Information Sciences 536 (2020) 185–204

Contents lists available at ScienceDirect

Information Sciences

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

Incremental feature selection based on fuzzy rough sets

Peng Ni[a,b], Suyun Zhao[a,b,*], Xizhao Wang[c], Hong Chen[a,b], Cuiping Li[a,b], Eric C.C. Tsang[d]

[a] Key Lab of Data Engineering and Knowledge Engineering of MOE Renmin University of China, Beijing, China
[b] School of Information, Renmin University of China, Beijing,100872, China
[c] Shenzhen University, Shenzhen, Guangdong, 518061, China
[d] Macau University of Sciences and Technology, Macau, China

ARTICLE INFO

Article history:
Received 10 September 2019
Revised 12 April 2020
Accepted 15 April 2020
Available online 18 May 2020

Keywords:
Feature selection
Fuzzy rough set
Incremental learning
Information measure

ABSTRACT

Incremental feature selection can improve learning of accumulated data. We focus on incremental feature selection based on rough sets, which along with their generalizations (e.g., fuzzy rough sets), reduce dimensionality without requiring domain knowledge, such as data distributions. By analyzing the basic concepts of fuzzy rough sets on incremental datasets, we propose incremental mechanisms of information measure. Moreover, we introduce a key instance set containing representative instances to select supplementary features when new instances arrive. As the key instance set is much smaller than the whole dataset, the proposed incremental feature selection mostly suppresses redundant computations. We experimentally compare the proposed method with various non-incremental and two state-of-the-art incremental methods on a variety of datasets. The comparison results demonstrate that the proposed method achieves compact results with reduced computation time, especially on high-dimensional datasets.

1. Introduction

In this era of big data, as data are increasingly accumulated over time, researchers should develop novel analysis methods based on incremental learning [14,29]. A learning algorithm is considered as incremental if it generates hypotheses h_0, h_1, ..., h_n based on corresponding training data t_1, t_2, ..., t_n, where h_{i+1} only depends on h_i and current training data t_i. However, the resulting hypothesis is only applicable to the available training data [11]. Incremental learning reduces the space and time complexities regarding storage and processing, respectively [7]. In the last decades, incremental learning has been widely studied, obtaining methods such as incremental classification [2,3,10,12,32], incremental clustering [1], and incremental feature extraction and selection [22,25,35].

Incremental feature selection allows to handle sequentially arriving data or large datasets divided into sequentially processed subsets. This type of selection is important in incremental learning and applies to streaming data collected over time to update the selected representative features [18]. Incremental feature selection fully leverages historical information to substantially reduce the size of the training set [37]. Moreover, arriving data are only processed once, and historical results are subsequently combined. Feature selection can be roughly divided into wrappers, filters, and embedded algorithms [13,19]. Most existing incremental feature selection methods use the filter approach, which selects features regardless of the

* Corresponding author. .
E-mail address: zhao.suyun@yahoo.com (S. Zhao).

https://doi.org/10.1016/j.ins.2020.04.038
0020-0255/

learning/mining model and often serves as preprocessing step [4]. Nevertheless, incremental wrapper and hybrid approaches have not been widely explored.

Currently, incremental feature selection mainly focuses on either streaming features or streaming instances [36]. For streaming features [18,26,34,47], the selection assumes a fixed number of instances in training data and variable number of features over time. For streaming instances [15,20,38], the selection approach updates and maintains a feature subset using representative features for discriminating new instances from its current surroundings. In addition, a presumed data distribution determines the selection effectiveness. Alternatively, rough sets [23,24], which do not assume a data distribution, can be adopted for granular computing [21,41,44,45] to perform feature selection on streaming instances.

Based on existing rough set concepts [23,24], various methods handle sequentially arriving data to perform incremental feature selection [15,16,20,38]. These methods include entropy-based [20], matrix-based [15,38], and positive-region-based incremental feature selection [16]. When only a new instance arrives, Hu et al. [15,16] perform incremental feature selection based on either the modified discernibility matrix or positive region. Likewise, Yang et al. [38] perform incremental feature selection by updating the discernibility matrix. Wei et al. [35] achieve incremental feature selection using a compact decision table to improve efficiency. When a group of instances arrives, Liang et al. [20] use information entropy to establish a state-of-the-art entropy-based incremental feature selection algorithm.

The aforementioned rough-based methods share a common assumption that instances are discretized [30,33,46]. In real-world applications, however, there are many continuous features in datasets. A fuzzy rough set [9,17], which supports continuous-valued data, has been proposed to handle such features. Various incremental feature selection methods based on fuzzy rough sets have been subsequently proposed, such as a matrix-based method [39,40]. Yang et al. [39] use fuzzy rough sets in an incremental feature selection algorithm by discarding irrelevant instances and selecting representative arriving instances. This method establishes a state-of-the-art matrix-based reduction algorithm (MIAR). However, most matrix-based algorithms store all the historical discernibility matrices/pairs, being inapplicable when memory is limited. Therefore, efficient and effective feature selection algorithms considering storage limitations should be developed.

We propose an incremental feature selection algorithm using fuzzy rough sets. Our main contributions can be summarized as follows.

- Incremental mechanisms of positive region and dependency function are devised using concepts from fuzzy rough sets on incremental datasets.
- The key instance set is introduced. This set contains instances that allow to select representative features by updating previously obtained feature subset when new instances arrive.
- The positive-region-based incremental reduction algorithm (PIAR) is then developed using the key instance set. PIAR preserves previously obtained features (i.e., it prevents catastrophic forgetting) and learns additional features from the key instance set.

As the key instance set is much smaller than the whole dataset, the corresponding incremental feature selection prevents some redundant computations and alleviates computation requirements of storage and processing.

An accelerated attribute reduction method [27] also performs recursive updating based on positive regions. The main differences between the method in [27] and the proposed method are summarized as follows.

- The methods have different objectives; the one in [27] accelerates reduction algorithms based on all the available data, whereas the proposed method updates reduction using accumulated data (i.e., data accumulated with subsequently arriving instances).
- The methods adopt different tools; the method in [27] uses classical rough sets, whereas the proposed method uses fuzzy rough sets.
- The methods handle different types of datasets; the method described in [27] processes discrete data, whereas the proposed method processes continuous-valued data.

The remainder of this paper is organized as follows. Section 2 briefly presents rough sets, fuzzy rough sets, and an overview of feature selection algorithms based on fuzzy rough sets. In Section 3, we propose the method to determine the key instance set for incremental feature selection. In Section 4, we present the fuzzy-rough-set-based incremental feature selection method and its proposed improvement. The proposed method is compared with non-incremental and state-of-the-art incremental feature selection methods in Section 5. Finally, we draw conclusions and provide directions of future work in Section 6.

2. Preliminaries

Rough set theory, initially proposed by Pawlak [23,24], is an effective method for feature selection, rule extraction, and knowledge discovery. In rough set philosophy, each feature is called an attribute, and then feature selection is called attribute reduction. Here, we review rough set theory and its generalization to fuzzy rough sets [9,17,31]. In addition, we review three classical feature selection (attribute reduction) algorithms [28,39,42] and two state-of-the-art incremental feature selection algorithms based on (fuzzy) rough set techniques [20,39].

Table 1
Example of fuzzy decision table.

Instance	A	b	c	class
x_1	−0.4	−0.3	−0.5	0
x_2	−0.4	0.2	−0.1	1
x_3	−0.3	−0.4	−0.3	0
x_4	0.3	−0.3	0	1

2.1. Rough sets and fuzzy rough sets

Usually, data are described as a decision table, denoted by $DT = (U, C \cup D)$, where $U = \{x_1, x_2, \ldots, x_n\}$, called universe, is a nonempty set with finite number of instances. Each instance in U is described by a nonempty finite set of attributes, denoted by $C \cup D$, where C denotes the set of condition attributes, D denotes the set of decision attributes, and $C \cap D = \emptyset$. Each attribute $r \in C \cup D$ corresponds to a map $U \to V_r$, in which V_r is the value set of r over U. With every subset of attributes, $B \subseteq C$, the universe is split into q equivalence classes $U/B = \{X_1, X_2, \ldots, X_q\}$, where $U = \bigcup_{i=1}^{q} X_i$ and $X_i \cap X_j = \emptyset$ for any $i \neq j$. U/B is called a partition of B on U. In addition, $\forall X \subseteq U$, $\underline{B}X = \{X_i | X \subseteq X \text{ and } X_i \in U/B\}$ and $\bar{B}X = \{X_i | X_i \cap X \neq \emptyset \text{ and } X_i \in U/B\}$. Pair $(\underline{B}X, \bar{B}X)$ is called a rough set of X on attribute subset B. Clearly, rough set theory is only suitable for crisp attributes. Fuzzy rough sets generalize rough sets by combining fuzzy sets [43] and rough sets, and they are suitable for both continuous and crisp attributes [9,31].

Let A be a fuzzy subset on U defined as a mapping A: $U \to [0, 1]$. Then, $\forall x \in U$, $A(x) \in [0, 1]$ is the fuzzy membership degree of x belonging to fuzzy set A [43]. If each attribute $r \in C$ corresponds to a map $U \to [0, 1]$, each attribute is fuzzy instead of crisp. As each continuous attribute can be converted into a fuzzy attribute, the decision table with continuous attributes is called a fuzzy decision table and denoted by FD.

Some concepts and properties of fuzzy rough sets are briefly reviewed below. More details can be found in [9,17,31].

Given attribute subset $B \subseteq C$ and triangular norm T (see the appendix for the properties of the triangular norm), a fuzzy similarity relation on attribute subset B is defined as $R_B(\cdot, \cdot)$ for every $x, y, z \in U$ and satisfying the following properties:

(1) Reflexivity ($R_B(x, x) = 1$),
(2) Symmetry ($R_B(x, y) = R_B(y, x)$),
(3) T-transitivity ($R_B(x, y) \geq T(R_B(x, z), R_B(z, y))$).

In fact, each similarity relation $R_B(\cdot, \cdot)$ corresponds to an attribute subset B.

The fuzzy rough set was first proposed by Dubois and Prade [9], and it is defined as follows.

Definition 1. Let U be a nonempty universe and $R_B(\cdot, \cdot)$ be a fuzzy similarity relation on U. A fuzzy rough set is an ordered pair $(\underline{R_B}A, \overline{R_B}A)$ of fuzzy set A on U such that for every $x \in U$,

(1) $\underline{R_B}A(x) = \inf_{u \in U} \max\{1 - R_B(x, u), A(u)\}$,
(2) $\overline{R_B}A(x) = \sup_{u \in U} \min\{R_B(x, u), A(u)\}$.

$\underline{R_B}A$ and $\overline{R_B}A$ are the lower and upper approximation operators of A on attribute subset B, respectively.

In most practical applications, only the decision attributes are crisp, whereas the condition attributes are usually continuous. Therefore, we mainly focus on fuzzy decision tables with crisp decision attributes hereinafter.

Proposition 1. *Given fuzzy decision table $FD = (U, C \cup D)$ and $\forall X \subseteq U$, the lower approximation operator of X on attribute subset B can be simplified, $\forall x \in U$, as follows:*

$$\underline{R_B}X(x) = \begin{cases} \min\limits_{u \in U,\ u \notin X} \{1 - R_B(x, u)\}, & x \in X \\ 0, & x \notin X \end{cases}.$$

Proposition 1 gives the topological meaning of the lower approximation operator. That is, the lower approximation value of $x \in X$ is the smallest distance from x to $u \notin X$ [17]. Thus, to find the minimal distance, it is necessary to go through all the instances in the universe. Consequently, fuzzy rough sets hinder or even impede the computation of the rough approximation on accumulated data. Below, we provide an example of fuzzy rough sets.

Example 1. Let $U = \{x_1, x_2, x_3, x_4\}$, $E = \{a, b, c\}$, $U/D = \{X_1, X_2\}$, $X_1 = \{x_1, x_3\}$, and $X_2 = \{x_2, x_4\}$. The lower and upper approximation operators of X_1 on attribute subset E are $\underline{R_E}X_1(x) = \inf_{u \in U} \max\{1 - R_E(x, u), X_1(u)\}$ and $\overline{R_E}X_1(x) = \sup_{u \in U} \min\{R_B(x, u), X_1(u)\}$, respectively.

