Artif Intell Rev (2018) 51:507–536
https://doi.org/10.1007/s10462-017-9570-6

Incremental reduction algorithm with acceleration strategy based on conflict region

Ge Hao[1,2] · Li Longshu[2] · Yang Chuanjian[3] · Ding Jian[1]

Published online: 28 July 2017

Abstract Attribute reduction plays a vital role in many areas of data mining and knowledge discovery. In the real world, several data sets may vary dynamically and many incremental reduction algorithms have been proposed to update reduct. Further improvement of the performance of the incremental reduction approach is an important task that can help to increase the efficiency of knowledge discovery in dynamic data systems. This paper researches incremental reduction algorithms via an acceleration strategy to compute new reduct based on conflict region. We firstly introduce the concepts and propositions of the conflict region and give a static reduction algorithm based on the conflict region. Consequently, incremental mechanisms based on the conflict region and an acceleration strategy for reduction are discussed. Then, two incremental reduction algorithms for updating new reduct when one single object and multi-objects are added to decision systems are developed. Finally, experiments on different data sets from UCI show the effectiveness and efficiency of the proposed algorithms in decision systems with the addition of objects.

Keywords Rough set · Incremental reduction · Dynamic data set · Conflict region

1 Introduction

Feature selection (also called attribute reduction) is one of the important research contents in data mining, machine learning, intelligent data analysis and pattern recognition. In the real world, information expands quickly and plenty of features are stored in datasets. Some of the

✉ Ge Hao
togehao@126.com

[1] School of Electronic and Electrical Engineering, Chuzhou University, Chuzhou 239000, China

[2] Key Laboratory of Computation Intelligence and Signal Processing of Education Ministry, Anhui University, Hefei 230601, China

[3] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

features in high-dimensional datasets are irrelevant or redundant, which typically deteriorates the performance of knowledge discovery algorithms.

Rough set theory is a powerful mathematical tool introduced by Pawlak (1982) to address imprecise, incomplete or vague information. Many researchers have contributed to its development and applications. Attribute reduction is one of the fundamental aspects of rough set theory, which can find a subset of attributes (denoted as the reduct) that provides the same descriptive, discernibility or classification ability as the original condition attribute set. In the last two decades, many attribute reduction techniques have been developed. These techniques mainly include the discernibility matrix-based reduction approaches (Skowron and Rauszer 1992; Wei et al. 2013; Yao and Zhao 2009; Zhang et al. 2003), the discernibility and indiscernibility-based reduction approaches (Li and Yao 2010; Qian et al. 2011; Susmaga and Słowiński 2015; Teng et al. 2010; Zhao et al. 2007), the positive region-based reduction approaches (Liu et al. 2009; Qian et al. 2010; Shen and Jensen 2004; Xu et al. 2006), the information entropy and its variation entropy-based reduction approaches (Hu et al. 2007; Liang and Xu 2002; Miao and Hu 1999; Qian et al. 2009), etc. These studies have offered interesting insights into attribute reduction and have been successfully applied to decision systems. However, most of the above reduction approaches can only be suitable for static data sets. When data sets vary with time, these algorithms must be re-implemented to obtain new reduct, which consumes a huge amount of computational time and space. It is very time-consuming or even infeasible to run repeatedly the static reduction algorithms. Hence, these static reduction algorithms are very inefficient when dealing with dynamic data tables.

Many scholars have noticed the shortcoming of the static reduction methods for dynamic data sets. Thereupon, based on rough set theory, some incremental learning approaches, instead of static reduction approaches, have been applied to obtain new reduct for dynamic systems. The incremental attribute reduction from dynamic decision systems can be broadly classified into three classes. The first class is the study on updating attribute reduction caused by variation of the object set (Chen et al. 2013; Fan et al. 2009; Huang et al. 2013; Jing et al. 2016a), the second class is the study on updating attribute reduction caused by variation of the attribute set (Jing et al. 2016b; Li et al. 2007, 2013; Zhang et al. 2012), and the third class is the study on updating attribute reduction caused by variation of the attribute values (Chen et al. 2010, 2012; Huang et al. 2016; Li and Li 2015).

Many papers concentrate on attribute reduction for dynamic variation of the object set in decision systems. Liu (1999) proposed an incremental reduction algorithm for the minimal reduct. This algorithm can only be applied to information systems without decision systems. Hu et al. (2005) presented a positive region-based attribute reduction algorithm to update the reduct via an incremental technique. Yang (2007) provided an incremental updating reduction algorithm based on an improved discernibility matrix in decision systems. However, it only considers the situation in which only a single object is entered into the systems. Xu et al. (2011) provided a dynamic attribute reduction algorithm based on integer programming in the decision systems. They considered multiple objects being entered into the system, but the entering of multiple objects is just seen as the cumulative change of a single object, which is inefficient when an object set is added into the decision systems. Regarding multiple objects being added into the decision systems, Liang et al. (2014) introduced incremental mechanisms for three representative information entropies and presented a group incremental approach to feature selection based on three information entropies. Wang et al. (2013a) developed incremental feature selection algorithms for decision systems that dynamically increase the attribute set based on three representative entropies. Moreover, Wang et al. (2013b) proposed attribute reduction algorithms for data sets with dynamically varying data values based on three representative entropies. Lang et al. (2014) presented an incremental approach to attribute reduction of dynamic set-valued information systems with respect to three aspects: variations of the attribute set, immigration and emigration of objects, and alterations of the attribute value. Shu and Shen (2014a) proposed incremental reduction algorithms for when an attribute set is added into and deleted from incomplete decision systems. In addition, Shu and Shen (2014b) presented efficient incremental feature selection algorithms for a single object and multiple objects with varying feature values, respectively. Zeng et al. (2015) presented a new hybrid distance for many types of data in hybrid information systems and researched updating mechanisms for reduction with variation of the attribute set and proposed two incremental feature selection algorithms with fuzzy rough set approaches.

From the above, the incremental technique improves the efficiency of reduction algorithms with dynamic systems. Can we further improve the performance of incremental reduction algorithms? This motivates us to develop efficient incremental reduction algorithms. In our previous work, we introduced the conflict region and proposed bidirectional heuristic reduction based on the conflict region (Ge et al. 2015). Moreover, it has been observed in many fields that the case of adding objects into decision systems is common. In this paper, we continue with this work and improve the approach to research incremental reduction algorithms with an acceleration strategy based on the conflict region for adding a single object and multiple objects. We firstly introduce the concept of the conflict region and the corresponding properties and give a static attribute reduction algorithm based on the conflict region. Then, we discuss the incremental mechanisms for adding a single object and multiple objects into decision systems based on the conflict region. Two incremental feature selection algorithms with an acceleration strategy for adding a single object and multiple objects are put forward. Finally, the performances of the incremental reduction algorithms are evaluated using several UCI datasets.

The contributions of this paper are as follows: (1) two incremental attribute reduction algorithms (IRACR-M and IRACR-S), used for adding multiple objects into decision systems rather than performing the static reduction algorithm repeatedly, are presented; (2) an acceleration strategy is presented by reducing the sort times to further increase the efficiency of the incremental reduction algorithms; and (3) the efficiency and effectiveness of the proposed algorithms are demonstrated on different UCI data sets.

The structure of the remainder of this paper is as follows. Section 2 briefly reviews preliminary notions in the rough set theory. Section 3 introduces the concept of the conflict region and gives a reduction algorithm based on the conflict region. Based on the conflict region, Sect. 4 designs the incremental reduction algorithm for adding a single object with the acceleration strategy and the incremental reduction algorithms for adding an object set with the acceleration strategy. Section 5 constructs a series of comparative experiments to evaluate the effectiveness and efficiency of our proposed incremental reduction algorithms. Finally, Sect. 6 gives conclusions that are drawn from this study.

2 Preliminaries of rough sets

In this section, we will review several basic concepts in rough set theory and the reduct of the positive region in the decision table.

2.1 Basic concepts of rough sets

Let $I = (U, A, V, f)$ be an information system, where $U = \{x_1, x_2, ..., x_n\}$ is a non-empty finite set of objects called the universe; $A = \{a_1, a_2, ..., a_m\}$ is a non-empty and finite set of attributes; $V = \{V_a | \forall a \in A\}$ is a set of a value domain of attributes, where V_a is a value set of the attribute a; $f : U \times A \to V$ is called an information function ($f(x, a) \in V_a$ for each $a \in A$).

For any $R \subseteq C$, the indiscernibility relation is defined as:

$$\mathrm{IND}(R) = \{(x, y) \in U \times U | f(x, a) = f(y, a), \quad \forall a \in R\}$$

That is, x and y are indiscernible with respect to R if and only if they have the same values for all attributes in R. The relation IND(R) is reflexive, symmetric and transitive and hence is an equivalence relation. $U/\mathrm{IND}(R) = \{[x]_R | x \in U\}$(just as U/R) indicates the partitions of U induced by R, which denotes the equivalence class determined by x with respect to R, i.e., $[x]_R = \{y | y \in U, (x, y) \in \mathrm{IND}(R)\}$.

For Pawlak's rough set model, given an equivalence relation R on the universe U and a subset $X \subseteq U$, we have the lower approximation and upper approximation of X as follows

$$R_-(X) = \bigcup\{x | [x]_R \subseteq X\},$$
$$R^-(X) = \bigcup\{x | [x]_R \cap X \neq \Phi\}.$$

The pair($R_-(X)$, $R^-(X)$) is referred to as the rough set approximation of X, where $R_-(X)$ is the smallest definable set containing X and $R^-(X)$is the largest definable set contained in X.

From the rough set approximation, the positive region, negative region and boundary region of X can be defined:

$$\mathrm{POS}_R(X) = R_-(X),$$
$$\mathrm{BND}_R(X) = R^-(X) - R_-(X),$$
$$\mathrm{NEG}_R(X) = U - R^-(X).$$

The positive region and negative region consist of objects whose descriptions allow for deterministic decisions regarding their membership in X. The boundary region consists of objects whose descriptions allow for non-deterministic decisions regarding their membership in X.

2.2 The positive region reduct of the decision table

A decision table can be denoted by $S = (U, C \bigcup D, V, f)$, where $C \bigcap D = \Phi$; C and D are called the condition attribute set and decision attribute set, respectively. Assume the objects are partitioned into r mutually exclusive crisp subsets $\{D_1, D_2, \ldots, D_r\}$ by the decision attributes set D. Regarding $R \subseteq C$, denoted by $\mathrm{POS}_R(D, U) = \bigcup_{i=1}^{r} R_- D_i$, it is called the positive region of D with respect to the condition attribute set R .

Given a decision table $S = (U, C \bigcup D, V, f)$, if $\exists x_i, x_j \in U (i \neq j)$ and $f(x_i, C) = f(x_j, C) \wedge f(x_i, D) \neq f(x_j, D)$, then the decision table S is an inconsistent decision table and x_i, x_j are called inconsistent objects; otherwise, the decision table S is a consistent decision table.

The attribute a is relatively indispensable in condition attribute set C if $\mathrm{POS}_{C-\{a\}}(D, U) \neq \mathrm{POS}_C(D, U)$; otherwise, a is said to be relatively dispensable in C. The set of relatively indispensable attributes is called core attribute set CORE(C).

Definition 1 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subset C$. For each $a \in C - R$, the joined significance measures of the attribute a based on the positive region in S can also be defined as

$$SIG_{\mathrm{POS}}{}^{+}(a, R, D, U) = \gamma_{R\cup\{a\}}(D, U) - \gamma_R(D, U), \text{ or}$$
$$SIG_{\mathrm{POS}}{}^{+}(a, R, D, U) = |\mathrm{POS}_{R\cup\{a\}}(D, U)| - |\mathrm{POS}_R(D, U)|$$

where $\gamma_R(D, U) = |\mathrm{POS}_R(D, U)|/|U|$.

Definition 2 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subseteq C$. For each $a \in R$, the weeded significance measures of the attribute a based on the positive region in S can also be defined as

$$SIG_{\mathrm{POS}}{}^{-}(a, R, D, U) = |\mathrm{POS}_R(D, U)| - |\mathrm{POS}_{R-\{a\}}(D, U)|$$

Proposition 1 *Given a decision table $S = (U, C \bigcup D, V, f)$, for $a \in C$, if $SIG_{\mathrm{POS}}{}^{-}(a, C, D, U) > 0$, then $a \in CORE(C)$.*

Attribute reduction is one of the applications for the rough set theory; it is a condition attributes sub-set that can provide the same information for classification purposes as the entire set of original condition attributes. Hu and Cercone (1995) *proposed a heuristic feature selection algorithm, called positive-region reduct, that keeps the positive region of the target decision unchanged. The classic positive region reduct of D with respect to C can be illustrated as follows.*

Definition 3 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subseteq C$. R is a positive region reduct of D with respect to C, which is satisfied with

(1) $\mathrm{POS}_C(D, U) = \mathrm{POS}_R(D, U)$;
(2) $\forall a \in R,\ \mathrm{POS}_R(D, U) \neq \mathrm{POS}_{R-\{a\}}(D, U)$.

