Incremental attribute reduction under variations of the attribute set based on conflict region

Chuanjian Yang¹, Gang Li²

¹ School of Computer and Information Engineering, Chuzhou University, Chuzhou, 239000, China
² School of Mechanical and Electrical Engineering Chuzhou University, Chuzhou, 239000, China

Email (✉): Gang Li (e-mail: gangli78@126.com)

ABSTRACT In real application, the attribute set of decision systems may vary with time. How to efficiently update the reduct becomes one of important tasks in the knowledge discovery. The static methods for updating knowledge need to recalculate when the attribute set varies every time, which makes it potentially very time-consuming to update knowledge, especially if the data sets are growled rapidly. Incremental learning method is an efficient technique for knowledge discovery in the dynamic system. In the dynamic system of the attribute set variations, there exist three kinds of attribute set changes: addition of the attribute set, deletion of the attributes and simultaneous variation of adding and deleting the attribute sets. The objective of this paper is to study the incremental reduction algorithm based on conflict region in the dynamic decision system. For three variations of the attribute set, we firstly introduce incremental mechanisms of updating conflict region; and then we research acceleration strategies for calculating significance measures of candidate attributes and eliminating redundant attributes. Consequently, a unified incremental reduction algorithm based on conflict region is presented for three variations of the attribute set. Finally, we design a series of experiments which are performed on 12 UCI data sets. The experimental results indicate that the proposed incremental methods can effectively to update the reduct with variations of the attribute set, and its efficiency is much better than that of static algorithms.

INDEX TERMS rough set; incremental reduction; dynamic decision system; conflict region

I. Introduction

Rough set theory, proposed by Pawlak in 1982 [1], is a powerful mathematical analysis methodology for handling uncertainty, imprecision and fuzziness information. It has been widely attention in data mining, machine learning, pattern recognition, knowledge discovery and so on. Attribute reduction is one of important applications of rough sets, whose objective is to eliminate superfluous attributes from the decision systems so as to obtain a simplified attribute set.

Attribute reduction is an important preprocessing step of knowledge discovery to find a minimums subset of the original attribute set for keeping the distinguishing power of the original decision systems. Many methods of attribute reduction approaches have been developed to improve knowledge learning, data mining and classifiers [2][3][4][5][6][7][8]. However, these methods are often applied in the static decision systems. In many practical applications, there exist many dynamic decision systems, that is, the data of decision systems changes over time. Since static reduction approaches need to recalculate from scratch to obtain a new reduct, they consume a great deal of calculation time and are not suitable for high-speed data processing. Apparently, static reduction methods are inefficient and not suitable for dealing with dynamic systems. Therefore, the incremental reduction techniques are efficient approaches to deal with the dynamic variations of the decision systems. Currently, in the dynamic decision systems, techniques of dynamic attribute reduction have attracted the attention of many scholars, whose researches mainly focus on three aspects: variation of the object set, variation of the attribute set and variation of attributes value.

With the variation of the object set, Hu et al. [10] researched an incremental technique for objects being added into the decision system and presented a reduction algorithm to update the reduct. Yang [12] improved the discernibility matrix and provided an incremental updating reduction algorithm based on the discernibility matrix. For multiple new objects being absorbed into the information
systems, Xu et al. [13] proposed a new reduction algorithm to incrementally update the reduct based on 0-1 integer programming. Liang et al. [15] researched incremental strategies for three representative information entropies and proposed a group incremental reduction algorithm, when the object set is added into the decision systems. For a new object adding into the decision table, Wei et al. [16] constructed an incremental reduction algorithm based on compacted decision table for updating three types of reduct. Shu et al. [17] presented the incremental updating to select a new reduct for the variation of multiple objects in complete decision systems. For adding and deleting signal object or multi-objects of incomplete decision systems, Shu et al. [18] researched updating mechanisms of positive region and designed dynamic reduction algorithms. From the perspective of knowledge granularity, Jing et al. [19] proposed an incremental reduction approach for objects varying dynamically. Based on the relative discernibility relations, Yang et al. [20] presented incremental algorithms for incrementally updating reduction when being added and deleted features of fuzzy rough set systems. For dominance-based rough set, Sang et al. [21] proposed incremental reduction approaches for dynamic ordered data in DRSA framework based on the matrix.

With the variation of the attribute set, Shu et al. [22] introduced updating mechanisms for the attribute set being added into or deleted from the incomplete decisions, and proposed incremental reduction algorithms. For hybrid fuzzy rough set systems, Zeng et al. [23] presented the hybrid distance for many types of data, researched updating mechanisms under the variation of the attribute set and proposed incremental reduction algorithms to update reduct. Based on three representative entropies, Wang et al. [24] presented the dimension incremental mechanisms of three entropies and developed a dimension incremental algorithm with incremental attribute set based on information entropy. When attributes are added to the decision system, Jing et al. [25] researched mechanisms of updating knowledge granularity and then proposed incremental reduction algorithms. Chen et al. [26] developed an incremental algorithm to obtain reduct based on discernibility objects when new attributes arriving.

With the variation of attribute values, Wang et al. [27] researched mechanisms of three representative entropies and proposed incremental reduction algorithm to update the reduct when attribute values being changed. For the incomplete decision systems, when the attribute values of single object or multiple objects are changed, Shu et al. [28] developed updating mechanisms of positive region, and proposed an incremental reduction algorithm. When values of single object and multiple objects are varied, Jing et al. [31] researched updating mechanisms of knowledge granularity, and designed two incremental reduction algorithms. For incomplete decision system with variation of attribute values, Xie et al. [29] analyzed three updating strategies of tolerance classes under three types of data changes, and proposed incremental reduction algorithms. For computing all reducts of the dynamic decision systems, Wei et al. [30] proposed an incremental reduction algorithm based on the discernibility matrix of the compacted decision system.

Additionally, for considering multi-dimensional variations of data sets, some scholars have also carried out research on this aspect. For the object and attribute sets change simultaneously, Jing et al. [32] researched updating strategies of computing reduct and proposed incremental reduction algorithms based on relative knowledge granulation. In terms of discernibility relation, Dong et al. [33] introduced updating mechanisms for adding objects and attributes simultaneously, and designed incremental attribute reduction to improve efficiency of the algorithm.

In the real applications, the dynamic variation of the attribute set is one of the most common scenes in the dynamic systems, such as clinical symptoms, environmental monitoring, insurance systems, etc. However, the attributes of actual decision-making systems often show a variety of changes. For example, in the insurance system, in order to track customers more accurately, customer related features will increase and some irrelevant features will be eliminated; Sometimes, features are added and removed at the same time. Hence, this paper focuses on the variations of the attribute set. Generally, the evolving characteristics of the attribute set include three aspects, i.e., addition of the attribute set, deletion of the attribute set, and simultaneous addition and deletion of the attribute set. More researches on first two variations of the attribute set have attracted much attention in recent years. For the third variation of the attribute set, only a few scholars pay attention to it, which is usually adopted by adding attributes and deleting attributes in turn. The operation is repetitive and inefficient. In addition, there is few reports on simultaneously researching three variations of the attribute set and scarcely unify incremental reduction framework, which inspires the research motivation of this paper. Therefore, this paper aims for investigating the mechanisms and implementations of incremental attribute reduction based on conflict region under three variations of the attribute set in the decision system. In order to address this, the corresponding concepts of conflict region are introduced. For three variations of the attribute set, three incremental strategies of updating conflict region for dynamic decision systems are researched, and the acceleration strategies for computing significance measures of candidate attributes and removing redundant attributes based on conflict region are discussed to improve algorithm performance. A unified incremental reduction algorithm is presented for three variations of the attribute set. Finally, a serial of experiments are performed on UCI data sets and the complexity analysis show that the proposed incremental methods are effective and efficient to update attribute reduct with variations of the attribute set. In general, the main contributions of this article can be summarized as follows. (1) We present a unified incremental reduction algorithm for three variations of the attribute set; (2) We introduce acceleration strategies to further improve the efficiency of the incremental algorithm; (3) In order to
illustrate the advantages of the proposed incremental methods, theoretical analysis and experimental verification are introduced.

The rest of this paper is organized as follows. In Section 2, some basic notions of rough sets are briefly reviewed and a static attribute reduction algorithm is presented based on conflict region; In Section 3, for three dynamic variations of the attribute set, the incremental strategies of updating conflict region are researched; In Section 4, an acceleration strategy for removing redundant attributes is discussed and a unified incremental reduction algorithms is presented for three variations of the attribute set; some examples are used to illustrated the proposed algorithm. In Section 5, we design some experiments to verify the proposed incremental reduction algorithm. Section 6 concludes this paper with some remarks and discussions.

II. Preliminaries

A. The concepts of Pawlak rough set

In this subsection, we review some basic concepts of rough set theory [1][9].

A decision system can be denoted as $S=(U, C \cup D, V, f)$, where $U=\{x_1, x_2, \ldots, x_n\}$ is a finite non-empty set of objects, which is called the universe; $C$ is the set of conditional attributes, and $D$ is the set of decision attributes; $V$ is the domain of attribute $a \in C \cup D$, and $V_a$ denotes the value domain of the attribute $a$; $f: U \times (C \cup D) \rightarrow V$ is an information function which associates an unique value of each attribute with every object belonging to $U$, such that for each $a \in A$ and $x \in U$, $f(x,a) \in V_a$. For simplicity, we suppose that the decision table has only one decision attribute.

For $R \subseteq C \cup D$, the indiscernibility relation (i.e., equivalence relation) on $R$ can be denoted as $IND(R) = \{(x,y) \in U \times U \mid \forall a \in R, f(x,a)=f(y,a)\}$. The objects $x$ and $y$ are indistinguishable under the equivalence relation $IND(R)$. The equivalence class including $x$ with respect to $R$ is defined as $[x]_R = \{y \mid (x,y) \in IND(R)\}$.

