This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

Active Sample Selection Based Incremental Algorithm for Attribute Reduction with Rough Sets

Yanyan Yang, Degang Chen, and Hui Wang

***Abstract*—Attribute reduction with rough sets is an effective technique for obtaining a compact and informative attribute set from a given dataset. However, traditional algorithms have no explicit provision for handling dynamic datasets where data present themselves in successive samples. Incremental algorithms for attribute reduction with rough sets have been recently introduced to handle dynamic datasets with large samples, though they have high complexity in time and space. To address the time/space complexity issue of the algorithms, this paper presents a novel incremental algorithm for attribute reduction with rough sets based on the adoption of an active sample selection process and an insight into the attribute reduction process. This algorithm first decides whether each incoming sample is useful with respect to the current dataset by the active sample selection process. A useless sample is discarded while a useful sample is selected to update a reduct. At the arrival of a useful sample, the attribute reduction process is then employed to guide how to add and/or delete attributes in the current reduct. The two processes thus constitute the theoretical framework of our algorithm. The proposed algorithm is finally experimentally shown to be efficient in time and space.**

***Index Terms*—Rough sets, attribute reduction, incremental learning, active sample selection.**

I. Introduction

Rough set theory [17]-[18] is a data analysis methodology that is well known for its ability in handling uncertainty, imprecision and vagueness. It has received considerable attention in data mining, machine learning and pattern recognition [3], [10], [57]-[59], [62]. One important application of rough sets is attribute reduction which aims to remove superfluous attributes from a decision table in order to obtain a compact and informative attribute set. Attribute reduction with rough sets can be viewed as a pure structural approach that only depends on the dataset without the need of any other knowledge. Highlighting the discernible ability of condition attributes related to decision labels is the essential difference between attribute reduction with rough sets and other feature selection methods. A variety of attribute reduction algorithms have been proposed and proven to be effective in improving the performance of learning algorithms [50]-[51], building some well-designed classifiers [9], [34], [39], and ranking attributes [52]-[53].

Traditional approaches to attribute reduction fall into two categories. The first category consists of structural methods for discerning samples. A typical representative is discernibility matrix based method which captures reducts with the reduced disjunctive form of the discernibility function [25]. Discernibility matrix based methods were summarized in [40] and the R package was used to implement these methods and other efficient ones. However, it has been argued that discernibility matrices are old-fashioned data structures and are not suitable for large volumes of data. Much effort has been made to improve this approach by trying to utilize the discernibility information in the discernibility matrix [2], [32]. For example, sample pair selection was proposed in [2] to locate all minimal elements without computing the whole discernibility matrix, so that the search space and time can be reduced. The second category consists of the significance measure oriented methods. Different kinds of reducts were defined by using significance measures [12], [18], [19], [20], and heuristic algorithms were proposed based on significance measures. For example, Shannon's entropy was introduced in [26] to define a new type of reduct. Wang et al. [41] proposed a heuristic algorithm for finding a reduct based on Shannon's conditional entropy. Permutation based heuristic algorithms were proposed in [42] to determine a reduct by employing an elimination process.

However, these traditional algorithms for attribute reduction are not designed to process datasets that are presented dynamically in successive samples (i.e., dynamic datasets with increasing samples), since they have no explicit principles of fully utilizing the information of the original dataset. They have to be run from scratch when new samples arrive, so that they are often computationally time-consuming and even impractical for dynamic datasets with large samples. To handle dynamic datasets, incremental techniques have been introduced into rough sets to address incremental rule acquisition [1], [21],

Manuscript received May 18, 2015; revised September, 30, 2015 and November, 29, 2015; accepted April 27, 2016. This work was supported by grants of NSFC (71471060), the Fundamental Research Funds for the Central Universities (JB2014204) and China Scholarship Council (201506730011).

Y. Y. Yang is with the School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China (e-mail: hrsinbba@163.com).

D. G. Chen is with the Department of Mathematics and Physics, North China Electric Power University, Beijing 102206, China (e-mail: chengdegang@263.net).

H. Wang is with the School of Computing and Mathematics, University of Ulster, Northern Ireland BT37 0QB, UK (e-mail: h.wang@ulster.ac.uk).

[43], incremental updating approximation [11], [15], [31], [61] and incremental attribute reduction [8], [13].

As one important research topic of rough sets, finding reducts from dynamic datasets has been considered from the perspective of the following three variations: attributes, attribute values and samples [13], [29], [35]. With the variation of attributes, there are some researches on incremental attribute reduction. For example, Zeng et al. [35] proposed incremental algorithms for feature selection with fuzzy rough set. Wang et al. [28] developed a dimension incremental strategy for attribute reduction based on the incremental computation of three measures of information entropy. Shu et al. [23] proposed an efficient algorithm for updating attribute reduction based on the incremental computation of the positive region in incomplete decision systems. With the variation of attribute values, Wang et al. proposed in [29] an incremental algorithm for attribute reduction based on incremental computation of three measures of entropy. Based on the incremental computation of positive region, Shu et al. [24] proposed two incremental algorithms for feature selection when attribute values of single sample and multiple samples vary.

In the sample variation type of studies, several incremental algorithms for attribute reduction have been proposed in the framework of rough sets. For example, an incremental algorithm for attribute reduction was proposed in [14] to find the minimal reduct, but it is only applicable for information systems without decision attribute. Two incremental algorithms were presented in [27], [54] to deal with dynamic decision table, but experimental results in [7] show both of them are very time-consuming. To improve the efficiency of updating one reduct, Hu et al. [7] presented an incremental algorithm for attribute reduction based on the positive region, which was shown experimentally to be more efficient than the two algorithms in [27], [54]. Based on the modified discernibility matrix, Hu et al. [8] proposed an incremental algorithm for finding all reducts. Yang [30] proposed an incremental algorithm for attribute reduction by updating the discernibility matrix. Guan [6] proposed an incremental algorithm for updating all reducts based on the discernibility matrix. Feng et al. [5] employed the incremental computation for attribute core to improve the efficiency of computing one reduct in rough sets. Shu et al. [22] presented an incremental algorithm for attribute reduction to compute a reduct from a dynamically-increasing incomplete decision system. To allow a group of samples to be added into a current dataset, Liang et al. [13] developed an efficient group incremental algorithm for attribute reduction by introducing incremental mechanisms for three measures of information entropy.

The above sample variation based incremental algorithms for attribute reduction with rough sets work by incrementally updating some intermediate steps of finding a reduct such as the positive region, information entropy and so on. Little attention has been paid to the issue of which attributes should be added into or deleted from a current reduct. On the other hand, these incremental algorithms passively employ whatever incremental samples arrive. Actually, not all incremental samples are contributive to the incremental computation. Some affect the incremental process of attribute reduction so they are useful, whereas some do not affect the incremental process so they are useless. The useless incremental samples should be discarded and the useful samples should be actively selected ---- this is a process of *active sample selection*. Several methods of active sample selection [44]-[48] were integrated into feature selection by conducting search not only in the feature space but also in the sample space. For example, Liu et al. [44]-[45] proposed a formalism of active feature selection called selective sampling by selecting informative samples based on some data characteristics. Relief [46] weighs each feature by searching two nearest samples of a randomly selected sample. ReliefF [47], a variant of Relief, is robust to noise and handles multiple classes. The SCRAP filter [48] is a conservative filtering scheme that tries to identify the features changing at two consecutive Points of Class Change and include them in the feature subset. However, they are not related to rough sets so they cannot be used to select samples for the incremental computation of attribute reduction. Hence, a method for actively selecting useful samples and filtering out useless samples in the incremental computation process is highly desirable to improve the space and time efficiency of the incremental algorithms for attribute reduction, especially when the dynamic datasets are large. The above considerations motivate us to study the incremental process of attribute reduction.