Considering Table 1, let $R_E(x, y) = \min_{r \in E}(R_r(x, y))$, where $R_r(x, y) = 1 - (\max(r(x), r(y)) - \min(r(x), r(y)))$, with $r(x)$, $r(y) \in [0, 1]$ representing the attribute values of instances x, y on attribute r, respectively.

For brevity, we only detail the computation steps for $R_E(x_1, x_2)$. $R_E(x_1, x_4)$ can be computed by following the same procedure. Denote $X(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases}$. Clearly, once $X_1(x_1) = 1$, $\max\{1 - R_E(x_1, x_1), X_1(x_1)\} = 1$ always holds.

$$
\begin{aligned}
&R_E(x_1, x_1) = R_E(x_1, x_3) = 1,\\
&R_a(x_1, x_2) = 1 - (\max(a(x_1), a(x_2)) - \min(a(x_1), a(x_2))) = 1 - (\max(-0.4, -0.4) - \min(-0.4, -0.4)) = 1,\\
&R_b(x_1, x_2) = 1 - (\max(b(x_1), b(x_2)) - \min(b(x_1), b(x_2))) = 1 - (\max(-0.3,\ 0.2) - \min(-0.3,\ 0.2)) = 0.5,\\
&R_c(x_1, x_2) = 1 - (\max(c(x_1), c(x_2)) - \min(c(x_1), c(x_2))) = 1 - (\max(-0.5, -0.1) - \min(-0.5, -0.1)) = 0.6,\\
&R_E(x_1, x_2) = \min_{r \in E}(R_r(x_1, x_2)) = \min\{R_a(x_1, x_2), R_b(x_1, x_2), R_c(x_1, x_2)\} = \min\{1,\ 0.5,\ 0.6\} = 0.5,\\
&R_E(x_1, x_4) = 0.3,
\end{aligned}
$$

$$
\begin{aligned}
&\underline{R_E}X_1(x_1) = \inf_{u \in U} \max\{1 - R_E(x_1, u), X_1(u)\}\\
&= \min\{\ \max\{1 - R_E(x_1, x_1), X_1(x_1)\}, \max\{1 - R_E(x_1, x_2), X_1(x_2)\}, \max\{1 - R_E(x_1, x_3), X_1(x_3)\}, \max\{1 - R_E(x_1, x_4), X_1(x_4)\}\ \}\\
&= \min\{\ \max\{1 - R_E(x_1, x_1),\ 1\},\ \max\{1 - 0.5, 0\},\ \max\{1 - R_E(x_1, x_3),\ 1\},\ \max\{1 - 0.3,\ 0\}\ \}\\
&= \min\{1,\ 0.5,\ 1,\ 0.7\}\\
&= 0.5,\\
&\overline{R_E}X_1(x_1) = \sup_{u \in U} \min\{R_E(x_1, u), X_1(u)\}\\
&= \max\{\min\{R_E(x_1, x_1), X_1(x_1)\}, \min\{R_E(x_1, x_2), X_1(x_2)\}, \min\{R_E(x_1, x_3), X_1(x_3)\}, \min\{R_E(x_1, x_4), X_1(x_4)\}\}\\
&= \max\{\ \min\{1,\ 1\},\ \min\{0.5,\ 0\},\ \min\{1,\ 1\},\ \min\{0.3,\ 0\}\ \}\\
&= 1.
\end{aligned}
$$

Definition 2. In fuzzy decision table $FD = (U, C \cup D)$, the positive region of D relative to C is defined as $POS_C^U(x) = \underline{R_C}([x]_D)(x)$ for every $x \in U$, and the dependency degree of D on C is defined as $\gamma_C^U = \sum_{x \in U} POS_C^U(x)/|U|$, where $[x]_D = \{y \in U :\ R_D(x, y) = 1\}$ represents the set containing the instances in U with the same decision classes of x.

Positive region $POS_C^U(x)$ measures the discernibility of D relative to C for each instance $x \in U$. Dependency function γ_C^U measures the discernibility of D relative to C on the universe.

Based on Proposition 1, the property of the positive region is described as follows.

Proposition 2. *Given fuzzy decision table $FD = (U, C \cup D)$, the positive region of D relative to C can be simplified as follows:* $\forall x \in U$, $POS_C^U(x) = \min_{u \in U,\ u \notin [x]_D} \{1 - R_C(x, u)\}$.

Proposition 2 describes the relation between the positive region and lower approximation.

Definition 3. In fuzzy decision table $FD = (U, C \cup D)$, $B \subseteq C$ is called a reduct of C with respect to D if B satisfies the following statements:

(1) $\gamma_C^U = \gamma_B^U$,
(2) for any $r \in B$, $\gamma_C^U \neq \gamma_{B-\{r\}}^U$.

To design a feature selection algorithm, it is necessary to determine the increment of the dependency degree with gradually increasing attributes. Thus, the dependency degree and positive region can be described by Proposition 3.

Proposition 3. *If $P \subseteq Q \subseteq C$, then*

(1) $POS_P^U(x) \leq POS_Q^U(x) \leq POS_C^U(x)$,
(2) $\gamma_P^U \leq \gamma_Q^U \leq \gamma_C^U$.

Proposition 3 shows that the dependency function is monotonic with gradually arriving attributes. Thus, Proposition 3 verifies the feasibility of the forward feature selection (attribute reduction) algorithm. Note that forward feature selection (attribute reduction) means to add the most representative attributes successively to the candidate reduct until the dependency degree reaches its maximum.

2.2. Existing static reduction algorithms and incremental feature selection algorithms

Here, we briefly review three known static reduction algorithms [28,39,42] and two state-of-the-art incremental feature selection algorithms [20,39]. By clarifying and comparing the characteristics of these algorithms, we determine the necessity to develop an incremental feature selection (attribute reduction) algorithm based on the positive region. Using Proposition 3, the dependency-function-based reduction algorithm (DAR) is detailed in Algorithm 1 [42].

Besides DAR, there exist two static feature selection algorithms, namely, entropy-based feature selection algorithm (EAR) [20,28] and discernibility matrix-based selection algorithm (MAR) [39]. Although EAR is based on three kinds of entropy calculations, we only consider the conditional combination entropy proposed by Qian and Liang [28] as specific case. More details about entropy and EAR can be found in [20,28]. Note that EAR is only suitable for decision tables with crisp attributes, denoted by *DT*, because entropy is only appropriate for classical rough sets.

Algorithm 1 DAR

Input: $FD = (U, C \cup D)$
Output: *red*
Step 1: $B \leftarrow \emptyset$, $\gamma_B^U \leftarrow 0$
Step 2: $lef \leftarrow C$
Step 3: Calculate γ_C^U
Step 4: While $(\gamma_B^U < \gamma_C^U)$, do
$a^* = \underset{a \in lef}{\arg\max} \gamma_{B\cup\{a\}}^U$
$B \leftarrow B \cup \{a^*\}$
$lef \leftarrow lef - \{a^*\}$
Calculate γ_B^U
End while
Step 5: Let $red \leftarrow B$, $i = 0$
Step 6: For $(i = 0$ *to* $|B| - 1)$
Take i th attribute b_i in B
if $\gamma_{red-\{b_i\}}^U = \gamma_C^U$, then $red \leftarrow red - \{b_i\}$
End for
Step 7: Return *red*

Algorithm 2 EAR

Input: $DT = (U, C \cup D)$
Output: *red*
Step 1: $red \leftarrow \emptyset$
Step 2: For each attribute a in C
if $Sig^{\text{inner}}(a, C, D) > 0$, then $red \leftarrow red \cup \{a\}$
End for
Step 3: $B \leftarrow red$
Step 4: While $(CE(D|B) \neq CE(D|C))$, do
$Sig^{\text{outer}}(a^*, B, D) = \max\{Sig^{\text{outer}}(a, B, D)\}$ for $a \in C - B$
$B \leftarrow B \cup \{a^*\}$
End while
Step 5: $red \leftarrow B$
Step 6: Return *red*

Definition 4. Consider decision table $DT = (U, C \cup D)$, $B \subseteq C$, and partitions $U/B = \{X_1, X_2, \ldots, X_m\}$ and $U/D = \{Y_1, Y_2, \ldots, Y_n\}$. A conditional entropy of B relative to D is defined as $CE(D|B) = \sum_{i=1}^{m} \left(\frac{|X_i|}{|U|} \frac{C_{|X_i|}^2}{C_{|U|}^2} - \sum_{j=1}^{n} \frac{|X_i \cap Y_j|}{|U|} \frac{C_{|X_i \cap Y_j|}^2}{C_{|U|}^2} \right)$, where $C_{|X_i|}^2$ denotes the number of pairs of instances which are not distinguishable from each other in X_i.

Definition 5. Consider decision table $DT = (U, C \cup D)$ and $B \subseteq C$. $\forall a \in B$, the inner significance of a in B is defined as $Sig^{\text{inner}}(a, B, D) = CE(D|B - \{a\}) - CE(D|B)$. If $Sig^{\text{inner}}(a, B, D) > 0$, then attribute a is indispensable.

Definition 6. Consider decision table $DT = (U, C \cup D)$ and $B \subseteq C$. $\forall a \in C - B$, the outer significance of a in B is defined as

$$Sig^{\text{outer}}(a, B, D) = CE(D|B) - CE(D|B \cup \{a\})$$

Definition 7. Given decision table $DT = (U, C \cup D)$, $B \subseteq C$ be a reduct denoted by *red* if and only if:

(1) $CE(D|B) = CE(D|C)$,
(2) $\forall a \in B$, $CE(D|C) \neq CE(D|B - \{a\})$.

The heuristic algorithm to find a reduct based on the conditional combination entropy [20] is detailed in Algorithm 2. And the incremental EAR (EIAR) is detailed in Algorithm 3 [20].

The incremental computation of conditional combination entropy $CE_{U \cup \Delta U}(D|B)$ is given in [20], where more details of EIAR can be found. As EAR and EIAR are only suitable for crisp attributes, datasets with continuous attributes should be discretized before their application at the expense of information loss. The third classical feature selection (attribute reduction) method is the discernibility-matrix-based reduction algorithm, MAR [39], which can handle a fuzzy decision table with continuous attributes. The discernibility matrix, which is a discernibility measure, is defined as follows.

Definition 8. Given fuzzy decision table $FD = (U, C \cup D)$, the discernibility matrices of attribute a and attribute set C with respect to D are respectively defined as

(1) $DM(a) = \{(x_i, x_j) \in U \times U | 1 - R_a(x_i, x_j) \geq \underline{R_C}\ [x_i]_D(x_i)\ ,\ x_j \notin [x_i]_D\}$,
(2) $DM(C) = \cup_{a \in C} DM(a)$.

Definition 8 shows that each entry of the discernibility matrix of C contains the instance pairs that can be discerned by C. Thus, the discernibility matrix contains the discernibility information of the fuzzy decision table.

