0057	*0.1581*	6.8403E−04
4	0.*1396*	1.2718E−04	3.0341E−04	3.7499E−04
5	1.7895E−06	1.4411E−04	7.3345E−05	1.6538E−06
6	*0.3283*	0.0020	*0.7899*	0.0180
7	0.0233	*0.1453*	1.3931E−07	*0.8025*
8	0.0033	0.0111	0.1041	0.0151
9	0.0452	0.0179	6.7574E−08	1.8366E−06
10	8.7520E−05	2.5621E−06	1.3150E−07	0.0011
11	1.0337E−05	6.8505E−07	3.4668E−05	3.5924E−06
12	*0.0810*	0.0158	*0.3549*	7.2576E−05
13	*0.6886*	*0.4186*	*0.8275*	*0.5231*
14	2.2741E−05	2.3327E−06	2.1720E−07	1.5154E−07
15	2.8673E−06	1.9152E−07	2.5667E−05	2.0981E−04
16	*0.4253*	0.0290	*0.0945*	*0.4671*

Table 15
Detailed explanations of different expressions.

Expression	Measure	Used algorithm
γ-CV-Algorithm 2	Approximation quality	Algorithm 2
γ-CV-Algorithm 3	Approximation quality	Algorithm 3
DI-CV-Algorithm 2	Neighborhood discrimination index	Algorithm 2
DI-CV-Algorithm 3	Neighborhood discrimination index	Algorithm 3
ENT-CV-Algorithm 2	Conditional entropy	Algorithm 2
ENT-CV-Algorithm 3	Conditional entropy	Algorithm 3
NDER-CV-Algorithm 2	Neighborhood decision error rate	Algorithm 2
NDER-CV-Algorithm 3	Neighborhood decision error rate	Algorithm 3

2 and Algorithm 3 perform equally well from the perspective of classification performances related to reducts.

4.7. Comparisons of stabilities of classification results

Following the stability of reduct discussed in Section 4.5, the stabilities of classification results [54] will be further explored in this section. As exhibited in Table 10, the joint distribution of classification results for computing such comparative evaluation is designed.

In Table 10, "$\mathrm{Pre}_{A_g}(x_i)$" denotes the predicted label of x_i over A_g. "e, f, m, n" denote the number of samples which satisfy the corresponding conditions, respectively. Given a decision system DS, assuming that universe U is partitioned into K disjoint groups $U_1, U_2, \ldots, U_K$, the stability of classification result is formulated as

$$Sta_{classification} = \frac{2}{K \cdot (K-1)} \sum_{g=1}^{K-1} \sum_{g'=g+1}^{K} \frac{e+n}{e+f+m+n}. \tag{9}$$

Following Eq. (9), one stability of classification result can be derived over one classifier based on one radius. For 20 radii, 20 stabilities of classification results can be derived over one classifier, then the average value of them is shown in Tables 11 and 12.

In Tables 11 and 12, for any two compared values, the greater value is in bold. With a deep investigation of Tables 11 and 12, it is not difficult to derive the following results: for four measures, the stabilities of classification results related to reducts which are derived by using Algorithm 3 are higher than that related to reducts derived by using Algorithm 2. Take the results on data set "Cardiotocography (ID: 3)" as an example, for CART classifier, the value over "γ-Algorithm 2" is 0.7548 while the value over "γ-Algorithm 3" is 0.8303; the value over "DI-Algorithm 2" is 0.8212 while the value over "DI-Algorithm 3" is 0.8377; the value over "ENT-Algorithm 2" is 0.7695 while the value over "ENT-Algorithm 3" is 0.7826; the value over "NDER-Algorithm 2" is 0.8082 while the value over "NDER-Algorithm 3" is 0.8394. Such results imply that Algorithm 3 can generate reducts with more stable classification results. This is mainly because the stabilities of reducts can be improved by using Algorithm 3, it follows that such reducts provide more stable classification results.

Table 16
Results of time consumption of computing different reducts based on cross-validation.

ID	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	**0.5391**	0.0872	**0.2524**	0.0500	**1.3383**	0.2093	**0.2094**	0.0763
2	**3.0574**	0.6310	**1.6449**	0.3888	**3.0706**	0.5155	**2.9483**	1.0331
3	**13.2292**	2.7365	**5.0260**	0.9983	**5.7041**	1.1783	**7.9143**	1.8768
4	**278.6294**	48.9546	**89.1961**	16.2685	**146.5335**	23.7283	**12.8118**	3.1157
5	**22.7400**	5.2907	**5.6875**	1.0347	**9.4233**	1.5687	**15.5793**	5.5459
6	**29.2240**	5.1506	**11.3624**	2.0151	**22.9309**	3.6336	**14.8601**	2.6141
7	**0.2175**	0.0655	**0.0660**	0.0158	**0.1470**	0.0377	**0.0615**	0.0148
8	**357.7228**	67.3920	**32.8784**	6.3782	**64.4396**	11.2307	**89.8733**	19.3233
9	**103.1420**	12.9710	**78.3460**	9.6356	**136.8171**	16.9710	**9.1409**	1.1243
10	**1.8820**	0.7645	**1.0006**	0.3854	**1.6681**	0.5427	**0.8784**	0.5350
11	**93.2912**	14.5389	**19.0712**	2.9877	**61.3743**	9.7366	**4.3882**	1.7825
12	**6.7916**	1.5099	**3.8411**	0.8183	**4.8288**	1.0226	**8.6085**	2.0928
13	**2.0756**	0.2865	**2.5672**	0.3485	**5.2878**	0.6833	**0.8334**	0.1360
14	**3.8198**	1.0512	**3.0712**	0.5888	**3.6378**	0.7542	**5.6348**	1.1476
15	**14.8402**	2.3928	**6.3219**	1.0332	**14.7640**	2.1385	**2.6316**	0.4515
16	**0.4608**	0.0922	**0.2528**	0.0468	**0.7095**	0.1346	**0.2831**	0.0618

Table 17
Results of stabilities of different reducts based on cross-validation.