For any $X \subseteq U$, the lower and the upper approximations of $X$ with respect to $R$ are denoted as follows, respectively.

$$R^- X = \{x \in U \mid [x]_R \subseteq X\}$$

$$R^+ X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$$

The positive, negative and boundary regions of $X$ with respect to $R$ are denoted as follows, respectively.

$$POS(X) = R^- X$$

$$BND(X) = R^- X - R^+ X$$

$$NEG(X) = U - R^+ X$$

The positive region of $D$ with respect to $R$ is defined as $POS(D) = \bigcup x \in U R^+ X$.

Given a decision table $S$, if $\exists x, y \in U$ and $f(x, C)=f(y, C)$ and $f(x, D)\neq f(y, D)$, then $S$ is an inconsistent decision table, and $x, y$ are called inconsistent objects; Otherwise, $S$ is a consistent decision table.

Proposition 1 Given a decision table $S=(U, C \cup D, V, f)$ and $R, P \subseteq C$, we have $U(R \cup P) \subseteq U/R$. [47]

Proposition 1 denotes that $R$ is coarser than $R \cup P$, or $R \cup P$ is finer than $R$.

B. The concepts of conflict region

Definition 1 Given a decision table $S=(U, C \cup D, V, f)$, $X \subseteq U$ and $R \subseteq C \cup D$. The quotients of the lower and upper approximations of $X$ are defined as follows, respectively. [34]

$$\tilde{R}_X = \{x \mid [x]_R \subseteq X\}$$

$$\tilde{R}^- X = \{x \mid [x]_R \cap X \neq \emptyset\}$$

where $\tilde{R}_X$ denotes the set of the equivalence classes defined by $U/R$, which are the subsets of $X$. $\tilde{R}^- X$ represents the set of the equivalence classes defined by $U/R$, which have a non-empty intersection with $X$. Definition 1 gives the expressions of the lower and upper approximations with regard to subsets of $U/R$ rather than subsets of $U$.

Definition 2 Given a decision table $S=(U, A=C \cup D, V, f)$, let $X \subseteq U$ and $R \subseteq A$. The quotient sets of the positive, boundary and negative regions of $X$ with respect to $R$ are defined as follows, respectively. [35]

$$QPOS(X) = \tilde{R}^- X$$

$$QBND(X) = \tilde{R}^- X - \tilde{R}^+ X$$

$$QNEG(X) = U - \tilde{R}^+ X$$

Definition 3 Given a decision table $S=(U, C \cup D, V, f)$ and $R \subseteq C$, the conflict region and the quotient set of conflict region of $D$ with respect to $R$ are defined as follows, respectively. [35]

$$\text{CON}(D) = \bigcup_{a \in D} \tilde{R}^- D_a$$

$$\text{QCON}(D) = \tilde{R}^- D_a \cup \bigcup_{a \notin D} \tilde{R}^- D_a = U - \text{QPOS}(D)$$

$$\text{CON}(D) = U - \text{QPOS}(D)$$

$$\text{QCON}(D) = \bigcup_{a \in D} \tilde{R}_a$$

Definition 4 Given a decision table $S=(U, A=C \cup D, V, f)$ and $R \subseteq C$, for $a \in C\cap R$, the joined significance measure of the attribute $a$ regarding to $R$ can be defined as. [35]

$$\text{SIG}_{CON} (a, R) = |\text{QCON}(D) - \text{QCON}(D-a) D)$$

Definition 5 Given a decision table $S=(U, A=C \cup D, V, f)$ and $R \subseteq C$, for $a \in R$, the weeded significance measure of the attribute $a$ regarding to $R$ can be defined as. [35]

$$\text{SIG}_{CON} (a, R) = |\text{QCON}(D) - \text{QCON}(D-a) D)$$

If $\text{SIG}_{CON} (a, R) > 0$, the attribute $a$ is indispensable; otherwise, if $\text{SIG}_{CON} (a, R) = 0$, it is dispensable. Therefore, according to definition of $\text{SIG}_{CON} (a, R)$, we can delete
dispensable attributes from the attribute set $R$. For $a \in C$, if $\text{Sig}_\text{CON}(a, R, D) > 0$, then $a \in \text{CORE}(C)$. [35]

**Definition 6** Given a decision table $S=(U, C \cup D, \mathcal{V}_f)$ and $R \subseteq C$, $R$ is the reduct of positive region, which satisfies with two conditions as follows.

1. $|\text{CON}_D| > |\text{CON}_C|$;
2. $\forall a \in R, |\text{CON}_{R \cup \{a\}}(D)| \neq |\text{CON}_D(D)|$.

The first condition of Definition 6 ensures that information obtained by $R$ include the same conflict information as that of the original attribute set $C$. The second condition of Definition 6 guarantees that there are no redundant attributes in $R$.

C. The static reduction algorithm based on conflict region

In the sub-section, a static heuristic attribute reduction algorithm based on conflict region is designed, which includes three components: (1) computing core attributes according to Definition 5; (2) generating the attribute subset; (3) deleting redundant attributes on the basis of Definition 5. The static algorithm can be described as follows.

**Algorithm 1** The static reduction algorithm based on conflict region (SRACR)

**Input:** A decision table $S=(U, C \cup D, \mathcal{V}_f)$;

**Output:** A reduct $R$;

Step 1: compute $\text{CON}_D(D)$, $R = \emptyset$;

Step 2: for ($\forall a \in C$)

if $(\text{Sig}_\text{CON} - (a, C, D) > 0)$

$\text{CORE}(C) = \text{CORE}(C)$;

else $\text{CORE}(C) = \text{CORE}(C)$;

Step 3: if ($R = \emptyset$) $\text{CON}_D(D) = U$;

else compute $\text{CON}_D(D)$ and $\text{QCON}_D(D)$;

Step 4: while ($|\text{CON}_D(D) - \text{CORE}(C)|$)

for ($\forall a_i \in C - R$)

$\text{Compute } \text{Sig}_\text{CON}(a_i, R, D)$;

$b = \text{argmax} \{\text{Sig}_\text{CON}(a_i, R, D)\}$

$R = R \cup \{b\}$;

Update $\text{CON}_D(D)$ and $\text{QCON}_D(D)$;

Step 5: for ($\forall a \in R - \text{CORE}(C)$)

if $(\text{Sig}_\text{CON}(a, R, D) = 0)$ $R = R - \{a\}$;

Step 6: Output the new reduct $R$.

Computing equivalence class is a key step of attribute reduction, which is realized by the method of radix sort in the paper. So, the time complexity of Step 1 is $O(|C|/|U|)$. Step 2 is to compute core attributes, and the complexity is $O(|C|^2/|U|)$. The time complexity of Step 3 is $O(|C|)$.

**Example 1** Table 1 illustrates a decision table, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $C = \{a, b, c, d, e\}$ is the condition attribute set and $D$ is the decision attribute set.

| $U$ | $a$ | $b$ | $c$ | $d$ | $e$ |
|-----|-----|-----|-----|-----|-----|
| $x_1$ | 1   | 0   | 0   | 0   | 1   |
| $x_2$ | 0   | 0   | 0   | 1   | 1   |
| $x_3$ | 0   | 0   | 0   | 1   | 1   |
| $x_4$ | 1   | 0   | 0   | 0   | 0   |
| $x_5$ | 1   | 0   | 0   | 0   | 0   |
| $x_6$ | 0   | 1   | 0   | 1   | 1   |
| $x_7$ | 0   | 0   | 0   | 1   | 1   |

We can obtain equivalence classes $U/C = \{\{x_1\}, \{x_2, x_3\}, \{x_4, x_5\}, \{x_6, x_7\}\}$, where $\{x_4, x_5\}$ and $\{x_6, x_7\}$ are two inconsistent object sets. Hence, we have $|\text{CON}_D(D)| = |\{x_1, x_2, x_3\}| = |\{x_4, x_5\}|$. Next, we can get $\text{CON}_D(D) = \text{CON}_C(D) = \text{CON}_{C \cup \{a\}}(D) = \text{CON}_{C \cup \{b\}}(D) = \text{QCON}_D(D) = \text{QCON}_{C \cup \{a\}}(D) = \text{QCON}_{C \cup \{b\}}(D) = \{x_1, x_2, x_3\}$ and $\text{QCON}_D(D) = \{x_4, x_5\}$. According to Definition 5, $\text{Sig}_\text{CON}(c, D) = |\text{CON}_{C \cup \{c\}}(D)| - |\text{CON}_D(D)| = 2 > 0$ and $\text{Sig}_\text{CON}(b, D) = |\text{QCON}_{C \cup \{b\}}(D)| - |\text{QCON}_D(D)| = 0$, one have $b, e \in \text{CORE}(C)$. Let $R = \{a, b\}$. Sequentially, we can get $|\text{CON}_D(D) - \text{CORE}(C)| = 0$, and $\text{CON}_D(D) = \{x_1, x_2, x_3\}$. And then, we have $|\text{QCON}_D(D)| = |\text{QCON}_{C \cup \{b\}}(D)|$. The attribute $a$ cannot be deleted from $R$. Hence, a reduct of $S$ is $R = \{a, b\}$, where $U/R = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, and $\text{QCON}_D(D) = \{x_4, x_5\}$.

III. Principles of updating the conflict region for variations of the attribute set

In real-world applications, the condition attribute set of decision systems often varies dynamically over time. The variations of the attribute set contains three scenarios: (1) addition of an attribute set, (2) deletion of an attribute set, and (3) simultaneous addition and deletion of the attribute sets. It will consume a lot of time when re-computing the new reduct. Incremental method is one of effective approaches for computing the new reduct based on previous knowledge to avoid repeated calculations in the dynamic systems.

In the previous discussion of attribute reduction, we know that the calculation of conflict region plays an important role based on conflict region, which directly affect the efficiency of reduction algorithm. In the section, we will discuss incremental principles for updating conflict region and significance measure of attributes under three variations of the attribute set respectively. Section 3.1 discusses the incremental mechanism to update conflict region with addition of the attribute set; Section 3.2 presents updating approaches of conflict region for deleting the attribute set; when adding and deleting attribute sets change simultaneously, the incremental update strategies of conflict region are researched in Section 3.3.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
A. Updating the conflict region for adding the attribute set

For a decision table, the knowledge granularity may be refined when an attribute set is added into the decision table and equivalence classes of the new decision table may become smaller than that of the original one.