The first condition ensures that the reduct has the same positive region information as the whole set of condition attributes. The second condition ensures that the reduct is minimal and there are no redundant attributes in the reduct. Namely, condition (1) states that the attributes in R are sufficient, while condition (2) implies that every attribute in R is necessary.

3 Attribute reduction algorithm based on the conflict region

In this section, we will introduce the concepts and propositions of the conflict region and give attribute significance measures based on the conflict region. Then, a static attribute reduction algorithm based on the conflict region will be proposed.

3.1 The concepts and propositions of the conflict region

The expression of the approximations, positive region, negative region and boundary region is a subset of U; however, we can use the granular structure information to express the above notions and redefine the lower approximation and upper approximation by using the equivalence classes of the universe, which are referred to as the quotient of the lower approximation and upper approximation of X.

Definition 4 Let $I = (U, A, V, f)$ be an information system, $X \subseteq U$ and $R \subseteq A$. The quotients of the lower approximation and upper approximation of X are defined as follows, respectively.

$$\widetilde{R}_{-}(X) = \{[x]_R | [x]_R \in U/R, [x]_R \subseteq X\},$$
$$\widetilde{R}^{-}(X) = \{[x]_R | [x]_R \in U/R, [x]_R \cap X \neq \Phi\}.$$

where $\widetilde{R}_{-}(X)$ is the set of the equivalence classes defined by U/R, which are the subsets of X. $\widetilde{R}^{-}(X)$ is the set of the equivalence classes defined by U/R, which have a non-empty intersection with X. According to Definition 4, we give an expression of the lower and upper approximations as subsets of U/R rather than subsets of U.

Definition 5 Let $I = (U, A, V, f)$ be an information system, $X \subseteq U$ and $R \subseteq A$. The quotient of the positive region, boundary region and negative region of X are defined as follows, respectively.

$$\begin{aligned} \mathrm{QPOS}_R(X) &= \widetilde{R}_{-}(X), \\ \mathrm{QBND}_R(X) &= \widetilde{R}^{-}(X) - \widetilde{R}_{-}(X), \\ \mathrm{QNEG}_R(X) &= U/R - \widetilde{R}^{-}(X) \end{aligned}$$

where $U/R = \{X_1, X_2, \ldots, X_r\}$ denotes the quotient set of U by R.

Definition 6 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subseteq C$. We define the conflict region of D with respect to the condition attribute set R as follows (Ge et al. 2012).

$$\mathrm{CON}_R(D, U) = U - \bigcup\nolimits_{i=1}^{r} R_{-} D_i = U - \mathrm{POS}_R(D, U)$$

$\mathrm{POS}_R(D, U)$ indicates all objects defined by U/R that for sure can induce a certain decision class. $\mathrm{CON}_R(D, U)$ indicates objects defined by U/R that can induce different decision class.

Definition 7 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subseteq C$. We give the quotient set of positive region and conflict region of D with respect to R as follows, respectively.

$$\begin{aligned} \mathrm{QPOS}_R(D, U) &= \bigcup\nolimits_{i=1}^{r} \widetilde{R}_{-} D_i, \\ \mathrm{QCON}_R(D, U) &= U/R - \bigcup\nolimits_{i=1}^{r} \widetilde{R}_{-} D_i = U/R - \mathrm{QPOS}_R(D, U). \end{aligned}$$

$\mathrm{QPOS}_R(D, U)$ indicates the set of all equivalence classes defined by U/R that can for sure induce a certain decision. $\mathrm{QCON}_R(D, U)$ indicates the set of all equivalence classes defined by U/R that can induce a partial decision.

Proposition 2 *Let $S = (U, C \bigcup D, V, f)$ be a decision table. If S is consistent, then $CON_R(D, U) = \Phi$ and $QCON_R(D, U) = \Phi$; otherwise, $CON_R(D, U) \neq \Phi$ and $QCON_R(D, U) \neq \Phi$.*

Proposition 3 *Let $S = (U, C \bigcup D, V, f)$ be a decision table and $P \subseteq R \subseteq C$. The following properties hold.*

(1) $POS_P(D, U) \subseteq POS_R(D, U)$ and $CON_R(D, U) \subseteq CON_P(D, U)$,
(2) $|POS_P(D, U)| \leq |POS_R(D, U)|$ and $|CON_R(D, U)| \leq |CON_P(D, U)|$,
(3) $|QPOS_P(D, U)| \leq |QPOS_R(D, U)|$ and $|QCON_P(D, U)| \leq |QCON_R(D, U)|$.

Definition 8 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subset C$. For each $a \in C - R$, the joined significance measures of the attribute a based on the conflict region in S can also be defined as (Ge et al. 2012):

$$SIG_{\mathrm{CON}}{}^{+}(a, R, D, U) = |\mathrm{CON}_R(D, U)| - |\mathrm{CON}_{R \cup \{a\}}(D, U)|$$

Definition 9 Let $S = (U, C \bigcup D, V, f)$ be a decision table and $R \subseteq C$. For each $a \in R$, the weeded significance measures of the attribute a based on the conflict region in S can also be defined as: (Ge et al. 2012)

$$SIG_{\mathrm{CON}}{}^{-}(a, R, D, U) = |\mathrm{CON}_{R-\{a\}}(D, U)| - |\mathrm{CON}_R(D, U)|$$

According to Definition 9, when an attribute is removed from the attribute set R, the significance measure of the attribute with respect to R can be obtained. If $SIG_{\mathrm{CON}}{}^{-}(a, R, D, U) > 0$, then the attribute a is indispensable; otherwise, if $SIG_{\mathrm{CON}}{}^{-}(a, R, D, U) = 0$, it is dispensable.

Therefore, according to Definition 9, we can delete dispensable attributes from the attribute set R while reserving other important attributes. Deleting redundant attributes from the obtained attribute subset guarantees the completeness of the attribute subset.

Proposition 4 *Given a decision table $S = (U, C \bigcup D, V, f)$, for $a \in C$, if $SIG_{\mathrm{CON}}{}^{-}(a, R, D, U) > 0$, then $a \in CORE(C)$* (Ge et al. 2012).

Definition 10 Given a decision table $S = (U, C \bigcup D, V, f)$ and $R \subseteq C$, R is the positive region reduct of S, which satisfies the following two conditions.

(1) $|\mathrm{CON}_R(D, U)| = |\mathrm{CON}_C(D, U)|$;
(2) $\forall a \in R, |\mathrm{CON}_{R-\{a\}}(D, U)| \neq |\mathrm{CON}_R(D, U)|$.

The first condition guarantees that the reduct R has the same conflict information as the whole attribute set C; the second condition guarantees that there are no redundant attributes in the reduct R.

Example 1 Table 1 illustrates a decision table S, where $U=\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$, $C = \{a, b, c, d\}$ is the condition attribute set, and D is the decision attribute set. Let $P = \{bc\}$ and $R = \{bcd\}$.

We can obtain $U/P = \{\{x_1, x_7\}, \{x_2, x_5\}, \{x_4, x_8\}, \{x_3, x_6, x_9\}\}$, $U/R = \{\{x_2\}, \{x_3, x_9\}, \{x_5\}, \{x_6\}, \{x_1, x_7\}, \{x_4, x_8\}\}$, $U/C = \{\{x_2\}, \{x_3, x_9\}, \{x_5\}, \{x_6\}, \{x_1, x_7\}, \{x_4, x_8\}\}$ and $U/D = \{\{x_4\}, \{x_1, x_5, x_6, x_8\}, \{x_2, x_3, x_7, x_9\}\}$, where $D_1 = \{x_4\}$, $D_2 = \{x_1, x_5, x_6, x_8\}$, $D_3 = \{x_2, x_3, x_7, x_9\}$ and $r = 3$.

Assume $X = \{x_1, x_2, x_4, x_5, x_7\}$, we can have
$\widetilde{P}_{-}(X) = \{\{x_1, x_7\}, \{x_2, x_5\}\}$ and $\widetilde{P}^{-}(X) = \{\{x_1, x_7\}, \{x_2, x_5\}, \{x_4, x_8\}\}$;
$\widetilde{R}_{-}(X) = \{\{x_1, x_7\}, \{x_2\}, \{x_5\}\}$ and $\widetilde{R}^{-}(X) = \{\{x_1, x_7\}, \{x_2\}, \{x_4, x_8\}, \{x_5\}\}$;
$\mathrm{QPOS}_P(X) = \widetilde{P}_{-}(X) = \{\{x_1, x_7\}, \{x_2, x_5\}\}$, $\mathrm{QBND}_P(X) = \widetilde{P}^{-}(X) - \widetilde{P}_{-}(X) = \{\{x_4, x_8\}\}$ and $\mathrm{QNEG}_P(X) = U/P - \widetilde{P}_{-}(X) = \{\{x_3, x_9\}\}$;
$\mathrm{QPOS}_R(X) = \widetilde{R}_{-}(X) = \{\{x_1, x_7\}, \{x_2\}, \{x_5\}\}$, $\mathrm{QBND}_R(X) = \widetilde{R}^{-}(X) - \widetilde{R}_{-}(X) = \{\{x_4, x_8\}\}$ and $\mathrm{QNEG}_R(X) = U/R - \widetilde{R}^{-}(X) = \{\{x_3, x_9\}\}$.

Moreover, we can obtain
$\mathrm{POS}_P(D, U) = \bigcup_{i=1}^{r} P_{-}D_i = \Phi$,
$\mathrm{CON}_P(D, U) = U - \bigcup_{i=1}^{r} P_{-}D_i = U - \mathrm{POS}_P(D, U) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$,
$\mathrm{QPOS}_P(D, U) = \bigcup_{i=1}^{r} \widetilde{P}_{-}D_i = \Phi$,
$\mathrm{QCON}_P(D, U) = U/P - \bigcup_{i=1}^{r} \widetilde{P}_{-}D_i = U/P - \mathrm{QPOS}_P(D, U) = \{\{x_1, x_7\}, \{x_2, x_5\}, \{x_4, x_8\}, \{x_3, x_6, x_9\}\}$,
$\mathrm{POS}_R(D, U) = \bigcup_{i=1}^{r} R_{-}D_i = \{x_2, x_3, x_5, x_6, x_9\}$,
$\mathrm{CON}_R(D, U) = U - \bigcup_{i=1}^{r} R_{-}D_i = U - \mathrm{POS}_R(D, U) = \{x_1, x_7, x_4, x_8\}$,
$\mathrm{QPOS}_R(D, U) = \bigcup_{i=1}^{r} \widetilde{R}_{-}D_i = \{\{x_2\}, \{x_3, x_9\}, \{x_5\}, \{x_6\}\}$,

Table 1 A decision table *s*

U	a	b	c	d	D
x_1	0	0	0	1	1
x_2	0	1	0	0	2
x_3	0	1	1	1	2
x_4	1	0	1	0	0
x_5	1	1	0	1	1
x_6	1	1	1	0	1
x_7	0	0	0	1	2
x_8	1	0	1	0	1
x_9	0	1	1	1	2

$\mathrm{QCON}_R(D, U) = U/R - \bigcup_{i=1}^{r} \widetilde{R}_{-} D_i = U/R - \mathrm{QPOS}_R(D, U) = \{\{x_1, x_7\}, \{x_4, x_8\}\}$,
$\mathrm{POS}_C(D, U) = \bigcup_{i=1}^{r} C_{-} D_i = \{x_2, x_3, x_5, x_6, x_9\}$,
$\mathrm{CON}_C(D, U) = U - \bigcup_{i=1}^{r} C_{-} D_i = U - \mathrm{POS}_C(D, U) = \{x_1, x_7, x_4, x_8\}$.

Since, $|\mathrm{CON}_R(D, U)| = |\mathrm{CON}_C(D, U)|$ and $\forall a \in R$, $|\mathrm{CON}_{R-\{a\}}(D, U)| \neq |\mathrm{CON}_R(D, U)|$, R is a reduct of S. Nevertheless, P is not a reduct of S.

3.2 Static attribute reduction algorithm based on the conflict region

According to the research of Sect. 3.1, in this subsection, we will design a static heuristic attribute reduction algorithm based on the conflict region. The reduction algorithm includes three components: core attributes computation, attribute subset generation and redundant attributes detection. The description of the algorithm is as follows.