ID	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	0.8501	**0.9784**	0.7263	**0.8825**	0.9369	**1.0000**	0.4826	**0.8557**
2	0.0719	**0.9370**	0.1660	**0.9348**	0.1772	**0.9080**	0.0643	**0.8559**
3	0.5898	**0.9513**	0.8456	**0.9817**	0.8642	**0.9791**	0.6904	**0.9418**
4	0.8648	**0.9996**	0.7914	**0.9509**	0.8844	**1.0000**	1.0000	**1.0000**
5	0.0985	**0.9733**	0.6010	**0.9947**	0.5357	**0.9940**	0.1040	**0.8459**
6	0.9789	**0.9934**	0.9612	**0.9854**	1.0000	**1.0000**	0.9100	**0.9965**
7	0.4566	**0.8609**	0.6513	**0.9753**	0.3512	**0.9406**	0.8390	**0.9880**
8	0.9551	**1.0000**	0.9522	**0.9982**	0.9530	**1.0000**	0.8180	**1.0000**
9	0.2191	**1.0000**	0.2950	**1.0000**	0.4403	**1.0000**	0.5700	**0.9980**
10	0.2410	**0.8497**	0.2621	**0.9482**	0.3043	**0.9643**	0.1442	**0.8368**
11	0.2920	**1.0000**	0.5406	**1.0000**	0.5344	**1.0000**	0.3080	**0.9423**
12	0.7270	**0.9754**	0.8854	**0.9919**	0.9628	**0.9943**	0.6179	**0.9555**
13	0.8724	**1.0000**	0.7873	**1.0000**	0.7898	**1.0000**	0.8226	**0.9987**
14	0.2804	**0.8981**	0.5164	**0.9570**	0.2345	**0.9555**	0.2120	**0.9443**
15	0.9781	**1.0000**	0.9193	**0.9805**	0.9780	**1.0000**	0.6955	**1.0000**
16	0.9585	**1.0000**	0.9579	**1.0000**	0.9687	**1.0000**	0.9237	**0.9922**

To further analyze the results of stabilities over classification results of different reducts from the viewpoint of statistics, the Wilcoxon signed rank test is employed. The significance level is set as 0.05. Similar to the computation of p-value elaborated in Section 4.5, the p-value related to stabilities of classification results can be derived. If the returned p-value is lower than 0.05, then these two algorithms perform significantly different from the perspective of stabilities over classification results of different reducts; otherwise, they perform equally well. The detailed results are shown in Tables 13 and 14.

In Tables 13 and 14, the p-values greater than 0.05 are in italic. With a careful investigation of Tables 13 and 14, it is not difficult to observe that for four measures, most of the returned p-values are lower than 0.05. Such result indicates that Algorithm 2 and Algorithm 3 do not perform equally well from the perspective of stabilities of classification results related to reducts.

4.8. The application of crosswise computing reduct on cross-validation

In this section, we will introduce Algorithm 2 and Algorithm 3 into the framework of cross-validation [56–59]. In our experiments, 5-fold cross-validation is employed. Specifically, the universe is partitioned into 5 groups randomly. For each round, 4 groups compose the training set for crosswise computing reduct, and the rest of 1 group is regarded as testing set for testing the classification performance of reduct. Similar to Sections 4.4–4.7, the time consumption of computing different reducts, the stabilities of different reducts, the classification accuracies related to different reducts and the stabilities of classification results related to different reducts will be compared.

To facilitate the understanding of the experimental results, some explanations of used expressions in this section are shown in Table 15.

In Table 15, the "Expression" denotes the results related to corresponding reduct, and such reduct is computed through using the "Used algorithm" and the "Measure" based on cross-validation. For example, "γ-Algorithm 2" denotes the results related to reduct, and such reduct is computed through using Algorithm 2 and the measure of approximation quality based on cross-validation.

Similar to Section 4.4, the time consumption of computing reducts by using Algorithm 2 and Algorithm 3 will be compared. The details about time consumption of computing reduct are shown in Table 16.

In Table 16, for the compared two values, the greater value is in bold. With a careful investigation of Table 16, it is not difficult to observe that the elapsed time of computing reducts by using Algorithm 2 based on cross-validation is less than that by using Algorithm 3. Take the results on data sets "Breast Tissue (ID: 1)" as an example, the value over "γ-CV-Algorithm 2" is 0.5391 while the value over "γ-CV-Algorithm 3" is 0.0872; the value of "DI-CV-Algorithm 2" is 0.2524 while the value over "DI-CV-Algorithm 3" is 0.0500; the value over "ENT-CV-Algorithm 2" is 1.3383 while the value over "ENT-CV-Algorithm 3" is 0.2093; the value over "NDER-CV-Algorithm 2" is 0.2094 while the value over "NDER-CV-Algorithm 3" is 0.0763. Such result implies that our Algorithm 3 can also improve the time efficiency of computing reducts in the framework of cross-validation.

Table 18
Results of classification accuracies w.r.t. different reducts based on cross-validation (CART classifier).