Given a decision table \(S=(U,C \cup D,V_f)\), let \(R \subseteq C\), \(U/R=\{X_1, \ldots, X_n\}\) and \(U/D=\{D_1, \ldots, D_m\}\), where, \(X_i \subseteq \text{QCON}(D)\) (\(1 \leq i \leq n\)) and \(X_j \subseteq \text{QCON}(D)\) (\(s+1 \leq j \leq n\)). Then, we have \(\text{QCON}_{R \cup a}(D)=\text{QCON}(D)\cdot \{x \in X_i \mid |X_i^a/D|=1\}\) (\(1 \leq s \leq s+1\)) and \(\text{QCON}_{R \cup a}(D)=\{X_j \mid |X_j^a/D|=1\}\) (\(1 \leq s \leq s+1\)).

**Proof:** When the attribute set \(P\) being added into the attribute set \(R\), \(X_i\) is refined into \(X_i^a\). There exist two situations for \(X_i\) as follows.

1. If \(s+1 \leq j \leq n\), then \(X_i \not\in \text{QCON}(D)\). From Proposition 2, we have \(X_i^a \not\in \text{QCON}_{R \cup a}(D)\) and \(\forall x \in X_i, x \not\in \text{QCON}_{R \cup a}(D)\).

2. If \(1 \leq j \leq s\), then \(X_i \in \text{QCON}(D)\) and \(\forall x \in X_i, x \not\in \text{CON}_{R \cup a}(D)\).

Proposition 2 can easily be proved on basis of Proposition 1.

Proposition 2 denotes that if the equivalence class is non-conflicting, the refined equivalence class is still non-conflicting after adding attribute set.

After adding the attribute set \(P\) into \(R\), the conflict region \(X_i(1 \leq i \leq s)\) is refined into \(X_i^a(1 \leq i \leq s)\), which includes four situations.

1. The granularity of the equivalent class \(X_i(1 \leq i \leq s)\) becomes finer, but the number of conflict objects becomes unchanged.
2. The granularity of the equivalent class \(X_i(1 \leq i \leq s)\) becomes finer, but the number of conflict objects reduces; that is, partial refined granules become non-collision granules.
3. The granularity of the equivalent class \(X_i(1 \leq i \leq s)\) becomes finer, and \(X_i^a(1 \leq i \leq s)\) are all no-conflict region (i.e., positive region).
4. The granularity of the equivalent class \(X_i(1 \leq i \leq s)\) remains unchanged.

For above four situations, we will research mechanisms of incrementally calculate \(\text{QCON}_{R \cup a}(D)\), \(\text{CON}_{R \cup a}(D)\) and \(\text{QCON}_{R \cup a}(D)\).

**Theorem 1** Given a decision table \(S=(U,C \cup D,V_f)\), let \(R \subseteq C\), \(U/R=\{X_1, \ldots, X_n\}\), and \(U/D=\{D_1, \ldots, D_m\}\). Suppose that \(P\) is the added attribute set \((C \cap P=\emptyset)\), and \(U/(R \cup P)=\{X_1^a, \ldots, X_n^a\}\), and \(X_j^a \subseteq \text{QCON}(D)\) (\(s+1 \leq j \leq n\)). Then, we have \(\text{QCON}_{R \cup a}(D)=\text{QCON}(D)\cdot \{x \in X_i^a \mid |X_i^a/D|=1\}\) (\(1 \leq s \leq s+1\)) and \(\text{QCON}_{R \cup a}(D)=\{X_j \mid |X_j^a/D|=1\}\) (\(1 \leq s \leq s+1\)).

**Proof:** When the attribute set \(P\) being added into the attribute set \(R\), \(X_i\) is refined into \(X_i^a\). There exist two situations for \(X_i\) as follows.
From Proposition 3, we can find that the joint significance measure of the attribute \( a \) with respect to \( R \) can be obtained from the results of partitioning \( QCON_a(D) \) by the attribute \( a \), thereby avoiding re-computing the significance measure in the whole universe, and the quotient set of \( CON_a(D) \) will also be continuously decrease in the process of reduction. This mechanism implies the idea of acceleration computation for the jointed significance measure, which will improve the performance of attribute reduction algorithm.

### B. Updating the conflict region for deleting the attribute set

When deleting the attribute set from the decision table, some of the knowledge granularity will be changed, while others will keep unchanged. The changed knowledge granularity will be coarse with the deletion of the attribute set.

For the decision table \( S=(U,C \cup D,V,f) \), let \( R, Q \subseteq C \), \( U/R=\{X_1, \ldots, X_s, X_{s+1}, \ldots, X_n\} \) and \( U/D=\{D_1, \ldots, D_m\} \). Suppose that \( Q \) is the deleted attribute set. The deletion operation of the attribute set may lead to coarsen the partition granularity. Let \( U/(R-Q) \) be expressed as:

\[
U/(R-Q) = \{X'_1, \ldots, X'_s, X'_{s+1}, \ldots, X'_n\}
\]

where \( X'_i \) (\( 1 \leq i \leq s \)) denotes the equivalence class whose granularity will keep unchanged, \( X' \) (\( 1 \leq i \leq s \)) in \( U/R \) represents the equivalence class whose granularity will be coarse, and \( X'_j \) (\( 1 \leq j \leq n \)) in \( U/(R-Q) \) denotes the equivalence class which some granules of \( X \) are combined into form a new equivalence class.

The coarsened granularity exist two different aspects: one situation is that the conflict object of \( Q \), \( CON \), the joined significance measure of the attribute \( \phi \) with respect to \( R \) can be defined as:

\[
\text{sig}_2^\alpha(b,R-Q,D)=|CON_{Q\phi}(D)|/|CON_{Q\phi}(b)|
\]

#### Definition 8
Given a decision table \( S=(U,C \cup D,V,f) \) and \( R \subseteq C \), \( Q \) is the deleted attribute set. For each \( b \in C-R-Q \), the joined significance measure of the attribute \( b \) with respect to \( R-Q \) can be defined as:

\[
\text{sig}_2^\alpha(b,R-Q,D)=|CON_{Q\phi}(D)|/|CON_{Q\phi}(b)|
\]

#### Proposition 4
Given a decision table \( S=(U,C \cup D,V,f) \), let \( R \subseteq C \), \( Q \) is the deleted attribute set. For each \( b \in C-R-Q \), the joined significance measure of the attribute \( b \) with respect to \( R-Q \) can be defined as:

\[
\text{sig}_2^\alpha(b,R-Q,D)=|CON_{Q\phi}(D)|/|CON_{Q\phi}(b)|
\]

Same as to Proposition 3, Proposition 4 implies the acceleration strategy for computing the jointed significance measure of the attribute \( b \) in \( C-Q-R \), which will improve the performance of reduction algorithm.

#### C. Updating the conflict region for adding and deleting the attribute sets simultaneously

In the real world, two variations of the attribute set (i.e., adding an attribute set and deleting an attribute set) may exist simultaneously. We can separately employ addition and deletion of the attribute sets to update the conflict region and compute the new reduct. However, many repetitive operations exist in two separate execution processes, which affect the efficiency of reduction algorithm. In the following, we consider a direct method to
deal with the case of adding and deleting attribute sets simultaneously.

For the decision table \( S = (U, C \cup D, V, f) \), let \( R \subseteq C \), \( U/R = \{X_1, \ldots, X_n\} \), and \( U/D = \{D_1, \ldots, D_m\} \). \( Q \) is the deleted attribute set and \( P \) is the added attribute set \((C \cap P = \Phi)\). Let \( R' = R \cup P \cdot Q \) and \( U/R' = \{X_1', \ldots, X_s', X_{s+1}', \ldots, X_n\} \), where \( X_j \) \((1 \leq j \leq n)\) in \( U/R \) denotes the equivalence class which will be changed, \( X_j \) \((s+1 \leq j \leq n)\) in \( U/R \) denotes the equivalence class which will be unchanged, and \( X_i \) \((1 \leq i \leq s)\) in \( U/R' \) denotes the equivalence class which has been changed.

**Theorem 3** Given a decision table \( S = (U, C \cup D, V, f) \), let \( R \subseteq C \), \( U/R = \{X_1, \ldots, X_n\} \) and \( U/D = \{D_1, \ldots, D_m\} \). \( Q \) is the deleted attribute set and \( P \) is the added attribute set \((C \cap P = \Phi)\). Let \( R' = R \cup P \cdot Q \) and \( U/R' = \{X_1', \ldots, X_s', X_{s+1}', \ldots, X_n\} \), we have \( \text{CON}_R(D) = \text{CON}_Q(D) \cup \{x \in X_i' | |X_i'/D| > 1\} \cup \{y \in X_j | |X_j/D| > 1\} \) \((1 \leq i \leq s, 1 \leq j \leq n)\) and \( \text{QCON}_R(D) = \{X_i' | |X_i'/D| > 1\} \cup \{X_j | |X_j/D| > 1\} \) \((1 \leq i \leq s, s+1 \leq j \leq n)\).

**Proof:** Similar to Theorem 2, when adding and deleting the attribute sets simultaneously, the equivalence class exist two scenarios: (1) the granularity of the equivalence class is changed; (2) the granularity of the equivalence class keep unchanged. The properties of the equivalence class will be changed for the first scenario and the properties of the equivalence class remain unchanged for the second scenario. For above two scenarios, we have the following analyses:

(1) In the first scenario, for each \( X_i' \in U/R' \) \((1 \leq i \leq s)\), there exist \( |X_i'/D| > 1 \) or \( |X_i'/D| = 1 \); and

(2) In the second scenario, for each \( X_j \in U/R' \) \((s+1 \leq j \leq n)\), there exist \( |X_j/D| > 1 \) or \( |X_j/D| = 1 \).