In this paper, we present a novel, space and time efficient incremental algorithm for attribute reduction from dynamic datasets with increasing samples, which consists of two processes----active sample selection and incremental attribute reduction. The active sample selection process is designed to classify each incoming sample as useless or useful with respect to a current dataset. For purpose of the space and time efficiency, useless samples are filtered out and not considered in the incremental computation. Useful samples are selected to update the reduct by the incremental attribute reduction process that determines which attributes should be added into and deleted from a current reduct. The active sample selection process is integrated into the attribute reduction process, resulting in our incremental algorithm for attribute reduction with active sample selection. Experimental evaluations demonstrate the effectiveness of our incremental algorithm in significant savings of memory space usage and run time.

The remainder of this paper is organized as follows. In Section II, we briefly review the basics of rough sets in order to facilitate subsequent discussions. In Section III, the incremental mechanisms of attribute reduction are presented in detail. In Section IV, our active sample selection based incremental algorithm for attribute reduction is presented. In Section V, experimental results are presented and analyzed. Conclusions are presented in Section VI.

II. Preliminaries

In this section, we review basic concepts related to rough sets [17]-[18], [55]-[56] as well as the discernibility matrix based approach to attribute reduction [2], [25].

A. Basic Concepts

Rough set theory is formally based on an information system defined by Pawlak [17]. For classification tasks, we consider a special information system, i.e., a decision table denoted by $(U, C\bigcup D)$ with $C\bigcap D=\phi$. Here, U termed the universe, is a non-empty finite set of samples, each sample is described by the condition attribute set C and $D=\{d\}$ is the decision attribute set. With each non-empty attribute subset $B\subseteq C$, we associate an equivalence relation defined as $IND(B)=\{(x,y)\in U\times U: a(x)=a(y), \forall a\in B\}$ in [17]. The equivalence relation $IND(B)$ partitions U into a family of equivalence classes denoted by $U/IND(B)=\{[x]_B : x\in U\}$, where $[x]_B=\{y\in U:(x,y)\in IND(B)\}$ is called the equivalence class of $IND(B)$ including x.

Assume U is partitioned into r decision classes $U/IND(D)=\{D_1,\cdots,D_r\}$ by D, where D_i is a subset of samples with the same decision value. For $D_i\in U/IND(D)$, $B-$ lower and $B-$upper approximations of D_i proposed in [17] are defined as $\underline{B}D_i=\bigcup\{[x]_B:[x]_B\subseteq D_i\}$ and $\overline{B}D_i=\bigcup\{[x]_B:[x]_B\bigcap D_i\neq\phi\}$. The samples in each equivalence class of $\underline{B}D_i$ share the same decision value, while the samples in each equivalence class of $\overline{B}D_i$ may have different decision values. This is related to the concepts of consistent and inconsistent samples, which are the theoretical basis of our active sample selection in this paper. The generalized decision of a sample $x\in U$ proposed in [55] is used to clearly represent the consistency and inconsistency [56] and defined as $d([x]_B)=\{d(y):y\in[x]_B\}$ which is actually the set of decision values of all samples in $[x]_B$. $x\in U$ is said to be consistent on $B\subseteq C$, if $|d([x]_B)|=1$; otherwise, $x\in U$ is inconsistent on B, where $|\bullet|$ is the cardinality of a set. As illustrated in [56], $(U, C\bigcup D)$ is a consistent decision table iff any $x\in U$ is consistent on C; otherwise, it is an inconsistent decision table. The union of $B-$ lower approximations of all decision classes is called the positive region of D with respect to B. For notational presentation in our paper, we denote the positive region by $POS_U(B,D)=\bigcup_{k=1}^{r}\underline{B}D_k$. $POS_U(B,D)$ is then the set of all consistent samples of U on B.

A real-life dataset usually contains some irrelevant and redundant attributes. The presence of such attributes may lead to a reduction in the useful information. Attribute reduction with rough sets proposed by Pawlak [17]-[18], can address the above issue. A subset $B\subseteq C$ is a reduct of C if it satisfies 1) $POS_U(B,D)=POS_U(C,D)$; 2) $\forall a\in B$, $POS_U(B-\{a\},D)\neq POS_U(B,D)$. The first condition indicates that B can retain all consistent samples of U on C, i.e., any sample satisfying $|d([x]_C)|=1$ must also meet $|d([x]_B)|=1$, and vice versa. The second one means that $B-\{a\}$ cannot retain all consistent samples of U on C, i.e., for $\forall a\in B$ there always exists $x\in POS_U(C,D)$ satisfying $|d([x]_{B-\{a\}})|>1$. The two conditions can guide us to add or delete attributes in our incremental computation of attribute reduction.

B. The Discernibility Matrix Based Approach to Attribute Reduction

As mentioned above, a variety of heuristic algorithms have been proposed to find a reduct. However, it has been noted in [25] that they could not find a proper reduct but an over-reduct or sub-reduct. To find proper reducts, the method of discernibility matrix was proposed in [25], by which a discernibility function could be constructed and all reducts could be found with its disjunctive form. Although finding all reducts with this technique is an NP-hard problem, this method provides the theoretical foundation for finding reducts from a decision table.

Let $U=\{x_1,\cdots,x_n\}$. For notational presentation in this paper, we denote $M=(c(x_i,x_j))_{n\times n}$ as the discernibility matrix of $(U, C\bigcup D)$ where $c(x_i,x_j)=\{a\in C: a(x_i)\neq a(x_j)\}$ if $(x_i,x_j)\in\Phi$, and $c(x_i,x_j)=\phi$ otherwise. Here, Φ is defined in the following way:

$$\begin{aligned}\Phi=\{(x_i,x_j)\in U\times U: &1)\ x_i\in POS_U(C,D), x_j\notin POS_U(C,D);\\ &2)\ x_i\notin POS_U(C,D), x_j\in POS_U(C,D);\\ &3)\ x_i,x_j\in POS_U(C,D), d(x_i)\neq d(x_j)\}.\end{aligned}$$

For $\forall(x_i,x_j)\in\Phi$, $c(x_i,x_j)\neq\phi$ clearly holds. It is necessary to discern the sample pair, in which one of two samples at least belongs to $POS_U(C,D)$ and neither of them belongs to $C-$ lower approximation of the same decision class. For $\forall(x_i,x_j)\notin\Phi$, $c(x_i,x_j)=\phi$ holds. It is unnecessary to discern the sample pair, in which two samples belong to $C-$lower approximation of the same decision class or do not belong to $POS_U(C,D)$. Thus, it is sufficient for our paper to only discern all sample pairs in Φ.

The discernibility matrix M is clearly symmetric, i.e., $c(x_i,x_j)=c(x_j,x_i)$, and $c(x_i,x_i)=\phi$. Sample pairs (x_i,x_j) and (x_j,x_i) are treated as one in this paper. An element $c(x_{i_0},x_{j_0})\in M$ is a minimal element of M if there is no $c(x_i,x_j)\in M$ such that $c(x_i,x_j)\subset c(x_{i_0},x_{j_0})$ [2]. Other elements can always be absorbed by the minimal elements in the discernibility matrix.

$f_U(C\bigcup D)=\wedge\{\vee c(x_i,x_j): c(x_i,x_j)\in M, c(x_i,x_j)\neq\phi\}$ is referred to as the discernibility function of $(U, C\bigcup D)$. By the distribution and absorption laws, $f_U(C\bigcup D)=\bigvee_{i=1}^{t}(\wedge A_i)$ is the minimal disjunctive normal form of $f_U(C\bigcup D)$ [25]. $\mathrm{Red}_U(C\bigcup D)=\{A_1,\cdots,A_t\}$ is then the set of all the reducts [25]. The intersection of all the reducts is denoted as $core_U=\bigcap\mathrm{Red}_U(C\bigcup D)$ which is called the core of C. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

many real-world applications, it is enough to find one reduct. The following theorems provide the basis for finding a reduct.