ID	γ-CV-All	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-All	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-All	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-All	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	0.6349 (13.47)	0.6590 (13.84)	**0.6749 (14.17)**	0.6462 (13.71)	0.6362 (13.36)	**0.6527 (13.71)**	0.6321 (13.41)	0.6751 (14.18)	**0.6880 (14.45)**	0.5693 (12.08)	0.5850 (12.29)	**0.5962 (12.52)**
2	**0.6033 (10.86)**	0.5346 (9.62)	0.5388 (9.70)	**0.6622 (11.92)**	0.5591 (10.06)	0.5697 (10.25)	**0.6417 (11.55)**	0.5514 (9.93)	0.5907 (10.63)	**0.6594 (11.87)**	0.6277 (11.30)	0.6223 (11.20)
3	0.7666 (325.98)	0.7634 (324.46)	**0.7713 (327.82)**	**0.7299 (310.36)**	0.7211 (306.45)	0.7264 (308.71)	**0.6218 (264.38)**	0.6126 (260.34)	0.6171 (262.28)	**0.7293 (310.08)**	0.7131 (303.08)	0.7206 (306.25)
4	**0.8737 (1894.98)**	0.8625 (1870.73)	0.8620 (1869.71)	**0.8702 (1887.57)**	0.8636 (1873.13)	0.8650 (1876.13)	**0.8713 (1889.75)**	0.8591 (1863.49)	0.8576 (1860.22)	0.7333 (1572.60)	0.7250 (1590.60)	**0.7333 (1590.60)**
5	**0.2316 (9.15)**	0.2199 (8.80)	0.2081 (8.32)	0.1895 (7.52)	**0.1901 (7.60)**	0.1855 (7.42)	**0.1496 (5.85)**	0.1463 (5.91)	0.1380 (5.52)	0.2526 (9.99)	0.2565 (10.26)	**0.2688 (10.75)**
6	**0.8755 (1733.59)**	0.8569 (1696.72)	0.8570 (1696.92)	**0.8621 (1707.20)**	0.8468 (1676.63)	0.8470 (1677.07)	**0.8780 (1738.60)**	0.8592 (1701.20)	0.8592 (1701.20)	**0.8566 (1696.32)**	0.8355 (1654.31)	0.8399 (1662.91)
7	0.8348 (58.60)	0.8838 (61.86)	**0.8909 (62.36)**	0.8318 (58.39)	0.8535 (59.75)	**0.8548 (59.83)**	0.8305 (58.30)	**0.8507 (59.55)**	0.8473 (59.31)	0.8226 (57.75)	0.8469 (59.28)	**0.8534 (59.74)**
8	**0.8542 (3416.71)**	0.8417 (3366.79)	0.8427 (3370.91)	**0.8553 (3421.06)**	0.8415 (3366.00)	0.8412 (3364.83)	**0.8542 (3416.62)**	0.8411 (3364.31)	0.8412 (3364.60)	**0.6975 (2789.80)**	0.6852 (2740.63)	0.6893 (2757.10)
9	**0.6388 (332.19)**	0.6250 (324.99)	0.6382 (331.85)	**0.7747 (402.85)**	0.7651 (397.85)	0.7721 (401.51)	**0.7005 (364.24)**	0.6806 (353.91)	0.6797 (353.45)	**0.5736 (298.26)**	0.5621 (292.31)	0.5582 (290.27)
10	0.7162 (68.18)	0.7139 (67.82)	**0.7241 (68.79)**	0.6950 (66.17)	0.7232 (68.71)	**0.7379 (70.10)**	0.6915 (65.84)	**0.7188 (68.29)**	0.7145 (67.88)	0.7055 (67.17)	0.7115 (67.59)	**0.7295 (69.31)**
11	**0.7958 (894.52)**	0.7835 (880.70)	0.7923 (890.57)	**0.8354 (938.97)**	0.8160 (917.18)	0.8233 (925.40)	**0.7719 (867.67)**	0.7571 (851.03)	0.7635 (858.15)	0.1248 (140.28)	0.1179 (132.50)	**0.1325 (148.88)**
12	0.8638 (399.09)	0.8616 (398.04)	**0.8698 (401.86)**	0.9043 (417.78)	**0.9051 (418.14)**	0.9045 (417.88)	0.8938 (412.92)	0.8976 (414.69)	**0.8993 (415.48)**	0.9344 (431.67)	0.9361 (432.48)	**0.9385 (433.57)**
13	**0.9967 (386.91)**	0.9948 (385.97)	0.9962 (386.53)	**0.9507 (369.08)**	0.9352 (362.86)	0.9469 (367.39)	**0.9297 (360.91)**	0.9162 (355.49)	0.9275 (359.87)	**0.7945 (308.41)**	0.7735 (300.10)	0.7871 (305.38)
14	0.4879 (65.86)	0.5136 (69.34)	**0.5196 (70.14)**	0.7633 (103.04)	0.7701 (103.96)	**0.7765 (104.83)**	0.4601 (62.12)	**0.4795 (64.73)**	0.4765 (64.33)	0.7721 (104.24)	0.7658 (103.38)	**0.7739 (104.48)**
15	**0.9928 (1083.34)**	0.9919 (1082.20)	0.9921 (1082.40)	0.9927 (1083.26)	0.9917 (1081.94)	0.9919 (1082.13)	**0.9928 (1083.37)**	0.9917 (1081.95)	0.9921 (1082.40)	0.8641 (942.89)	0.8576 (935.60)	**0.8655 (944.28)**
16	**0.9683 (387.33)**	0.9634 (385.38)	0.9633 (385.32)	**0.9683 (387.30)**	0.9636 (385.44)	0.9635 (385.41)	**0.9677 (387.08)**	0.9613 (384.52)	0.9602 (384.09)	**0.9211 (368.45)**	0.9118 (364.73)	0.9190 (367.60)

Table 19
Results of classification accuracies w.r.t. different reducts based on cross-validation (KNN classifier).