Algorithm 1 Static attribute reduction algorithm based on the conflict region(SRACR)

```
Input: A decision table $S=(U,C\cup D,V,f)$;
Output: a reduct $R$.
Step 1: for $\forall a\in C$, if ($SIG_{\mathrm{CON}}^{-}(a,C,D,U)>0$) CORE($C$)=CORE($C$)$\bigcup\{a\}$, and let $R$=CORE($C$);
Step 2: if($R=\Phi$) then $\mathrm{CON}_R(D,U)=U$;
            else compute $\mathrm{CON}_R(D,U)$;
Step 3: while ($|\mathrm{CON}_R(D,U)| \neq |\mathrm{CON}_C(D,U)|$)
        { for $k$=1 to $|C-R|$ Do Compute $SIG_{\mathrm{CON}}^{+}(a_k,R,D,U)$;
          select attribute $b$ which satisfy equation: $SIG_{\mathrm{CON}}^{+}(b,R,D,U)=\max\{SIG_{\mathrm{CON}}^{+}(a_k,R,D,U)\}$.
          if the attribute like that is not only one, we choose an attribute arbitrarily.
          $R=R\bigcup\{b\}$;
          update $\mathrm{CON}_R(D,U)$;
        }//end_while
Step 4: for $\forall a\in R$-CORE(C) Do
            if ($SIG_{\mathrm{CON}}^{-}(a,R,D,U)=0$) then $R=R-\{a\}$;
Step 5: output $R$.
```

The time complexity analysis The radix sort algorithm is used to compute equivalence classes. The time complexity of computing the core attributes is $O(|C|^2|U|)$ in Step 1. Step 2 is to compute conflict region $\mathrm{CON}_R(D, U)$, whose time complexity are $O(|\mathrm{CORE}||U|)$ in the worst case. In Step 3, the indispensable attributes are computed and added into the selected attribute subset in turn, the time complexity is $\sum_{i=1}^{|C-\text{CORE}|}|U|(|C-\text{CORE}|-i+1)$. In Step 4, in the worst case, the time complexity of deleting redundant attributes from the obtained attribute subset R is $O(|C|^2|U|)$. Hence, the time complexity of Algorithm SRACR is $2O(|C|^2|U|)+O(|\text{CORE}||U|)+\sum_{i=1}^{|C-\text{CORE}|}|U|(|C-\text{CORE}|-i+1)$.

Example 2 Table 1 illustrates a decision table S, where $U=\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9\}$, $C=\{a,b,c,d\}$ is the condition attribute set and D is the decision attribute set.

We have $U/C=\{\{x_1,x_7\},\{x_2\},\{x_3,x_9\},\{x_4,x_8\},\{x_5\},\{x_6\}\}$, where $\{x_1,x_7\}$ and $\{x_4,x_8\}$ are two inconsistent object set. Hence, Table 1 is an inconsistent decision table, and we have $\text{POS}_C(D,U)=\{x_2,x_3,x_5,x_6,x_9\}$ and $\text{CON}_C(D,U)=\{x_1,x_7,x_4,x_8\}$.

We can get $\text{CON}_{C-\{a\}}(D,U)=\text{CON}_{C-\{c\}}(D,U)=\text{CON}_{C-\{d\}}(D,U)=\{x_1,x_7,x_4,x_8\}$, and $\text{CON}_{C-\{b\}}(D,U)=\{x_1,x_7,x_4,x_8,x_6\}$. Since $SIG_{\text{CON}}{}^{-}(b,C,D,U)=|\text{CON}_{C-\{b\}}(D,U)|-|\text{CON}_C(D,U)|=1>0, b\in\text{CORE}(C)$. Let $R=\{b\}$. We get $SIG_{\text{CON}}{}^{+}(a,R,D,U)$=6 and $SIG_{\text{CON}}{}^{+}(c,R,D,U)=SIG_{\text{CON}}{}^{+}(d,R,D,U)=0$. We should select a to add into $R=\{ab\}$ and get $\text{CON}_R(D,U)=\{x_1,x_7,x_4,x_8\}$. And then, we have $|\text{CON}_R(D,U)|=|\text{CON}_C(D,U)|$. The attribute a can not be deleted from R. Hence, a reduct of S is $R=\{ab\}$, where $U/R=\{\{x_1,x_7\},\{x_2,x_3,x_9\},\{x_4,x_8\},\{x_5,x_6\}\}$.

4 Incremental attribute reduction algorithm based on the conflict region when adding one object

With variation of the decision systems, new objects are added into the original decision system due to the arrival of new information, which results in the existing reduct possibly becoming invalid. The static (non-incremental) attribute reduction approach is to re-perform Algorithm 1 to acquire new reduct, which is often very costly or even intractable. To overcome this shortcoming, it is necessary to develop the incremental reduction algorithm to avoid re-computation by utilizing previous results.

In this section, we focus on discussing attribute reduction updating mechanisms and an acceleration strategy when adding one object into the decision system.

4.1 Incremental updating mechanisms based on the conflict region when adding one object

When a new object is added into the decision system, instead of re-computing the reduct, the incremental reduction approach can be applied to obtain the new reduct to improve the performance of the attribute reduction algorithm. In this sub-section, we introduce the incremental mechanisms for updating the conflict region when adding one object instead of re-computing the new decision table. To further improve the efficiency of the incremental reduction algorithm, we research an acceleration strategy of reducing the number of the radix sort for equivalence partition. Finally, we design an incremental reduction algorithm with the acceleration strategy based on the conflict region when adding one object and list some examples to show the effectiveness of the proposed algorithm.

Given a decision table $S=(U,C\bigcup D,V,f)$ and a reduct R, we can get $U/R=\{X_1,X_2,\ldots,X_m\}$ and $U/C=\{Y_1,Y_2,\ldots,Y_n\}$. Assume that a new object x_{new} is added into the decision table S. We can get $S'=(U\bigcup\{x_{new}\},C\bigcup D,V,f)$. Let $U'=U\bigcup\{x_{new}\}$, $U'/R=\{X_1,\ldots,X'_i,\ldots,X_m\}$ and $U'/C=\{Y_1,\ldots,Y'_j,\ldots,Y_n\}$. Four cases exist as follows for adding the object x_{new} into the decision table S.

Case (a) $:\exists j \in [1, \ldots, n], Y_j' = Y_j \bigcup \{x_{new}\}, |Y_j/D| = 1$ and $|Y_j'/D| = 1$;
Case (b) $:\exists j \in [1, \ldots, n], Y_j' = Y_j \bigcup \{x_{new}\}, |Y_j/D| \neq 1$ and $|Y_j'/D| \neq 1$;
Case (c) $:\exists j \in [1, \ldots, n], Y_j' = Y_j \bigcup \{x_{new}\}, |Y_j/D| = 1$ and $|Y_j'/D| \neq 1$;
Case (d) $: j = n + 1$, i.e., $Y_j' = \{x_{new}\}$.

Case(a) shows that the object x_{new} belongs to an existing non-conflict class; Case(b) shows that the object x_{new} belongs to an existing conflict class; Case(c) shows that the object x_{new} results in the generation of a new conflict region; Case(d) shows that the object x_{new} produces a new non-conflict region.

Proposition 5 *Given a decision table S and $R \subseteq C$, let $CON_R(D, U)$ be the conflict region of D with respect to R. A new object x_{new} is added into S to get $S' = (U', C \bigcup D, V, f)$, where $U' = U \bigcup \{x_{new}\}$ and $U'/R = \{X_1, \ldots, X_i', \ldots, X_m\}$. Let $\exists i \in [1, \ldots, m]$, $X_i' = X_i \bigcup \{x_{new}\}$; we have the following propositions.*

(1) $|X_i/D| = 1$ and $|X_i'/D| = 1$, then $CON_R(D, U') = CON_R(D, U)$, $QCON_R(D, U') = QCON_R(D, U)$ and $|CON_R(D, U')| = |CON_R(D, U)|$;
(2) $|X_i/D| = 1$ and $|X_i'/D| \neq 1$, then $CON_R(D, U') = CON_R(D, U) \bigcup X_i'$, $QCON_R(D, U') = QCON_R(D, U) \bigcup X_i$' and $|CON_R(D, U')| = |CON_R(D, U)| + |X_i'|$;
(3) $|X_i/D| \neq 1$ and $|X_i'/D| \neq 1$, then $CON_R(D, U') = CON_R(D, U) \bigcup \{x_{new}\}$, $QCON_R(D, U') = QCON_R(D, U) - X_i \bigcup X_i'$ and $|CON_R(D, U')| = |CON_R(D, U)| + 1$.
(4) $i = m + 1$, then $CON_R(D, U') = CON_R(D, U)$, $QCON_R(D, U') = QCON_R(D, U)$ and $|CON_R(D, U')| = |CON_R(D, U)|$;

Proposition 6 *Given a decision table S, let $CON_C(D, U)$ be the conflict region of D with respect to C. A new object x_{new} is added into S to get $S' = (U', C \bigcup D, V, f)$, where $U' = U \bigcup \{x_{new}\}$ and $U'/C = \{Y_1, \ldots, Y_j', \ldots, Y_n\}$. Let $\exists j \in [1..n]$, $Y_j' = Y_j \bigcup \{x_{new}\}$; we have the following propositions.*

(1) $|Y_j/D| = 1$ and $|Y_j'/D| = 1$, then $CON_C(D, U') = CON_C(D, U)$ and $|CON_C(D, U')| = |CON_C(D, U)|$;
(2) $|Y_j/D| = 1$ and $|Y_j'/D| \neq 1$, then $CON_C(D, U') = CON_C(D, U) \bigcup Y_j'$ and $|CON_C(D, U')| = |CON_C(D, U)| + |Y_j'|$;
(3) $|Y_j/D| \neq 1$ and $|Y_j'/D| \neq 1$, then $CON_C(D, U') = CON_C(D, U) \bigcup \{x_{new}\}$ and $|CON_C(D, U')| = |CON_C(D, U)| + 1$;
(4) $j = n+1$, then $CON_C(D, U') = CON_C(D, U)$ and $|CON_C(D, U')| = |CON_C(D, U)|$.

Theorem 1 *Given a decision table S, R is a reduct of S. Let $CON_R(D, U)$ and $CON_C(D, U)$ be the conflict region of D with respect to R and C, respectively. A new object x_{new} is added into the decision table S to get $S' = (U', C \bigcup D, V, f)$, where $U' = U \bigcup \{x_{new}\}$, $U'/R = \{X_1, \ldots, X_i', \ldots, X_m\}$ and $U'/C = \{Y_1, \ldots, Y_j', \ldots, Y_n\}$. Let $\exists i \in [1..m]$, $X_i' = X_i \bigcup \{x_{new}\}$, and $\exists j \in [1..n]$, $Y_j' = Y_j \bigcup \{x_{new}\}$; the following properties hold.*

(1) If x_{new} satisfies Case(a), then $|CON_R(D, U')| = |CON_C(D, U')|$;
(2) If x_{new} satisfies Case(b), then $|CON_R(D, U')| = |CON_C(D, U')|$;
(3) If x_{new} satisfies Case(c), then $|CON_R(D, U')| \geq |CON_C(D, U')|$;
(4) If x_{new} satisfies Case(d), then $|CON_R(D, U')| \geq |CON_C(D, U')|$.

Proof (1) In this case, new conflict objects with respect to C and R are not generated, respectively. Thus, $\mathrm{CON}_C(D, U) = \mathrm{CON}_C(D, U')$ and $\mathrm{CON}_R(D, U) = \mathrm{CON}_R(D, U')$. Hence, $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$.

(2) In this case, $\exists x \in \mathrm{CON}_C(D, U)$ satisfies $f(x, C) = f(x_{new}, C) \wedge f(x, D) \neq f(x_{new}, D)$ such that $\mathrm{CON}_C(D, U') = \mathrm{CON}_C(D, U) \bigcup \{x_{new}\}$. Likewise, $\exists y \in \mathrm{CON}_R(D, U)$ satisfies $f(y, R) = f(x_{new}, R) \wedge f(y, D) \neq f(x_{new}, D)$ such that $\mathrm{CON}_R(D, U') = \mathrm{CON}_R(D, U) \bigcup \{x_{new}\}$. Hence, $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$.

(3) In this case, $\exists x \in Y_j$ satisfies $f(x, C) = f(x_{new}, C) \wedge f(x, D) \neq f(x_{new}, D)$, such that $\mathrm{CON}_C(D, U') = \mathrm{CON}_C(D, U) \bigcup Y_j \bigcup \{x_{new}\}$. Suppose $x_{new} \in X_i'$; there exist two cases as follows.