Theorem 2.1 ([25]) $core_U = \{a : c(x_i, x_j) = \{a\} \in M\}$.

Theorem 2.2 ([25]) $B \subseteq C$ is a reduct of C if and only if the following conditions hold: 1) for $\forall c(x_i, x_j) \neq \phi$, $B \cap c(x_i, x_j) \neq \phi$; 2) for $\forall a \in B$, there exists $c(x_i, x_j) \neq \phi$ such that $(B - \{a\}) \cap c(x_i, x_j) = \phi$.

Theorem 2.1 implies the core is the union of all singletons in the discernibility matrix, whereas Theorem 2.2 provides a convenient way to test if a subset of attributes is a reduct [25]. In Theorem 2.2, the first condition states that it is sufficient to employ a reduct to together distinguish all sample pairs in Φ; the second one states that it is necessary for each attribute in a reduct to discern the sample pair in Φ, i.e., for $\forall a \in B$ there always exists a sample pair which can be only distinguished by a, but cannot be distinguished by $B - \{a\}$. In a word, a reduct is a minimal subset of attributes that together distinguishes all sample pairs in Φ. Theorem 2.2 provides the basis for our attribute addition and deletion criterions of this paper.

The discernibility matrix based approach has to compute and store all elements in the discernibility matrix. As a result, it is not suitable for large volumes of data, even when finding a reduct. Rather than finding the whole discernibility matrix, sample pair selection was proposed in [2] to locate all minimal elements in the discernibility matrix by searching the corresponding sample pairs, so that the search space and time of reducts are reduced effectively. Experiments in [2] have shown that this algorithm can find reducts effectively. Thus, sample pair selection will be the springboard of our research in this paper. Interested readers may consult [2] for more details on how to employ sample pair selection to compute minimal elements and reduct.

III. Incremental mechanisms of attribute reduction

As aforementioned, algorithms in [2] aim to locate all minimal elements in the discernibility matrix by selecting certain sample pairs and then find reducts by only using minimal elements. This idea is the theoretical foundation of our incremental mechanisms of attribute reduction in this section, i.e., instead of updating the whole discernibility matrix, we only update minimal elements in order to enhance the time efficiency of updating one reduct.

In this section, the scheme of active sample selection is first studied to classify each incoming sample as a useless or useful sample with respect to a current dataset. The useless samples do not contribute a bit to the incremental computation of attribute reduction, so they can be ignored or filtered out to save space and time in the incremental computation. The useful samples are selected to update the current reduct. At the arrival of a useful sample, the attribute reduction process is then developed via the attribute addition and deletion criterions that reveal which attributes should be added into and deleted from the current reduct.

A. Notations

To precisely describe the incremental mechanisms of attribute reduction, we introduce some symbols here.

We assume $(U, C \cup D)$ is a current dataset with a reduct red computed by Algorithm 3 in [2]. $ME = \{c_1, \cdots, c_r\}$ is the minimal element set in the discernibility matrix. $KP = \{p_1, \cdots, p_r\}$ is a family of sample pair sets corresponding to the minimal elements, where p_i is the sample pair set corresponding to c_i ($i = 1, \cdots, r$), i.e., p_i is the locations of the minimal element c_i in the discernibility matrix. Now, a sample x is added into $(U, C \cup D)$. Starting from red, we can incrementally obtain a reduct red_x of $(U \cup \{x\}, C \cup D)$. $ME' = \{c_1', \cdots, c_{r'}'\}$ is the minimal element set in the discernibility matrix of $(U \cup \{x\}, C \cup D)$. $KP' = \{p_1', \cdots, p_{r'}'\}$ is the family of sample pair sets corresponding to the minimal elements in $(U \cup \{x\}, C \cup D)$.

B. Useless and Useful Samples

Suppose x is an incremental sample and $[x]_C$ is the equivalence class of $IND(C)$ including x in $(U \cup \{x\}, C \cup D)$, then $[x]_C - \{x\}$ is the sample set in U that shares the same condition description with x. At the arrival of the incremental sample x, there are four possible cases:

Case 1: The new sample x shares the same condition description with some inconsistent samples of U on C, which implies that $|d([x]_C - \{x\})| > 1$ holds for $(U, C \cup D)$.

Case 2: The new sample x shares the same condition and decision descriptions with some consistent samples of U on C, which implies that $|d([x]_C - \{x\})| = 1$ holds for $(U, C \cup D)$ and $|d([x]_C)| = 1$ holds for $(U \cup \{x\}, C \cup D)$.

Case 3: The new sample x shares the same condition description and different decision values with some consistent samples of U on C, which implies $|d([x]_C - \{x\})| = 1$ holds for $(U, C \cup D)$ and $|d([x]_C)| > 1$ holds for $(U \cup \{x\}, C \cup D)$.

Case 4: The new sample x shares different condition descriptions with any sample of U, which means that $[x]_C = \{x\}$ holds for $(U \cup \{x\}, C \cup D)$.

If the new sample is in Case 1 or Case 2, clearly, the minimal elements and the reducts of the new decision table are identical with those of the current dataset. Moreover, the new sample is dispensable to the incremental computation of attribute reduction, since by definition there is always a sample in the current dataset that replaces the new sample in the incremental computation of attribute reduction. Thus, a new sample is said to be useless with respect to the current dataset, if it is in Case 1 or Case 2. Conversely, a new sample is useful with respect to the current dataset if it is in Case 3 or Case 4. We will discuss in detail how to deal with Case 3 and Case 4 in the next subsection.

Based on the definition of useless sample, a scheme named active sample selection will be designed in Section IV to decide whether each incoming sample can be filtered out prior to performing the incremental computation. It seems that the conditions satisfied by the useless sample are simple, but it is of great significance to discard these useless samples. The reason is as follows. For one thing, the space can be saved, and the time efficiency of the incremental computation can be also improved, which will be demonstrated by experimental results in Section V. For another, such a mechanism to filter out useless samples has not yet been developed in the incremental computation of attribute reduction, since the existing incremental attribute reduction algorithms passively employ all incoming samples. Therefore, it is highly desirable to design such a scheme of active sample selection.

C. Incremental Mechanism of Attribute Reduction for Case 3

In this section we study the incremental mechanism of attribute reduction when a new sample satisfying Case 3 arrives. We first discuss how to incrementally compute the minimal elements of the new decision table and their corresponding sample pairs. Based on the updated minimal elements, we then develop the attribute addition and deletion criterions that determine how to add attributes into and delete attributes from the current reduct.

1) Incremental computing of minimal elements

When an incremental sample x in Case 3 is added into a current dataset, we first determine which elements in ME may not be the minimal elements in the discernibility matrix of the new decision table. We then locate all possible new minimal elements in the discernibility matrix of the new decision table.