ID	γ-CV-All	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-All	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-All	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-All	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	0.7026 (14.93)	0.6961 (14.62)	**0.7049 (14.80)**	**0.6725 (14.29)**	0.6514 (13.68)	0.6718 (14.11)	0.7003 (14.88)	0.7096 (14.90)	**0.7131 (14.98)**	0.5511 (11.70)	0.5523 (11.60)	**0.5528 (11.61)**
2	**0.5911 (10.64)**	0.5390 (9.70)	0.5031 (9.06)	**0.6550 (11.79)**	0.5823 (10.48)	0.5858 (10.54)	**0.6589 (11.86)**	0.5560 (10.01)	0.6009 (10.82)	**0.6694 (12.05)**	0.6098 (10.98)	0.6349 (11.43)
3	0.7561 (321.50)	0.7446 (316.44)	**0.7565 (321.51)**	0.7352 (312.59)	0.7336 (311.80)	**0.7435 (315.98)**	**0.6205 (263.84)**	0.6023 (255.99)	0.6038 (256.60)	0.7136 (303.43)	0.7088 (301.26)	**0.7303 (310.37)**
4	**0.9495 (2059.42)**	0.9429 (2045.23)	0.9437 (2046.83)	0.9423 (2043.87)	0.9434 (2046.16)	**0.9455 (2050.70)**	**0.9469 (2053.77)**	0.9397 (2038.12)	0.9401 (2039.06)	0.7436 (1612.80)	0.7503 (1627.48)	**0.7503 (1627.48)**
5	**0.2344 (9.27)**	0.1995 (7.98)	0.1856 (7.42)	**0.1346 (5.33)**	0.1317 (5.27)	0.1309 (5.23)	**0.1113 (4.39)**	0.0806 (3.22)	0.0655 (2.62)	**0.1855 (7.34)**	0.1764 (7.05)	0.1846 (7.38)
6	**0.9017 (1785.54)**	0.8896 (1761.33)	0.8897 (1761.56)	**0.8936 (1769.46)**	0.8829 (1748.14)	0.8831 (1748.53)	**0.9024 (1787.00)**	0.8901 (1762.40)	0.8901 (1762.40)	**0.8867 (1755.90)**	0.8711 (1724.80)	0.8741 (1730.63)
7	**0.8058 (56.58)**	0.7949 (55.64)	0.7989 (55.93)	0.8399 (58.96)	0.8663 (60.64)	**0.8698 (60.89)**	0.8563 (60.11)	**0.8830 (61.81)**	0.8725 (61.07)	0.8263 (58.01)	0.8449 (59.14)	**0.8516 (59.61)**
8	**0.9520 (3807.82)**	0.9496 (3798.53)	0.9517 (3806.85)	**0.9532 (3812.94)**	0.9498 (3799.34)	0.9495 (3798.08)	0.9524 (3809.47)	0.9487 (3794.74)	**0.9494 (3797.59)**	**0.7373 (2949.15)**	0.7283 (2913.21)	0.7355 (2942.09)
9	0.5924 (308.07)	0.5892 (306.41)	**0.5928 (308.24)**	0.7492 (389.57)	0.7541 (392.15)	**0.7570 (393.65)**	**0.6412 (333.42)**	0.6348 (330.08)	0.6310 (328.14)	0.5455 (283.65)	**0.5566 (289.42)**	0.5551 (288.65)
10	0.7335 (69.83)	0.7167 (68.09)	**0.7380 (70.11)**	0.6890 (65.59)	**0.7331 (69.65)**	0.7288 (69.23)	0.7073 (67.34)	0.7382 (70.13)	**0.7398 (70.28)**	0.6952 (66.18)	0.7034 (66.83)	**0.7214 (68.54)**
11	**0.8622 (969.14)**	0.8516 (957.17)	0.8597 (966.32)	**0.8946 (1005.53)**	0.8854 (995.18)	0.8936 (1004.42)	**0.8554 (961.46)**	0.8420 (946.39)	0.8531 (958.86)	**0.1129 (126.93)**	0.1055 (118.53)	0.1119 (125.80)
12	0.8580 (396.41)	0.8638 (399.09)	**0.8723 (403.01)**	0.9148 (422.65)	**0.9253 (427.51)**	0.9252 (427.43)	0.9063 (418.70)	0.9060 (418.58)	**0.9080 (419.49)**	0.9429 (435.61)	0.9473 (437.64)	**0.9482 (438.08)**
13	0.9953 (386.39)	0.9938 (385.61)	**0.9955 (386.26)**	**0.9484 (368.17)**	0.9314 (361.37)	0.9420 (365.50)	**0.9280 (360.24)**	0.9140 (354.64)	0.9259 (359.23)	**0.7680 (298.14)**	0.7369 (285.93)	0.7628 (295.97)
14	0.4884 (65.93)	**0.5152 (69.55)**	0.5113 (69.03)	0.7802 (105.33)	0.7779 (105.02)	**0.7835 (105.77)**	0.4495 (60.68)	0.4679 (63.17)	**0.4683 (63.21)**	**0.7881 (106.40)**	0.7756 (104.70)	0.7792 (105.19)
15	**0.8730 (952.64)**	0.8548 (932.55)	0.8549 (932.66)	**0.8715 (951.02)**	0.8532 (930.88)	0.8530 (930.64)	**0.8732 (952.80)**	0.8549 (932.74)	0.8549 (932.67)	**0.8436 (920.54)**	0.8376 (913.80)	0.8431 (919.84)
16	**0.9853 (394.13)**	0.9784 (391.36)	0.9790 (391.58)	**0.9852 (394.08)**	0.9785 (391.40)	0.9788 (391.53)	**0.9849 (393.97)**	0.9760 (390.39)	0.9752 (390.09)	0.8846 (353.84)	0.8768 (350.74)	**0.8884 (355.36)**