Hence, \( \text{CON}_R(D) = \{x \in X_i' | |X_i'/D| > 1\} \cup \{y \in X_j | |X_j/D| > 1\} \) \((1 \leq i \leq s, 1 \leq j \leq n)\) and \( \text{QCON}_R(D) = \{X_i' | |X_i'/D| > 1\} \cup \{X_j | |X_j/D| > 1\} \) \((1 \leq i \leq s, s+1 \leq j \leq n)\). And, according to relevant properties of set operations, we have \( \text{CON}_R(D) \cap \text{QCON}_R(D) = \{x \in X_i' | |X_i'/D| > 1\} \cup \{X_j | |X_j/D| > 1\} \) \((1 \leq i \leq s, s+1 \leq j \leq n)\).

**Definition 9** Given a decision table \( S = (U, C \cup D, V, f) \), let \( R \subseteq C \), \( Q \) is the deleted attribute set and \( P \) is the added attribute set \((C \cap P = \Phi)\). For each \( c \in C \cup P - R - Q \), the joined significance measure of the attribute \( c \) with respect to \( R - Q \) can be defined as.

\[
\text{Sig}^*_{\text{CON}^c}(c, R - Q, D) = |\text{CON}_R(D) \cap \text{QCON}_{R - c}(D)|
\]

**Proposition 5** Given a decision table \( S = (U, C \cup D, V, f) \), let \( R \subseteq C \), \( Q \) is the deleted attribute set and \( P \) is the added attribute set \((C \cap P = \Phi)\). For each \( c \in C \cup P - R - Q \), the joined significance measure of the attribute \( c \) with respect to \( R - Q \) can be defined as.

\[
\text{Sig}^2_{\text{CON}^c}(c, R - Q, D) = |\{ x \in Z | W \in \text{QCON}_R(D) \land Z \in W[c] \land |Z/D| = 1\}|
\]

Same as to Proposition 3, Proposition 5 implies the acceleration mechanism for calculating the joined significance measure of the attribute \( c \in C \cup P - R - Q \), which will improve the performance of reduction algorithm.

**IV. Incremental attribute reduction for variations of the attribute set**

From the discussion of the previous section, we know that variations of the attribute set lead the changes of conflict region and the operation of updating region directly affects the efficiency of reduction algorithm. In this section, to improve the efficiency of attribute reduction, we will develop a unified incremental attribute reduction algorithm for three variations of the attribute set.

When variations of the attribute sets of the original decision system, first, based on the corresponding incremental strategies, the incremental algorithm for updating conflict regions is executed. Then, we select the most significance attribute from candidate attributes to add into the selected attribute subset and repeat above process until the new selected attribute subset satisfies the termination condition. Finally, we use the deletion manner to remove redundant attributes from the selected attribute subset and obtain the new reduction. The process of incrementally updating reduce with variations of the attribute sets is displayed in Fig. 1. In Fig. 1, the orange operation modules indicate that incremental method and acceleration strategy are used. These methods will be specifically analyzed in Fig 1.
A. The acceleration strategy for removing redundant attributes

According to Definition 6, condition(1) denotes that the attributes in \( R \) are sufficient, and condition(2) ensures that the selected attribute of \( R \) is necessary for preserving the conflict region of target decisions. For an attribute \( a \) of \( R \), if \( \text{Sig}_{\text{CON}}(a,R,D)=0 \), the attribute \( a \) is redundant and should be removed from \( R \). Generally, the time complexity of distinguishing the redundancy of every attribute in \( R \) is \( O(|R||U|) \) and the number of radix sort is \( |R|(|R|-1) \), which is similar to step 4 of algorithm SRACR. This operation of checking necessary of every attribute in \( R \) will consume plenty of computational time for quadratic time of radix sort.

According to the previous analysis, it is known that radix sort is a pivotal and frequent operation in the process of attribute algorithm based on conflict region. The computation of reducing the number of the sort can improve the efficiency of reduction algorithm. In order to improve the performance of reduction algorithms, we research an acceleration strategy by reducing sort times in the process of checking redundant attributes.

For the decision table \( S=(U,C \cup D,V,f) \), \( C \) is the condition attribute set which includes \( c_1,c_2,\ldots,c_{|C|} \), etc. \( U/C \) is a partition of the universe with respect to \( C \). We use radix sort to compute the equivalence classes of \( U/C \). For \( U/(C-{c_i}) \) and \( U/(C-{c_i+1}) \), \(|C|-2 \) times sorts are repetitive (i.e., \( c_{i+1},...,c_{C-1},c_1,\ldots,c_i \)) and have only one different sort. We all know that radix sort is stable; thus we can obtain \( U/(C-{c_i+1}) \) derived from \( U/(C-{c_i}) \) [59][36].

Proposition 6 For the decision table \( S=(U,C \cup D,V,f) \), \( \text{RoS}(C-{c_i+1})=\text{RoS}(C-{c_i}) \).

Proof: \( \text{RoS}(C-{c_i+1}) \) represents the result of radix sort with regard to the attributes order \( c_{i+2},...,c_{C-1},c_1,\ldots,c_i \). \( \text{RoS}(C-{c_i}) \) can be obtained by two steps. Firstly, \( \text{RoS}(C-{c_i}) \) can be calculated by radix sort in regard to the attributes order \( c_{i+1},...,c_{C-1},c_1,\ldots,c_i \). Next, based on \( \text{RoS}(C-{c_i}) \), we execute radix sort by the attribute \( c_i \). Hence, the result of radix sort by the attributes order \( c_{i+1},c_{i+2},\ldots,c_{C-1},c_1,\ldots,c_i \) is equivalent to combine above two steps of sort operations. Since radix sort is a stable sort, \( \text{RoS}(C-{c_i+1}) \) is identical with \( \text{RoS}(\text{RoS}(C-{c_i}),(c_i)) \). Therefore, Proposition 6 holds. □

According to Proposition 6, we can effectively reduce the number of the radix sort between \( U/(C-{c_i}) \) and \( U/(C-{c_i+1}) \) and eliminate redundant attributes from the selected attribute set to accelerate the efficiency of incremental reduction algorithm.

B. A unified incremental reduction algorithm for variations of the attribute set

In the sub-section, we construct a unified incremental reduction algorithm IARA_\( \Delta \), which can deal with three variations of the attribute set in a uniform approach.

Algorithm 2 The unified incremental reduction algorithm for the variations of the attribute set (IARA_\( \Delta \))

Input: \( S=(U,C \cup D,V,f) \), a reduct \( R \), \( \text{CON}_R(D) \), \( \text{QCON}_R(D) \), the deleted attribute set \( Q \) and the added attribute set \( P \) \((C \cap P)=\Phi\).

Output: a new reduct.

Step 1: if \((Q=\Phi \text{ and } P=\Phi) \land \Delta=1\); let \( R'=R \) and \( G=C \).

Step 2: \( U/P-R \); //addition of the attribute set

Step 3: if \((Q=\Phi \text{ and } P=\Phi) \land \Delta=2\); let \( R'=R\cup Q \) and \( G=C \).

Step 4: \( U/P-Q \); //deletion of the attribute set

Step 5: while \((|\text{CON}_R(D)|<|\text{CON}_F(D)|) \}

Step 6: let \( R'=(a_1,a_2,\ldots,a_{|R'|}) \), \( \text{flag}=0 \); // check redundant attributes

Step 7: for \((1 \leq i \leq |R'|-1) \}

Step 8: Output the new reduct \( R' \).

In the algorithm IARA_\( \Delta \), Step 1 is used to identify which of three variations. Step 3 and Step 4 are to update \( \text{CON}_R(D) \), \( \text{CON}_F(D) \) and \( \text{QCON}_F(D) \), whose time complexity are \( O(|C||U|) \) and \( O(|R'|||U|) \) in the worst case, respectively; Step 5 is used to select the most significance attribute to add the selected attribute subset by the mechanism of acceleration, whose time complexity is \( \sum_{i=1}^{|	ext{CON}_F(D)|} |U|(|G|-i+1) \); In Step 6-7, the selected attributes are checked to remove redundant attributes with the acceleration strategy, whose time complexity is \( O(|C||U|) \) in the worst case. Hence, in the worst case, the total time complexity are \( O(|C||U|) \) and \( O(|R'|||U|) \) respectively.
C. Analyses and comparisons of time complexity

Table 2  Comparisons of the time complexity of obtaining a new reduct for three variations of the attribute set

| Steps                                      | SRACR                                                                 | IARA_Δ                                                                 |
|--------------------------------------------|----------------------------------------------------------------------|----------------------------------------------------------------------|
| Adding the attribute set                   | \(O(C \cup P[U]) + O(C |\text{CORE}||U)| + 2O(C |\text{CORE}||U)| + \) \(20 \times O(C \cup P[U])\)     |
| Deleting the attribute set                 | \(O(C \cup Q[U]) + O(C |\text{CORE}||U)| + 2O(C |\text{CORE}||U)| + \) \(20 \times O(C \cup Q[U])\)     |
| Deleting and adding the attribute sets simultaneously | \(O(C \cup P-Q[U]) + O(C |\text{CORE}||U)| + 2O(C \cup P-Q[U]) + \sum_{i=1}^{Q-C} |U|(|C \cup P - R| - i + 1)\) | \(O(C \cup P-Q[U]) + O(C \cup P-Q[U]) + \sum_{i=1}^{Q-C} |U|(|C \cup P - Q - R| - i + 1)\) |

From Table 2, we can find that the operation of computing core attributes consumes plenty of time in the algorithm SRACR. However, algorithm IARA_Δ avoids computing core attributes and initializes the new attribute reduct from the original attribute reduce. Such that, in the process of selecting the most significance attribute from candidate attributes, the number of loop for selecting candidate attributes of algorithm IARA_Δ is \(|C \cup P-Q-R|\), and is less than that of algorithm SRACR which is \(|C \cup P-Q|\). For checking the redundancy of the selected attributes, the time complexity of algorithm SRACR is \(O(C \cup P-Q[U])\), while the time complexity of IARA_Δ is only \(O(C \cup P-Q[U])\). Hence, it is clear that the time complexity of algorithm IARA_Δ is much smaller than that of algorithm SRACR.