For Case 3, samples in $[x]_C - \{x\}$ are inconsistent in $(U \bigcup \{x\}, C \bigcup D)$, but they are consistent in $(U, C \bigcup D)$. This fact implies $POS_{U\bigcup\{x\}}(C,D) = POS_U(C,D) - ([x]_C - \{x\})$. By the definition of discernibility matrix, it is unnecessary for $(U \bigcup \{x\}, C \bigcup D)$ to discern (x_i, x_j), where $\forall x_i \in [x]_C - \{x\}$ and $\forall x_j \in U - POS_U(C,D)$. However, it is necessary for $(U, C \bigcup D)$ to discern the sample pairs, and they may also determine some minimal elements in ME. Let $\omega_1 = \{(x_i, x_j) : x_i \in [x]_C - \{x\}, x_j \in U - POS_U(C,D)\}$ and $p_k^* = p_k - \omega_1$ for $\forall p_k \in KP$. Then sample pairs in p_k^* need to be discerned and can still determine $c_k \in ME$ in $(U \bigcup \{x\}, C \bigcup D)$. $p_k^* = \phi$ implies it is unnecessary for $(U \bigcup \{x\}, C \bigcup D)$ to discern all sample pairs in p_k, i.e., the element $c_k \in ME$ determined by p_k is not a minimal element in the discernibility matrix of $(U \bigcup \{x\}, C \bigcup D)$. Let $M^* = ME - \{c_k \in ME : p_k^* = \phi\}$, then it is necessary for $(U \bigcup \{x\}, C \bigcup D)$ to discern sample pairs corresponding to each element in M^*, i.e., elements in M^* may still be the minimal elements in the discernibility matrix of the new decision table.

Next, we locate all possible new minimal elements in the discernibility matrix of the new decision table. For Case 3, samples in $[x]_C$ are inconsistent on C, which means it is necessary for $(U \bigcup \{x\}, C \bigcup D)$ to discern the sample pair (x_i, x_j) where $\forall x_j \in POS_{U\bigcup\{x\}}(C,D)$ and $\forall x_i \in [x]_C$. Since by definition $c(x, x_j) = c(x_k, x_j)$ holds for $\forall x_k \in [x]_C$, we only compute $c(x, x_j) = \{a \in C : a(x) \neq a(x_j)\}$ for $\forall (x, x_j) \in \omega_2 = \{(x, x_j) : x_j \in POS_{U\bigcup\{x\}}(C,D)\}$. Moreover, by definition, the minimal elements of $\{c(x, x_i) : (x, x_i) \in \omega_2\}$ can be computed as $\{c_1^{**}, \cdots, c_s^{**}\}$, and $p_i^{**} \subseteq \omega_2$ is the sample pair set corresponding to c_i^{**} ($i = 1, \cdots, s$). Obviously, the minimal element set in the discernibility matrix of $(U \bigcup \{x\}, C \bigcup D)$ is contained in $M^* \bigcup \{c_1^{**}, \cdots, c_s^{**}\}$, i.e., $ME' \subseteq M^* \bigcup \{c_1^{**}, \cdots, c_s^{**}\}$. To obtain the new minimal element set ME', the following theorem is given based on the definition of minimal element.

Theorem 3.1 For $\forall c_i^{**}$, the following statements hold:

1) If $\exists c_k \in M^*$ such that $c_k \subset c_i^{**}$, $c_k \in ME'$ and $c_i^{**} \notin ME'$ hold for $(U \bigcup \{x\}, C \bigcup D)$;

2) If $\exists c_k \in M^*$ such that $c_i^{**} \subset c_k$, $c_k \notin ME'$ and $c_i^{**} \in ME'$ hold for $(U \bigcup \{x\}, C \bigcup D)$;

3) For $\forall c_k \in M^*$, $c_i^{**} \not\subset c_k$ and $c_k \not\subset c_i^{**}$ imply that $c_i^{**}, c_k \in ME'$ hold for $(U \bigcup \{x\}, C \bigcup D)$;

4) If $\exists c_k \in M^*$ such that $c_i^{**} = c_k$, $c_k \in ME'$ and its corresponding sample pair set is $p_k^* \bigcup p_i^{**}$.

In Theorem 3.1, Statement 1) states that c_i^{**} can be absorbed by an element in $M^* \subseteq ME$, so that it is not a minimal element of $(U \bigcup \{x\}, C \bigcup D)$; Statement 2) means that c_i^{**} can absorb a minimal element c_k in $M^* \subseteq ME'$, so that c_k is not a minimal element of $(U \bigcup \{x\}, C \bigcup D)$, but c_i^{**} is a minimal element of $(U \bigcup \{x\}, C \bigcup D)$; Statement 3) means both c_i^{**} and c_k are minimal elements of $(U \bigcup \{x\}, C \bigcup D)$; Statement 4) implies that c_i^{**} is identical with c_k.

The concrete steps of the incremental computing of the minimal elements are as follows.

Step 1: Find $\omega_1 = \{(x_i, x_j) : x_i \in [x]_C - \{x\}, x_j \in U - POS_U(C,D)\}$ and $p_k^* = p_k - \omega_1$ for $\forall p_k \in KP$, and let $M^* = ME - \{c_k \in ME : p_k^* = \phi\}$;

Step 2: Find $c(x, x_i) = \{a \in C : a(x) \neq a(x_i)\}$ for $\forall x_i \in POS_{U\bigcup\{x\}}(C,D)$, and then compute their minimal elements $\{c_1^{**}, \cdots, c_s^{**}\}$ and the corresponding sample pairs $\{p_1^{**}, \cdots, p_s^{**}\}$;

Step 3: By Theorem 3.1, compute the minimal elements $ME' \subseteq M^* \bigcup \{c_1^{**}, \cdots, c_s^{**}\}$ and KP'.

2) Attribute addition and deletion criterions

Based on the incremental computing of minimal elements, this subsection develops the attribute addition and deletion criterions to reveal which attributes should be added into and deleted from a current reduct.

For Case 3, the current reduct red either keeps the discernibility of all condition attributes, or does not, i.e. $POS_{U\bigcup\{x\}}(C,D)=POS_{U\bigcup\{x\}}(red,D)$ or $POS_{U\bigcup\{x\}}(C,D)\supset POS_{U\bigcup\{x\}}(red,D)$. This fact implies that there are two possibilities: $[x]_C=[x]_{red}$ and $[x]_C\subset[x]_{red}$. In terms of the two possibilities, we study the incremental mechanisms of attribute reduction for Case 3.

Theorem 3.2 For Case 3, $[x]_C=[x]_{red}$ indicates red contains a reduct of $(U\bigcup\{x\},C\bigcup D)$.

Proof: For Case 3, we have

$$POS_{U\bigcup\{x\}}(C,D)=POS_U(C,D)-([x]_C-\{x\}),\quad (1)$$

$$POS_{U\bigcup\{x\}}(red,D)=POS_U(C,D)-([x]_{red}-\{x\}).\quad (2)$$

$[x]_C=[x]_{red}$ implies that $POS_{U\bigcup\{x\}}(C,D)=POS_{U\bigcup\{x\}}(red,D)$ holds for $(U\bigcup\{x\},C\bigcup D)$. Thus, red contains a reduct of $(U\bigcup\{x\},C\bigcup D)$.

Theorem 3.2 shows that the current reduct red is either a reduct of the new decision table, or properly contains a reduct of the new decision table. The following criterion derived from Theorem 2.2, is used to find a reduct red_x of the new decision table.

Attribute Deletion Criterion 1: Attribute $a\in red$ can be deleted from red if the following statement holds: $(red-\{a\})\bigcap c_k'\neq\phi$ for $\forall c_k'\in ME'$.

Attribute Deletion Criterion 1 means that $red-\{a\}$ can discern sample pairs that need to be discerned in the new decision table, so that the attribute a can be deleted from red according to Theorem 2.2. If the attribute deletion criterion does not hold for $\forall a\in red$, i.e., red is a minimal attribute subset discerning sample pairs that need to be discerned in the new decision table, red is just a reduct of the new decision table. Otherwise, if the criterion holds for $a\in red$, any sample pair discerned by the attribute a can be also discerned by the attributes in $red-\{a\}$, which implies that a can be deleted from red. We can continue applying the attribute deletion criterion until the criterion does not hold. Thus, a reduct red_x can be obtained.