Algorithm 3 EIAR

Input: $DT = (U, C \cup D)$, ΔU, red
Output: *newred*
Step 1: $B \leftarrow red$, compute $U/B = \{X_1^B, X_2^B, ..., X_m^B\}$, $U/C = \{X_1^C, X_2^C, ..., X_s^C\}$, $\Delta U/B = \{M_1^B, M_2^B, ..., M_{m'}^B\}$, $\Delta U/C = \{M_1^C, M_2^C, ..., M_{s'}^C\}$
Step 2: Compute $(U \cup \Delta U)/B = \{X_1'^B, X_2'^B, ..., X_k'^B, X_{k+1}^B, X_{k+2}^B, ..., X_m^B, M_{k+1}^B, M_{k+2}^B, ..., M_{m'}^B\}$, $(U \cup \Delta U)/C = \{X_1'^C, X_2'^C, ..., X_{k'}'^C, X_{k'+1}^C, X_{k'+2}^C, ..., X_s^C, M_{k'+1}^C, M_{k'+2}^C, ..., M_{s'}^C\}$
Step 3: If $k = 0$ and $k' = 0$, go to step 4; otherwise, go to step 5
Step 4: Compute $CE_{\Delta U}(D|B)$ and $CE_{\Delta U}(D|C)$. If $CE_{\Delta U}(D|B) = CE_{\Delta U}(D|C)$, go to step 7; otherwise, go to step 5.
Step 5: While $(CE_{U\cup\Delta U}(D|B) \neq CE_{U\cup\Delta U}(D|C))$, do
$Sig^{outer}_{U\cup\Delta U}(a^*, B, D) = \max\{Sig^{outer}_{U\cup\Delta U}(a, B, D)\}$ for $a \in C - B$
$B \leftarrow B \cup \{a^*\}$
End while
Step 6: For each attribute a in B
if $Sig^{inner}_{U\cup\Delta U}(a, B, D) = 0$, then $B \leftarrow B - \{a\}$
End for
Step 7: *newred* $\leftarrow B$
Step 8: Return *newred*

Algorithm 4 MAR

Input: $FD = (U, C \cup D)$
Output: *red*
Step 1: $\forall x_i \in U$, compute $\underline{R_C}[x_i]_D(x_i)$
Step 2: For each condition attribute $a \in C$, compute its fuzzy discernibility matrix $DM(a)$ and $DM(C)$
Step 3: $\text{Core}_D(C) \leftarrow \emptyset$
For each $a \in C$, compute $DM(C - \{a\})$
If $DM(C - \{a\}) \neq DM(C)$, then $\text{Core}_D(C) \leftarrow \text{Core}_D(C) \cup \{a\}$
Step 4: $red \leftarrow \text{Core}_D(C)$, and $DM(a) \leftarrow DM(a) - DM(red)$ for $\forall a \notin red$
Step 5: While $(DM(red) \neq DM(C))$, do
add attribute $a^* \in C - red$ satisfying $|DM(a^*)| = \max_{a\in C-red} |DM(a)|$ into red
$DM(red) \leftarrow DM(red) \cup DM(a^*)$ and $DM(a) \leftarrow DM(a) - DM(a^*)$ for $\forall a \in C - red$
End while
Step 6: Return *red*

Algorithm 5 MIAR

Input: $FD = (U, C \cup D)$, ΔU, red
Output: *newred*
Step 1: $B \leftarrow red$
Step 2: For each condition attribute $a \in C$, compute its fuzzy discernibility matrix $DM'(a)$ and $DM'(C)$ in fuzzy decision table $(U \cup \Delta U, C \cup D)$.
Step 3: If $DM'(B) = DM'(C)$, go to **Step 5**
Step 4: While $(DM'(B) \neq DM'(C))$, do
add attribute $a^* \in C - B$ satisfying $|DM'(B \cup \{a^*\})| = \max_{a\in C-B} |DM'(B \cup \{a\})|$ into B
$DM'(B) \leftarrow DM'(B) \cup DM'(a^*)$, for $\forall a \in C - red$
End while
Step 5: While $(DM'(B) = DM'(C))$, do
select attribute $a \in B$ satisfying $DM'(B) = DM'(B - \{a\})$, and let $B = B - \{a\}$
End while
Step 6: *newred* $\leftarrow B$
Step 7: Return *newred*

Proposition 4. $\text{Core}_D(C) = \{a \in C : DM(C - \{a\}) \neq DM(C)\}$.

Proposition 4 implies that if an instance pair can be discerned by attribute a but not by the attributes in $C - \{a\}$, then attribute a is a core attribute. The reduct can be analogously defined by using a fuzzy discernibility matrix as follows.

Definition 9. Given fuzzy decision table $FD = (U, C \cup D)$, $B \subseteq C$ is a reduct if and only if:

(1) $DM(B) = DM(C)$,
(2) $\forall a \in B$, $DM(B - \{a\}) \neq DM(C)$.

The heuristic algorithm to find a reduction result using a fuzzy discernibility matrix [39] is detailed in Algorithm 4. And the incremental version of MAR (MIAR) is detailed in Algorithm 5 [39].

The incremental computation of relative discernibility relation $DM'(B)$ is defined in [39], where more details are available. Table 2 lists the characteristics of the three kinds of reduction algorithms. The computation time of DAR is high, thus being inefficient or even unfeasible on accumulated data. As no mature incremental dependency-degree-based reduction algorithm is currently available for notably speeding up DAR, this problem remains to be addressed.

Algorithm 6 DIAR (dependency-function-based attribute reduction algorithm)

Input: $FD = (U, C \cup D)$, ΔU, red, POS^U_{red}, POS^U_C
Output: *newred*
Step 1: $B \leftarrow red$
Step 2: Calculate $POS^{U\cup\Delta U}_B$, $POS^{U\cup\Delta U}_C$ $\gamma^{U\cup\Delta U}_B$, and $\gamma^{U\cup\Delta U}_C$ using Theorem 1
Step 3: Calculate ΔS_B
Step 4: While $(\gamma^{U\cup\Delta U}_B < \gamma^{U\cup\Delta U}_C)$, do
  $lef \leftarrow C - B$
  $a^* = \arg\max_{a_i \in lef} \gamma^{U\cup\Delta U}_{B\cup\{a_i\}}$
  $B \leftarrow B \cup \{a^*\}$
End while
Step 5: $newred \leftarrow B$
Step 6: For $(i = 0\ to\ |B| - 1)$
  Take i th attribute b_i in B
  If $\gamma^{U\cup\Delta U}_{newred-\{b_i\}} \geq \gamma^{U\cup\Delta U}_C$, then $newred \leftarrow newred - \{b_i\}$
End for
Step 7: Return *newred*.

Algorithm 7 PIAR (positive-region-based attribute reduction algorithm)

Input: $FD = (U, C \cup D)$, ΔU, red, POS^U_{red}, POS^U_C
Output: *newred*
Step 1: $B \leftarrow red$
Step 2: Calculate $POS^{U\cup\Delta U}_B$ and $POS^{U\cup\Delta U}_C$ using Theorem 1
Step 3: Calculate ΔS_B
Step 4: While $(|\Delta S_B| \neq 0)$, do
  $lef \leftarrow C - B$
  $a^* = \arg\max_{a \in lef} |\Delta t_{(B\cup\{a\})}|$
  $\Delta S_B \leftarrow \Delta S_B - \Delta t_{(B\cup a^*)}$
  $B \leftarrow B \cup \{a^*\}$
End while
Step 5: $newred \leftarrow B$
Step 6: For $(i = 0\ to\ |B| - 1)$
  Take i th attribute b_i in B
  if $\forall\ x \in (U \cup \Delta U)$, $POS^{U\cup\Delta U}_{newred-\{b_i\}}(x) \geq POS^{U\cup\Delta U}_C(x)$
then $newred \leftarrow newred - \{b_i\}$
End for
Step 7: Return *newred*.

Table 2
Comparison of reduction algorithms.

Algorithm	Discretization	Storage requirement	Computation time	Reduction size	Accuracy	Efficient incremental algorithm
MAR	No	High	Low	Suitable	Very high	Yes
EAR	Yes	Low	High	Small	Low	Yes
DAR	No	Low	**High**	Suitable	High	**No**

3. Key instances in dynamic fuzzy decision table

Let U denote the original universe, ΔU denote the set of arriving instances, and $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ denote a dynamicuzzy decision table. In this table, the lower approximation and reduct are different from those of the original table. Given a dynamic fuzzy decision table, the positive region and dependency degree can be recomputed on the whole decision table by using the static algorithm (Algorithm 1), but this solution is time consuming. As just some not all instances are key to conduct such computation, we propose the key instance set to quickly update the dependency degree and reduct.

3.1. Key instance set

In a dynamic fuzzy decision system, positive region $POS^{U\cup\Delta U}(x)$ has the following properties.

Theorem 1. *Given dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$, we have*

(1) if $x \in U$, then

$$POS^{U\cup\Delta U}_C(x) = \begin{cases} \min\limits_{u\in\Delta U, u\notin [x]_D} \{1 - R_C(x,u)\}, & \text{if } POS^U_C(x) > \min\limits_{u\in\Delta U, u\notin[x]_D} \{1 - R_C(x,u)\} \\ POS^U_C(x), & \text{otherwise} \end{cases}.$$

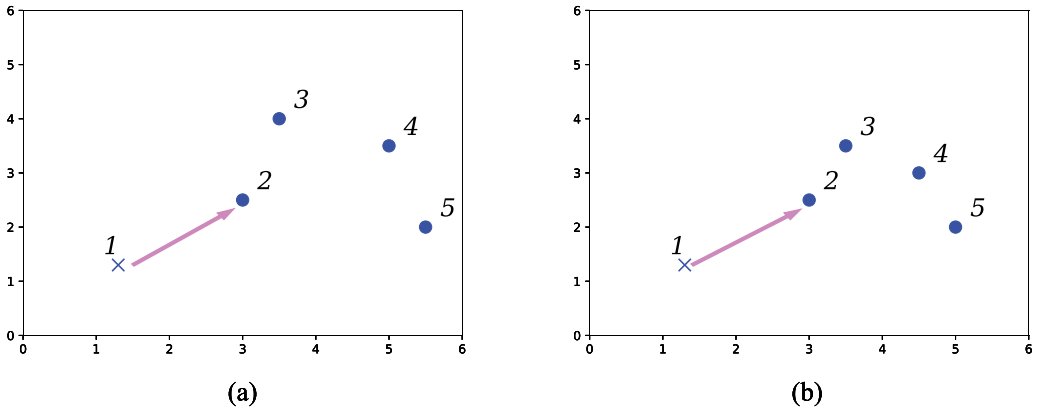

Fig. 1. Positive region before and after reduction for (a) C and (b) $B \subseteq C$.

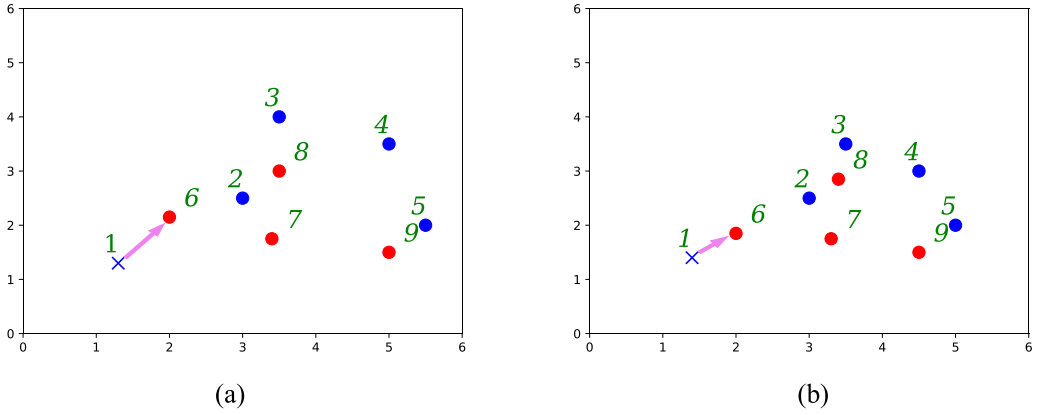

Fig. 2. Key instance set for (a) C and (b) $B \subseteq C$.

(2) if $x \in \Delta U$, then $POS_C^{U\cup\Delta U}(x) = \min\limits_{u\in(U\cup\Delta U),u\notin[x]_D}\{1-R_C(x,u)\}$.

Proof.

(1) By Proposition 2, if $x \in \Delta U$, the result is straightforward.
(2) If $x \in U$,

$$POS_C^{U\cup\Delta U}(x) = \min_{u\in(U\cup\Delta U),u\notin[x]_D}\{1-R_C(x,u)\} = \min\left\{\min_{u\in U,u\notin[x]_D}\{1-R_C(x,u)\}, \min_{u\in\Delta U,u\notin[x]_D}\{1-R_C(x,u)\}\right\}.$$

If $POS_C^U(x) > \min\limits_{u\in\Delta U,u\notin[x]_D}\{1-R_C(x,u)\}$, $POS_C^{U\cup\Delta U}(x) = \min\limits_{u\in\Delta U,u\notin[x]_D}\{1-R_C(x,u)\}$; otherwise, $POS_C^{U\cup\Delta U}(x) = POS_C^U(x)$.■

Theorem 1 mainly describes the change of the positive region according to the arriving instances. The positive region does not always change with these instances, and Theorem 1 allows to quickly update the positive region and prevent recomputing the positive region on the whole dataset. Moreover, $POS_C^{U\cup\Delta U}(x)$ is not always different from $POS_C^U(x)$ on all instances. Therefore, we collect the instances on which $POS_C^{U\cup\Delta U}(x)$ is different from $POS_C^U(x)$ to form a special set.