Table 3  The comparison of time complexity for static and incremental algorithms by algorithm execution steps

| Steps                                      | SRACR                                                                 | IARA_Δ                                                                 |
|--------------------------------------------|----------------------------------------------------------------------|----------------------------------------------------------------------|
| Step(1)                                    | \(O(C \cup P-Q[U]) + O(C \cup P-Q[U]) + O(C |\text{CORE}||U)| + \) \(O(C \cup P-Q[U])\)       |
| Step(2)                                    | \(\sum_{i=1}^{Q-C} |U|(|C \cup P - R| - i + 1)\)                                   |
| Step(3)                                    | \(O(C \cup P-Q[U])\)                                                |
| Total                                      | \(O(C \cup P-Q[U]) + 2O(C \cup P-Q[U]) + O(C |\text{CORE}||U)| + \) \(20 \times O(C \cup P-Q[U])\)     |
|                                           | \(\sum_{i=1}^{Q-C} |U|(|C \cup P - R| - i + 1)\)                                   |

To further illustrate the efficiency of algorithm SRACR and IARA_Δ, taking the situation of adding and deleting the attribute sets simultaneously as an example, we compare the time complexity according to execution steps of algorithms. The execution steps of two algorithms can be summarized into three steps shown in Table 3. Step(1) is to compute the core attribute set and CON_c(D) for SRACR, and is to update CON_c(D) and CON_p(D) for incremental algorithm IARA_Δ. Step(2) is to add the attributes into the candidate attribute set; Step(3) is to remove the redundant attributes from the candidate attribute set.

Compared with algorithm SRACR, we can find that algorithm IARA_Δ is accelerated from the following three aspects. The first (i.e., step(1)) is that algorithm SRACR need to re-compute core attributes and conflict region, so its time consumption is \(O(C \cup P-Q[U])\) while algorithm IARA_Δ only need to update conflict region CON_c(D) and CON_p(D), so that its time consumption is \(O(C \cup P-Q[U])\). The second (i.e., step(2)) is that the number of loop for algorithm SRACR is \(|C \cup P-Q|\) for candidate attributes, while that of algorithm IARA_Δ is only \(|C-Q-R|\) on the basis of the original reduce R. Obviously, the number of iteration of algorithm IARA_Δ for selecting candidate attributes is less than that of algorithm SRACR. The third (i.e., step(3)) is that because of using the acceleration strategy for removing redundant attributes, the time complexity of IARA_Δ is only \(O(C \cup P-Q[U])\), while that of SRACR is \(O(C \cup P-Q[U])\). From above analyses, it can further illustrate that the efficiency of algorithm IARA_Δ is better than that of algorithm SRACR.

D. Analyses of illustrations for incremental attribute reduction

In the following, in order to clearly illustrate the incremental attribute reduction algorithm IARA_Δ, three examples are employed to analyze algorithm IARA_Δ for three variations of the attribute set, respectively. Example 2 denotes to add an attribute set; Example 3 represents to delete an attribute set; Example 4 denotes to simultaneously add and delete attribute sets. Specific analyses of three instances are as follows.

Example 2  (Continued from Example 1) \(R = \{a, b, e\}\), we have \(U/C\} = \{\{x_1, x_2, x_3, x_5\}, \{x_1, x_4, x_5\}, \{x_4, x_5\}\} and \(U/R\} = \{\{x_1, x_2, x_3, x_5\}, \{x_1, x_4, x_5\}, \{x_4, x_5\}\} and get \(CON_c(D) = \{x_4, x_5\}\), \(QCON_p(D) = \{x_4, x_5\}\), \(CON_p(D) = \{x_4, x_5\}\).
= \{x_4, x_5, x_6, x_7\} and QCON(D) = \{\{x_4, x_5\}, \{x_6, x_7\}\}.

Suppose \(P = \{f, g\}\) is the added attribute set, where \(g = \{1, 0, 1, 1, 1, 1, 1\}\) and \(h = \{0, 1, 0, 1, 0, 0, 0\}\).

According to Theorem 1, for \(\forall X \in QCON(D)\), we have \(\{x_1, x_2, x_3\}/P = \{x_1, x_2, x_3\}\) and \(\{x_6, x_7\}/P = \{x_6, x_7\}\). And we can get \(U/C \cup P = \{x_4, x_5\}\), \(\{x_1, x_2, x_3\}/d = \{x_1, x_2, x_3\}\), \(\{x_6, x_7\}/d = \{x_6, x_7\}\), \(\{x_4, x_5\}/g = \{x_4, x_5\}\), \(\{x_6, x_7\}/g = \{x_6, x_7\}\)
and \(\{x_4, x_5\} \cup \{x_6, x_7\} = \{x_4, x_5, x_6, x_7\}\).

For the original reduct, \(R\), we have \(\text{CON}(D) = \{x_4, x_5, x_6, x_7\}\) and QCON(D) = \{\{x_4, x_5\}, \{x_6, x_7\}\}.

Obviously, \(\text{CON}(D) \subseteq \text{QCON}(D)\).

For \(C \cup P = \{c, d, f, g\}\), by Theorem 1, we have \(\{x_1, x_2, x_3\}/c = \{x_1, x_2, x_3\}\), \(\{x_1, x_2, x_3\}/d = \{x_1, x_2, x_3\}\), \(\{x_4, x_5\}/f = \{x_4, x_5\}\), \(\{x_4, x_5\}/g = \{x_4, x_5\}\), \(\{x_6, x_7\}/c = \{x_6, x_7\}\), \(\{x_6, x_7\}/d = \{x_6, x_7\}\), \(\{x_6, x_7\}/f = \{x_6, x_7\}\)
and \(\{x_6, x_7\}/g = \{x_6, x_7\}\)

We can get \(\text{Sig}^1_{\text{CON}}(c,R,D) = \text{Sig}^1_{\text{CON}}(d,R,D) = \text{Sig}^1_{\text{CON}}(f,R,D) = 0\) and \(\text{Sig}^1_{\text{CON}}(g,R,D) = 2\); and we select g to add into \(R = \{\text{abc}\}\) and get \(\text{CON}(D) = \{x_4, x_5, x_6, x_7\}\).

And then, we have \(\text{QCON}(D) = \{x_4, x_5, x_6, x_7\}\) and every attribute in \(R\) cannot be removed. Hence, a reduct of the new decision table is \(R = \{\text{abc}\}\).

\[\text{Example 3 (Continued from Example 1) } R = \{\text{abc}\}\]

We have \(U/C = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}\) and \(U/R = \{x_4, x_5, x_6, x_7\}\), and get \(\text{CON}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}\), \(\text{QCON}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}\), \(\text{CON}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}\) and QCON(D) = \{\{x_4, x_5\}, \{x_6, x_7\}\}.

Suppose \(Q = \{d, e\}\) is the deleted attribute set and \(P = \{f, g\}\) is the added attribute set, where \(f = \{1, 0, 1, 1, 1, 1, 1\}\) and \(g = \{0, 1, 0, 1, 0, 0, 0\}\).

We have \(U/(C \cup P = \{x_1, x_2, x_3, x_4, x_5\}, \{x_6, x_7, x_8\}\), \(\{x_1, x_2, x_3\}\). Compared with the previous granularity of the equivalence class, we can find that \(\{x_1, x_2, x_3\}\), \(\{x_6, x_7\}\) and \(\{x_8\}\) are the equivalence classes which are changed, and the granularities of the equivalence classes \(\{x_1, x_2, x_3\}\) and \(\{x_6, x_7\}\) remain unchanged. According to Theorem 3, we have \(\text{CON}(C \cup P, Q) = \{x_1, x_4, x_5\}, \{x_2, x_3, x_4, x_5\}\), \(\{x_6, x_7\}\) and \(\{x_8\}\).

Similarly, we have \(U/(R = \{x_1, x_4, x_5, x_6, x_7, x_8\}\), \(\{x_2, x_3, x_4, x_5\}\), \(\{x_6, x_7\}\) and \(\{x_8\}\) remain unchanged. According to Theorem 2, we have \(\text{CON}(C \cup P, Q) = \{x_1, x_4, x_5\}, \{x_2, x_3, x_4, x_5\}\), \(\{x_6, x_7\}\) and QCON(D) = \{\{x_4, x_5\}, \{x_6, x_7\}\}. Then, \(\text{CON}(C \cup P, Q) < \text{QCON}(D)\).

For \(C \cup P = \{c, d, f, g\}\), we have \(\{x_1, x_2, x_3\}/c = \{x_1, x_2, x_3\}\), \(\{x_1, x_2, x_3\}/d = \{x_1, x_2, x_3\}\), \(\{x_1, x_2, x_3\}/f = \{x_1, x_2, x_3\}\), \(\{x_1, x_2, x_3\}/g = \{x_1, x_2, x_3\}\), \(\{x_4, x_5\}/c = \{x_4, x_5\}\), \(\{x_4, x_5\}/d = \{x_4, x_5\}\), \(\{x_4, x_5\}/f = \{x_4, x_5\}\)
and \(\{x_4, x_5\}/g = \{x_4, x_5\}\).

We can get \(\text{Sig}^1_{\text{CON}}(c,R,Q,D) = 1, \text{Sig}^1_{\text{CON}}(d,R,Q,D) = 0, \text{Sig}^1_{\text{CON}}(g,R,Q,D) = 2\); and we select g to add into \(R = \{\text{abc}\}\) and get \(\text{CON}(D) = \{x_1, x_4, x_5, x_6, x_7\}\).