   (1) $\exists y \in X_i$ and $X_i \supset Y_j$, then $\mathrm{CON}_R(D, U') = \mathrm{CON}_R(D, U) \bigcup X_i \bigcup \{x_{new}\}$, such that $|\mathrm{CON}_R(D, U')| > |\mathrm{CON}_C(D, U')|$;
   (2) $\exists y \in X_i$ and $X_i = Y_j$, then $\mathrm{CON}_R(D, U') = \mathrm{CON}_R(D, U) \bigcup X_i \bigcup \{x_{new}\}$, such that $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$.

(4) In this case, x_{new} is a whole new object that is not the same as any objects of S such that $\mathrm{CON}_C(D, U') = \mathrm{CON}_C(D, U)$. However, there may exist three cases as follows.

   (a) $i = m + 1$, then $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$;
   (b) $i \leq m$, $|X_i/D| = 1$ and $|X_i'/D| = 1$, then $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$;
   (c) $i \leq m$, $|X_i/D| \neq 1$ and $|X_i'/D| \neq 1$, then $\mathrm{CON}_R(D, U') = \mathrm{CON}_R(D, U) \bigcup X_i'$ such that $|\mathrm{CON}_R(D, U')| > |\mathrm{CON}_C(D, U')|$.

Proposition 7 *Given a decision table S, a new object x_{new} is added into S to get $S' = (U', C \bigcup D, V, f)$, $CON_R(D, U')$ and $QCON_R(D, U')$, where $U' = U \bigcup \{x_{new}\}$. Let $R \subset C$; for $a \in C - R$, $CON_{R \bigcup \{a\}}(D, U')$ can be illustrated as follows.*

$$|CON_{R \bigcup \{a\}}(D, U')| = |\bigcup \{Z | W \in QCON_R(D, U') \wedge Z \in W/a \wedge |Z/D| \neq 1\}|$$

Proposition 8 *Given a decision table S, let $R \subset C$; a new object x_{new} is added into S to obtain S'. For each $a \in C - R$, the joined significance measures of the attribute a based on the conflict region in S' can also be defined as follows.*

$$\begin{aligned} SIG_{\mathrm{CON}}{}^{+}(a, R, D, U') &= |CON_R(D, U')| - |CON_{R \bigcup \{a\}}(D, U')| \\ &= |CON_R(D, U')| - |\bigcup \{Z | W \in QCON_R(D, U') \\ &\qquad \wedge Z \in W/a \wedge |Z/D| \neq 1\}| \end{aligned}$$

4.2 The acceleration strategy for deleting redundant attributes

As we know, if a subset $R*$ of C only satisfies condition(1) of the reduct definition, the attributes in $R*$ are sufficient. Namely, the condition(1) is not a necessary condition for the reduct of S and $R*$ is only a superset of a reduct. There may exist one real reduct R of S and $R \subseteq R*$. Hence, we should check whether there exist redundant attributes in $R*$.

Generally, for an attribute a of reduct-superset $R*$, if $SIG_{\mathrm{CON}}{}^{-}(a, R*, D, U') = 0$, the attribute a is redundant and should be deleted from the attribute set $R*$. The time complexity of this operation is $O(|R|^2|U|)$ and the number of radix sort is $|R|(|R| - 1)$. This operation of checking every attribute in $R*$ will require plenty of computational time. Can we further improve the performance of incremental reduction algorithms? This motivates us to research an acceleration approach by reducing sort times in the process of checking redundant attributes to improve efficiency. In the following, we discuss the acceleration strategy based on the conflict region for updating reduct.

Given a decision table $S = (U, C \bigcup D, V, f)$, C is a condition attribute set that includes $c_1, c_2, \ldots, c_i, \ldots$,etc. U/C is a partition of the universe with respect to C. We can 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., $c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_{i-1}$) and have only one different sort. We all know that the radix sort is stable; thus we can obtain $U/(C - \{c_{i+1}\})$ derived from $U/(C - \{c_i\})$ (Qian et al. 2011; Zhou et al. 2007).

Proposition 9 *For a decision table* $S = (U, C \bigcup D, V, f)$*, RS(*$U, C - \{c_{i+1}\}$*) = RS(RS(*$U, C - \{c_i\}$*),* $\{c_i\}$*).*

Proof RS($U, C - \{c_{i+1}\}$) denotes the result of the radix sort with respect to the attributes order $c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_i$. We divide two steps to obtain RS(RS($U, C - \{c_i\}$), $\{c_i\}$). That is, firstly, we can obtain RS($U, C - \{c_i\}$) by radix sort with respect to the attributes order $c_{i+1}, \ldots, c_{|C|}, c_1, \ldots, c_{i-1}$; then, based on RS($U, C - \{c_i\}$) we complete the radix sort by the attribute c_i. Therefore, the result of radix sort with respect to the attributes order $c_{i+1}, c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_i$ is equivalent to merging the above two steps of sort operations. Because the radix sort is a stable sort, RS($U, C - \{c_{i+1}\}$) is equivalent to RS(RS($U, C - \{c_i\}$), $\{c_i\}$). Hence, Proposition 9 holds.

From Proposition 9 we can significantly reduce the number of the radix sort between $U/C - \{c_i\}$ and $U/C - \{c_{i+1}\}$. Hence, we can employ Proposition 9 to carry out the operation of checking redundant attributes from the previous result of attribute reduction to accelerate the efficiency of incremental reduction algorithms.

4.3 Incremental reduction algorithm based on the conflict region when adding one object

Based on the above discussion, we present an incremental reduction algorithm (IRACR-O) with the acceleration strategy when adding one object into the decision table. The algorithm includes three main steps:

(1) The first step (Steps 1–2 in algorithm IRACR-O) is to update $\text{CON}_R(D, U')$, $\text{QCON}_R(D, U')$ and $\text{CON}_C(D, U')$ when adding one object;
(2) The second step (Step 3 in algorithm IRACR-O) is to select new attributes from the remaining attributes to add into the attribute subset, until the new attribute subset satisfies $|\text{CON}_R(D, U')| = |\text{CON}_C(D, U')|$;
(3) The third step (Steps 4–5 in algorithm IRACR-O) is to delete the redundant attributes of the new attribute subset with the acceleration strategy.

Algorithm 2 Incremental reduction algorithm with the acceleration strategy based on the conflict region when adding one object (IRACR-O)

```
Input: A decision table $S=(U,C\cup D,V,f)$, a reduct $R$, $U/R=\{X_1,\ldots,X_m\}$,$U/C=\{Y_1,\ldots,Y_n\}$ and a new incremental object $x_{new}$;
Output: the new reduct $R$ and the new decision table $S'$.
Step 1: let $U'=U\cup\{x_{new}\}$, compute $U'/R=\{X_1,\ldots,X_i',\ldots,X_m\}$ and $U'/C=\{Y_1,\ldots,Y_j',\ldots,Y_n\}$;
Step 2: if ($i=m+1$, $X_i'=\{x_{new}\}$) goto Step6;
        else if ($X_i'=X_i\cup\{x_{new}\}$, $|X_i/D|=1\wedge|X_i'/D|=1$) goto Step 6;
        else if ($X_i'=X_i\cup\{x_{new}\}$, $|X_i/D|\neq1\wedge|X_i'/D|\neq1$) update $\mathrm{CON}_R(D,U')$, $\mathrm{QCON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$;
        else if ($X_i'=X_i\cup\{x_{new}\}$, $|X_i/D|=1\wedge|X_i'/D|\neq1$) update $\mathrm{CON}_R(D,U')$, $\mathrm{QCON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$;
Step 3: while($|\mathrm{CON}_R(D,U')|\neq|\mathrm{CON}_C(D,U')|$)
     { for $\forall a_k\in C-R$, compute $SIG_{\mathrm{CON}}^{+}(a_k,R,D,U')$;
       select the attribute $b$ which satisfies equation: $SIG_{\mathrm{CON}}^{+}(b,R,D,U')=\max\{SIG_{\mathrm{CON}}^{+}(a_k,R,D,U')\}$;
       if the attribute like that is not only one, we choose an attribute arbitrarily;
       $R=R\cup\{b\}$;
       update and $\mathrm{QCON}_R(D,U')$;
     }
Step 4: let $R=\{a_1,a_2,\ldots,a_{|R|}\}$, flag=0;   // check redundant attributes
        Sort $S$ by $R$, obtain $\mathrm{RS}(U',R)$ and $\mathrm{CON}_{R-\{a_1\}}(D,U')$;
        if ($SIG_{\mathrm{CON}}^{-}(a_1,R,D,U')=0$ ) { $R=R-\{a_1\}$; flag=1; } else flag=0;
Step 5: for $i=1$ to $|R|-1$ do
     { if (flag) compute $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$; // $a_i$ can be deleted
         else       // $a_i$ cannot be deleted
             sort $S$ by $a_i$ based on the previous sort result to obtain $\mathrm{RS}(U',R-\{a_{i+1}\})$ and $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$;
        if ($SIG_{\mathrm{CON}}^{-}(a_{i+1},R,D,U')=0$ ) { $R=R-\{a_{i+1}\}$; flag=1; } else flag=0;
     }
Step 6: output the new reduct $R$ and $S'$.
```

The time complexity analysis In Step 1, when adding one object into the decision system, the time complexity of computing the partition of the whole object set under the attribute sets R and C is $O(|C||U|)$; the time complexity of Step 2 is $O(|X_i'|)$ in the worst case. In general, we only execute the 'while' loop structure once and can select an attribute to satisfy the termination condition of the loop structure. Hence, the time complexity of Step 3 is $O(|C-R||X_i'|)$; In Steps 4–5, the selected attributes are checked to delete redundant attributes with the acceleration strategy, the time complexity of Step 4 –5 is $O(|C||U'+1|)$. Hence, the total time complexity of algorithm IRACR-O is $2O(|C||U'|)+O(|X_i'|)+O(|C-R||X_i'|)$.

The performance analysis From IRACR-O, we can find that the incremental reduction algorithm improves the performance of the reduction algorithm compared with the non-incremental reduction algorithm SARCR when adding one object.

(1) In Step 2, the new conflict regions ($\mathrm{CON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$) are computed via an incremental approach based on the existing conflict region.
(2) In Step 3, the new attribute subset is generated from the original reduct, which generally only needs to execute the loop structure once to get the attributes subset R that satisfies $|\mathrm{CON}_R(D,U')|=|\mathrm{CON}_C(D,U')|$.
(3) From Steps 4–5, the acceleration strategy with reduction of the radix sort time is used to improve the performance of deleting redundant attributes.

When multiple objects immigrate into the decision system, we need to execute Steps 1–3 of Algorithm 2 repeatedly and execute Step 4 once to update reduct. The incremental reduction algorithm for adding multiple objects is as follows.

Algorithm 3 Incremental reduction algorithm with the acceleration strategy based on the conflict region when adding mult-objects (IRACR-M)

Input: A decision table $S=(U,C\bigcup D,V,f)$, a reduct R, $U/R=\{X_1,\ldots,X_m\},U/C=\{Y_1,\ldots,Y_n\}$ and a new incremental object set U_{new};

Output: a new reduct R and the new decision table S'.

```
Step 1: for $\forall x_{new}\in U_{new}$
        { let $U'=U\bigcup\{x_{new}\}$, compute $U'/R=\{X_1,\ldots,X_i',\ldots,X_m\}$ and $U'/C=\{Y_1,\ldots,Y_j',\ldots,Y_n\}$;
          if ($i=m+1$, $X_i'=\{x_{new}\}$) continue;
           else if ($X_i'=X_i\bigcup\{x_{new}\}$, $|X_i/D|=1\wedge|X_i'/D|=1$) continue;
           else if ($X_i'=X_i\bigcup\{x_{new}\}$, $|X_i/D|\neq1\wedge|X_i'/D|\neq1$) update $\mathrm{CON}_R(D,U')$, $\mathrm{QCON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$;
           else if ($X_i'=X_i\bigcup\{x_{new}\}$, $|X_i/D|=1\wedge|X_i'/D|\neq1$) update $\mathrm{CON}_R(D,U')$, $\mathrm{QCON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$;
          while($|\mathrm{CON}_R(D,U')|\neq|\mathrm{CON}_C(D,U')|$)
          { for $\forall a_k\in C-R$, compute $SIG_{\mathrm{CON}}^{+}(a_k,R,D,U')$;
             select the attribute $b$ which satisfies: $SIG_{\mathrm{CON}}^{+}(b,R,D,U')=\max\{SIG_{\mathrm{CON}}^{+}(a_k,R,D,U')\}$;
             if the attribute like that is not only one, we choose an attribute arbitrarily;
             $R=R\cup\{b\}$;
             update $\mathrm{CON}_R(D,U')$ and $\mathrm{QCON}_R(D,U')$;
           }// end_while
          } //end_Step1
Step 2: let $R=\{a_1,a_2,\ldots,a_{|R|}\}$, flag=0;    // check redundant attributes
          sort $S$ by $R$, obtain $\mathrm{RS}(U',R)$ and $\mathrm{CON}_{R-\{a_1\}}(D,U')$;
          if ($SIG_{\mathrm{CON}}^{-}(a_1,R,D,U')=0$ ) { $R=R-\{a_1\}$; flag=1; } else flag=0;
Step 3: for $i=1$ to $|R|-1$ do
        { if (flag) compute $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$; // $a_i$ can be deleted
             else        // $a_i$ cannot be deleted
                sort $S$ by $a_i$ based on the previous Sort result to obtain $\mathrm{RS}(U',R-\{a_{i+1}\})$ and $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$;
           if ($SIG_{\mathrm{CON}}^{-}(a_{i+1},R,D,U')=0$ ) { $R=R-\{a_{i+1}\}$; flag=1; } else flag=0;
           } //end_for
Step 4: output the new reduct $R$ and $S$'.
```

The time complexity analysis Suppose that l objects are added into the decision system. From the analysis of the time complexity of algorithm 2, we know that the time complexity of Step 1 is $l^*(O(|C||U|)+O(|X_i'|)+O(|C-R||X_i'|))$; In Steps 2–3, the operations of checking the redundant attributes of obtain selected attribute subset R and deleting the redundant attributes form R with the acceleration strategy are performed, whose time complexity is $O(|C||U'+1|)$. Hence, the time complexity of algorithm IRACR-M is $l^*(O(|C||U|)+O(|X_i'|)+O(|C-R||X_i'|))+O(|C||U'+1|)$.