Definition 10. In dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U\cup\Delta U, C\cup D)$, $\Delta S_B = \{x \in U\cup\Delta U | POS_B^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)\}$ is called the key instance set of B in $FD^{U\cup\Delta U}$.

Definition 10 shows that the key instance set of B has the instances whose positive region values on B change when some instances arrive. Fig. 1 and 2 illustrate Definition 10.

In Fig. 1 and 2, the crossing points denote instances with positive label, and the dot points denote instances with negative label. The purple lines represent the minimal distance from instance 1 (with the x mark) to the instance represented by a dot. In Fig. 2, the blue and red points represent the original and arriving instances, respectively.

Point distributions on all attributes C and reduct B are respectively shown in Fig. 1(a) and (b). Fig. 1 shows that the minimal distance (i.e., purple line) does not change before and after reduction. Therefore, distinguishability does not change due to reduction. In contrast, the minimal distances change when new instances arrive, as shown in Fig. 2, in which the

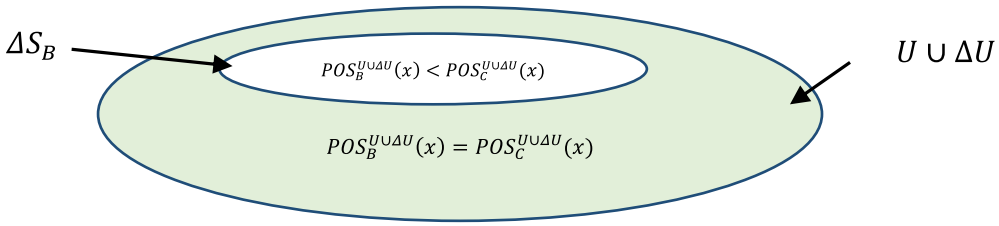

Fig. 3. Relation between key instance set and positive region.

minimal distances, indicated by purple line, become smaller. Therefore, B is not the reduct of C in Fig. 2, and it should be updated by supplementing new attributes for the purple lines in Fig. 2(a) and (b) to be equal. Thus, instance 1 is key for the selection of new attributes, and all such instances compose the key instance set.

3.2. Properties of key instance set

The key instance set has the following properties.

Proposition 5. *Given dynamic fuzzy decision table* $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ *and* $B \subseteq C$,

(1) $\forall x \in U \cup \Delta U - \Delta S_B$, $POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$,
(2) $POS_B^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$ if and only if $x \in \Delta S_B$.

Proof.

(1) By Proposition 3, we get $\forall x \in U\cup\Delta U$, $POS_B^{U\cup\Delta U}(x) \le POS_C^{U\cup\Delta U}(x)$. By Definition 10, $\forall x \in \Delta S_B$, we get $POS_B^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$. From these results, if $x \notin \Delta S_B$, then $POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$ (i.e., $\forall x \in U \cup \Delta U - \Delta S_B$, $POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$).
(2) By Definition 10, the result is straightforward. ■

Proposition 5 clearly shows that the key instance set consists of all instances that do not reach the maximum positive region values, as illustrated in Fig. 3.

Proposition 6. *Given dynamic fuzzy decision table* $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ *and* $B, P \subseteq C$, *the following statements always hold:*

(1) $\forall x \in U \cup \Delta U - \Delta S_B$, if $B \subseteq P$, then $POS_B^{U\cup\Delta U}(x) = POS_P^{U\cup\Delta U}(x)$,
(2) $\forall x \in \Delta S_B$, if $P \subseteq B$, then $POS_P^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$,
(3) $\forall x \in \Delta S_B$, if $B \subseteq P$, then $POS_B^{U\cup\Delta U}(x) \le POS_P^{U\cup\Delta U}(x) \le POS_C^{U\cup\Delta U}(x)$.

Proof.

(1) By Proposition 5, we get, $\forall x \in U \cup \Delta U - \Delta S_B$, $POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$. By Proposition 3, $\forall x \in U \cup \Delta U - \Delta S_B$ and $B \subseteq P$, $POS_B^{U\cup\Delta U}(x) \le POS_P^{U\cup\Delta U}(x)$. As $POS_B^{U\cup\Delta U}(x)$ already reaches its maximum, then $POS_P^{U\cup\Delta U}(x)$ must be equal to $POS_B^{U\cup\Delta U}(x)$.
(2) By Definition 10, we get, $\forall x \in \Delta S_B$, $POS_B^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$. By Proposition 3, $\forall x \in \Delta S_B$ and $P \subseteq B$, $POS_P^{U\cup\Delta U}(x) \le POS_B^{U\cup\Delta U}(x)$. Thus, $POS_P^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$ always holds.
(3) By Proposition 3, the result is straightforward.■

Proposition 6 shows the effect of adding and removing attributes on the positive region values. Proposition 6(1) guarantees that $\forall x \notin \Delta S_B$, $POS_B^{U\cup\Delta U}(x)$ already reaches its maximum and does not change with more attributes added to B. Thus, we can call $U \cup \Delta U - \Delta S_B$ as a positive region invariant set, which is the complementary of the key instance set. Furthermore, the statements (2)&(3) in Proposition 6 show that $\forall x \in \Delta S_B$, $POS_B^{U\cup\Delta U}(x)$ cannot reach its maximum with any attribute deleted from B. It is possible and feasible that $POS_B^{U\cup\Delta U}(x)$ reaches its maximum with one or more attributes added to B.

Proposition 7. *Given dynamic fuzzy decision table* $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ *and* $B, P, Q \subseteq C$, *the following two statements always hold:*

(1) $\emptyset \subseteq \Delta S_B \subseteq U \cup \Delta U$,
(2) if $P \subseteq Q \subseteq B$, then $\Delta S_P \supseteq \Delta S_Q \supseteq \Delta S_B$.

Proof.

(1) By Definition 10, the result is straightforward.
(2) By Proposition 3(1), $\forall x \in U\cup\Delta U$, if $P \subseteq Q \subseteq B$, then $POS_P^{U\cup\Delta U}(x) \le POS_Q^{U\cup\Delta U}(x) \le POS_B^{U\cup\Delta U}(x)$ and $POS_B^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x) \Rightarrow POS_Q^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x) \Rightarrow POS_P^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$. ■

Proposition 7(1) indicates the lower and upper boundaries of the key instance set. Proposition 7(2) shows that the key instance set is anti-monotonic with the addition of attributes. Hence, the key instance set becomes smaller with more attributes.

Based on the key instance set, the relative contribution of the attributes with respect to the obtained reduct in a dynamic fuzzy decision table can be measured using a new approach.

Definition 11. Given dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$, if ΔS_B is the key instance set of B, then $\forall a \in C - B$,

(1) $\forall t_{(B\cup\{a\})} = \{x \in \Delta S_B \mid POS^{U\cup\Delta U}_{B\cup\{a\}}(x) = POS^{U\cup\Delta U}_{C}(x)\}$ is called the incremental discernible instance set of a in B with respect to D,
(2) the cardinality of $\Delta t_{(B\cup\{a\})}$ is called the relative significance degree of a in B with respect to D.

Definition 11. shows that the incremental discernible instance set is composed of some instances from key instance set ΔS_B. In addition, the positive region values of $\Delta t_{(B\cup\{a\})}$ reach their maxima when attribute a is added to B. Therefore, the cardinality of the incremental discernible instance set reflects the relative significance degree of a in B with respect to D. Based on this concept, the incremental reduction algorithm based on positive region can be designed without recomputing on the universe.

3.3. Incremental mechanism designed on key instance set

Theorem 2 allows to determine if a reduct in the original fuzzy decision table remains a reduct when some incremental instances arrive.

Theorem 2. *Given the dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ and $B\subseteq C$, if B is a reduct on fuzzy decision table $FD^{U} = (U, C \cup D)$, then*

(1) $\gamma^{U\cup\Delta U}_{B} \le \gamma^{U\cup\Delta U}_{C}$,
(2) If $Q\subseteq B\subseteq P\subseteq C$, then $\gamma^{U\cup\Delta U}_{Q} \le \gamma^{U\cup\Delta U}_{B} \le \gamma^{U\cup\Delta U}_{P} \le \gamma^{U\cup\Delta U}_{C}$.

Proof.

As B is a reduct on $FD^{U} = (U, C \cup D)$, we have $B\subseteq C$. By Definition 2 and Proposition 3, the results in Theorem 2(1) and (2) are straightforward.■

Theorem 2 shows that B may not be a reduct anymore on $FD^{U\cup\Delta U}$ and that a forward strategy should be adopted to update the original reduct. The incremental mechanism of the dependency function in the dynamic fuzzy decision table is presented in Theorem 3.

Theorem 3. *(Incremental mechanism of dependency degree). Consider dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$, $B\subseteq C$, and key instance set ΔS_B of B on $FD^{U\cup\Delta U}$. When an attribute $a \subseteq C - B$ is added to B, then*

(1) $\gamma^{U\cup\Delta U}_{B\cup\{a\}} = \dfrac{|U\cup\Delta U|\gamma^{U\cup\Delta U}_{B} - \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B}(x) + \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|}$,
(2) $\Delta\gamma^{U\cup\Delta U}_{B\cup\{a\}} = \gamma^{U\cup\Delta U}_{B\cup\{a\}} - \gamma^{U\cup\Delta U}_{B} = \dfrac{\sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x) - \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B}(x)}{|U\cup\Delta U|}$.

Proof.

(1) By the definition of dependency function, we get

$$\gamma^{U\cup\Delta U}_{B\cup\{a\}} = \frac{\sum_{x\in U\cup\Delta U} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|} = \frac{\sum_{x\in U\cup\Delta U-\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x) + \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|}.$$

By Proposition 5, we get $\forall x \in U\cup\Delta U - \Delta S_B$, $POS^{U\cup\Delta U}_{B}(x) = POS^{U\cup\Delta U}_{C}(x)$. Thus,

$$\sum_{x\in U\cup\Delta U-\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x) = \sum_{x\in U\cup\Delta U-\Delta S_B} POS^{U\cup\Delta U}_{B}(x),$$

and we have

$$\begin{aligned}\gamma^{U\cup\Delta U}_{B\cup\{a\}} &= \frac{\sum_{x\in U\cup\Delta U-\Delta S_B} POS^{U\cup\Delta U}_{B}(x) + \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|}\\ &= \frac{\sum_{x\in U\cup\Delta U} POS^{U\cup\Delta U}_{B}(x) - \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B}(x) + \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|}\\ &= \frac{|U\cup\Delta U|\gamma^{U\cup\Delta U}_{B} - \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B}(x) + \sum_{x\in\Delta S_B} POS^{U\cup\Delta U}_{B\cup\{a\}}(x)}{|U\cup\Delta U|}.\end{aligned}$$

(1) The result is straightforward from Theorem 3(1). ■

Theorem 3 shows that when computing $\gamma^{U\cup\Delta U}_{B\cup\{a\}}$, we should compute $POS^{U\cup\Delta U}_{B}(x)$ and $POS^{U\cup\Delta U}_{B\cup\{a\}}(x)$ on the key instance set ΔS_B.

3.4. Main theorem of key instance set

Lemma 1 is necessary to propose the main theorem of the key instance set.

Lemma 1. *In fuzzy decision table $FD = (U, C \cup D)$, $P \subseteq C$ is called a reduct of C with respect to D if P satisfies the following statements:*

(1) $\forall x \in U, POS_P^U(x) = POS_C^U(x)$,
(2) $\forall r \in P, \exists x \in U,\ POS_{P-\{r\}}^U(x) \neq POS_C^U(x)$.

Proof.