Table 20
Results of stabilities of classification results w.r.t. different reducts based on cross-validation (CART classifier).

ID	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	0.8086	**0.8158**	0.7996	**0.8138**	**0.8094**	0.7633	0.7632	**0.8400**
2	0.7200	**0.8538**	0.8028	**0.8822**	0.7749	**0.8006**	0.7466	**0.8306**
3	0.7793	**0.8289**	0.8174	**0.8417**	0.7736	**0.7821**	0.8159	**0.8362**
4	0.8743	**0.8824**	0.8759	**0.8834**	0.8733	**0.8836**	**0.8868**	0.8853
5	0.8189	**0.881**1	0.8452	**0.8773**	0.8523	**0.8863**	0.8009	**0.8530**
6	0.8436	**0.8450**	0.8299	**0.8313**	**0.8480**	0.8470	0.8309	**0.8390**
7	0.8944	**0.9241**	0.8838	**0.9174**	0.8296	**0.8851**	0.9081	**0.9449**
8	0.8660	**0.8702**	0.8681	**0.8704**	**0.8656**	0.8631	0.8547	**0.8963**
9	0.5963	**0.6507**	0.7417	**0.7628**	0.6778	**0.6870**	0.7258	**0.8098**
10	0.6685	**0.7309**	0.6875	**0.7709**	0.7002	**0.7537**	0.6770	**0.7796**
11	0.7540	**0.8480**	0.8149	**0.8700**	0.7526	**0.8333**	0.8440	**0.9266**
12	0.9058	**0.9200**	0.9119	**0.9198**	0.9084	**0.9271**	0.9262	**0.9487**
13	0.9931	**0.9973**	0.9292	**0.9531**	0.9126	**0.9312**	0.9325	**0.9649**
14	0.7386	**0.8095**	0.8180	**0.8560**	0.6796	**0.7811**	0.7946	**0.8631**
15	0.9936	**0.9941**	0.9930	**0.9939**	0.9935	**0.9956**	0.9180	**0.9436**
16	0.9760	**0.9770**	0.9763	**0.9771**	0.9755	**0.9766**	0.9693	**0.9737**

Table 21
Results of stabilities of classification results w.r.t. different reducts based on cross-validation (KNN classifier).

ID	γ-CV-Algorithm 2	γ-CV-Algorithm 3	DI-CV-Algorithm 2	DI-CV-Algorithm 3	ENT-CV-Algorithm 2	ENT-CV-Algorithm 3	NDER-CV-Algorithm 2	NDER-CV-Algorithm 3
1	0.8494	**0.8549**	0.8361	**0.8540**	0.8508	**0.8565**	0.7379	**0.8182**
2	0.7430	**0.8856**	0.8289	**0.9026**	0.7966	**0.9134**	0.7792	**0.9037**
3	0.7939	**0.8738**	0.8410	**0.8875**	0.8325	**0.8573**	0.8169	**0.8753**
4	0.9644	**0.9781**	0.9577	**0.9734**	0.9635	**0.9773**	0.9257	**0.9257**
5	0.8146	**0.9283**	0.8714	**0.9240**	0.8701	**0.9248**	0.7808	**0.8606**
6	0.9294	**0.9300**	0.9260	**0.9269**	0.9302	**0.9302**	0.9189	**0.9276**
7	0.8395	**0.8806**	0.8802	**0.9250**	0.8526	**0.9392**	0.8856	**0.9076**
8	0.9624	**0.9668**	0.9634	**0.9673**	0.9621	**0.9669**	0.9134	**0.9518**
9	0.5634	**0.7958**	0.7133	**0.8594**	0.6235	**0.8094**	0.6872	**0.7851**
10	0.7119	**0.8351**	0.7008	**0.8561**	0.7418	**0.8551**	0.6844	**0.8246**
11	0.8345	**0.9493**	0.8850	**0.9595**	0.8501	**0.9445**	**0.9170**	0.8985
12	0.9224	**0.9416**	0.9560	**0.9638**	0.9518	**0.9560**	0.9533	**0.9698**
13	0.9923	**0.9980**	0.9398	**0.9762**	0.9309	**0.9665**	0.9467	**0.9835**
14	0.7499	**0.8544**	0.8486	**0.9102**	0.6930	**0.8525**	0.8193	**0.9143**
15	0.9321	**0.9360**	0.9205	**0.9316**	0.9321	**0.9361**	0.9096	**0.9470**
16	0.9919	**0.9934**	0.9919	**0.9934**	0.9913	**0.9932**	0.9776	**0.9901**

Similar to Section 4.5, the stabilities of reducts which are derived by using Algorithm 2 and Algorithm 3 based on cross-validation are compared. The details are shown in Table 17.