For Case 3 and $[x]_C\subset[x]_{red}$, the incremental mechanism of attribute reduction is analyzed below. By equations (1) and (2), $POS_{U\bigcup\{x\}}(C,D)=POS_{U\bigcup\{x\}}(red,D)\bigcup([x]_{red}-[x]_C)$ and $POS_{U\bigcup\{x\}}(red,D)\bigcap([x]_{red}-[x]_C)=\phi$ hold for $(U\bigcup\{x\},C\bigcup D)$. This fact implies that red can keep samples in $POS_{U\bigcup\{x\}}(red,D)$ consistent on C, but cannot keep samples in $[x]_{red}-[x]_C$ consistent on C. To find red_x starting from red, we need to add attributes into red until samples in $[x]_{red}-[x]_C$ are consistent in $(U\bigcup\{x\},C\bigcup D)$. Since $[x]_{red}=([x]_{red}-[x]_C)\bigcup[x]_C$ holds and $[x]_{red}$ can be divided into some smaller equivalence classes by adding attributes into red, we only need to add attributes into red until the positive region of the decision sub-table $([x]_{red},C\bigcup D)$ are just $[x]_{red}-[x]_C$. Thus, we have the following attribute addition criterion.

Attribute Addition Criterion 1: Attribute subset $B\subseteq C-red$ can be added to red if B is a minimal addition subset satisfying $POS_{[x]_{red}}(B\bigcup red,D)=[x]_{red}-[x]_C$.

According to the above criterion, $[x]_{red}-[x]_C$ is all consistent samples of $[x]_{red}$ with respect to $B\bigcup red$. Since samples in $POS_{U\bigcup\{x\}}(red,D)$ are also consistent on $B\bigcup red$, $POS_{U\bigcup\{x\}}(B\bigcup red,D)=POS_{U\bigcup\{x\}}(red,D)\bigcup([x]_{red}-[x]_C)$ holds, which implies $POS_{U\bigcup\{x\}}(C,D)=POS_{U\bigcup\{x\}}(B\bigcup red,D)$. Thus, a reduct red_x of the new decision table is a subset of $B\bigcup red$, i.e., $red_x\subseteq B\bigcup red$. Since B is a minimal addition attribute set satisfying the condition that samples in $[x]_{red}-[x]_C$ are consistent after adding attributes into red, there always exists $x_0\in[x]_{red}-[x]_C$ satisfying $\left|d([x_0]_{red\bigcup(B-\{a\})})\right|>1$ for $\forall a\in B$. This fact implies that each attribute in B is necessary, i.e., $B\subseteq red_x$. However, there may be redundant attributes in red due to the addition of attribute subset B. Thus, we only need to delete redundant attributes from red. According to Theorem 2.2, we have the following attribute deletion criterion.

Attribute Deletion Criterion 2: Attribute $a\in red$ can be deleted from red if the following statement holds: $(B\bigcup(red-\{a\}))\bigcap c_k'\neq\phi$ for $\forall c_k'\in ME'$.

If the criterion above does not hold for $\forall a\in red$, there always exists a sample pair (x_i,x_j) that can be only distinguished by the attribute a. This fact indicates that there are no redundant attributes in red, i.e., $B\bigcup red$ is a reduct of the new decision table. If the criterion holds for $a\in red$, the attribute a should be deleted from red. By continuing applying the attribute deletion criterion, we can obtain red_x that is the union of B and the remaining attributes in red.

To sum up, if a new sample is in Case 3 and $[x]_C=[x]_{red}$, we can apply Attribute Deletion Criterion 1 to obtain a reduct of the new decision table; if it is in Case 3 and $[x]_C\subset[x]_{red}$, we first employ Attribute Addition Criterion 1 to add some attributes into the current reduct and then use Attribute Deletion Criterion 2 to delete redundant attributes from the current reduct until a reduct of the new decision table is obtained. The flowchart of the incremental attribute reduction mechanism for Case 3 is shown in Fig. 1.

In the following, we employ an example to illustrate the above incremental mechanisms of attribute reduction for Case 3.

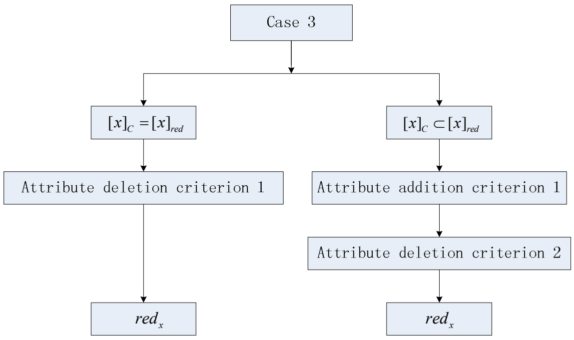

Fig. 1. Incremental mechanism of attribute reduction for Case 3.

TABLE I
A DECISION TABLE

U	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	1	0	1	1	0	1	1
x_2	0	0	1	1	0	1	0
x_3	1	1	0	1	0	1	0
x_4	1	1	0	1	0	1	1
x_5	0	0	0	0	1	0	0
x_6	1	0	1	1	0	0	0
x_7	1	0	0	0	1	1	1
x_8	1	1	1	0	0	0	1

Example 3.1 A decision table is shown in Table I, where $U = \{x_1, \cdots, x_8\}$ is the set of samples, $C = \{a_1, \cdots, a_6\}$ is the set of condition attributes and $D = \{d\}$ is the decision attribute set.

By the method of [2], the minimal element set and their corresponding sample pair family are computed as $ME = \{c_1 = \{a_1\}, c_2 = \{a_2, a_3\}, c_3 = c\{a_6\}, c_4 = \{a_2, a_4\}\}$ and $KP = \{p_1, p_2, p_3, p_4\}$, where $p_1 = \{(x_2, x_1)\}$, $p_2 = \{(x_3, x_1), (x_4, x_1)\}$, $p_3 = \{(x_6, x_1)\}$, $p_4 = \{(x_8, x_6)\}$. By Algorithm 3 in [2], we can obtain a reduct $red = \{a_1, a_2, a_6\}$.

Suppose $x = [1,0,1,1,0,1,0]$ is added to Table I. Then we have $[x]_C = \{x, x_1\}$ and $|d([x]_C)| = 2$, which implies that x is in Case 3. By Step 1, we have $\omega_1 = \{(x_3, x_1), (x_4, x_1)\}$, $M^* = \{c_1, c_3, c_4\}$, $p_1^* = p_1$, $p_2^* = \phi$, $p_3^* = p_3$ and $p_4^* = p_4$. By Step 2, we find $c(x, x_2) = \{a_1\}$, $c(x, x_5) = \{a_1, a_3, a_4, a_5, a_6\}$, $c(x, x_6) = \{a_6\}$, $c(x, x_7) = \{a_3, a_4, a_5\}$ and $c(x, x_8) = \{a_2, a_4, a_6\}$. By the definition of minimal element, we have $c_1^{**} = \{a_1\}$, $c_2^{**} = \{a_6\}$, $c_3^{**} = \{a_3, a_4, a_5\}$, $p_1^{**} = \{(x, x_2)\}$, $p_2^{**} = \{(x, x_6)\}$ and $p_3^{**} = \{(x, x_7)\}$. By Step 3, we obtain $ME' = \{c_1', c_2', c_3', c_4'\}$, where $c_1' = \{a_1\}$, $c_2' = \{a_6\}$, $c_3' = \{a_2, a_4\}$, $c_4' = \{a_3, a_4, a_5\}$. $KP' = \{p_1', p_2', p_3', p_4'\}$ is the sample pair family, where $p_1' = \{(x_2, x_1), (x, x_2)\}$, $p_2' = \{(x_6, x_1), (x, x_6)\}$, $p_3' = \{(x_8, x_6)\}$, $p_4' = \{(x, x_7)\}$.