(1) If P is a reduct of C, then $\gamma_C^U = \gamma_P^U \Leftrightarrow \sum_{x\in U} POS_C^U(x)/|U| = \sum_{x\in U} POS_P^U(x)/|U|$. If $P \subseteq C$, then $POS_P^U(x) \leq POS_C^U(x)$. Assume $\exists x \in U$ for which $POS_P^U(x) < POS_C^U(x)$, then

$$\sum_{x\in U} POS_P^U(x)/|U| < \sum_{x\in U} POS_C^U(x)/|U|,$$

which contradicts the first condition. Thus, $\forall x \in U, POS_P^U(x) = POS_C^U(x)$.

(1) By Definition 3(2) and Proposition 3, if P is a reduct of C, then, $\forall r \in P$,

$\gamma_C^U \neq \gamma_{P-\{r\}}^U \Leftrightarrow \sum_{x\in U} POS_C^U(x)/|U| \neq \sum_{x\in U} POS_{P-\{r\}}^U(x)/|U| \Leftrightarrow \exists x \in U, POS_{P-\{r\}}^U(x) \neq POS_C^U(x)$.■

Lemma 1 shows that reduct P is the minimal subset of C in which all positive region values reach their maxima.

Theorem 4. *(Main theorem of key instance set). Given dynamic fuzzy decision table $FD^{U\cup\Delta U} = (U \cup \Delta U, C \cup D)$ and $B \subseteq C$, the following statements are always equivalent:*

(1) $\Delta S_B = \emptyset$,
(2) $\exists P \subseteq B$ that is a reduct in $FD^{U\cup\Delta U}$.

Proof.

(1)⇒ (2) By Proposition 5 and Definition 10, $\Delta S_B = \emptyset \Leftrightarrow \forall x \in U \cup \Delta U, POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$. By Lemma 1, $B \subseteq C$ contains a reduct in $FD^{U\cup\Delta U}$. Clearly, $P \subseteq B$ is a reduct in $FD^{U\cup\Delta U}$ if and only if P is a minimal subset satisfying the conditions of Lemma 1.

(2)⇒(1) By Lemma 1, $\forall x \in U \cup \Delta U, POS_P^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$. By Proposition 3, $P \subseteq B \Rightarrow POS_P^{U\cup\Delta U}(x) \leq POS_B^{U\cup\Delta U}(x)$. Thus, $\forall x \in U \cup \Delta U, POS_B^{U\cup\Delta U}(x) = POS_C^{U\cup\Delta U}(x)$. By Definition 10, we can easily prove that $\Delta S_B = \emptyset$■

Theorem 4 shows that B contains one reduct when $\Delta S_B = \emptyset$. Therefore, $\Delta S_B = \emptyset$ can set a stop criterion for feature selection (attribute reduction) by using the forward strategy.

4. Incremental feature selection based on key instance set

4.1. Incremental feature selection based on dependency function

Based on the incremental mechanism of the dependency function given in Theorem 3, we detail the proposed incremental feature selection (attribute reduction) method in Algorithm 6.

In DIAR, step 4 uses the incremental mechanism of the dependency function, thus reducing redundant computation. However, the increment of the dependency function is always computed on a fixed key instance set. Intuitively, this set should be updated and become smaller with arriving attributes. As the key instance set can be updated using the positive region, we use the positive region for improving DIAR.

4.2. Incremental feature selection based on positive region

Theorem 4 allows to improve DIAR by using the incremental mechanism of the positive region instead of that of the dependency function, as detailed in Algorithm 7.

Lemma 1 and Theorem 4 ensure that the reduct obtained from PIAR is also a reduct of both DAR and DIAR. PIAR provides the following improvements over DIAR:

- The key instance set becomes smaller with arriving attributes in PIAR, whereas the set remains fixed in DIAR.
- Steps 6 in PIAR and DIAR are different. For example, if attribute p_i is not redundant, PIAR only conducts computation on instance $x \in (U \cup \Delta U)$ satisfying $POS_{newred-\{b_i\}}^{U\cup\Delta U}(x) < POS_C^{U\cup\Delta U}(x)$. However, DIAR computes $\gamma_{newred-\{b_i\}}^{U\cup\Delta U}$. Therefore, PIAR reduces the computation cost compared with DIAR.

In real applications, PIAR, DIAR, and DAR share the limitation of being sensitive to noise. Consequently, we add threshold $\alpha \in [0, 1]$ in the measure of information to mitigate the effect of noise in practice. Specifically, step 3 uses $POS_B^{U\cup\Delta U}(x) + \alpha < POS_C^{U\cup\Delta U}(x)$, while step 4 uses $POS_{B\cup\{a\}}^{U\cup\Delta U}(x) + \alpha \geq POS_C^{U\cup\Delta U}(x)$, and step 6 uses $POS_{newred-\{b_i\}}^{U\cup\Delta U}(x) + \alpha \geq POS_C^{U\cup\Delta U}(x)$.

Table 3
Time complexities of DAR, DIAR, and PIAR.

DAR	$O(\sum_{i=1}^{m}(\|C\|+1-i)\|U\cup\Delta U\|^2)$
DIAR	$O(\sum_{i=1}^{l}(\|lef\|+1-i)\|\Delta S\|\|U\cup\Delta U\|)$
PIAR	$O(\sum_{i=1}^{k}(\|lef\|+1-i)\|\Delta S_{i-1}\|\|U\cup\Delta U\|)$

$0 \le m \le |C|$, $0 \le l \le |lef|$, and $0 \le k \le |lef|$ represent the number of while loops in DAR, DIAR, and PIAR, respectively.

Table 4
Time complexity per algorithm step.

Algorithm	Steps 2 and 3	Step 4	Step 6
DAR	$O(\|C\|\|U\cup\Delta U\|^2)$	$O(\sum_{i=1}^{m}(\|C\|+1-i)\|U\cup\Delta U\|^2)$	$O(\|B\|^2\|U\cup\Delta U\|^2)$
DIAR	$O(\|C\|(\|U\|\|\Delta U\|+\|\Delta U\|\|U\cup\Delta U\|))$	$O(\sum_{i=1}^{l}(\|lef\|+1-i)\|\Delta S\|\|U\cup\Delta U\|)$	$O(\|B\|^2\|U\cup\Delta U\|^2)$
PIAR	$O(\|C\|(\|U\|\|\Delta U\|+\|\Delta U\|\|U\cup\Delta U\|))$	$O(\sum_{i=1}^{k}\sum(\|lef\|+1-i)\|\Delta S_{i-1}\|\|U\cup\Delta U\|)$	$O(\|B\|^2\|U\cup\Delta U\|^2)$

B is the candidate reduct before step 6. The time complexities of unlisted steps are equal in the three algorithms.

Table 5
Specifications of selected datasets to evaluate feature selection.

Dataset	Attribute type	Number of attributes	Number of instances	Number of classes
Waveform	Real	21	5000	3
Letter	Integer	16	20,000	26
Shuttle	Integer	9	58,000	7
Credit	Integer	23	30,000	2
Gene9	Real	12,600	203	5
Gene12	Real	9182	174	11
Gene14	Real	3312	203	5
FPS-5	Real	3208	3600	6
FPS-7	Real	4813	3600	6

4.3. Scalability analysis

DAR, DIAR, and PIAR have the same space complexity of $O(|U\cup\Delta U|)$, and their worst time complexities are listed in Table 3.

As $|lef|$is smaller than $|C|$, $|\Delta S|$ is usually much smaller than $|U\cup\Delta U|$, $|\Delta S_{i-1}| \ge |\Delta S_i|$, and $|\Delta S_0| = |\Delta S|$, it is easy to conclude that the computation time of the proposed DIAR and PIAR is smaller than that of DAR. Although PIAR and DIAR have similar time complexities, PIAR is notably faster than DIAR in simulations. To clarify this difference, we list the worst time complexities per algorithm step in Table 4.

In steps 2 and 3 of DIAR and PIAR, $POS_B^{U\cup\Delta U}$ and $POS_C^{U\cup\Delta U}$ should be computed using the incremental mechanism of the positive region. Thus, DIAR and PIAR are faster than DAR in this step.

In step 4 of DIAR and PIAR, only $POS_{B\cup\{a\}}^{U\cup\Delta U}(x)$should be computed for every $x \in \Delta S_B$, $a \in lef$, $|\Delta S_B| \le |U\cup\Delta U|$. Note that $|\Delta S_B|$ reduces when adding a new attribute into B in PIAR, whereas ΔS remains invariant in DIAR. Thus, PIAR requires less time than DIAR to compute this step, because the values of l and k are usually similar.

In step 6, redundant attributes are deleted. In PIAR, we check whether any attribute b_i is redundant without computing all $POS_{B-\{b_i\}}^{U\cup\Delta U}(x), x \in U\cup\Delta U$. However, all $POS_{B-\{b_i\}}^{U\cup\Delta U}(x), x \in U\cup\Delta U$ should be computed in DIAR.

Overall, PIAR and DIAR may have the same worst time complexity. However, PIAR is substantially faster than DIAR in the detailed analyses. Thus, we consider PIAR instead of DIAR to compare the proposed method with various non-incremental and incremental feature selection algorithms.

5. Experimental evaluations

We compared the proposed PIAR with a classical algorithm (DAR) [42], an intuitive non-incremental algorithm (NonIAR), and two state-of-the-art rough-set-based incremental algorithms on real datasets [20,39] to verify its performance.

5.1. Experimental setup

All experiments were conducted on a computer with Ubuntu release 16.0, Intel Core i7-4790 CPU at 3.60 GHz, and 8 GB RAM and implemented by C++. We considered the nine UCI datasets [8] listed in Table 5, which differ considerably regarding numbers of instances and features. There are three main types of data, namely, data with high dimensionality and

few instances, with low dimensionality and several instances, and with high dimensionality and several instances. Therefore, these datasets allow to comprehensively analyze the algorithm performance in terms of dimensionality and instances.

To simulate dynamical datasets, we equally split the original datasets into six subsets with the same distribution as the original dataset. These subsets were provided successively and incrementally to the algorithms.

To illustrate the calculation of fuzzy similarity relation $R_B(\cdot, \cdot)$, we considered the bounded intersection (also called the Łukasiewicz T-norm) given by

$$T_{\mathrm{L}}(\mathrm{a}, \mathrm{b}) = \max\{0, a + b - 1\}, \ \ a, b \in [0, 1]$$

as special case of triangular norm T. The similarity degree satisfying T_{L} can be calculated as follows:

$$\forall x, y \in U, \ R_B(x, y) = \min_{r \in B}(R_r(x, y)),$$

where $R_r(x, y) = 1 - (\max(r(x), r(y)) - \min(r(x), r(y)))$ with $r(x)$, $r(y) \in [0, 1]$ represents the attribute values of instances x, y on attribute r, respectively. More details can be found in [31].

We selected the computation time, speedup ratio, reduction ratio, and classification accuracy as measures of the algorithm effectiveness and efficiency.

The speedup ratio is defined as

$$SpeedupRatio = \frac{T_{\mathrm{baseline}}}{T},$$

where T_{baseline} is the computation time of the DAR classical reduction algorithm and T is the computation time of the evaluated algorithm. If the classical algorithm did not work on certain datasets, we considered the maximal running time of the evaluated as T_{baseline}. The speedup ratio can take values in $[0,\infty]$.

The reduction ratio is defined as

$$ReductionRatio = \frac{Reduct}{Attributes},$$

Where *Reduct* is the size of the reduct and *Attributes* is the number of condition attributes. The reduction ratio can take values in [0,1].

We used the *K*-nearest neighbors (KNN) [6] (with usual value of $K = 3$), support vector machine (SVM), and extreme gradient boosting (XGB) [5] to measure the classification performance of the reduced datasets [33]. After obtaining a reduct for the dataset, 5-fold cross-validation was applied to ensure the fairness and stability of the classification results.

5.2. Comparison with non-incremental feature selection algorithms

From the available non-incremental feature selection algorithms, such as DAR, MAR, and EAR, we selected DAR as baseline for comparison, because the proposed algorithm, PIAR, is based on the dependency function or positive region. Moreover, we considered an intuitive non-incremental feature selection (attribute reduction) algorithm, NonIAR, that does not recompute DAR from the beginning of an empty set but uses the historical reduct as original candidate of the new reduct. Specifically, $B \leftarrow red$ is used as step 1 of DAR, whereas the other steps do not change.