In Table 17, for the compared values, the greater value is in bold. With a careful investigation of Table 17, it is not difficult to observe that the stabilities of reducts derived by using Algorithm 3 based on cross-validation are higher than that derived by using Algorithm 2 based on cross-validation. Take the results on data set "Cardiotocography (ID: 3)" as an example, the value over "γ-CV-Algorithm 2" is 0.5898 while the value over "γ-Algorithm 3" is 0.9513; the value over "DI-Algorithm 2" is 0.8456 while the value over "DI-Algorithm 3" is 0.9817; the value over "ENT-Algorithm 2" is 0.8642 while the value over "ENT-Algorithm 3" is 0.9791; the value over "NDER-Algorithm 2" is 0.6904 while the value over "NDER-Algorithm 3" is 0.9418. Such result implies that Algorithm 3 can also generate reducts with higher stabilities in the framework of cross-validation.

Similar to Section 4.6, the classification accuracies related to different reducts based on cross-validation are compared. CART and KNN classifiers are employed to classify the testing samples. The details about classification accuracies related to different reducts are shown in Tables 18 and 19. Notably, "γ-CV-All", "DI-CV-All", "ENT-CV-All" and "NDER-CV-All" denote the classification accuracies related to reducts over the whole training set which are derived through using approximation quality, neighborhood discrimination index, conditional entropy and neighborhood decision error rate based on cross-validation, respectively.

In Tables 18 and 19, the value in parentheses is the number of testing samples which are classified correctly, and for compared values, the greater value is in bold. With a careful investigation of Tables 18 and 19, it is not difficult to derive the following results: (1) the classification accuracies related to reducts over different subsets of training samples based on cross-validation are similar with that related to reducts over all of the training samples based on cross-validation; (2) the classification accuracies related to reducts derived by using Algorithm 3 based on cross-validation are similar with that related to reducts derived by using Algorithm 2 based on cross-validation. Take the results on data set "Breast Tissue (ID: 1)" as an example, for CART classifier, the values over "γ-CV-All", "γ-CV-Algorithm 2" and "γ-CV-Algorithm 3" are 0.6349, 0.6590 and 0.6749, respectively; the values over "DI-CV-All", "DI-CV-Algorithm 2" and "DI-CV-Algorithm 3" are 0.6462, 0.6362 and 0.6527, respectively; the values over "ENT-CV-All", "ENT-CV-Algorithm 2" and "ENT-CV-Algorithm 3" are 0.6321, 0.6751 and 0.6880, respectively; the values over "NDER-CV-All", "NDER-CV-Algorithm 2" and "NDER-CV-Algorithm 3" are 0.5693, 0.5850 and 0.5962, respectively.

Similar to Section 4.7, the stabilities of classification results related to reducts which are derived through using Algorithm 2 and Algorithm 3 based on cross-validation are compared. The details are shown in Tables 20 and 21.

In Tables 20 and 21, the greater value is in bold. With a careful investigation of Tables 20 and 21, it is not difficult to observe that the stabilities of classification results related to reducts which are derived by using Algorithm 3 based on cross-validation are higher than that related to reducts which are derived by Algorithm 2 based on cross-validation. Take the results on data set "Brain

Tumor (ID: 2)" as an example, for CART classifier, the value over "γ-Algorithm 2" is 0.7200 while the value over "γ-Algorithm 3" is 0.8538; the value over "DI-Algorithm 2" is 0.8028 while the value over "DI-Algorithm 3" is 0.8822; the value over "ENT-Algorithm 2" is 0.7749 while the value over "ENT-Algorithm 3" is 0.8006; the value over "NDER-Algorithm 2" is 0.7466 while the value over "NDER-Algorithm 3" is 0.8306. Such result implies that Algorithm 3 can generate reducts with more stable classification results.

5. Conclusions and future perspectives

In this paper, to acquire multiple reducts for evaluating the performances related to stabilities of reducts and searching strategy, the method of crosswise computing reduct is proposed. Different from the previous approach which derives one and only one reduct over the whole data, the mechanism of our proposed strategy is to partition the whole data into several different groups, and then computing reducts over those crosswise selected groups. Moreover, to speed up the process of crosswise computing reduct, an acceleration strategy is designed. Different from the naive approach which repeats the greedy searching over each subset of data, our proposed acceleration strategy is realized through considering that the reduct over the whole data may provide guidance for the computation of reducts over different subsets of data. Furthermore, the experimental results over 16 data sets have demonstrated superiorities of our acceleration strategy: (1) the elapsed time can be decreased significantly; (2) the stability of reduct can be improved; (3) the generated reduct can provide more stable classification results without poorer classification performance.

The following topics deserve our further researches.

1. The proposed acceleration strategy is only used in general process of computing reduct, such a strategy will be further explored for alleviating over-fitting in the process of computing reduct.
2. The proposed acceleration strategy only speeds up the process of searching attributes without considering the perspectives of samples and attributes in data. Therefore, the acceleration strategy which considers both samples and attributes will be further addressed.

CRediT authorship contribution statement

Zehua Jiang: Software, Validation, Writing - original draft. **Keyu Liu:** Investigation, Writing - Review & Editing. **Jingjing Song:** Data curation, Funding acquisition. **Xibei Yang:** Conceptualization, Methodology, Formal analysis, Resources, Writing - Review & Editing, Project administration, Funding acquisition. **Jinhai Li:** Supervision. **Yuhua Qian:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Natural Science Foundation of China (No. 62076111, 61906078), the Postgraduate Research & Practice Innovation Program of Jiangsu Province, PR China (No. KYCX20_3162) and the Key Laboratory of Data Science and Intelligence Application, Fujian Province University, PR China (No. D1901).