Example 3 (Continuation of Example 1) For Table 1, $R=\{ab\}$ is a reduct of the decision table S, and we can get $U/R=\{\{x_1,x_7\},\{x_2,x_3,x_9\},\{x_4,x_8\},\{x_5,x_6\}\}$ and $U/C=\{\{x_1,x_7\},\{x_2\},\{x_3,x_9\},\{x_4,x_8\},\{x_5\},\{x_6\}\}$, where $X_1=\{x_1,x_7\}$, $X_2=\{x_2,x_3,x_9\}$, $X_3=\{x_4,x_8\}$, $X_4=\{x_5,x_6\}$, $Y_1=\{x_1,x_7\}$, $Y_2=\{x_2\}$, $Y_3=\{x_3,x_9\}$, $Y_4=\{x_4,x_8\}$, $Y_6=\{x_5\}$ and $Y_6=\{x_6\}$. A new object $y_1=(0,1,0,0,1)$ is added to Table 1.

Execute IRACR-O; we can find that $X_2=\{x_2,x_3,x_9\}$ and $X_2'=\{x_2,x_3,x_9,y_1\}$, where $|X_2/D|=1$ and $|X_2'/D|\neq1$. Hence, we update $\mathrm{CON}_R(D,U')=\{x_1,x_7,x_4,x_8,x_2,x_3,$

$x_9, y_1\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_4, x_8, x_2, y_1\}$ and can get $|\mathrm{CON}_R(D, U')| = 8$, $|\mathrm{CON}_C(D, U')| = 6$ and $|\mathrm{CON}_R(D, U')| \neq |\mathrm{CON}_C(D, U')|$. Thus, we need to execute Step3 to add attributes for $C - R$ into R. In the first circulation, we can get $SIG_{\mathrm{CON}}{}^{+}(c, R, D, U') = 2$ and $SIG_{\mathrm{CON}}{}^{+}(d, R, D, U') = 2$. Then, we select c into R, and have $R = \{ab\} \bigcup \{c\} = \{abc\}$. Thus, we have $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')| = 6$. Redundant attributes in R do not exist. Hence, the new reduct is $R = \{abc\}$.

Example 4 (Continuation of Example 1) For Table 1, $R = \{ab\}$ is a reduct of the decision table S, and the three objects (i.e., $y_1 = (0, 1, 0, 0, 1)$, $y_2 = (0, 1, 0, 1, 1)$ and $y_3 = (1, 0, 1, 0, 2)$) are added into Table 1.

(1) We suppose that the adding order is y_1, y_2, y_3 and execute IRACR-M. From Example 3, we can get $R = \{abc\}$, $\mathrm{CON}_R(D, U') = \mathrm{CON}_C(D, U') = \{x_1, x_7, x_4, x_8, x_2, y_1\}$ and $\mathrm{QCON}_R(D, U') = \{\{x_1, x_7\}, \{x_4, x_8\}, \{x_2, y_1\}\}$.

Then, adding y_2 into the decision table, we can find that $X_2 = \{x_2, y_1\}$ and $X_2' = \{x_2, y_1, y_2\}$, where $|X_2/D| \neq 1$ and $|X_2'/D| \neq 1$. We update $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_4, x_8, x_2, y_1, y_2\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_4, x_8, x_2, y_1\}$; $|\mathrm{CON}_R(D, U')| \neq |\mathrm{CON}_C(D, U')|$. We select d into R and have $R = \{abc\} \bigcup \{d\} = \{abcd\}$.

Next, adding y_3 into the decision table, we have $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_2, y_1, x_4, x_8, y_3\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_2, y_1, x_4, x_8, y_3\}$; $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$.

Finally, according to Steps 2–3, we find that a in R can be deleted. Hence, the new reduct is $R = \{bcd\}$.

(2) We suppose that the adding order is y_3, y_2, y_1 and execute the algorithm IRACR-M.

Similar to the above analysis, when y_3 is added into the decision table, we can get $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_4, x_8, y_3\}$ and $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_4, x_8, y_3\}$ such that $R = \{ab\}$.

Then, adding y_2 into the decision table, we can have $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_4, x_8, y_3, x_2, y_2\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_4, x_8, y_3\}$, where $|\mathrm{CON}_R(D, U')| \neq |\mathrm{CON}_C(D, U')|$. We have $SIG_{\mathrm{CON}}{}^{+}(c, R, D, U') = 0$ and $SIG_{\mathrm{CON}}{}^{+}(d, R, D, U') = 2$. We select d into R and have $R = \{ab\} \bigcup \{d\} = \{abd\}$, $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_4, x_8, y_3\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_4, x_8, y_3\}$.

Next, adding y_1 into the decision table, we have $\mathrm{CON}_R(D, U') = \{x_1, x_7, x_2, y_1, x_4, x_8, y_3\}$ and $\mathrm{CON}_C(D, U') = \{x_1, x_7, x_2, y_1, x_4, x_8, y_3\}$; $|\mathrm{CON}_R(D, U')| = |\mathrm{CON}_C(D, U')|$.

Finally, redundant attributes in R do not exist. Hence, the new reduct is $R = \{abd\}$.

For Example 4, we can find that the results of the two reduction processes are not coincident because of the different orders in which objects are added.

5 Incremental attribute reduction based on the conflict region when adding the object set

In real-life applications, it is not often the case that only one object is added into the decision system. Generally, an object set (i.e., multiple objects) is added into the decision system. From Algorithm 2, obviously, immigration of multiple objects can be seen as the cumulative immigration of a single object. If we re-execute Algorithm 2 many times to update the reduct, it will consume plenty of time. In this sub-section, to improve the efficiency of the incremental attribute reduction algorithm when adding multiple objects, we will develop an incremental reduction algorithm in an efficient manner to obtain the new reduct that avoids repeated execution.

For a decision table $S = (U, C \bigcup D, V, f)$, a reduct R, $U/R = \{X_1, X_2, \ldots, X_m\}$ and $U/C = \{Y_1, Y_2, \ldots, Y_n\}$, we assume that an object set U_{new} is added into the decision table S; we can get $S' = (U \bigcup U_{new}, C \bigcup D, V, f)$. Let $U' = U \bigcup U_{new}$, $U'/R = \{X'_1, \ldots, X'_s, X'_{s+1}, \ldots, X'_m\}$ and $U'/C = \{Y'_1, \ldots, Y'_t, Y'_{t+1}, \ldots, Y'_n\}$, where $\forall i \in [1, \ldots, s]$, $|X'_i/D| \neq 1$; $\forall i \in [s+1, \ldots, m]$, $|X'_i/D| = 1$; $\forall j \in [1, \ldots, t]$, $|Y'_j/D| \neq 1$; $\forall j \in [t+1, \ldots, n]$, $|Y'_j/D| = 1$. Hence, we have $\mathrm{CON}_R(D, U') = \bigcup\{X'_i | i \in [1, \ldots, s]\}$ and $\mathrm{CON}_C(D, U') = \bigcup\{Y'_j | i \in [1, \ldots, t]\}$.

Proposition 10 *Given a decision table S, R is a reduct of S and the object set* U_{new} *is added into S. Let* $U' = U' \bigcup U_{new}$, $U'/R = \{X'_1, \ldots, X'_s, X'_{s+1}, \ldots, X'_{m'}\}$ *and* $U'/C = \{Y'_1, \ldots, Y'_t, Y'_{t+1}, \ldots, Y'_{n'}\}$. *For each* $a \in C - R$, *the joined significance measures of the attribute a based on the conflict region can also be defined as follows.*

$$SIG_{\mathrm{CON}}{}^{+}(a, R, D, U') = |CON_R(D, U')| - |\bigcup\{Z | W \in QCON_R(D, U') \wedge Z \in W/a \wedge |Z/D| \neq 1\}|$$

In the following, we present an incremental reduction algorithm with the acceleration strategy when adding an object set into the decision table.

Algorithm 4 Incremental reduction algorithm with the acceleration strategy based on the conflict region when adding an object set (IRACR-S)

```
Input: A decision table $S=(U,C\bigcup D,V,f)$, a reduct $R$, $U/R=\{X_1,X_2,\ldots,X_m\}$, $U/C=\{Y_1,Y_2,\ldots,Y_n\}$ and an incremental object set $U_{new}$;
Output: the new reduct $R$ and the new decision table $S'$.
Step 1: let $U'=U\bigcup U_{new}$, compute $U'/R=\{X_1',\ldots,X_s',X_{s+1}',\ldots,X_m'\}$ and $U'/C=\{Y_1',\ldots,Y_t',Y_{t+1}',\ldots,Y_{n'}'\}$ and obtain $\mathrm{CON}_R(D,U')$, $\mathrm{QCON}_R(D,U')$ and $\mathrm{CON}_C(D,U')$;
Step 2: if ($\mathrm{CON}_R(D,U')=\Phi$) goto Step 6;
Step 3: while($|\mathrm{CON}_R(D,U')| \neq |\mathrm{CON}_C(D,U')|$)
    { for $\forall a_k \in C-R$, compute $SIG_{\mathrm{CON}}{}^{+}(a_k,R,D,U')$;
      select the attribute $b$ which satisfies equation: $SIG_{\mathrm{CON}}{}^{+}(b,R,D,U')=\max\{SIG_{\mathrm{CON}}{}^{+}(a_k,R,D,U')\}$.
      if the attribute like that is not only one, we choose an attribute arbitrarily.
      $R=R\cup\{b\}$;
      update $\mathrm{CON}_R(D,U')$ and $\mathrm{QCON}_R(D,U')$;
    }
Step 4: let $R=\{a_1,a_2,\ldots,a_{|R|}\}$, flag=0;
    sort $S$ by $R$, obtain $\mathrm{RS}(U',R)$ and $\mathrm{CON}_{R-\{a_1\}}(D,U')$;
    if ($SIG_{\mathrm{CON}}{}^{-}(a_1,R,D,U')=0$ ) { $R=R-\{a_1\}$; flag=1; } else flag=0;
Step 5: for $i$ =1 to $|R|$-1 do
    { if (flag) compute $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$;   // $a_i$ can be deleted
      else       // $a_i$ cannot be deleted
        sort $S$ by $a_i$ based on the previous Sort result to obtain $\mathrm{RS}(U',R-\{a_{i+1}\})$ and $\mathrm{CON}_{R-\{a_{i+1}\}}(D,U')$;
      if ($SIG_{\mathrm{CON}}{}^{-}(a_{i+1},R,D,U')=0$ ) { $R=R-\{a_{i+1}\}$; flag=1; } else flag=0;
    }
Step 6: output the new reduct $R$ and $S'$.
```

The time complexity analysis Step 1 is to compute $\mathrm{CON}_R(D, U')$, $\mathrm{QCON}_R(D, U')$ and $\mathrm{CON}_C(D, U')$ when adding multiple objects, whose time complexity is $O(|C||U'|)$. Step 3 is to check the selected attribute whether satisfies the stopping condition, whose time complexity 3 is $O(|C - R|^2|\mathrm{CON}_R(D, U')|)$; In Steps 4–5, the selected attributes are checked to delete redundant attributes with the acceleration strategy. So the time complexity of Steps 4–5 is $O(|C||U'|)$. Hence, the total time complexity of algorithm IRACR-S is $3O(|C||U'|) + O(|C-R|^2|\mathrm{CON}_R(D,U')|)$.