5.2.1. Computation time

We obtained the computation time of PIAR, NonIAR, and DAR as data subsets subsequently arrived. The evolution of the computation time is shown in Fig. 4.

Fig. 4 shows that the DAR trend grows up dramatically, indicating that DAR spends increasingly more time as instances arrive, especially on datasets with high dimensionality and several instances. For example, DAR spends about 9 days and 23 h (860,597 s) on dataset FPS-5, and we omit the DAR trend for dataset FPS-7, as it exceeds 10 days. Therefore DAR does not perform well on datasets with high dimension and several instances.

The computation time of NonIAR remains below that of DAR, showing its higher speed as new instances arrive and the effectiveness to initialize a candidate reduct by using the original reduct when new instances arrive. However, NonIAR is still time consuming when new instances arrive, because as a non-incremental algorithm, it performs calculations on the whole dataset. Thus, dataset size affects the computation time of NonIAR. As a result, it is necessary to propose an incremental algorithm, which need not be executed on all available data when new instances arrive.

Overall, the computation times of DAR and NonIAR are substantially higher than that of PIAR, showing the high efficiency of PIAR as new instances arrive, given its processing on some instead of all instances. Thus, PIAR prevents redundant calculations and substantially reduces the computation time.

5.2.2. Speedup ratio

The average speedup ratio for each dataset is shown in Fig. 5. Datasets with low dimensionality and several instances are denoted by I, those with high dimensionality and few instances by II, and those with high dimensionality and several instances by III. Given the excessively long execution time of DAR on dataset FPS-7 of over 10 days, we consider only FPS-5 as dataset with high dimensionality and several instances.

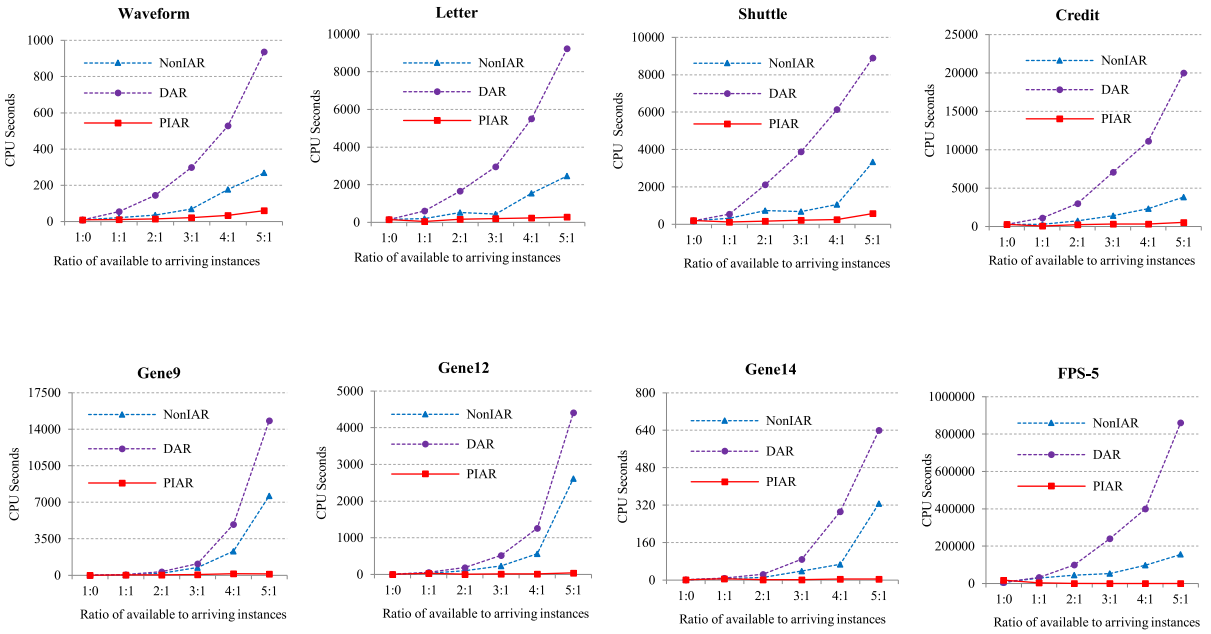

Fig. 4. Computation time in CPU seconds according to arriving instances.

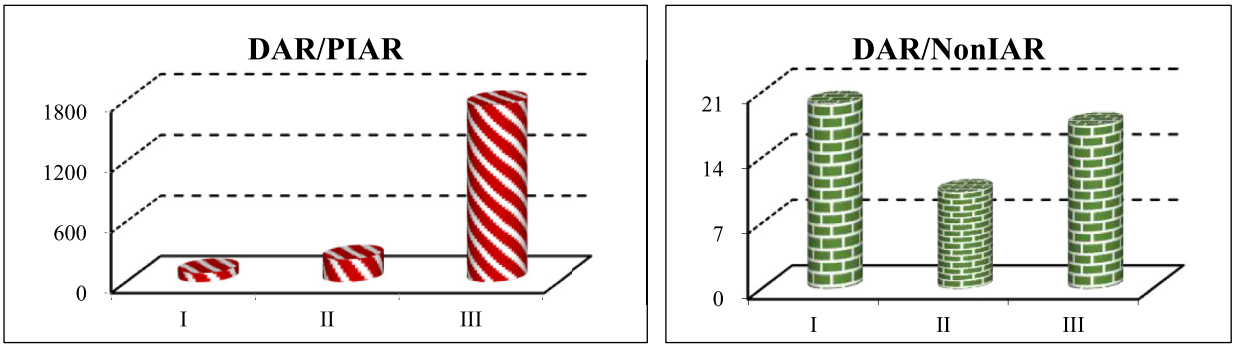

Fig. 5. Average speedup ratios for DAR/PIAR and DAR/NonIAR.

Fig. 5 shows the high speedup ratio of DAR/PIAR, being higher on datasets with high dimensionality and several instances. Therefore, PIAR is more efficient than DAR, especially on such datasets. In addition, NonIAR is faster than DAR.

Fig. 6 shows the speedup ratios of NonIAR and PIAR. When the ratio of available-to-arriving data is 1:1, the PIAR speedup ratio increases more sharply than that of NonIAR on datasets Waveform, Letter, Shuttle, and Credit. These datasets have low dimensionality and several instances. However, the trends of PIAR and NonIAR increase similarly on datasets Gene9, Gene12, Gene14, and FPS-5, which have high dimensionality and few instances. Therefore, when the size of arriving data is comparable to that of available data, PIAR provides higher speedup than NonIAR on datasets with several instances.

5.3. Comparison with state-of-the-art rough-set-based incremental feature selection algorithms

We also compared two state-of-the-art rough-set-based incremental feature selection algorithms, namely, MIAR [39] and EIAR [20], with the proposed PIAR.

5.3.1. Comparison between piar and eiar

Although PIAR and EIAR are rough-set-based incremental attribute reduction methods, their procedures are very different. First, they use different information measures, as PIAR is based on positive region, whereas EIAR is based on entropy. Second, PIAR and EIAR consider fuzzy rough sets and rough sets, respectively.

As EIAR cannot be applied to datasets with real-valued attributes, we use the fuzzy C-means to discretize data. We omitted the preprocessing time of EIAR, and thus, the computation time of EIAR is longer than that presented in this paper.

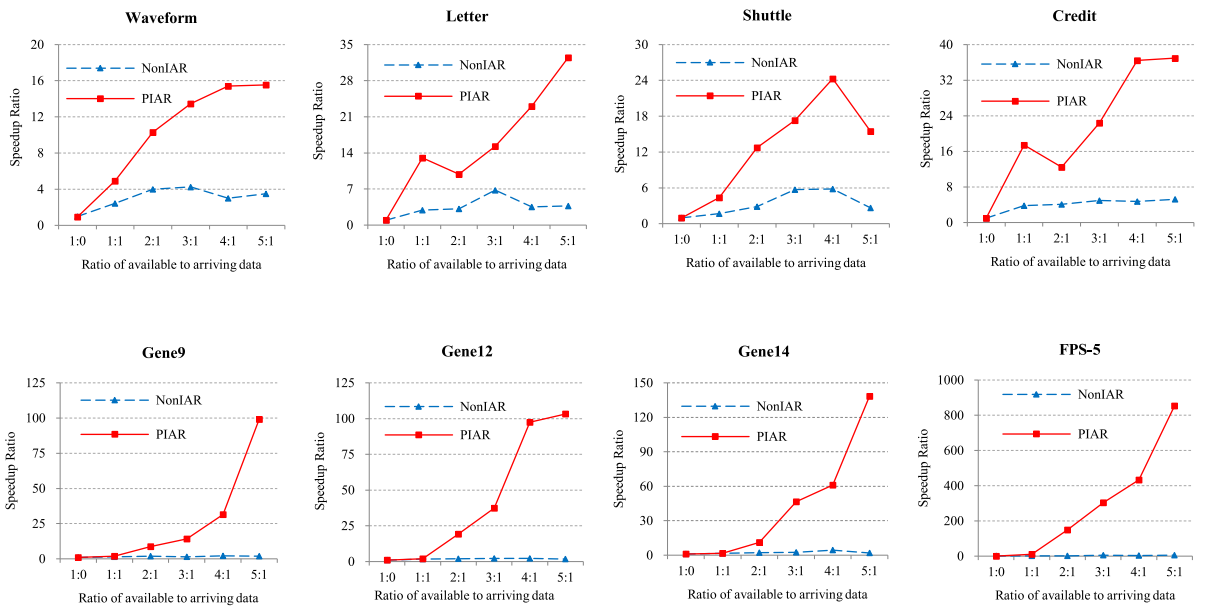

Fig. 6. Speedup ratio of NonIAR and PIAR as data subsets arrive.
When the ratio of available-to-arriving data is high, such as 4:1 or 5:1, the PIAR speedup increases more sharply than that of NonIAR. Hence, as the size of arriving data is lower than that of available data, the efficiency of PIAR is substantially better than that of NonIAR. Therefore, PIAR more suitably handles accumulating data than NonIAR.

Table 6
Computation time (CPU seconds) of PIAR and EIAR on data with low dimensionality and several instances.

Dataset	PIAR	EIAR
Waveform	141.90	48.83
Letter	931.30	317.50
Shuttle	1341.24	28.94
Credit	1467.95	863.06
Average	970.60	314.58

Table 7
Reduction of PIAR and EIAR on the datasets with low dimensionality and several instances.

Dataset	PIAR		EIAR		All attributes	
	No. red.	Red. ratio	No. red.	Red. ratio	No. attr.	Red. ratio
Waveform	15	0.714	14	0.667	21	1
Letter	9	0.563	11	0.688	16	1
Shuttle	6	0.667	6	0.667	9	1
Credit	13	0.565	15	0.652	23	1
Average	10.75	0.627	11.5	0.668	17.25	1

Overall, we compared PIAR and EIAR on nine datasets with different characteristics and divided the datasets into three types: datasets with low dimensionality and several instances, those with high dimensionality and few instances, and those with high dimensionality and several instances. The corresponding results show that neither PIAR nor EIAR are efficient on every type of dataset, and we analyze the advantages and drawbacks of each method.

We first compare PIAR and EIAR on datasets with low dimensionality and several instances. The computation time and reduction results are listed in Tables 6 and 7, respectively, and the classification performance is shown in Fig. 7. All classification results were obtained from the original datasets, that is, although EIAR runs on discretized datasets, the accuracy was computed with respect to the original datasets.

Table 6 shows that the computation time of PIAR is often higher than that of EIAR, thus being less efficient on datasets with low dimensionality and several instances. Table 7 and Fig. 7 show that PIAR and EIAR obtain reducts with comparable classification accuracies. Overall, EIAR is much faster than PIAR and provides similar reduction performance on the evaluated datasets. Hence, PIAR is not as suitable for datasets with low dimensionality and several instances as EIAR.