References

[1] D.G. Chen, Y.Y. Yang, Attribute reduction for heterogeneous data on the combination of classical and fuzzy rough set models, IEEE Trans. Fuzzy Syst. 22 (2014) 1325–1334.
[2] Z.H. Jiang, K.Y. Liu, X.B. Yang, H.L. Yu, H. Fujita, Y.H. Qian, Accelerator for supervised neighborhood based attribute reduction, Internat. J. Approx. Reason. 119 (2020) 122–150.
[3] K.Y. Liu, X.B. Yang, H.L. Yu, X.J. Chen, Supervised information granulation strategy for attribute reduction, Int. J. Mach. Learn. Cybern. 11 (2020) 2149–2163.
[4] Y.H. Qian, J.Y. Liang, W. Wei, Consistency-preserving attriute reduction in fuzzy rough set framework, Int. J. Mach. Learn. Cybern. 4 (2013) 287–299.
[5] J.J. Song, C.C. Tsang Eric, D.G. Chen, X.B. Yang, Minimal decision cost reduct in fuzzy decision-theoretic rough set model, Knowl.-Based Syst. 126 (2017) 104–112.
[6] W. Wei, J.Y. Liang, Information fusion in rough set theory: an overview, Inf. Fusion 48 (2019) 107–118.
[7] J.H. Li, Y. Ren, C.L. Mei, Y.H. Qian, X.B. Yang, A comparative study of multigranulation rough sets and concept lettices via rule acquisition, Knowl.-Based Syst. 91 (2016) 152–164.
[8] A. Skowron, A. Jankowski, Rough sets and and interactive granular computing, Fund. Inform. 147 (2016) 371–385.
[9] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognit. Lett. 24 (2003) 833–849.
[10] W. Pedrycz, Granular bidirectional and multidirectional associative memories: towards a collaborative buildup of granular mappings, J. Inf. Process. Syst. 13 (2017) 435–447.
[11] W. Pedrycz, M. Krawczak, S. Zadrozny, Computational intelligence techniques for decision support, data mining and information searching, Inform. Sci. 460–461 (2018) 374–376.
[12] Y.H. She, X.L. He, H.X. Shi, Y.H. Qian, A multi-valued logic approach for multigranulation rough set model, Internat. J. Approx. Reason. 82 (2017) 270–284.
[13] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowiński (Ed.), Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, 1992, pp. 331–362.
[14] S. Wang, T.R. Li, C. Luo, H.M. Chen, H. Fujita, Domain-wise approaches for updating approximations with multi-dimensional variation of ordered information systems, Inform. Sci. 478 (2019) 100–124.
[15] W.H. Xu, W.X. Sun, Y.F. Liu, W.X. Zhang, Fuzzy rough set models over two universes, Int. J. Mach. Learn. Cybern. 4 (2013) 631–645.
[16] X. Yang, T.R. Li, D. Liu, H.M. Chen, C. Luo, A unified model of sequential three-way probabilistic rough sets, Inform. Sci. 420 (2017) 126–147.
[17] C. Zhang, D.Y. Li, J.Y. Liang, Hesitant fuzzy linguistic rough set over two universe model and its applications, Int. J. Mach. Learn. Cybern. 9 (2018) 577–588.
[18] J.Z. Li, X.B. Yang, X.N. Song, J.H. Li, P.X. Wang, D.J. Yu, Neighborhood attribute reduction: a multi-criterion approach, Int. J. Mach. Learn. Cybern. 10 (2019) 731–742.
[19] Y.D. Zhang, Z.J. Yang, H.M. Lu, X.X. Zhou, P. Phillips, Q.M. Liu, S.H. Wang, Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation, Emot.-Aware Mob. Comput. 4 (2016) 8375–8385.
[20] Q.H. Hu, D.R. Yu, Z.X. Xie, Neighborhood classifiers, Expert Syst. Appl. 34 (2008) 866–876.
[21] Q.H. Hu, W. Pedrycz, D.R. Yu, J. Lang, Selecting discrete and continuous features based on neighborhood decision error minimization, IEEE Trans. Cybern. 40 (2010) 137–150.
[22] J.H. Dai, Q. Wu, W.T. Wang, H.W. Tian, Conditional entropy for incomplete decision systems and its application in data minging, Int. J. Gen. Syst. 41 (2012) 713–728.
[23] C.Z. Wang, Q.H. Hu, X.Z. Wang, D.G. Chen, Y.H. Qian, Z. Dong, Feature selection based on neighborhood discrimination index, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 2986–2999.
[24] X. Zhang, C.L. Mei, D.G. Chen, J.H. Li, Feature selection in mixed data: a method using a novel fuzzy rough set-based information entropy, Pattern Recognit. 56 (2016) 1–15.
[25] M.J. Cai, G.M. Lang, H. Fujita, Z.Y. Li, T. Yang, Incremental approaches to updating reducts and under dynamic covering granularity, Knowl.-Based Syst. 172 (2019) 130–140.
[26] D.G. Chen, Y.Y. Yang, Z. Dong, An incremental algorithm for attribute reducition with variable precision rough sets, Appl. Soft Comput. 45 (2016) 129–149.
[27] J. Fan, Y.L. Jiang, Y. Liu, Quick attribute reduction with generalized indiscernibility models, Inf. Sci. 397-398 (2017) 15–36.
[28] G.M. Lang, D.Q. Miao, M.J. Cai, Z.F. Zhang, Incremental approaches for updating reducts in dynamic covering information systems, Knowl.-Based Syst. 134 (2017) 85–104.