Furthermore, we have $[x]_C \subset [x]_{red}$. Thus, we first apply Attribute Addition Criterion 1 to obtain the minimal addition subset $B = \{a_3\}$. By applying Attribute Deletion Criterion 2, we then find any attribute in red cannot be deleted, which implies $red_x = B \cup red = \{a_1, a_2, a_3, a_6\}$ is a reduct of the new decision table.

D. Incremental Mechanism of Attribute Reduction for Case 4

In this section we study the incremental mechanism of attribute reduction when a new sample satisfying Case 4 arrives. We discuss how to incrementally compute the minimal elements. On the basis of the updated minimal elements, we also develop the attribute addition and deletion criterions that reveal how to add attributes into and delete attributes from a current reduct.

1) Incremental computing of minimal elements

When a new sample x satisfying Case 4 is added into $(U, C \cup D)$, the positive region of D with respect to C is $POS_{U \cup \{x\}}(C, D) = POS_U(C, D) \cup \{x\}$. In $(U \cup \{x\}, C \cup D)$, $M^* = ME$ is thus the minimal elements of the discernibility attribute sets corresponding to sample pairs in $\{(x_i, x_j) : \forall x_i, x_j \in U\}$. Besides, it is necessary for $(U \cup \{x\}, C \cup D)$ to discern $(x, x_i) \in \psi$, where $\psi = \{(x, x_i) : U - POS_U(C, D)\ or\ x_i \in POS_U(C, D), d(x_i) \neq d(x)\}$. Thus, we need to compute $c(x, x_i) = \{a \in C : a(x) \neq a(x_i)\}$ for $\forall (x, x_i) \in \psi$. By definition, the minimal elements of $\{c(x, x_i) : (x, x_i) \in \psi\}$ can be computed as $\{c_1^{**}, \cdots, c_s^{**}\}$, and $p_i^{**} \subseteq \psi$ is the sample pair set corresponding to c_i^{**} ($i = 1, \cdots, s$). So, the minimal element set ME' of the new decision table is contained in $M^* \cup \{c_1^{**}, \cdots, c_s^{**}\}$. By Theorem 3.1, we can find minimal elements of the new decision table.

For completeness, we also develop the following steps to find ME'.

Step 1: For $\forall (x, x_i) \in \psi = \{(x, x_i) : x_i \in U - POS_U(C, D)\ or\ x_i \in POS_U(C, D), d(x_i) \neq d(x)\}$, calculate $c(x, x_i) = \{a \in C : a(x) \neq a(x_i)\}$, and then compute their minimal elements $\{c_1^{**}, c_2^{**}, \cdots, c_s^{**}\}$ and the corresponding sample pairs $\{p_1^{**}, p_2^{**}, \cdots, p_s^{**}\}$;

Step 2: By Theorem 3.1, compute the minimal elements $ME' \subseteq ME \cup \{c_1^{**}, c_2^{**}, \cdots, c_s^{**}\}$ and KP'.

2) Attribute addition and deletion criterions

In this section we discuss which attributes should be added into and deleted from a current reduct based on the updated minimal elements, when a new sample satisfying Case 4 is added. For Case 4, the current reduct red either keeps the discernibility of all condition attributes, or does not, i.e., $POS_{U \cup \{x\}}(C, D) = POS_{U \cup \{x\}}(red, D)$ or $POS_{U \cup \{x\}}(red, D) = POS_{U \cup \{x\}}(C, D) - [x]_{red}$. Therefore, there are two possibilities: $|d([x]_{red})| = 1$ and $|d([x]_{red})| > 1$. We consider the two possibilities below.

Theorem 3.3 If a new sample is in Case 4 and $|d([x]_{red})| = 1$, red is just a reduct of $(U \cup \{x\}, C \cup D)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

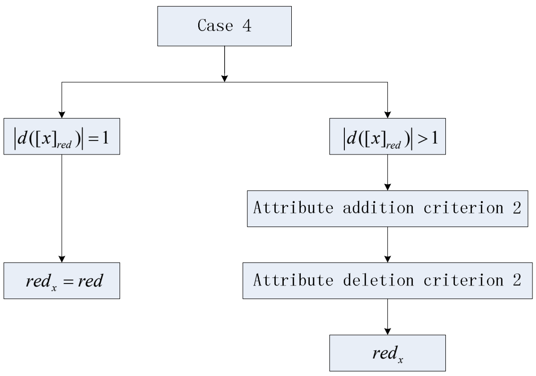

Fig. 2. Incremental mechanism of attribute reduction for Case 4.

Proof: On the one hand, $[x]_C = \{x\}$ implies $POS_{U\bigcup\{x\}}(C,D) = POS_U(C,D)\bigcup\{x\}$ and $|d([x]_{red})|=1$ implies $POS_{U\bigcup\{x\}}(red,D) = POS_U(red,D)\bigcup\{x\}$. Thus, we have $POS_{U\bigcup\{x\}}(C,D) = POS_{U\bigcup\{x\}}(red,D)$. On the other hand, since red is a reduct of $(U, C\bigcup D)$, there exists $x_i \in POS_U(C,D) \subset POS_{U\bigcup\{x\}}(C,D)$ satisfying $|d([x_i]_{red-\{a\}})|>1$ for $\forall a \in red$. Therefore, red is also a reduct of $(U\bigcup\{x\}, C\bigcup D)$.

According to Theorem 3.3, the current reduct is just the reduct of the new decision table when the new sample is in Case 4 and $|d([x]_{red})|=1$. Now we discuss the variation of attribute reduction when adding a new sample satisfying Case 4 and $|d([x]_{red})|>1$. For Case 4, $|d([x]_{red})|>1$ implies x is inconsistent on red. To find red_x starting from red, we first need to add attributes into red until x is consistent. Thus, the following attribute addition criterion is given.

Attribute Addition Criterion 2: An attribute subset $B \subseteq C - red$ can be added to red if B is a minimal addition subset satisfying $|d([x]_{B\bigcup red})|=1$.

According to this criterion, x is consistent on $B\bigcup red$, which implies $POS_{U\bigcup\{x\}}(C,D) = POS_{U\bigcup\{x\}}(B\bigcup red, D)$. Therefore, a reduct red_x of the new decision table is a subset of $B\bigcup red$, i.e., $red_x \subseteq B\bigcup red$. Since B is a minimal addition attribute set that keeps the sample x consistent, $|d([x]_{(B-\{a\})\bigcup red})|>1$ holds for $\forall a \in B$. This fact implies each attribute in B is necessary, i.e., $B \subseteq red_x$. In addition, there may be redundant attributes in red due to the addition of B. By using Attribute Deletion Criterion 2, we can delete redundant attributes from red to obtain a reduct red_x that is the union of B and the remaining attributes in red.

In Case 4, if $|d([x]_{red})|=1$, the current reduct red is just the reduct red_x of the new decision table; if $|d([x]_{red})|>1$, we can first use Attribute Addition Criterion 2 to add attributes into red until the new sample is consistent, and then apply Attribute Deletion Criterion 2 to delete redundant attributes from red until red_x is obtained. The flowchart of the attribute reduction process is shown in Fig.2.

The following example is given to illustrate the above incremental mechanisms.

Example 3.2 Consider Example 3.1. Suppose the new sample $x = [1,1,1,1,0,1,0]$ is added into Table I. Then, we have $[x]_C = \{x\}$, which implies that the new sample is in Case 4.