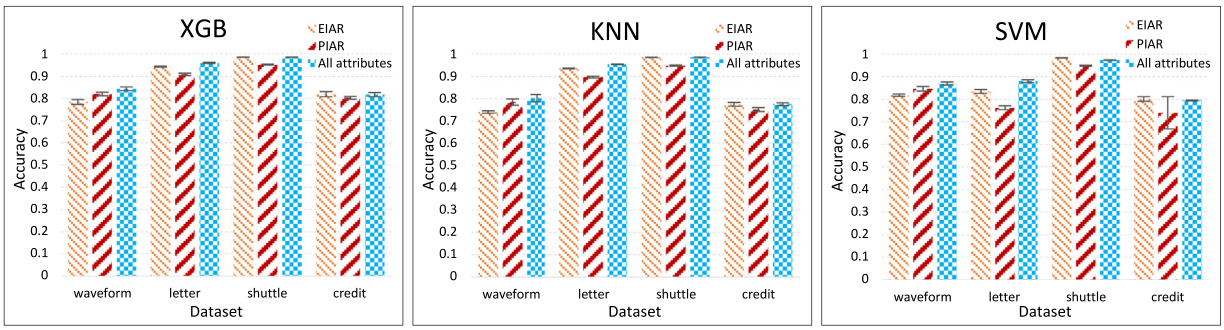

Fig. 7. Classification accuracy of PIAR and EIAR on datasets with low dimensionality and several instances.

Table 8
Computation time (CPU seconds) of PIAR and EIAR on data with high dimensionality and few instances.

Dataset	PIAR	EIAR
Gene9	460.73	129.04
Gene12	109.80	85.90
Gene14	19.41	19.23
Average	196.65	78.06

Table 9
Reduction of PIAR and EIAR on datasets with high dimensionality and few instances.

Dataset	PIAR		EIAR		All attributes	
	No. red.	Red. ratio	No. red.	Red. ratio	No. attr.	Red. ratio
Gene9	27	0.002	7	0.001	12,600	1
Gene12	22	0.002	8	0.001	9182	1
Gene14	14	0.004	7	0.002	3312	1
Average	21	0.003	7.33	0.001	8364.67	1

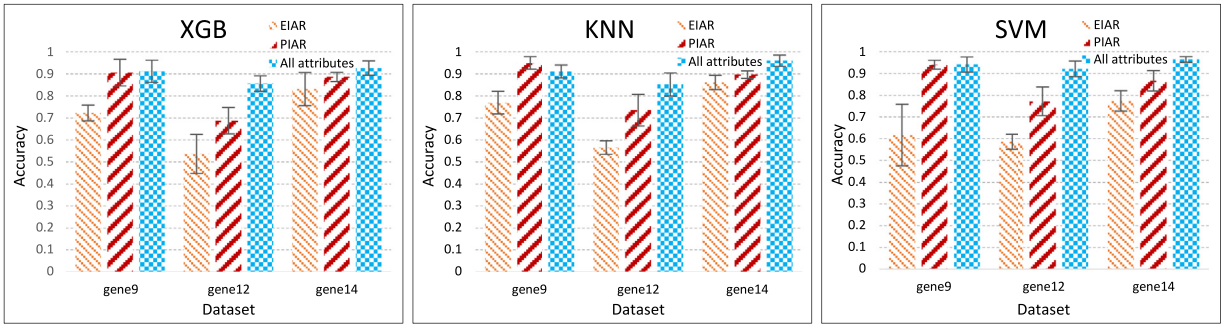

Fig. 8. Classification accuracy of PIAR and EIAR on datasets with high dimensionality and few instances.

Table 8 lists the efficiency of PIAR and EIAR on datasets with high dimensionality and few instances. Again, EIAR is faster than PIAR on such datasets.

The reduction and performance comparison of PIAR and EIAR are provided in Table 9 and Fig. 8, respectively. EIAR can obtain a reduct that is slightly more compact, but its classification performance is lower than that of PIAR. Moreover, the reduction results of EIAR speed up learning at the expense of information loss, because EIAR requires discretization. Comparatively, PIAR can obtain a reduct with an acceptable reduction ratio and high classification accuracy on the datasets with high dimensionality and few instances.

We finally compare EIAR and PIAR on datasets with high dimensionality and several instances. The efficiency in terms of computation time is listed in Table 10, while the reduction and classification accuracy are provided in Table 11 and Fig. 9, respectively. The computation time of EIAR is higher than that of PIAR. Therefore, PIAR is faster than EIAR on datasets with high dimensionality and several instances.

Table 10
Computation time (CPU seconds) of PIAR and EIAR on data with high dimensionality and several instances.

Dataset	PIAR	EIAR
FPS-5	6585.27	6840.74
FPS-7	13,368.93	241,457.62
Average	9977.10	124,149.18

Table 11
Reduction of PIAR and EIAR on datasets with high dimensionality and several instances.

Dataset	PIAR		EIAR		All attributes	
	No. red.	Red. ratio	No. red.	Red. ratio	No. attr.	Red. ratio
FPS-5	50	0.016	51	0.016	3208	1
FPS-7	47	0.010	47	0.010	4813	1
Average	48.5	0.013	49	0.013	4010.5	1

Table 12
Computation time (CPU seconds) of PIAR and MIAR as instances arrive.

Dataset	PIAR	MIAR
Waveform	141.90	278.14
Gene9	460.73	348.66
Gene12	109.80	155.41
Gene14	19.41	85.13
Letter	931.30	*Out of memory*
Shuttle	1341.24	*Out of memory*
Credit	1467.95	*Out of memory*
FPS-5	6585.27	*Out of memory*
FPS-7	13,368.93	*Out of memory*
Average	182.96	216.84

The average results of PIAR are computed on datasets for which MIAR does not run out of memory.

Table 13
Reduction of PIAR and MIAR.

Dataset	PIAR		MIAR		All attributes	
	No. red.	Red. ratio	No. red.	Red. ratio	No. attr.	Red. ratio
Waveform	15	0.714	13	0.619	21	1
Gene9	27	0.002	27	0.002	12,600	1
Gene12	22	0.002	12	0.001	9182	1
Gene14	14	0.004	12	0.004	3312	1
Letter	9	0.563	–	–	16	1
Shuttle	6	0.667	–	–	9	1
Credit	13	0.565	–	–	23	1
FPS-5	50	0.016	–	–	3208	1
FPS-7	47	0.010	–	–	4813	1
Average	19.5	0.181	16	0.157	6278.75	1

The average results of PIAR and all attributes are computed on datasets for which MIAR does not run out of memory.

Table 14
Characteristics of incremental feature selection algorithms evaluated in this study.

Criterion	PIAR	EIAR	MIAR
Storage requirements	Low	Low	**High**
Performance on datasets with low dimensionality and high number of instances	**Slow**	Fast	**Out of memory**
Performance on datasets with high dimensionality and low number of instances	Acceptable	Fast	**Slow**
Performance on datasets with high dimensionality and high number of instances	Fast	Acceptable	**Out of memory**
Classification accuracy	High	**Acceptable or poor**	Good
Requires data discretization	No	**Yes**	No

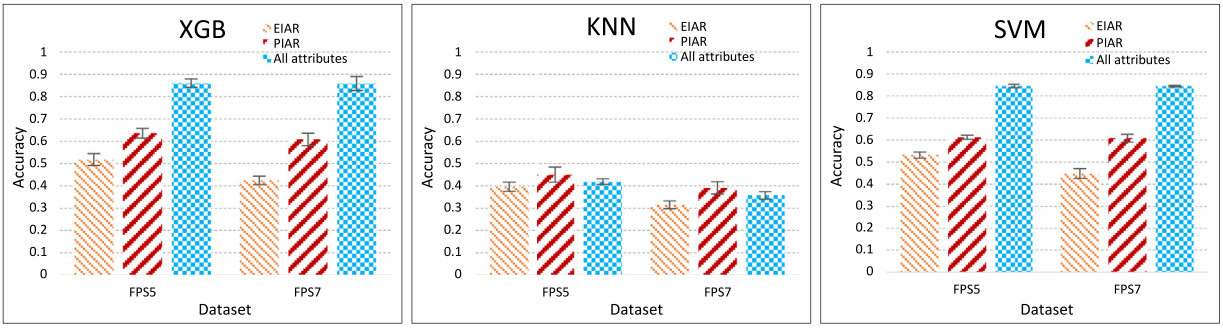

Fig. 9. Classification accuracy of PIAR and EIAR on datasets with high dimensionality and several instances.

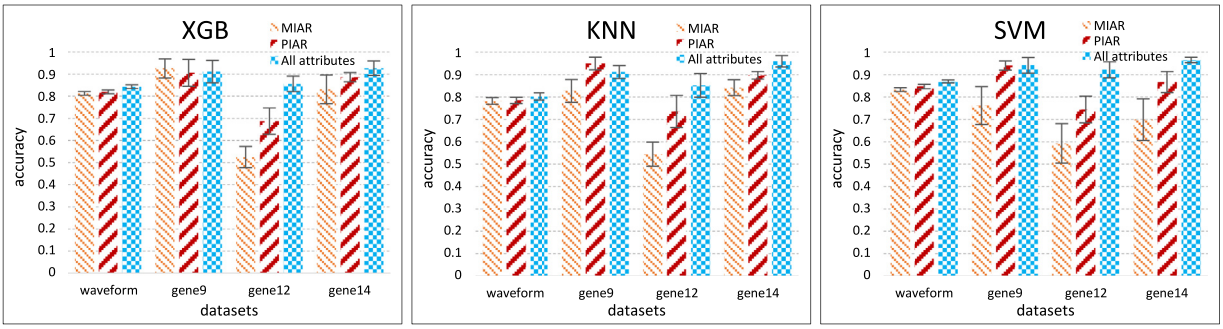

Fig. 10. Classification accuracy of PIAR and MIAR.

Table 11 shows the reduction of PIAR and EIAR on the datasets with high dimensionality and several instances. For comparison, some thresholds were considered for EIAR to obtain similar size of reducts as those from PIAR. Fig. 9 shows that PIAR has better classification accuracy than EIAR.

Overall, the comparison between PIAR and EIAR mainly unveils two aspects:

(1) EIAR obtains reduction with lower classification performance by data discretization that leads to information loss.
(2) PIAR is more efficient on the datasets with high dimensionality, whereas EIAR is more efficient on those with low dimensionality.

5.3.2. Comparison between piar and miar

We also compare PIAR and MIAR in terms of computation time and classification accuracy. Although both PIAR and MIAR are based on fuzzy rough sets, they have different characteristics. First, PIAR is based on the positive region, whereas MIAR is based on the discernibility matrix. Second, MIAR requires more storage than PIAR. From the comparison, we unveil the advantages and drawbacks of PIAR and MIAR.

The computation times of PIAR and MIAR are listed in Table 12. MIAR runs out of memory on datasets Letter, Shuttle, Credit, FPS-5, and FPS-7, because it is storage-intensive on datasets with several instances. Thus, only the comparison results on the other four datasets (i.e., Waveform, Gene9, Gene12, and Gene14) are listed for MIAR in Table 12. PIAR often works faster than MIAR. More importantly, PIAR requires much less storage than MIAR, and hence it can handle datasets with high dimensionality and several instances.

Regarding performance, Table 13 shows that PIAR and MIAR have similar reduction ratios. For example, the average reduction ratios of PIAR and MIAR are 16 and 19.5, respectively, indicating the ability of both algorithms to reduce dimensionality. Fig. 10 shows that PIAR has better classification accuracy than MIAR, while its reduction ratios mostly remain below 0.4%.

Overall, the comparison between PIAR and MIAR mainly unveils two aspects:

(1) Unlike PIAR, MIAR cannot handle datasets with several instances.
(2) When the arriving instances are fewer than the available ones, PIAR is substantially faster than MIAR.

5.3.3. Overall comparison of incremental feature selection methods

We summarize the characteristics of the incremental feature selection algorithms, namely, PIAR, EIAR, and MIAR, in Table 14. The entries highlighted in bold indicate the unsuitability of the algorithm to the indicated criterion. In practice, we should choose the appropriate incremental feature selection algorithm according to the available hardware and dataset

characteristics. For example, MIAR is preferred if the necessary memory is available. If computation time is restricted, EIAR is preferred, and for high-dimensional data, PIAR is preferred.