[29] K.Y. Liu, X.B. Yang, H.L. Yu, J.S. Mi, P.X. Wang, X.J. Chen, Rough set based semisupervised feature selection via ensemble selector, Knowl.-Based Syst. 165 (2019) 282296.
[30] F. Min, H.P. He, Y.H. Qian, W. Zhu, Test-cost-sensitive attribute reduction, Inform. Sci. 181 (2011) 4928–4942.
[31] 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, Inform. Sci. 179 (2009) 4140–4150.
[32] W. Wei, X.Y. Wu, J.Y. Liang, J.B. Cui, Y.J. Sun, Discernibility matrix based incremental attribute reduction for dynamic data, Knowl.-Based Syst. 140 (2018) 142–157.
[33] S.P. Xu, X.B. Yang, H.L. Yu, D.J. Yu, J.Y. Yang, E.C.C. Tsang, Multi-label learning with label-speci?c feature reduction, Knowl.-Based Syst. 104 (2016) 52–61.
[34] X.B. Yang, Y.S. Qi, X.N. Song, J.Y. Yang, Test cost sensitive multigranulation rough set: model and minimal cost selection, Inform. Sci. 250 (2013) 184–199.
[35] Y. Chen, K.Y. Liu, J.J. Song, H. Fujita, X.B. Yang, Y.H. Qian, Attribute group for attribute reduction, Inform. Sci. 535 (2020) 64–80.
[36] Y.H. Qian, Q. Wang, H.H. Chen, J.Y. Liang, C.Y. Dang, Fuzzy-rough feature selection accelerator, Fuzzy Sets and Systems 258 (2015) 61–78.
[37] X.S. Rao, X.B. Yang, X. Yang, X.J. Chen, D. Liu, Y.H. Qian, Quickly calculating reduct: an attribute relationship based approach, Knowl.-Based Syst. 200 (2020) 106014.
[38] H.R. Ju, H.X. Li, X.B. Yang, B. Huang, Cost-sensitive rough set: a multi-granulation approach, Knowl.-Based Syst. 123 (2017) 137–153.
[39] D. Liu, D.C. Liang, Incremental learning researches on rough set theory: status and future, Int. J. Rough Sets Data Anal. 1 (2014) 99–112.
[40] J.S. Mi, W.Z. Wu, W.X. Zhang, Approaches to knowledge reduction based on variable precision rough set model, Inform. Sci. 159 (2004) 255–272.
[41] Z. Pawlak, Rough set, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[42] X.B. Yang, S.C. Liang, H.L. Yu, S. Gao, Y.H. Qian, Pseudo-label neighborhoood rough set: measures and attrtibute reductions, Internat. J. Approx. Reason. 105 (2019) 112–129.
[43] Z.H. Jiang, X.B. Yang, H.L. Yu, D. Liu, P.X. Wang, Y.H. Qian, Accelerator for multi-granularity attribute reduction, Knowl.-Based Syst. 177 (2019) 145–158.
[44] K.Y. Liu, X.B. Yang, H. Fujita, D. Liu, X. Yang, Y.H. Qian, An efficient selector for multi-granularity attribute reduction, Inform. Sci. 505 (2019) 457–472.
[45] Y.Y. Yao, Y. Zhao, J. Wang, On reduct construction algorithms, Trans. Comput. Sci. 2 (2008) 100–117.
[46] J.H. Zhai, X.Z. Wang, X.H. Pang, Voting-based instance selection from large data sets with mapreduce and random weight networks, Inform. Sci. 367–368 (2016) 1066–1077.
[47] P.F. Zhu, W.C. Zhu, Q.H. Hu, C.Q. Zhang, W.M. Zuo, Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (2017) 364–374.
[48] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php.
[49] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, et al., Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (2002) 436–442.
[50] X. Zhang, Cancer Gene Expression Datasets, Zenodo, 2015, http://dx.doi.org/10.5281/zenodo.21712.
[51] J.H. Dai, W.T. Wang, H.W. Tian, L. Liu, Attribute selection base on a new conditional entropy for incomplete decision systems, Knowl.-Based Syst. 39 (2013) 207–213.
[52] Q.H. Hu, L. Zhang, D.G. Chen, W. Pedrycz, D.R. Yu, Gaussian kernel based fuzzy rough sets: model, uncertainty measures and applications, Internat. J. Approx. Reason. 51 (2010) 453–471.
[53] Y. Liu, W.L. Huang, Y.L. Jiang, Z.Y. Zeng, Quick attribute reduct algorithm for neighborhood rough set model, Inform. Sci. 271 (2014) 65–81.
[54] X.B. Yang, Y.Y. Yao, Ensemble selector for attribute reduction, Appl. Soft Comput. 70 (2018) 1–11.
[55] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[56] G.X. Jiang, W.J. Wang, Error estimation based on variance analysis of *k*-fold cross-validation, Pattern Recognit. 69 (2017) 94–106.
[57] G.X. Jiang, W.J. Wang, Markov cross-validation for time series model evaluations, Inform. Sci. 375 (2017) 219–233.
[58] J.H. Kim, Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap, Comput. Statist. Data Anal. 53 (2009) 3735–3745.
[59] T.T. Wong, Performance evaluation of classification algorithms by *k*-fold and leave-one-out cross validation, Pattern Recogonition 48 (2015) 2839–2846.