And then, we have \(\text{CON}(Q,D) = \{\text{CON}(C \cup P, Q)\}\) and every attribute in \(R\) cannot be removed. Hence, a reduct of \(S\) is \(R = \{\text{abc}\}\).

\[\text{V. Experimental analysis}\]

In the section, we carry out a series of experiments to demonstrate the effectiveness and efficiency for the proposed incremental algorithms for the reduct under the variations of the attribute set. In the experiments, data sets are select from UCI (http://archive.ics.uci.edu/ml/datasets.html) and listed in Table 4. All experiments have been performed on a personal computer with Intel i7-4790 3.6 GHz, RAM 4 G and Windows 7. The used programming environment is Microsoft Visual Studio.NET 2005 platform and the programming language is C++.

| ID | Data sets | Samples | Attributes | Classes |
|----|-----------|---------|------------|---------|
| 1  | Audiology  | 200     | 69         | 24      |
| 2  | Dermatology| 366     | 34         | 6       |
| 3  | Soybean    | 307     | 35         | 19      |
| 4  | Sineceon   | 1539    | 256        | 10      |
| 5  | Chess      | 3196    | 36         | 2       |
| 6  | Gene       | 3190    | 60         | 3       |
| 7  | Mushroom   | 8124    | 22         | 2       |
| 8  | Landsett   | 6435    | 36         | 7       |
| 9  | Handswritten| 5620  | 64         | 10      |
| 10 | Tidan2000  | 5822    | 85         | 2       |
| 11 | Connect-4  | 67557   | 42         | 3       |
| 12 | Cosytype   | 581012  | 54         | 7       |

We design three groups of experiments. The first one is that the object set increase dynamically, while the attribute set keeps fixed change; The second one is that the
attribute set changes dynamically, while the object set remains unchanged as the original object set; the third one is the comparisons of reduction results and the classification accuracy of static and incremental algorithms when three variations of the attribute set.

To illustrate the feasibility, in this section, IARA_A denotes the incremental reduction algorithm for adding the attribute set, IARA_D represents the incremental reduction algorithm for deleting the attribute set, and IARA_AD denotes the incremental reduction algorithm for adding and deleting the attribute sets simultaneously. Shu et al. [22] developed the positive region-based attribute reduction algorithms UARA and UARD, in which UARA is for the addition of attribute set and UARD is for the reduction of attribute set. In order to handle the simultaneous addition and deletion of attribute sets, Shu et al. [22] adopted the sequential operation of algorithm UARA and algorithm UARD, called UARA+UARD (UAR_AD for short).

A. Performance comparisons between static and incremental algorithms with variation of the attribute set

In this subsection, we compare the performance of static and incremental algorithms when the variance ratio of the attribute set keeps unchanged and the number of the object increases on 6 data sets which are described in Table 4. In each of data sets, we select 50% objects from the original data sets as the basis data sets, and 10%, 20%, ..., 100% objects are take out from the remaining 50% objects as the incremental objects. In what follows, we perform dynamic algorithm and static algorithm to compare the computational time.

a. The comparison of static and incremental algorithms when adding the fixed attribute set

We carry out algorithms SRACR, UARA and IARA_A on 12 data sets with the objects increasing but the adding attribute set unchanging. For each data set, 2/3 of the attributes of the original attribute set are the initial attribute set, and the remaining attributes are the added attribute set. With the increasing size of the data set, the trends of comparative results are depicted in Fig. 2. The x-coordinate pertains to the scale of the test set from 5% to 50% (that is, the value of x-axis represents the proportion of the added objects to the whole universe), whereas the y-coordinate pertains to the computing time of algorithms. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UARA and the star line means the consuming time of algorithm IARA_A.

![Fig. 2 Comparison of computation time for SRACR, UARA and IARA_A when adding objects](image-url)
As shown in each subfigures of Fig. 2, it is clear that the consuming time of two algorithms SRACR, UARA and IARA_A rises with the increase of the number of objects. However, with the increase of data set, we can find that the performance of UARA is better than SRACR and the calculation efficiency of IARA_A is much better than SRACR and UARA. Furthermore, the larger the data set increase, the performance advantage of algorithm IARA_A becomes more obvious. The main reason is that algorithm SRACR needs a lot of repeated computation to get new reduct, and algorithms UARA and IARA_A are the incremental algorithm, which avoids recalculating and uses the previous results. In addition, algorithm IARA_A uses the acceleration strategies to compute conflict region and obtain the reduct, such that the computational time of IARA_A is greatly reduced and is better than that of UARA. Therefore, with the increase of the number of objects and fixed incremental attributes, computational efficiency of algorithm IARA_A outperforms that of algorithms SRACR and UARA.

b. The comparison of static and incremental algorithms when deleting the fixed attribute set

We compare performance of algorithms SRACR, UARD and IARA_D on 12 data sets shown in Table 4 with the objects increasing but the deleting attribute set unchanging. For each data set, the original attribute set is the initial attribute set, and 1/3 of the attributes are selected as the deleted attribute set. The comparative results are depicted in Fig. 3 with the increasing size of the data set. The x-coordinate is the scale of the test set from 5% to 50% (that is, the value of x-axis represents the proportion of the added objects to the whole universe), whereas the y-coordinate is the computing time of algorithms. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UARD and the star line means the consuming time of algorithm IARA_D.

From subfigures of Fig. 3, we can find that the consuming time of algorithms SRACR, UARD and
IARA_D grows monotonically with the number of objects increasing. However, the calculation time of algorithms UARD and IARA_D are much lower than that of algorithm SRACR, and the performance of algorithm UARD is better than that of SRACR. The main reason is that algorithms IARA_D and UARD are the incremental algorithms and avoid recalculating by using the previous results. Additionally, IARA_D adopts the acceleration strategies to obtain conflict region and the reduct, such that the computational time of IARA_D is greatly reduced and is superior to that of UARD. Therefore, with the increase of the number of objects and fixed diminished attributes, we draw a conclusion that algorithm IARA_D can greatly reduce consuming time to obtain a reduct compared with algorithms UARD and SRACR.

c. The comparison of static and incremental algorithms when adding and deleting the fixed attribute sets simultaneously

When the adding attributes and the deleting attribute are fixed respectively and the number of objects increases in turn, we compare algorithms SRACR, UARD and IARA_D on 12 data sets shown in Table 4. We select 2/3 of the original attribute set as the initial attribute set, the remaining attributes as the added attribute set, and 1/2 of the initial attribute set as the deleted attribute set. The comparison results of algorithms under these data sets are shown in Fig. 4. The x-coordinate denotes the scale of the test set from 5% to 50% (that is, the value of x-axis represents the proportion of the added objects to the whole universe), whereas the y-coordinate pertains the computing time of algorithms. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UARD and the star line means the consuming time of algorithm IARA_AD.

![Fig. 4 Comparison of computation time for SRACR, UAR_A and IARA_AD when adding objects](image)

According to subfigures of Fig. 4, it is obvious that the consuming time of algorithms SRACR, UAR_A and IARA_AD rises monotonically with the increase of the number of objects. However, compared with algorithms SRACR and UAR_A, the calculation time of algorithm IARA_AD increases very slowly, and the performance of algorithm IARA_AD is obviously better than that of algorithms SRACR and UAR_AD when the data sets...
become larger. The main reason is that when the data set changes dynamically, algorithm SRACR needs to recalculate attribute reduction. Although both algorithms UAR_AD and IARA adopt incremental methods to update the reduct and avoid recalculation, IARA also improves efficiency through acceleration strategy. Therefore, we draw a conclusion that IARA_AD can greatly reduce calculation time to obtain a reduct when adding and deleting specific attributes from the original attribute set simultaneously.

B. Performance comparisons between static and incremental algorithms under different variance ratios of the attribute set

In this sub-section, we investigate the performance of static and incremental algorithms under the attribute set varying but the size of the object set unchanging on 6 data sets. The experiments are divided into three aspects: (1) adding the attribute set, (2) deleting the attribute set, (3) adding and deleting the attribute sets simultaneously.

a. The performance comparison between static and incremental algorithms when adding different the attribute set

To verify the performance of algorithms SRACR, UARA and IARA_A when adding different number of attributes, from 12 data sets shown in Table 4, we select 50% condition attributes as the basic attribute set and the rest attribute set is roughly divided into 8 parts of equal size. The remaining attributes of eight parts are successively added into the base attribute set. The first part considered as the first adding attributes, the combination of the 1st adding attributes and the second part is viewed as 2nd adding attributes, the combination of the second adding attributes and the third part is regarded as 3rd adding attributes, ..., the combination of all eight parts is treated as 8th adding attributes.

With addition of the attribute set, we carry out algorithms SRACR, UARA and IARA_A to obtain the new reduct respectively, and the results of computational time are exhibited in Fig. 5. In each sub-figures of Fig. 5, x-coordinate denotes the ordinal number of adding attributes and y-coordinate is the computational time of reduction algorithms. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UARA and the star line means the consuming time of algorithm IARA_A.

Fig. 5 Comparison of computation time for SRACR, UARA and IARA_A when adding attributes
As shown in each sub-figures of Fig. 5, it shows that although the computation time of algorithms SRACR, UARA and IARA_A increases with increment of attributes monotonically, the calculation time of algorithms UARA and IARA_A is much less than that of SRACR and the performance of IARA_A is better than that of UARA. The reason depicts the fact that IARA_A and UARA avoid recalculation and IARA_A also uses acceleration strategies. In addition, we can find that the curve shapes of algorithm SRACR approximately appear quadratic, while the curve shapes of UARA and IARA_A are approximately linear. Therefore, the results of the experiment shows that algorithm IARA_A outperforms algorithm UARA and SRACR to obtain the new reduct when attributes are added into the decision systems.

b. The performance comparison between static and incremental algorithms when deleting different the attribute set

To verify the performance of algorithms SRACR, UARD and IARA_D when deleting different number of attributes, from 6 data sets described in Table 4, 100% condition attributes of data set is taken as the basic attribute set and construct eight groups of deleted attribute sets, whose numbers of deleting attributes is 1/16, 2/16, ..., 8/16 of the original attribute set respectively. When deleting the attribute set, we carry out algorithms SRACR and IARA_D to obtain the new reduct respectively, and the results of computational time are exhibited in Fig. 6. In each sub-figure of Fig. 6, x-coordinate pertains to the ordinal number of deleting attributes and y-coordinate is the consuming time of algorithms. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UARD and the star line means the consuming time of algorithm IARA_D.