The performance analysis We can find that IRACR-S improves the efficiency of the incremental reduction algorithm compared with IRACR-M when adding multiple objects into the decision system.

(1) For IRACR-S, we only need to execute Step 1 at a time to update the conflict region.
(2) For adding multiple objects, in Step 3, we only need to run the 'while' loop-structure once to obtain the selected attributes, while in IRACR-M, for adding every object of the object set, we should run the 'while' loop-structure once.

Example 5 (Continuation of Example 1) For Table 1, $R = \{ab\}$ is a reduct of the decision table S; we have $U/R = \{\{x_1, x_7\}, \{x_2, x_3, x_9\}, \{x_4, x_8\}, \{x_5, x_6\}\}$ and $U/C = \{\{x_1, x_7\}, \{x_2\}, \{x_3, x_9\}, \{x_4, x_8\}, \{x_5\}, \{x_6\}\}$. Let $X_1 = \{x_1, x_7\}$, $X_2 = \{x_2, x_3, x_9\}$, $X_3 = \{x_4, x_8\}$, $X_4 = \{x_5, x_6\}$, $Y_1 = \{x_1, x_7\}$, $Y_2 = \{x_2\}$, $Y_3 = \{x_3, x_9\}$,$Y_4 = \{x_4, x_8\}$, $Y_5 = \{x_5\}$ and $Y_6 = \{x_6\}$; we add an object set $U_{new} = \{y_1, y_2, y_3\}$ into Table 1, where $y_1 = (0, 1, 0, 0, 1)$, $y_2 = (0, 1, 0, 1, 1)$, $y_3 = (1, 0, 1, 0, 2)$.

For IRACR-S, we have $U' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, y_1, y_2, y_3\}$ and can get $U'/R = \{\{x_5, x_6\}, \{x_1, x_7\}, \{x_4, x_8, y_3\}, \{x_2, x_3, x_9, y_1, y_2\}\}$ and $U'/C = \{\{x_6\}, \{y_2\}, \{x_3, x_9\}, \{x_1, x_7\}, \{x_2, y_1\}, \{x_4, x_8, y_3\}\}$. Obviously, $U'/R \bigcup \{c\} = \{\{x_5\}, \{x_6\}, \{x_3, x_9\}, \{x_1, x_7\}, \{x_2, y_1, y_2\}, \{x_4, x_8, y_3\}\}$ and $U'/R \bigcup \{d\} = \{\{x_5\}, \{x_6\}, \{x_1, x_7\}, \{x_4, x_8, y_3\}, \{x_2, x_3, x_9\}, \{y_1, y_2\}\}$. We can get $SIG_{\mathrm{CON}}{}^{+}(c, R, D, U') = 2$ and $SIG_{\mathrm{CON}}{}^{+}(d, R, D, U') = 0$. Thus, we select c to merge into $R = \{ab\} \bigcup \{c\} = \{abc\}$ and update $\mathrm{CON}_R(D, U')$ to get $|\mathrm{CON}_R(D, U')| = 8$. Next, we have $U'/\{abc\} = \{\{x_5\}, \{x_6\}, \{x_3, x_9\}, \{x_1, x_7\}, \{x_4, x_8, y_3\}, \{x_2, y_1, y_2\}\}$ and select d to merge into $R = \{abc\} \bigcup \{d\} = \{abcd\}$. According to Steps 4–5, we find that the attribute a in R is a redundant attribute and must be deleted. Hence, the new reduct is $R = \{bcd\}$.

Table 2 gives the comparison of the time complexity for static and incremental algorithms when adding multiple objects.

Step(1) is to compute the core attribute set and $\mathrm{CON}_C(D, U)$ for SRACR and to update $\mathrm{CON}_C(D, U')$ and $\mathrm{CON}_R(D, U')$ for IRACR-M and IRACR-S; Step(2) is to add the attributes into the subset R; Step(3) is to delete the redundant attributes from the subset R.

From Table 2, when adding l objects into the decision table, we can find that the time complexity of IRACR-M is smaller than that of SRACR and that the time complexity of IRACR-S is smaller than that of IRACR-M. Hence, we can draw the conclusions that the proposed incremental algorithm IRACR-S is more efficient than the non-incremental algorithm SRACR and is also more efficient than the proposed incremental algorithm IRACR-M.

To further illustrate the efficiency of the proposed incremental reduction algorithms with the acceleration strategy, we present comparisons of the sorting number for four incremental reduction algorithms (IARC-S (Liang et al. 2014), GIARC-S (Liang et al. 2014), IRACR-M and IRACR-S) in Table 3.

In Table 3, for convenience, let θ denote $\theta = \sum_{i=0}^{|C-R|-1} (|C| - (|R| + i))$. Step(1) is to update $ME(D|C)$ and $ME(D|B)$ for IARC-S and GIARC-S and to update $\mathrm{CON}_C(D, U')$ and $\mathrm{CON}_R(D, U')$ for IRACR-M and IRACR-S; Step(2) is to add the attributes into the subset R; Step(3) is to delete the redundant attributes from the subset R.

From Table 3, because of using the acceleration strategy, the sorting numbers of IRACR-M and IRACR-S are both $2|R|$ in step(3), which are less than one of IARC-S and GIARC-S. Hence, the sorting number of IRACR-M is less than one of IARC-S and the sort time of

Table 2 The comparison of time complexity for static and incremental algorithms when adding *l* objects

Steps	SRACR	IRACR-M	IRACR-S
Step(1)	$O(\vert C\vert^2\vert U'\vert) + O(\vert\text{CORE}\vert\vert U'\vert)$	$l^* O(\vert C\vert\vert U\vert) + O(\vert X'_i\vert)$	$O(\vert C\vert\vert U'\vert)$
Step(2)	$\sum_{i=1}^{\vert C-\text{CORE}\vert} \vert U'\vert(\vert C - \text{CORE}\vert\text{-}i + 1)$	$l^* O(\vert C - R\vert\vert X'_i\vert)$	$O(\vert C - R\vert^2\vert\text{CON}_R(D, U')\vert)$
Step(3)	$O(\vert C\vert^2\vert U'\vert)$	$O(\vert C\vert\vert U' + 1\vert)$	$O(\vert C\vert\vert U'\vert)$
Total	$2O(\vert C\vert^2\vert U'\vert) + O(\vert\text{CORE}\vert\vert U'\vert) + \sum_{i=1}^{\vert C-\text{CORE}\vert} \vert U'\vert(\vert C-\text{CORE}\vert\text{-}i+1)$	$l^*(O(\vert C\vert\vert U\vert)+O(\vert X'_i\vert)+ O(\vert C-R\vert\vert X'_i\vert))+O(\vert C\vert\vert U'+1\vert)$	$3O(\vert C\vert\vert U'\vert)+ O(\vert C - R\vert^2\vert\text{CON}_R(D, U')\vert)$

Table 3 Comparison of sorting number among four incremental reduction algorithms

Steps	IARC-S	GIARC-S	IRACR-M	IRACR-S
Step(1)	$2l^{*}\lvert C\rvert$	$2\lvert C\rvert$	$2l^{*}\lvert C\rvert$	$2\lvert C\rvert$
Step(2)	$l^{*}(\lvert C-R\rvert+\theta)$	$\lvert C-R\rvert+\theta$	$l^{*}(\lvert C-R\rvert+\theta)$	$\lvert C-R\rvert+\theta$
Step(3)	$\lvert R\rvert^{2}$	$\lvert R\rvert^{2}$	$2\lvert R\rvert$	$2\lvert R\rvert$
Total	$l^{*}(3\lvert C\rvert-\lvert R\rvert+\theta)+\lvert R\rvert^{2}$	$3\lvert C\rvert+\lvert R\rvert^{2}-\lvert R\rvert+\theta$	$l^{*}(3\lvert C\rvert-\lvert R\rvert+\theta)+2\lvert R\rvert$	$3\lvert C\rvert+\lvert R\rvert+\theta$

Table 4 Descriptions of data sets

ID	Data set	# of objects	# of attributes	# of core attributes	# of classes
a	Led17	2000	22	14	10
b	Chess	3196	36	27	2
c	Landsat-trn	4435	37	0	7
d	Gene	3190	60	0	3
e	Semeion	1593	256	0	10
f	Mushroom	8124	22	0	2
g	Handwritten	5620	64	0	10
h	Ticdata2000	5822	85	9	2
i	Connect-4	67, 557	42	0	3

IRACR-S is lower than the sorting numbers of the other three algorithms. It can be known that the sort for the decision systems is a pivotal and frequent operation in the process of attribute algorithms. The computation of reducing the number of the sort can improve the efficiency of reduction algorithms. Hence, the proposed incremental reduction algorithms IARC-S and GIARC-S make use of the acceleration to further improve the efficiency of reduction algorithms.

6 Experiments

In this section, we present a series of experiments to demonstrate the effectiveness and efficiency of the proposed algorithms. The experiment data use 11 UCI data sets (https://archive.ics.uci.edu/ml/datasets.html) whose basic information is outlined in Table 4. All of the experiments have been carried out on a personal computer with Windows XP, Pentium Dual-CORE CPU i5-3470 3.20GHZ and 3.47GB memory. The software used is Microsoft Visual Studio 2005, and the programming language is C++.

6.1 A comparison of static and incremental reduction algorithms when adding data

To evaluate the efficiency of the proposed incremental reduction algorithms, we compare the incremental reduction algorithms (IRACR-M, IRACR-S) with the static reduction algorithm SRACR when adding data sets. For each data set in Table 4, 50% of the objects of the original data set are selected as the basic data set. Then, we divide the remaining 50% of the objects into ten equal parts. The first part is regarded as the 1st data set, the combination of the first part and the second part is viewed as the 2nd data set, the combination of the 2nd data set and the third part is regarded as the 3rd data set, $\cdots$, the combination of all eight parts is viewed as the tenth data set, i.e., 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50% of the universe. Next, we add ten data sets into the basic data set and run the static reduction algorithm (SRACR) and two incremental reduction algorithms (IRACR-M and IRACR-S). The results are shown in Fig. 1. In each sub-figure (a–i) of Fig. 1, the x-coordinate pertains to the ratio of adding the data set, while the y-coordinate concerns the computational time.

It is easy to see from Fig. 1 that for each data set, the computational time of IRACR-M is much smaller than that of SRACR and the computational time of IRACR-S is much smaller

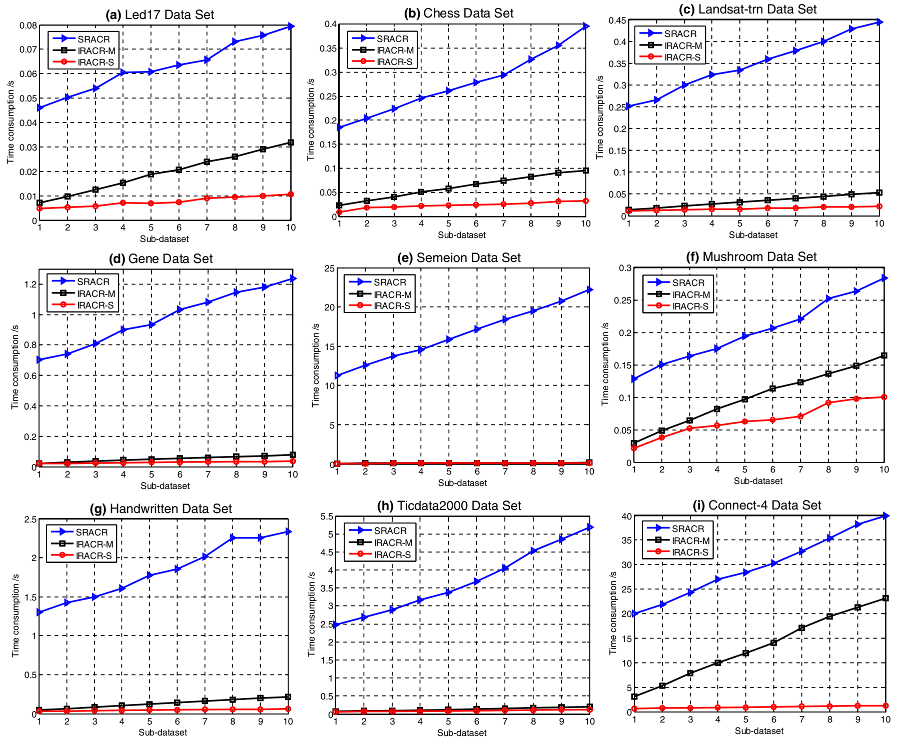

Fig. 1 The comparison of static and incremental algorithms when adding objects

than that of the incremental reduction algorithm IRACR-M. Namely, the performance of IRACR-M is higher than that of SRACR and the performance of IRACR-S is most efficient among the three algorithms.