6. Conclusions

We propose an incremental feature selection algorithm based on fuzzy rough sets. In addition, we obtain various insights on incrementally selected relevant features with successively arriving instances. Our main contributions can be summarized as follows:

(1) Incremental mechanisms, such as the incremental positive region and key instance set, are proposed with strict mathematical reasoning and serve as bases to design an incremental feature selection algorithm, PIAR.
(2) The proposed algorithm fully leverages historical feature selection results and prevents recomputing on all available data.
(3) Although the proposed algorithm may achieve a relatively low performance on datasets with low dimensionality and several instances, it exhibits high performance on datasets with high dimensionality.

Note that feature selection for datasets with low dimensionality is not necessary, because their data have many redundant instances but few or no redundant attributes. Consequently, the proposed algorithm spends much time identifying redundant instances from arriving instances, but few or no new features are added to the original selection on such datasets. In contrast, datasets with high dimensionality have many redundant or even conflicting features. Thus, arriving instances are often informative and may lead to determination of features not included in the original feature selection. Therefore, the proposed algorithm is suitable to handle datasets with high dimensionality.

From this study and its results, we provide future directions of research:

(1) Explore incremental feature selection with streaming features based on fuzzy positive regions.
(2) Besides the incremental solutions mainly focused on streaming features and streaming instances, incremental class labels should be investigated.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Peng Ni: Conceptualization, Methodology, Software, Validation, Data curation, Writing - original draft. **Suyun Zhao:** Investigation, Conceptualization, Methodology, Data curation, Writing - original draft, Supervision. **Xizhao Wang:** Investigation, Methodology, Supervision. **Hong Chen:** Supervision, Methodology. **Cuiping Li:** Supervision, Methodology. **Eric C.C. Tsang:** Writing - review & editing.

Acknowledgments

This work is supported by the National Key Research & Development Plan of China (2018YFB1004401, 2017YFB1400700, 2016YFB1000702), NSFC (No. 61702522, 61772536, 61772537, 61732006, 61532021), NSSFC (No. 12\&ZD220), National Basic Research Program of China (973) (No.2014CB340402), National High-Technology Research and Development Program of China (863) (No.2014AA015204), and Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (15XNLQ06). This study was partially done when the authors worked in SA Center for Big Data Research in RUC. This Center is funded by a Chinese National 111 Project Attracting.

Appendix

A triangular norm is an operator T: $[0, 1]^2 \rightarrow [0, 1]$ satisfying:

(1) Boundary condition: $T(a, 1) = a, a \in [0, 1]$,
(2) Commutativity: $T(a, b) = T(b, a),\ \ a, b \in [0, 1]$,
(3) Associativity: $T(T(a, b), c) = T(a, T(b, c))$ for $a, b, c \in [0, 1]$,
(4) Monotonicity: $a < \alpha,\ b < \beta \Rightarrow T(a, b) \leq T(\alpha, \beta)$ for $a, b, \alpha, \beta \in [0, 1]$.

References

[1] M. Ackerman, S. Dasgupta, Incremental clustering: the case for extra clusters, in: Advances in Neural Information Processing Systems, 2014, pp. 307–315.
[2] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, D.B. R.osen, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, IEEE. Trans. Neural. Netw. 3 (5) (1992) 698–713, doi:10.1109/72.159059.
[3] G. Cauwenberghs, T. Poggio, Incremental and decremental support vector machine learning, in: Proceedings of the 2001 Neural Information Processing Systems, 2001, pp. 409–415.

[4] X.Y. Che, D.G. Chen, J.S. Mi, A novel approach for learning label correlation with application to feature selection of multi-label data, Inf. Sci. 512 (2020) 795–812, doi:10.1016/j.ins.2019.10.022.
[5] T.Q. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi:10.1145/2939672.2939785.
[6] D. Coomans, D.L. Massart, Alternative *k*-nearest neighbour rules in supervised pattern recognition: part 1. *k*-nearest neighbour classification by using alternative voting rules, Anal. Chim. Acta 136 (1982) 15–27, doi:10.1016/S0003-2670(01)95359-0.
[7] A.K. Das, S. Sengupta, S. Bhattacharyya, A group incremental feature selection for classification using rough set theory based genetic algorithm, Appl. Soft Comput. 65 (2018) 400–411, doi:10.1016/j.asoc.2018.01.040.
[8] D. Dua, E.K. Taniskidou, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2018 https://archive.ics.uci.edu/ml/datasets.html/.
[9] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (2–3) (1990) 191–209, doi:10.1080/03081079008935107.
[10] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM. Comput. Surv. 46 (4) (2014) 1–37, doi:10.1145/2523813.
[11] C. Giraud-Carrier, A note on the utility of incremental learning, AI. Commun. 13 (4) (2000) 215–223.
[12] S.U. Guan, F. Zhu, An incremental approach to genetic-algorithms-based classification, IEEE. Trans. Syst., Man, Cybern., Part B: Cybern 35 (2) (2005) 227–239, doi:10.1109/TSMCB.2004.842247.
[13] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[14] Donatella Frimani, Massimo Mecella, Monica Scannapieco, Carlo Batini, On the meaningfulness of 'Big Data quality'. 1 (2016) 6–20, doi: 10.1007/s41019-015-0004-7.
[15] F. Hu, J. Dai, G.Y. Wang, Incremental algorithms for attribute reduction in decision table, Control. Decis. 22 (3) (2007) 268.
[16] F. Hu, G. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets, in: International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, 2005, pp. 185–193, doi:10.1007/11548669_20.
[17] Q.H. Hu, L. Zhang, S. An, D. Zhang, D.R. Yu, On robust fuzzy rough set models, IEEE Trans. Fuzzy Syst. 20 (4) (2011) 636–651, doi:10.1109/TFUZZ.2011.2181180.
[18] X.G. Hu, P. Zhou, P.P. Li, J. Wang, X.D. Wu, A survey on online feature selection with streaming features, Front. Comput. Sci. 12 (3) (2018) 479–493, doi:10.1007/s11704-016-5489-3.
[19] H.X. Li, L. Zhang, B. Huang, X.Z. Zhou, Cost-sensitive dual-bidirectional linear discriminant analysis, Inf. Sci. 510 (2020) 283–303, doi:10.1016/j.ins.2019.09.032.
[20] J. Liang, F. Wang, C. Dang, Y. Qian, A group incremental approach to feature selection applying rough set technique, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 294–308, doi:10.1109/TKDE.2012.146.
[21] K.Y. Liu, X.B. Yang, H. Fujita, D. Liu, X. Yang, Y.H. Qian, An efficient selector for multi-granularity attribute reduction, Inf. Sci. 505 (2019) 457–472, doi:10.1016/j.ins.2019.07.051.
[22] L.P. Liu, Y. Jiang, Z.H. Zhou, Least square incremental linear discriminant analysis, in: 9th IEEE International Conference on Data Mining, 2009, pp. 298–306, doi:10.1109/ICDM.2009.78.
[23] Z. Pawlak, Rough sets: Theoretical aspects of Reasoning About Data, 9, Springer Science & Business Media, Berlin, Germany, 2012.
[24] Z. Pawlak, A. Skowron, Rough sets: some extensions, Inf. Sci. 177 (1) (2007) 28–40, doi:10.1016/j.ins.2006.06.006.
[25] S. Perkins, K. Lacker, J. Theiler, Grafting: fast, incremental feature selection by gradient descent in function space, J. Mach. Learn. Res. 3 (2003) 1333–1356.
[26] S. Perkins, J. Theiler, Online feature selection using grafting, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 592–599.
[27] Y.H. Qian, J.Y. Liang, W. Pedrycz, C. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (9–10) (2010) 597–618, doi:10.1016/j.artint.2010.04.018.
[28] Y.H. Qian, J.Y. Liang, Combination entropy and combination granulation in rough set theory, Int. J. Uncertain. Fuzz. Knowl. Syst. 16 (2) (2008) 179–193, doi:10.1142/S0218488508005121.
[29] J.C. Schlimmer, R.H. Granger, Incremental learning from noisy data, Mach. Learn. 1 (3) (1986) 317–354, doi:10.1007/BF00116895.
[30] K. Selvakumar, M. Karuppiah, L. SaiRamesh, S.H. Islam, M.M. Hassan, G. Fortino, K.K.R. Choo, Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs, Inf. Sci. 497 (2019) 77–90, doi:10.1016/j.ins.2019.05.040.
[31] E.C.C. Tsang, D.G. Chen, D.S. Yeung, X.Z. Wang, J.W. Lee, Attributes reduction using fuzzy rough sets, IEEE. Trans. Fuzzy. Syst. 16 (5) (2008) 1130–1141, doi:10.1109/TFUZZ.2006.889960.
[32] P.E. Utgoff, Incremental induction of decision trees, Mach. Learn. 4 (2) (1989) 161–186, doi:10.1023/A:1022699900025.
[33] C.Z. Wang, Y.L. Qi, M.W. Shao, Q.H. Hu, D.G. Chen, Y.H. Qian, Y.J. L.in, A fitting model for feature selection with fuzzy rough sets, IEEE. Trans. Fuzzy. Syst. 25 (4) (2017) 741–753, doi:10.1109/TFUZZ.2016.2574918.
[34] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, X. Wu, Online feature selection with group structure analysis, IEEE. Trans. Knowl. Data. Eng. 27 (11) (2015) 3029–3041, doi:10.1109/TKDE.2015.2441716.
[35] W. Wei, P. Song, J.Y. Liang, X.Y. Wu, Accelerating incremental attribute reduction algorithm by compacting a decision table, Int. J. Mach. Learn. Cybern. 10 (9) (2018) 2355–2373, doi:10.1007/s13042-018-0874-x.
[36] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1178–1192, doi:10.1109/TPAMI.2012.197.
[37] J. Xu, C. Xu, B. Zou, Y.Y. Tang, J.T. Peng, X.G. You, New incremental learning algorithm with support vector machines, IEEE. Trans. Syst. Man. Cybern. Syst. 49 (11) (2018) 1–12, doi:10.1109/TSMC.2018.2791511.
[38] M. Yang, An incremental updating algorithm for attribute reduction based on improved discernibility matrix, Chin. J. Comput. 30 (5) (2007) 815–822.
[39] Y.Y. Yang, D.G. Chen, H. Wang, E.C.C. Tsang, D.L. Zhang, Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving, Fuzzy. Sets. Syst 312 (2017) 66–86, doi:10.1016/j.fss.2016.08.001.
[40] Y.Y. Yang, D.G. Chen, H. Wang, X.Z. Wang, Incremental perspective for feature selection based on fuzzy rough sets, IEEE Trans. Fuzzy Syst. 26 (3) (2018) 1257–1273, doi:10.1109/TFUZZ.2017.2718492.
[41] J.T. Yao, A.V. Vasilakos, W. Pedrycz, Granular computing: perspectives and challenges, IEEE Trans. Cybern. 43 (6) (2013) 1977–1989, doi:10.1109/TSMCC.2012.2236648.
[42] Y.Y. Yao, Y. Zhao, J. Wang, On reduct construction algorithms, Trans. Comput. Sci. 2 (2008) 100–117, doi:10.1007/978-3-540-87563-5_6.
[43] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353, doi:10.1016/S0019-9958(65)90241-X.
[44] H.Y. Zhang, H.J. Song, S.Y. Yang, Feature selection based on generalized variable-precision (ϑ,σ)-fuzzy granular rough set model over two universes, Int. J. Mach. Learn. Cybern. 10 (5) (2019) 913–924, doi:10.1007/s13042-017-0770-9.
[45] L.B. Zhang, H.X. Li, X.Z. Zhou, B. Huang, Sequential three-way decision based on multi-granular autoencoder features, Inf. Sci. 507 (2020) 630–643, doi:10.1016/j.ins.2019.03.061.
[46] X. Zhang, C.L. Mei, D.G. Chen, J.H. Li, Feature selection in mixed data: a method using a novel fuzzy rough set-based information entropy, Pattern. Recognit 56 (2016) 1–15, doi:10.1016/j.pat cog.2016.02.013.
[47] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 7 (2006) 1861–1885.