From each sub-figures of Fig. 6, for each data set, it can be seen that the computational time of algorithms SRACR, UARD and IARA_D monotonically decrease with the increase of the number of deleted attributes. However, the computational time of algorithm IARA_D is much less than that of algorithms UARD and SRACR a. The reason should be owed to the fact that algorithms UARD and IARA_D avoid re-computing on the basis of the previous
results and IARA_D also uses acceleration strategies, so that the efficiency of IARA_D is improved. In addition, we can observe that the curve shapes of algorithm SRACR approximately appear quadratic, while the curve shapes of algorithms UARD and IARA_D is approximately linear. Therefore, we can draw a conclusion that algorithm IARA_D can greatly reduce consuming time to obtain a new reduct when deleting the attribute set from the original attribute set.

c. The performance comparison between static and incremental algorithms when adding and deleting different the attribute set simultaneously

To test the performance of algorithm IARA_AD when adding and deleting the attribute set simultaneously, for 12 data sets shown in Table 4, we select 50% attributes of the original data set as the basic attribute set and construct eight groups of added and deleted attribute sets, the number of deleting of 6 data sets is 1/8, 2/8,..., 8/8 of the basic attribute set, the number of added attributes of 6 data sets is 1/8, 2/8,..., 8/8 of the remaining attributes.

When deleting and adding the attribute sets simultaneously, the reduct can be calculated by algorithms SRACR and IARA_AD, the experimental results of computational time versus the size of the incremental attribute set are shown in Fig. 7. In each sub-figure of Fig. 7, x-coordinate pertains to the ordinal number of adding and deleting attributes and y-coordinate is the computational time of algorithms SRACR, UAR-AD and IARA-AD. The hollow triangle line represents the consuming time of algorithm SRACR, the hollow circle line denotes the consuming time of algorithm UAR_AD and the star line means the consuming time of algorithm IARA_AD.

From each sub-figure of Fig. 7, for each data set, it can be seen that the calculation time of algorithms SRACR and UAR_AD monotonically decrease with the increase of the number of adding and deleting attributes, and the change of computation time of algorithm IARA_AD is relatively gentle. The consuming time of algorithm IARA_AD is much less than that of algorithms SRACR and UAR_AD. The main reason is that algorithm IARA_AD
C. Comparisons of running time, reduction results and classification accuracies on different data sets

In this subsection, we compare running time, reduction results and classification accuracies on different data sets when the attribute set keeps the specific variance ratio on 12 data sets shown in Table 4.

a. Comparisons of running time and reduction results

(1) Comparisons of computational time and reduction results for adding the attribute set

For the case of adding the attribute set, we select 50% attributes from the original condition attribute set as the basis attribute set, and the remaining 50% condition attributes are taken out as the added attribute set. In the experiment, we carry out algorithms SRACR, UARA and IARA_A to obtain the reduct of each data set. Table 5 shows the results of reduction, the number of selected attributes (RL) and consuming time of algorithms in each data set.

From Table 5, we notice that results of the reduct and RL obtained by incremental algorithms UARA and IARA_A are relatively close or identical to that of the static algorithm SRACR. However, compared with algorithm SRACR, the computational time of algorithm IARA_A is much lower than that of algorithms SRACR and UARA. Specifically, the reduct average length of algorithms SRACR, UARA and IARA_A are 14.17, 15.58 and 15.33 respectively, and the average consuming time of algorithms SRACR, UARA and IARA_A are 25.1962 s, 7.1476 s and 1.7548 s respectively.

(2) Comparisons of computational time and reduction results for deleting the attribute set

For the case of deleting the attribute set, 100% condition attributes of the data set take out as the basic attribute set and 50% condition attributes are selected as deleted attributes. We use algorithms SRACR, UARD and IARA_D to calculate the reduct of each data set. The results of reduction, the number of selected attributes (RL) and consuming time of algorithms are depicted in Table 6.
From Table 6, it can be seen that the reduct and RL by incremental algorithms UARA and IARA_D are relatively close or identical to that of the static algorithm SRACR. However, compared with algorithms SRACR and UARA, algorithm IARA_D can update the reduct in a much shorter time. Specifically, the reduct average length of algorithms SRACR, UARA and IARA_A are 11.33, 17.75 and 11.58 s respectively, and the average consuming time of algorithms SRACR, UARA and IARA_A are 11.33, 17.75 and 11.58 s respectively, and the average consuming time of algorithms SRACR, UARA and IARA_A are 6.4933 s, 1.809 s and 0.4450 s respectively.

To compare performance of different reduction algorithms, based on ten-fold cross-validations, we select 12 data sets of Table 4 and use C4.5, NB and KNN classifiers to train data and analyze classification accuracies by original data and the reduced data by static algorithm, algorithms of [18] and the proposed algorithms for three variations of the attribute set. Table 8 - Table 10 show the results of the classification accuracies in term of the attributes and 'Ave' denotes the average classification accuracy of reducts by different algorithms.

(3) Comparisons of computational time and reduction results for adding and deleting the attribute sets simultaneously

When adding and deleting the attribute sets simultaneously, 70% attributes of data set are taken out as the original attribute set, the remaining 30% condition attributes are taken out as the new added attributes and 30% condition attributes are selected as the deleted attributes. We use algorithms SRACR, UAR_AD and IARA_AD to compute the reduct of each data set. The results of reduction, the number of selected attributes (RL) and consuming time of algorithms in each data set are shown in Table 7.

It can be observed from Table 7 that the reduct and RL by incremental algorithms UAR_AD and IARA_AD are relatively close or identical to that of the static algorithm SRACR. However, compared with algorithms SRACR and UAR_AD, algorithm IARA_AD can update the new reduct in a much shorter time. Specifically, the reduct average length of algorithms SRACR, UARA and IARA_A are 13.33, 13.58 and 13.42 respectively, and the average consuming time of algorithms SRACR, UARA and IARA_A are 11.3103 s, 3.7665 s and 1.1503 s respectively.

### Table 7 Comparisons of running time and reduction results for adding and deleting attributes simultaneously

| Data sets     | Audiology | Dermatology | Soybean | Simeion | Chess          | Gene         | Mushroom | Landsat | Handwritten | Ticdata2000 | Connect-4 | Covtype | AVE    |
|---------------|-----------|-------------|---------|---------|----------------|--------------|----------|---------|-------------|------------|-----------|---------|--------|
| UAR_AD        | 0.1593    | 0.0117      | 0.0349  | 0.0248  | 0.0179         | 0.2072       | 0.0122   | 0.5478  | 0.1441      | 0.0685     | 0.0218    | 0.1324  | 6.4933 |
| IARA_AD       | 0.0220    | 0.0086      | 0.0179  | 0.3756  | 0.6614         | 0.0618       | 0.0122   | 0.4578  | 0.0513      | 0.3924     | 0.0281    | 0.5294  | 6.4933 |
| SRACR         | 0.5246    | 0.0050      | 0.0197  | 3.7654  | 0.8881         | 0.2417       | 0.1593   | 0.2378  | 0.5411      | 3.7654     | 0.3921    | 0.5294  | 6.4933 |
| Ave           | 0.2417    | 0.0122      | 0.0197  | 3.7654  | 0.8881         | 0.2417       | 0.1593   | 0.2378  | 0.5411      | 3.7654     | 0.3921    | 0.5294  | 6.4933 |

---

**b. Comparison of classification accuracies**

To compare performance of different reduction algorithms, based on ten-fold cross-validations, we select 12 data sets of Table 4 and use C4.5, NB and KNN algorithms to train data and analyze classification accuracies by original data and the reduced data by static algorithm, algorithms of [18] and the proposed algorithms for three variations of the attribute set. Table 8 - Table 10 show the results of the classification accuracies in term of the attributes and 'Ave' denotes the average classification accuracy of reducts by different algorithms.

---

**This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/**
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3128879, IEEE Access

As shown in Table 8 - Table 10, we can see that, in most cases, the classification accuracies of the reduced data sets are improved compared with the unreduced data sets. The classification accuracy of the reduct selected by incremental algorithms (algorithms of [22] and the proposed algorithms) is very close or even identical to that of the static algorithm SRACR on 12 data sets. In addition, we can find that there are some differences in the comparison results of the average classification accuracy of three algorithms. Compared with the other two incremental algorithms, SRACR algorithm has the highest average classification accuracy. However, the average classification accuracy difference of the three reduction algorithms under the three classifiers is very small. Hence, the experimental results demonstrate that the proposed algorithms IARA_A, IARA_D and IARA_AD can get better classification performance by removing redundant or irrelevant attributes, and can find effective reducts without losing classification performance.

### VI. Conclusions

In real-life applications, the attributes of decision system will change rapidly with time. How to efficiently update the reduct becomes an important task in knowledge discovery, data dimming and other fields. In this paper, we introduce the approach for computing attribute reduction based on conflict region. And then, we disuses incremental mechanisms to update conflict region for three variations of the attribute set, and a unified incremental reduction algorithm IARA_A is developed based on conflict region; examples are used to illustrate effectiveness of the incremental algorithm. Finally, we construct a series of experiments to evaluate the efficiency and effectiveness of the proposed algorithm. The experimental results showed that, for three variations of the attribute set, algorithm...
IARA_A can select a new reduct in much shorter time without losing the classification performance compared to the non-incremental reduction algorithm.