From Fig. 1 it is easy to find that with the increment of the condition attributes number, the slope of the curve for algorithms IRACR-M and IRACR-S decreases. For example, the slopes of the data sets Gene, Ticdata2000 and Semeion are close to zero due to including more condition attributes.

To more clearly compare the efficiency between IRACR-M and IRACR-S, for data sets Landsat-trn, Gene, Semeion, Ticdata2000 and Handwritten, Fig. 2 gives the more detailed trend of IRACR-M and IRACR-S as the size of the data sets increases. We can see from these figures that IRACR-S takes less time to select a new feature subset than IRACR-M. For example, the computational time of IRACR-M achieves 0.076 s for the Gene data set, while IRACR-S needs 0.0345 s. The main reason is that IRACR-M needs to be carried out repeatedly to handle multiple objects, while IRACR-S only deals with multiple objects at a time instead of repetitive operations.

6.2 Comparison with reduct results of static and incremental reduction algorithms

To compare the results of incremental and static reduction algorithms, for each data set in Table 4, we randomly select 50% of the objects as the basic data set. Then, the remaining

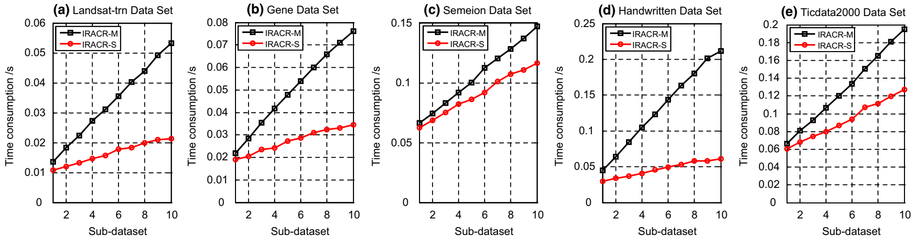

Fig. 2 The comparison of two incremental algorithms when adding objects

50% of the objects are added into the corresponding basic data set. The generated reducts are shown in Table 5.

It is easy to see from Table 5 that the results of reducts for SRACR, IRACR-M and IRACR-S are not entirely identical; the reason is that the different orders of objects being added into the decision table result in different significances of the selected features in SRACR, IRACR-M and IRACR-S such that the feature subsets selected by the reduction algorithms are not wholly the same.

Moreover, we can observe that IRACR-S can find relatively smaller or equal-sized attributes reduct than IRACR-M. For example, for data sets Mushroom, Ticdata2000 and Semeion, the sizes of attributes in the reduct by IRACR-S are less than that of IRACR-M. This is because for IRACR-S, the selected most significant attributes apply to the whole adding object set, while the selected most significant attributes only rely on the current one object for IRACR-M. Relative to IRACR-M, IRACR-S can more fully select the significant attributes from the integrity of the adding object set, which results in one needing to select smaller attributes than IRACR-M.

6.3 Performance comparison of the essential step for the static and incremental reduction algorithm

When an object set is added, the static reduction algorithm includes the following three steps.

Step 1 Compute the core attributes set;

Step 2 Select the most significant attributes to add into the alternative attribute subset;

Step 3 Delete the redundant attributes from the alternative attribute subset.

The mains steps of the incremental reduction algorithm are as follows.

Step 1 Select the most significant attributes to add into the alternative attribute subset;

Step 2 Delete the redundant attributes from the alternative attribute subset.

To concretely illustrate the performance of the static algorithm and incremental algorithms when an object set is entered into the decision system, we compare the time consumed for the main steps of three attribute reduction algorithms (SRACR, IRACR-M and IRACR-S) for each data set in Table 4. We select 60% of the objects of the original data set as the basic data set and add 10% of the objects of the universe into the basic data set. The details of the computational time of the main steps of the static and incremental reduction algorithms for nine data sets are shown in Table 6 and Fig. 3 . In Table 6, 'Core' denotes the time consumption of computing core attributes, 'Add' denotes the time consumption of adding the attributes into the reduct, 'Del' denotes the time consumption of deleting the redundant attributes, and 'Total' denotes the whole time consumption of the reduction algorithm.

Table 5 Comparison of reduct results for three algorithms

Data sets	Initial reduct		SRACR		IRACR-M		IRACR-S	
	Nos	reduct	Nos	reduct	Nos	reduct	Nos	reduct
Led17	17	{2,3,5,8,9,10,12,13,14,15,16,17, 18,19,20,21,22}	18	{1,2,3,7,8,9,10,11,12,13,14, 15,16,17,18,19,20,21}	18	{1,4,5,8,9,10,11,12,13,14,15, 16,17,18,19,20,21,22}	18	{2,3,7,8,9,10,11,12,13,14,15, 16,17,18,19,20,21,22}
Chess	23	{1,3,4,5,6,7,10,11,12,13,15,16,17, 18,21,23,26,30,31,33,34,35,36}	29	{1,3,4,5,6,7,9,10,12,13,15,16, 17,18,20,21,23,24,25,26,27, 28,30,31,32,33,34,35,36}	29	{1,2,3,4,5,6,7,9,10,11,12,13, 15,16,17,18,20,21,23,24,25, 26,27,30,31,33,34,35,36}	29	{1,3,4,5,6,7,9,10,11,12,13,15, 16,17,18,20,21,23,24,25,26, 27,28,30,31,33,34,35,36}
Landsat-trn	7	{1,3,17,27,30,34,36}	6	{4,9,12,18,25,36}	8	{2,3,4,17,27,30,34,36}	8	{2,3,4,17,27,30,34,36}
Mushroom	1	{5}	5	{3,5,13,18,20}	7	{1,2,3,4,5,7,9}	5	{3,5,13,18,20}
Gene	10	{2,4,14,21,28,29,31,32,35,36}	10	{5,11,13,21,25,29,30,32,35,36}	10	{1,2,4,14,21,28,29,32,35,36}	10	{1,2,4,14,21,28,29,32,35,36}
Handwritten	7	{5,6,19,36,42,47,51}	8	{3,4,14,27,42,51,52,54}	8	{3,4,5,6,19,42,47,51}	8	{1,5,6,19,36,42,47,51}
Ticdata2000	18	{1,2,4,9,15,18,44,47,59,66,68,70, 75,76,78,80,82,83,85}	22	{1,2,7,18,38,39,44,45,47,55, 59,68,69,70,75,78,79,80,82, 83,84,85}	25	{1,2,4,7,9,15,16,18,44,47,48, 55,58,59,63,66,68,70,75,76, 78,80,82,83,85}	20	{1,2,4,8,9,15,18,44,47,59,66, 68,70,75,76,78,80,82,83,85}
Semeion	19	{5,8,12,36,44,69,107,112,114, 135,152,154,158,162,207,230, 246,253,256}	22	{1,4,16,20,31,44,49,89,106, 109,111,113,135,139,145,153, 165,172,193,196,220,222}	27	{2,3,5,8,10,12,14,15,16,22, 25,36,44,69,107,112,114, 135,152,154,158,162,207, 230,246,253,256}	20	{1,5,8,12,36,44,69,107,112, 114,135,152,154,158,162, 207,230,246,253,256}
Connect-4	32	{1,2,3,4,6,7,8,9,10,11,13,14,15, 16,17,19,20,21,22,23,25,26,27, 29,31,32,33,35,37,38,39,40}	35	{1,2,3,4,6,7,8,9,10,12,13,14,15, 16,18,19,20,21,22,24,25,26,27, 28,29,31,32,33,34,36,37,38,39, 40,42}	32	{1,2,3,4,6,7,8,9,10,11,13,14, 15,16,17,19,20,21,22,23,25, 26,27,29,31,32,33,35,37,38, 39,40}	32	{1,2,3,4,6,7,8,9,10,11,13,14, 15,16,17,19,20,21,22,23,25, 26,27,29,31,32,33,35,37,38, 39,40}

Table 6 Comparison of performances of three reduction algorithms

Data sets	SRACR				IRACR-M			IRACR-S		
	Core	Add	Del	Total	Add	Del	Total	Add	Del	Total
Led17	0.0408	0.0159	0.0227	0.0794	0.0268	0.0049	0.0317	0.0059	0.0046	0.0105
Chess	0.1400	0.1914	0.0642	0.3956	0.0805	0.0153	0.0958	0.0182	0.0143	0.0325
Landsat-trn	0.2932	0.1136	0.0376	0.4445	0.0481	0.0050	0.0532	0.0162	0.0049	0.0212
Gene	0.7052	0.3620	0.1706	1.2380	0.0651	0.0109	0.0760	0.0272	0.0072	0.0344
Semeion	13.7412	5.3291	3.1365	22.2066	0.1275	0.0197	0.1472	0.0994	0.0169	0.1164
Mushroom	0.1820	0.0721	0.0299	0.2840	0.1569	0.0072	0.1642	0.0953	0.0050	0.1004
Handwritten	1.7050	0.5030	0.1274	2.3354	0.2019	0.0099	0.2118	0.0515	0.0094	0.0610
Ticdata2000	3.6927	1.2407	0.2569	5.1904	0.1598	0.0354	0.1952	0.0998	0.0273	0.1271
Connect-4	14.3963	15.8668	9.6327	39.8959	12.1743	1.0418	13.2162	0.5892	0.6885	1.2778

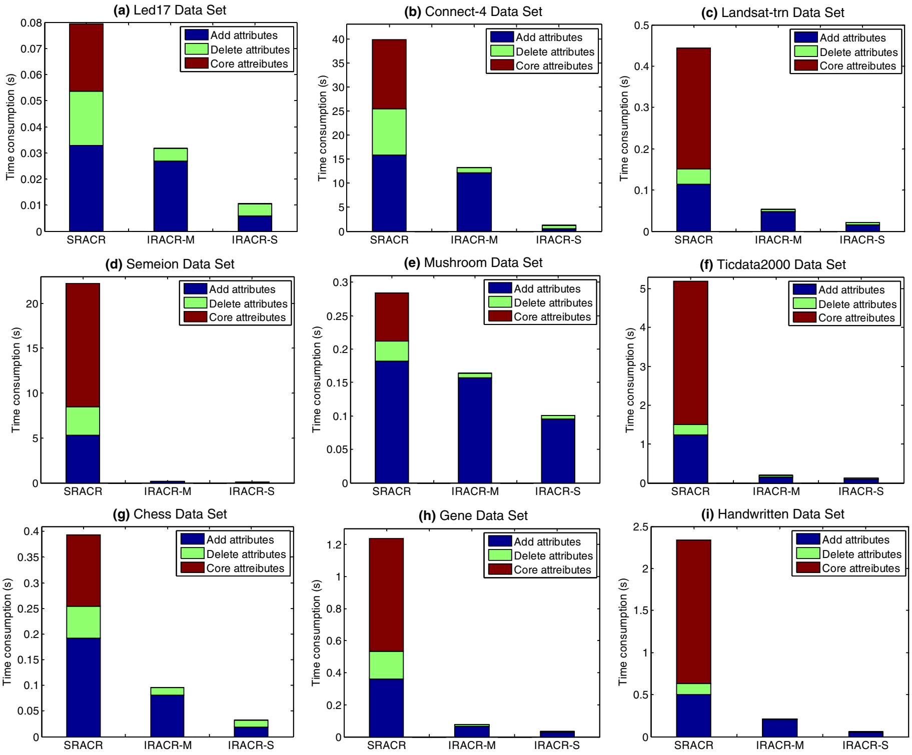

Fig. 3 The comparison of the running time for the static and incremental algorithms of the main steps

Each column denotes the whole computational time of the reduction algorithm. In the sub-figures of Fig. 3, the red part denotes the running time of computing core attributes for the static algorithm, the blue part represents the running time of selecting attributes in the reduct, and the green part denotes the running time of deleting redundant attributes.

It is clear that the static reduction algorithm spends a lot of time on computing core attributes. Meanwhile, for the incremental reduction algorithm, the computational time of selecting attributes and deleting redundant attributes is less than that of the static reduction algorithm. These reasons illustrate that the incremental reduction algorithm is faster than the static reduction algorithm with respect to finding the reduct.