By Step 1, we have $c(x,x_1) = \{a_2\}$, $c(x,x_3) = c(x,x_4) = \{a_3\}$, $c(x,x_7) = \{a_2,a_3,a_4,a_5\}$, $c(x,x_8) = \{a_4,a_6\}$. By the definition of minimal element, we have $c_1^{**} = \{a_2\}$, $c_2^{**} = \{a_3\}$, $c_3^{**} = \{a_4,a_6\}$, $p_1^{**} = \{(x,x_1)\}$, $p_2^{**} = \{(x,x_3),(x,x_4)\}$ and $p_3^{**} = \{(x,x_8)\}$. By Step 2, the updated minimal elements are computed as $ME' = \{c_1^{'}, c_2^{'}, c_3^{'}, c_4^{'}\}$, where $c_1^{'} = \{a_1\}$, $c_2^{'} = \{a_2\}$, $c_3^{'} = \{a_3\}$, $c_4^{'} = \{a_6\}$. The corresponding sample pair family is $KP = \{p_1^{'}, p_2^{'}, p_3^{'}, p_4^{'}\}$, where $p_1^{'} = \{(x_2,x_1)\}$, $p_2^{'} = \{(x,x_1)\}$, $p_3^{'} = \{(x,x_3),(x,x_4)\}$ and $p_4^{'} = \{(x_6,x_1)\}$.

Moreover, since $|d([x]_{red})|=2$ holds, we can use Attribute Addition Criterion 2 to compute the addition attribute subset that is $B = \{a_3\}$. Applying Attribute Deletion Criterion 2, we then find no attribute in red can be deleted. Therefore, a reduct of $(U\bigcup\{x\}, C\bigcup D)$ is $red_x = B\bigcup red = \{a_1,a_2,a_3,a_6\}$.

IV. Active Sample Selection Based Incremental Algorithm for Attribute Reduction

In this section we present our active sample selection based incremental algorithm for attribute reduction which works by integrating a scheme of actively selecting samples into the attribute reduction process. The scheme of active sample selection is first designed to determine whether each incoming sample is useless or useful with respect to a current dataset. The incremental attribute reduction algorithm is then developed, which actively filters out useless samples and selects useful samples to perform the incremental computation.

A. Active Sample Selection

Based on the definition of the useless sample, our active sample selection algorithm is designed as follows.

Algorithm 4.1. Active sample selection

Input: An original dataset $(U, C\bigcup D)$, a new sample x.
Output: cr, which is 0 if x is a useless sample, or is 1 if it is a useful sample.
Initialize: $cr = 1$.
1. Compute $|d([x]_C - \{x\})|$ and $|d([x]_C)|$;
2. If $|d([x]_C - \{x\})|>1$, %Case 1
3. $cr = 0$;
4. Elseif $|d([x]_C - \{x\})|=1$ and $|d([x]_C)|=1$, %Case 2
5. $cr = 0$;
6. End if.
7. Return cr.

The time complexity of Algorithm 4.1 is $O(|U|)$, where $|U|$ is the number of samples in the current dataset. $cr = 0$ implies the new sample is a useless sample with respect to the current

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

dataset, and thus it will be discarded before performing the incremental computation; $cr=1$ implies it is a useful sample, so it will be selected to perform the incremental computation.

B. Active Sample Selection Based Incremental Algorithm for Attribute Reduction

In this section, we first present the incremental algorithm for computing minimal elements, which is the basis of our incremental algorithm for attribute reduction. We then present our active sample selection based incremental algorithm for attribute reduction. In this algorithm only useful samples are selected to perform the incremental computation of attribute reduction, while useless samples will be filtered out.

Algorithm 4.2. Incremental algorithm for computing minimal elements

Input: An original dataset $(U, C\bigcup D)$, $POS_U(C,D)$, $ME=\{c_1,\cdots,c_r\}$, $KP=\{p_1,\cdots,p_r\}$, and a new sample x.

Output: $ME'=\{c_1^{'},\cdots,c_{r'}^{'}\}$, $KP'=\{p_1^{'},\cdots,p_{r'}^{'}\}$ and $POS_{U\bigcup\{x\}}(C,D)$.

Initialize: $M^*=ME$, $K^*=KP$.

1. Compute $[x]_C$, $|d([x]_C-\{x\})|$ and $|d([x]_C)|$;
2. If $|d([x]_C-\{x\})|=1$ and $|d([x]_C)|>1$, %Case 3
3. Compute $\omega_1=\{(x_i,x_j): x_i\in[x]_C-\{x\}, x_j\in U-POS_U(C,D)\}$ and $POS_{U\bigcup\{x\}}(C,D)=POS_U(C,D)-([x]_C-\{x\})$;
4. For each $p_i\in K^*$, let $p_i=p_i-\omega_1$;
5. Let $M^*=M^*-\{c_k\in M^*: p_k=\phi\}$;
6. For each $x_i\in POS_{U\bigcup\{x\}}(C,D)$, compute $c(x,x_i)=\{a\in C: a(x)\neq a(x_i)\}$;
7. Let $M=\{c(x,x_i): \forall x_i\in POS_{U\bigcup\{x\}}(C,D)\}$ and turn to Step 16;
8. End if
9. If $[x]_C=\{x\}$, %Case 4
10. Let $POS_{U\bigcup\{x\}}(C,D)=POS_U(C,D)\bigcup\{x\}$;
11. Compute $\omega_2=\{x_i: x_i\in U-POS_{U\bigcup\{x\}}(C,D); or\ x_i\in POS_{U\bigcup\{x\}}(C,D), d(x)\neq d(x_i)\}$;
12. For each $x_i\in\omega_2$, compute $c(x,x_i)=\{a\in C: a(x)\neq a(x_i)\}$;
13. Let $M=\{c(x,x_i): \forall x_i\in\omega_2\}$ and turn to Step 16;
14. End if.
15. Let $M^{**}=\phi$, $KP^{**}=\phi$;
16. While ($M\neq\phi$),
17. Select $c(x,x_{i_0})$ satisfying $|c(x,x_{i_0})|=\min\{|c(x,x_i)|: c(x,x_i)\in M\}$;
18. Let $M^{**}=[M^{**}; c(x,x_{i_0})]$ and $p^{**}=(x,x_{i_0})$;
19. For each $c(x,x_i)\in M$,
20. If $c(x,x_i)\supset c(x,x_{i_0})$, let $c(x,x_i)=\phi$; if $c(x,x_i)=c(x,x_{i_0})$, let $p^{**}=p^{**}\bigcup\{(x,x_i)\}$ and $c(x,x_i)=\phi$;
21. End for.
22. End while.
23. Let $ME'=M^*\bigcup M^{**}$ and $KP'=K^*\bigcup KP^{**}$;
24. For each $c_i^{**}\in M^{**}$,
25. For each $c_j\in M^*$,
26. If $c_i^{**}\subset c_j$, let $ME'=ME'-c_j$ and $KP'=KP'-p_j$;
27. If $c_i^{**}\supset c_j$, let $ME'=ME'-c_i^{**}$ and $KP'=KP'-p_i^{**}$;
28. If $c_i^{**}=c_j$, let $ME'=ME'-c_i^{**}$, $KP'=KP'-p_i^{**}$, and $p_j=p_j\bigcup p_i^{**}$.
29. End for.
30. End for.
31. Output ME', KP' and $POS_{U\bigcup\{x\}}(C,D)$.

The time complexity of Algorithm 4.2 is $O(|U|(|ME|+|U|))$. By the incremental computing of minimal elements, our active sample selection based incremental algorithm for attribute reduction is developed as follows.

Algorithm 4.3. Active sample selection based incremental algorithm for attribute reduction (ASS-IAR)

Input: An original dataset $(U, C\bigcup D)$, $POS_U(C,D)$, $ME=\{c_1,\cdots,c_r\}$, $KP=\{p_1,\cdots,p_r\}$, and a new sample x.