Reference

[1] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341–356.
[2] J. Qian, P. Ly-X. Yue, C. Liu C, et al., Hierarchical attribute reduction algorithms for big data using mapreduce. Knowledge-Based Systems 73 (2015) 18–31.
[3] J.S. Mi, W.Z. Wu, W.X. Zhang, Approaches to knowledge reduction based on variable precision rough set model, Information Science 159 (2004) 255–272.
[4] C.Z. Wang, Y. Huang, M.S. Shao, et al. Fuzzy rough set-based attribute reduction using distance measures. Knowledge Based Systems 164 (2019) 205-212.
[5] C.Z. Wang, Y. Huang, W.P. Ding, Z.H. Cao, Attribute reduction with fuzzy rough set based features reduction, Information Sciences, 549 (2021) 68-86.
[6] C.Z. Wang, Y. Wang, M.W. Shao, Y.H. Qian, D.G. Chen, Fuzzy rough attribute reduction for categorical data, IEEE Transactions on Fuzzy Systems, 28(5) (2020) 818-830.
[7] H.M. Chen, T.R. Li, X. Fan , et al., Feature selection for imbalanced data based on neighborhood rough sets. Information Sciences, 483 (2019) 1-20.
[8] N. N. Nguyen, W. Sartra, An efficient stripped cover-based accelerator for reduction of attributes in incomplete decision tables, Expert Systems With Applications 143 (2020) 113076.
[9] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowinski (Ed.), Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, Academic, Dordrecht, 1992, pp. 331–362.
[10] F. Hu, G.Y. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets, in: Proceeding of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Regina, Canada, in: LNCS, vol. 3641, 2005, pp. 185–193.
[11] X. Yang, T.R. Li, D. Liu, et al., A unified framework of dynamic three-way probabilistic rough sets, Information Sciences 420 (2017) 126–147.
[12] X. Yang, T.R. Li, H. Fujita, D. Liu, Y.Y. Yao, A unified model of sequential three-way decisions and multilevel incremental processing, Knowledge-Based Systems 134 (2017) 172–188.
[13] Y.T. Xu, L.S. Wang, R.Y. Zhang, A dynamic attribute reduction algorithm based on 0-1 integer programming, Knowledge-Based Systems 24 (8) (2011) 1341–1347.
[14] M. Yang, An incremental updating algorithm for attribute reduction based on improved discernibility matrix, Chinese Journal of Computer 30 (5) (2007) 815–822.
[15] J.Y. Liang, F. Wang, C.Y. Dang, Y.H. Qian, A Group incremental approach to feature selection applying rough set technique, IEEE Transaction on Knowledge and Data Engineering 26 (2) (2014) 294–308.
[16] W. Wei, P. Song , J.Y. Liang, X.Y. Wu, Accelerating incremental attribute reduction algorithm by compacting a decision table, International Journal of Machine Learning and Cybernetics, 10 (9) (2019) 2355-2373.
[17] W.H. Shu, W.B. Qian, Y.H. Xie, Incremental approaches for feature selection from dynamic data with the variation of multiple objects, Knowledge-Based Systems 163 (2019) 320–331.
[18] W.H. Shu, W.B. Qian, An incremental approach to attribute reduction from dynamic incomplete decision systems in rough set theory, Data & Knowledge Engineering 100 (2015) 116–132.
[19] Y.G. Jing, T.R. Li, F. Fujita, Z. Yu, B. Wang, An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view, Information Sciences 411 (2017) 23–38.
[20] Y.Y. Yang, D.G. Chen, H. Wang, X.Z. Wang, Incremental perspective for feature selection based on fuzzy rough sets, IEEE Transactions on Fuzzy Systems 26 (3) (2018) 1257-1273.
[21] B.B. Sang, H.M. Chen, L.Y. Yang, D.P. Zhou, T.R. Li, W.H. Xu, Incremental attribute reduction approaches for ordered data with time-evolving objects, Knowledge-Based Systems 212 (2021) 106583.
[22] W.H. Shu, H. Shen, Updating attribute reduction in incomplete decision systems with the variation of attribute set, International Journal of Approximate Reasoning 55 (2014) 867–884.
[23] A.P. Zeng, T.R. Li, D. Liu, J.B. Zhang, H.M. Chen, A fuzzy rough set approach for incremental feature selection on hybrid information systems, Fuzzy Sets and Systems 258 (1) (2015) 39-40.
[24] F. Wang, J.Y. Liang, Y.H. Qian, Attribute reduction: A dimension incremental strategy, Knowledge-Based Systems 39 (2013) 95–108.
[25] Y. Jing, T. Li, J. Huang, et al., An incremental attribute reduction approach based on knowledge granularity under the attribute generalization, International Journal of Approximate Reasoning 76 (2016) 80–95.
[26] D.G. Chen, J.L. Dong, J.S. Mi, Incremental mechanism of attribute reduction based on discernible relations for dynamically increasing attribute, Soft Computing 24 (1) (2020) 321-332.
[27] F. Wang, J.Y. Liang, C.Y. Dang, Attribute reduction for dynamic data sets, Applied Soft Computing 13 (1) (2013) 676–689.
[28] W.H. Shu, H. Shen, Incremental feature selection based on rough set in dynamic incomplete data, Pattern Recognition 47 (2014) 3890–3906.
[29] X.J. Xie, X.L. Qian, A novel incremental attribute reduction approach for dynamic incomplete decision systems, International Journal of Approximate Reasoning 93 (2018) 443–462.
[30] W. Wei, X.Y. Wu, J.Y. Liang, J.B. Cai, Y.J. Sun, Discernibility matrix based incremental attribute reduction for dynamic data, Knowledge-Based Systems 140 (2018) 142–157.
[31] Y.G. Jing, T.R. Li, J.F. Huang, H.M. Chen, A group incremental reduction algorithm with varying data values, International Journal of Intelligent Systems 32 (9) (2016) 900–925.
[32] Y.G. Jing, T.R. Li, H. Fujita, et al., An incremental attribute reduction method for dynamic data mining, Information Sciences 465 (2018) 202–218.
[33] L.J. Dong, D.G. Chen, Incremental attribute reduction with rough set for dynamic datasets with simultaneously increasing samples and attributes, International Journal of Machine Learning and Cybernetics 11 (6) (2020) 1339-1355.
[34] H. Ge, L.S. Li, C.J. Yang, J. Ding, Incremental reduction algorithm with acceleration strategy based on conflict region,
Artificial Intelligence. Review 51 (4) (2019) 507-536.

[35] H. Ge, L.S. Li, Y. Xu, et al., Bidirectional heuristic attribute reduction based on conflict region, Soft Computing 19 (7) (2015) 1973–1986.

[36] H. Ge, L.S. Li, Y Xu, C.Z. Yang, Quick general reduction algorithms for inconsistent decision tables, International Journal of Approximate Reasoning 82 (2017) 56-80.

[37] H.M. Chen, T.R. Li, C. Luo, S.J. Horng, G.Y. Wang, A rough set-based method for updating decision rules on attribute values’ coarsening and refining, IEEE Transactions on Knowledge and Data Engineering 26 (12) (2014) 2886–2899.

[38] X.H. Hu, N. Cercone, Learning in relational databases a rough set approach, International Journal of Computational Intelligence 11 (2) (1995) 323–338.

[39] Q.H. Hu, D.R. Yu, Entropies of fuzzy indiscernibility relation and its operations, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 12 (5) (2004) 575–589.

[40] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy rough model and information granulation, Pattern Recognition 40 (2007) 3509–3521.

[41] C.C. Huang, T.L. Tseng, Y.N. Fan, Y.N, Fan, C.H. Hsu, Alternative rule induction methods based on incremental object using rough set theory, Applied Soft Computing 13 (1) (2013) 372–389.

[42] R. Jensen, Q. Shen, Fuzzy-rough attribute reduction with application to web categorization, Fuzzy Sets and Systems 141 (2004) 469–485.

[43] R. Jensen, Q. Shen, Fuzzy-rough sets assisted attribute selection, IEEE Transactions on Fuzzy Systems 15 (1) (2007) 73–89.

[44] B. Jerzy, R. Slowinski, Incremental induction of decision rules from dominance-based rough approximations, Electronic Notes in Theoretical Computer Science 84 (4) (2003) 40–51.

[45] F. Li, Y.Q. Yin, Approaches to knowledge reduction of covering decision systems based on information theory, Information Sciences 179 (11) (2009) 1694–1704.

[46] T.R. Li, D. Ruan, W. Geert, J. Song, Y. Xu, A rough sets based characteristic relation approach for dynamic attribute generalization in data mining, Knowledge-Based Systems 20 (5) (2007) 485–494.

[47] J.Y. Liang, F. Wang, C.Y. Dang, Y.H. Qian, A group incremental approach to feature selection applying rough set technique, IEEE Transactions Knowledge and Data Engineering . 26 (2) (2014) 294–308.

[48] J.Y. Liang, J.R. Mi, W. Wei, F. Wang, An accelerator for attribute reduction based on perspective of objects and attributes, Knowledge-Based Systems 44 (2013) 90–100.

[49] D. Liu, T.R. Li, D. Ruan, J.B. Zhang, Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, Journal of Global Optimization 51 (2) (2011) 325–344.

[50] S.H. Liu, Q.J. Seng, B. Wu, Z.Z. Shi, F. Hu, Research on efficient algorithms for Rough set methods, Chinese Journal of Computers 26 (5) (2003) 524–529.

[51] C. Luo, T.R. Li, Y.Y. Yao, Dynamic probabilistic rough sets with incomplete data, Information Sciences 417 (2017) 39–54.

[52] D.Q. Miao, Y. Zhao, Y.Y. Yao, H.X. Li, F.F. Xu, Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model, Information Sciences 179 (2009) 4140–4150.

[53] D.Q. Miao, G. R. Hu. A heuristic algorithm for reduction of knowledge, Journal of Computer Research and Development 36 (6) (1999) 681–684.

[54] E.C.C. Tsang, D.G. Chen, D.S. Yeung, X.Z. Wang, J.W.T. Lee, Attribute reduction using fuzzy rough sets, IEEE Transactions on Fuzzy Systems 16 (5) (2008) 1130–1141.

[55] Z.Y. Xu, Z.P. Liu, B.R. Yang, W. Song, A quick attribute reduction algorithm with complexity of max(\(|C||U|\),O(|C|^2|U|/C)), Chinese Journal of Computers 29 (3) (2006) 611–615.

[56] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, Information Sciences 179 (2009) 867–882.

[57] J.B. Zhang, T.R. Li, H.M. Chen, Composite rough sets for dynamic data mining, Information Sciences 257 (4) (2014) 81–100.

[58] Z. Zheng, G.Y. Wang, Y. Wu, A Rough set and rule tree based incremental knowledge acquisition algorithm, Fundamenta Informaticae 59 (2-3) (2004) 299–313.

[59] J. Qian, D.Q. Miao, Z.H. Zhang, W. Li Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. International Journal of Approximate Reasoning 52 (2) (2011) 212–230.