6.4 The performance comparison with other incremental algorithms

To further illustrate the efficiency of the IRACR-M and IRACR-S algorithms, in this subsection, our proposed algorithms are compared with other existing incremental reduction algorithms. We select the incremental reduction algorithm based on the positive region in (Hu et al. 2005), denoted as IRPR, and only select four incremental reduction algorithms (IARC-S and GIARC-S, IARC-C and GIARC-C) (Liang et al. 2014). IARC-S and IARC-C are incremental algorithms for reduct based on Shannon's entropy and the combination entropy, respectively. GIARC-S and GIARC-C are group incremental algorithms for reduct based on Shannon's entropy and the combination entropy, respectively. Actually, the algorithms IRACR-M, IARC-S and IARC-C are the incremental reduction algorithms for adding one

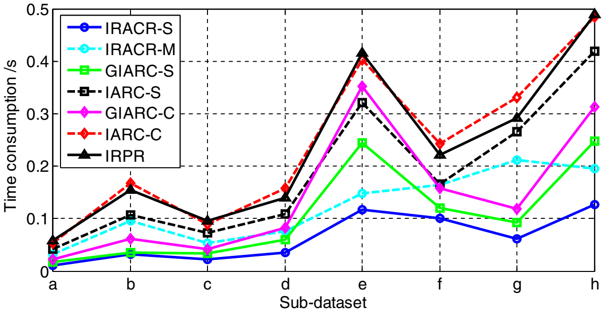

Fig. 4 The comparison of the running time with other incremental algorithms

object, and the algorithms IRACR-S, GIARC-S and GIARC-C are the group incremental reduction algorithms for adding the object set.

For each data set in Table 4, we randomly select 50% of the objects as the basic data set. Then, the remaining 50% of objects are added to the corresponding basic data set. The comparison results of the computational time are shown in Table 7 and Fig. 4.

From Table 7 and Fig. 4, we can easily find that the computational time of IRACR-M is much less than those of IARC-S and IARC-C and that the computational time of IRACR-S is much less than those of GIARC-S and GIARC-C. The computational times of IRACR-M and IRACR-S are both much less than that of IRPR. Hence, the experimental results indicate that IRACR-M is much more efficient than IARC-S and IARC-C and that IRACR-S is much more efficient than GIARC-S and GIARC-C.

6.5 Comparison of classification accuracies for five algorithms

To evaluate the classification performance with respect to generated reducts by the proposed incremental reduction algorithms (IRACR-M, IRACR-S), we use the results of the corresponding reduction algorithms to train the C4.5 classifier based on tenfold cross-validation method. IRACR-M, IRACR-S are used to select the attribute reducts when randomly selecting 50% of the objects as the basic data set and the remaining 50% of the objects as adding objects. The classification accuracy with respect to the raw data and the reduct result generated by different algorithms are shown in Table 8.

From Table 8, it can be seen that when adding objects into the decision systems, the average classification accuracy of the reduct selected by algorithms IRACR-M and IRACR-S is closed to that of algorithm SRACR on most data sets and the classification accuracy of IRACR-S are better than that of Initial reduct and SRACR on many data sets. The experimental results indicate that algorithm IRACR-M and IRACR-S can find a feasible reduct as algorithm SRACR and selecting a feasible reduct based on the incremental algorithms IRACR-M and IRACR-S can consume much shorter computational time.

7 Conclusions

In many real-world tasks, the objects of decision systems may vary dynamically. Quickly updating attribute reduction is a significant task for knowledge discovery. In this paper, firstly, the concept and properties of the conflict region are presented. When a single object and an object set are added into the decision systems, the incremental mechanisms for the conflict

Table 7 Comparison with other incremental algorithms

Algorithms	Led17	Chess	Landsat-trn	Gene	Semeion	Mushroom	Handwritten	Ticdata2000	Connect-4
IRACR-S	0.01054	0.03256	0.02128	0.03446	0.11643	0.10047	0.06103	0.12716	1.27786
IRACR-M	0.03176	0.11864	0.05323	0.07607	0.14729	0.16420	0.21186	0.19528	23.18633
GIARC-S	0.01663	0.03515	0.03352	0.06034	0.24349	0.11939	0.09287	0.24746	10.47731
IARC-S	0.04177	0.10637	0.07306	0.10929	0.32127	0.16655	0.26610	0.41945	34.14973
GIARC-C	0.02157	0.06185	0.04161	0.08312	0.35262	0.15769	0.11914	0.31274	11.35989
IARC-C	0.05357	0.16673	0.08860	0.15750	0.40285	0.24285	0.33035	0.48378	37.23140
IRPR	0.05785	0.15382	0.09621	0.13905	0.41619	0.22143	0.29115	0.48916	41.94814

Table 8 Comparison of classification accuracy for three algorithms

Data sets	Initial reduct (%)	SRACR (%)	IRACR-M (%)	IRACR-S (%)
Led17	99.6812	99.7368	99.7603	**99.7831**
Chess	**99.4379**	**99.4379**	97.3094	**99.4379**
Landsat-trn	78.4456	**79.4363**	75.5584	77.4379
Mushroom	**100.000**	**100.000**	99.7361	**100.000**
Gene	61.9442	**74.5465**	72.8683	74.3578
Handwritten	63.9382	77.9141	**78.6121**	78.5691
Ticdata2000	89.7284	90.0213	89.9816	**90.0382**
Semeion	68.3203	71.3241	**71.6941**	71.4639
Connect-4	79.3531	80.1704	79.3642	**80.9251**

The bold numbers denote the highest accuracy

region are discussed. To further improve the efficiency of incremental reduction algorithms, we give an acceleration strategy and develop incremental feature selection algorithms with the acceleration strategy based on the conflict region in terms of adding a single object and an object set. Theoretical analysis and experimental results for UCI datasets have shown that the proposed algorithms can effectively reduce computational time to improve the performance of updating reduct. In the future, we will extend the acceleration strategy to the variation of the attribute set and discuss quickly incremental reduction algorithms when the attribute set varies.

Acknowledgements Research on this work is partially supported by the grants from the National Science Foundation of China (Nos. 51307011, 61402005), by the funds from Anhui Provincial Natural Science Foundation (Nos. 1508085MF126, 1508085MF127), by the fund from Anhui Education Natural Science Foundation (No. KJ2013A015), by the fund from Outstanding young talent foundation of Chuzhou University (No. 2013RC03), and by the fund from Initial Scientific Research of Chuzhou University (No. 2016qd07).

References

Chen HM, Li TR, Qiao SJ, Ruan D (2010) A rough sets based dynamic maintenance approach for approximations in coarsening and refining attribute values. Int J Intell Syst 25(10):1005–1026

Chen HM, Li TR, Ruan D (2012) Maintenance of approximations in incomplete ordered decision information systems. Knowl Based Syst 31:140–161

Chen HM, Li TR, Ruan D, Lin JH, Hu CX (2013) A rough set based incremental approach for updating approximations under dynamic maintenance environments. IEEE Trans Knowl Data Eng 25(2):274–284

Fan YN, Tseng TL, Chern CC, Huang CC (2009) Rule induction based on an incremental rough set. Expert Syst Appl 36(9):11439–11450

Ge H, Li LS, Yang CJ (2012) A efficient attribute reduction algorithm based on conflict region. Chin J Comput 35(2):342–350

Ge H, Li LS, Xu Y, Yang CJ (2015) Bidirectional heuristic attribute reduction based on conflict region. Soft Comput 19(7):1973–1986

Hu XH, Cercone N (1995) Learning in relational databases a rough set approach. Int J Comput Intell 11(2):323–338

Hu F, Wang GY, Huang H, Wu Y (2005) Incremental attribute reduction based on elementary sets. In:10th International conference on rough sets fuzzy sets data mining and granular computing, pp 185–193

Hu QH, Xie ZX, Yu DR (2007) Hybrid attribute reduction based on a novel fuzzy rough model and information granulation. Pattern Recognit 40:3509–3521

Huang CC, Tseng TL, Fan YN, Hsu CH (2013) Alternative rule induction methods based on incremental object using rough set theory. Appl Soft Comput 13(1):372–389

Huang CC, Tseng TL, Tang CY (2016) Feature extraction using rough set theory in service sector application from incremental perspective. Comput Ind Eng 91:30–41

Jing YG, Li TR, Luo C, Horng SJ, Wang GY, Yu Z (2016a) An incremental approach for attribute reduction based on knowledge granularity. Knowl Based Syst 104:24–38

Jing YG, Li TR, Huang JF, Zhang YY (2016b) An incremental attribute reduction approach based on knowledge granularity under the attribute generalization. Int J Approx Reason 76:80–95

Lang GM, Ling QG, Yang T (2014) An incremental approach to attribute reduction of dynamic set-valued information systems. Int J Mach Learn Cybern 5(5):775–788

Li TR, Ruan D, Geert W, Song J, Xu Y (2007) A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowl Based Syst 20:485–494

Li RP, Yao YY (2010) Indiscernibility and similarity in an incomplete information table. In: 5th International conference of rough set and knowledge technology, pp 15–17

Li SY, Li TR, Liu D (2013) Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set. Knowl Based Syst 40:17–26

Li SY, Li TR (2015) Incremental update of approximations in dominance-based rough sets approach under the variation of attribute values. Inf Sci 294:348–361

Liang JY, Xu ZB (2002) The algorithm on knowledge reduction in incomplete information systems. Int J Uncertain Fuzziness 10(1):95–103

Liang JY, Wang F, Dang CY, Qian YH (2014) A group incremental approach to feature selection applying rough set technique. IEEE Trans Knowl Data Eng 26(2):294–308

Liu ZT (1999) An incremental arithmetic for the smallest reduction of attributes. Acta Electronica Sinica 27(11):96–98

Liu Y, Xiong R, Chu J (2009) Quick attribute reduction algorithm with hash. Chin J Comput 32(8):1493–1499

Miao DQ, Hu GR (1999) An heuristic algorithm of knowledge reduction. Chin J Comput Res Dev 36(6):681–684

Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356

Qian YH, Liang JY, Wang F (2009) A new method for measuring the uncertainty in incomplete information systems. Int J Uncertain Fuzziness 17(6):855–880

Qian YH, Liang JY, Pedrycz W, Dang CY (2010) Positive approximation: an accelerator for attribute reduction in rough set theory. Artif Intell 174:597–618

Qian J, Miao DQ, Zhang ZH, Li W (2011) Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. Int J of Approx Reason 52(2):212–230

Shen Q, Jensen R (2004) Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring. Pattern Recognit 37(7):1351–1363

Shu WH, Shen H (2014a) Updating attribute reduction in incomplete decision systems with the variation of attribute set. Int J Approx Reason 55:867–884

Shu WH, Shen H (2014b) Incremental feature selection based on rough set in dynamic incomplete data. Pattern Recognit 47:3890–3906

Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. Intell Decis Support 11:331–362

Susmaga R, Słowiński R (2015) Generation of rough sets reducts and constructs based on inter-class and intra-class information. Fuzzy Set Syst 274:124–142

Teng SH, Zan DC, Sun JX (2010) Attribute reduction algorithm based on common discernibility degree. Chin J Pattern Recognit Artif Intell 23(1):630–638

Wang F, Liang JY, Qian YH (2013a) Attribute reduction: a dimension incremental strategy. Knowl Based Syst 39:95–108

Wang F, Liang JY, Dang CY (2013b) Attribute reduction for dynamic data sets. Appl Soft Comput 13:676–689

Wei W, Liang JY, Wang JH, Qian YH (2013) Decision-relative discernibility matrixes in the sense of entropies. Int J of Gen Syst 42(7):721–738

Xu YT, Wang LS, Zhang RY (2011) A dynamic attribute reduction algorithm based on 0–1 integer programming. Knowl Based Syst 24(8):1341–1347

Xu ZY, Liu ZP, Yang BY, Song W (2006) A quick attribute reduction algorithm with complexity of $\max(O(|C||U|)O(|C|^2|U/C|))$. Chin J Comput 29(3):611–615

Yang M (2007) An incremental updating algorithm for attributes reduction based on the improved discernibility matrix. Chin J Comput 30(5):815–822

Yao YY, Zhao Y (2009) Discernibility matrix simplification for constructing attribute reducts. Inf Sci 179(7):867–882

Zeng AP, Li TR, Liu D, Chen HM (2015) A fuzzy rough set approach for incremental feature selection on hybrid information systems. Fuzzy Set Syst 258(1):39–60

Zhang WX, Mi JS, Wu WZ (2003) Approaches to knowledge reductions in inconsistent systems. Int J Intell Syst 18(9):989–1000

Zhang JB, Li TR, Ruan D, Liu D (2012) Rough sets based matrix approaches with dynamic attribute variation in set valued information systems. Int J Approx Reason 53(4):620–635

Zhao Y, Yao YY, Luo F (2007) Data analysis based on discernibility and indiscernibility. Inf Sci 177(22):4959–4976

Zhou JW, Feng BQ, Liu Y (2007) New algorithm for quick computing core. Chin J Xi’an Jiaotong Univ 41(6):688–691