Output: A new reduct red_x.

Initialize: $red_x=red$.

1. Compute cr by Algorithm 4.1;
2. If $cr=0$,
3. Let $U=U$, $red_x=red$, and return red_x;
4. Else
5. Compute $[x]_C$, $|d([x]_C-\{x\})|$, $|d([x]_C)|$, $[x]_{red_x}$ and $|d([x]_{red_x})|$;
6. Compute the minimal elements ME' by Algorithm 4.2 and let $U=U\bigcup\{x\}$;
7. Let $A=red_x$, $B=\phi$ and $S=C-red_x$;
8. If $|d([x]_C-\{x\})|=1$ and $|d([x]_C)|>1$, %Case 3
9. If $[x]_C=[x]_{red_U}$,
10. If $\exists a\in A$ satisfying $(red_x-\{a\})\bigcap c_i^{'}\neq\phi$ for $\forall c_i^{'}\in ME'$,
11. Let $red_x=red_x-\{a\}$, and turn to Step 10;
12. Else
13. Return red_x;
14. End if.
15. Else
16. While ($|POS_{[x]_{red_x}}(B\bigcup red_x, D)|\neq|[x]_{red_x}-[x]_C|$),
17. For each $a_i\in S$, compute $POS_{[x]_{red_x}}(B\bigcup red_x\bigcup\{a_i\}, D)$;
18. Select a_k satisfying $|POS_{[x]_{red_x}}(B\bigcup red_x\bigcup\{a_k\}, D)|=\max_{a_i\in S}\{|POS_{[x]_{red_x}}(B\bigcup red_x\bigcup\{a_i\}, D)|\}$;
19. Let $B=[B, a_k]$ and $S=S-\{a_k\}$;
20. End while.
21. Turn to Step 35.
22. End if.
23. End if.
24. If $[x]_C=\{x\}$, %Case 4
25. If $|d([x]_{red_U})|=1$,
26. Return red_x;
27. Else
28. While ($|d([x]_{B\bigcup red_x})|>1$),
29. For each $a_i\in S$, compute $|d([x]_{\{a_i\}\bigcup B\bigcup red_x})|$;
30. Select a_k such that $|d([x]_{\{a_k\}\bigcup B\bigcup red_x})|$ is minimum, and let $B=[B, a_k]$ and $S=S-\{a_k\}$;
31. End while.
32. Turn to Step 35;
33. End if.
34. End if.
35. If $\exists a\in A$ satisfying $(B\bigcup(red_x-\{a\}))\bigcap c_i^{'}\neq\phi$ for $\forall c_i^{'}\in ME'$,

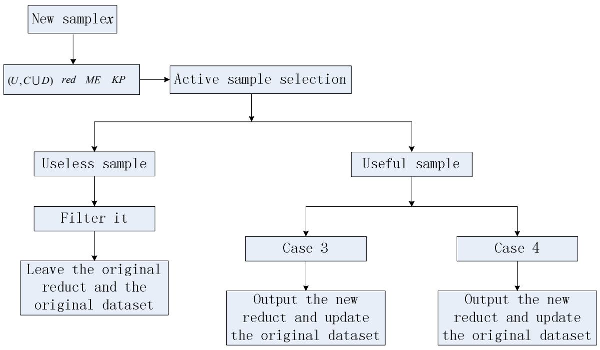

Fig. 3. The process of ASS-IAR.

36. Let $red_x = red_x - \{a\}$ and turn to Step 35;
37. Else
38. Let $red_x = B \cup red_x$ and return red_x;
39. End if.

In Algorithm 4.3 (i.e., ASS-IAR), each incremental sample is first vetted as to whether it is useless or useful with respect to the current dataset. If it is useless, it will be filtered out without being involved in the incremental computation. If it is useful, it will be used in the incremental computation. The time complexity of ASS-IAR is $O(|U|)$ in most optimistic case where the new sample is useless. Its time complexity is $O(|U|(|U||C|+|ME|))$ in the most pessimistic case where the new sample is useful. The flowchart of ASS-IAR is shown in Fig. 3.

Remark: Bireduct [36]-[37] is an attempt in rough sets to mix the process of reducing attributes and the process of selecting samples that are discernible by those attributes. In [38], it has been successfully applied to data stream, where data samples occur consecutively. It seems that the bireduct method resembles ASS-IAR since they both share the scheme of adding/removing samples and attributes. However, there are mainly the following differences between bireduct and ASS-IAR. The first one is they select samples in different fashions. Bireduct adds the newly joined sample with removing the oldest samples (i.e., samples of a current dataset), which cannot be discerned with the new sample by using the attribute set of the temporal bireduct. ASS-IAR, which employs our active sample selection to evaluate each newly joined sample based on its usefulness, filters out useless incoming samples and selects useful incoming samples to perform the incremental computation. The second one is they reduce attributes in different modes. Bireduct selects a minimal attribute subset discerning the sample set of a temporal bireduct, i.e., a reduct for the sample set in the bireduct. ASS-IAR adds attributes when a current reduct is incapable to keep the consistency of new dataset (i.e., the dataset after adding the newly joined sample), and removes redundant attributes in a current reduct due to the addition of attributes. The third one is the attribute subset obtained by the bireduct method is a temporal reduct, while the reduct obtained by ASS-IAR is a reduct of the whole dataset.

ASS-IAR can filter out useless samples to save memory space as well as runtime. As illustrated in B of Section III, active sample selection is very simple but highly effective. In order to demonstrate the effectiveness of active sample selection, we experimentally compare it with a variant of the algorithm, denoted by IAR. IAR passively employs all incremental samples and is constructed by only changing Step 3 of ASS-IAR into "Let $U = U \cup \{x\}$, $ME' = ME$, $KP' = KP$ and return $red_x = red$". Obviously, the only difference between ASS-IAR and IAR is that ASS-IAR filters out useless samples, whereas IAR does not.

V. Experimental Comparisons

In this section, we experimentally evaluate the time efficiency of ASS-IAR by comparing with several feature selection methods on nine UCI datasets. These methods can be roughly divided into two types. One type is rough-set-based feature selection methods. They are two incremental attribute reduction algorithms, i.e., IAR and GIARC [13]. The other type is non-rough-set feature selection methods. They are RELIEFF [47], SCRAP [48], and CONSISTENCY [49].

Before presenting our experiments, the following fact is noted. As the preprocessing step of the incremental algorithms in our experiments, Algorithm 3 in [2] is only run on the original dataset to obtain the original minimal elements, their sample pairs and the original reduct, which are applied to the incremental algorithms. ASS-IAR will not be compared with Algorithm 3 in [2] since it has been shown experimentally that this algorithm is often stopped due to out of memory in current software and hardware environments when dealing with large datasets.

A. Experimental Setup

The experiments in this section are set up as follows.

The hardware environment: Windows 7PC and Intel (R) Xeon (R) CPU E5-2620 0 @ 2.00 GHz 2.00 GHz and 80 GB memory.

Dataset: Nine datasets from University of California, Irvine (UCI) Machine Learning Repository [16] are used (Table II), where we replace the missed values with the most frequently value of an symbolic attribute and the mean value of a real-valued attribute on two missing-valued datasets 'Soybean' and 'Spam'.

Dataset discretization: The fuzzy C-mean clustering algorithm proposed in [33] is used to discretize real-valued condition attributes.

Dataset Split: When using algorithms ASS-IAR, IAR and GIARC, each dataset in Table II is divided into several parts (see the 4th and 5th columns of Table II), where the 4th column refers to the number of samples in the original dataset and the 5th column means the number of parts of equal size in the remaining set of samples. The first m samples of each dataset in Table II are chosen as the original dataset, and the remaining samples are divided into n parts of equal size. The first part is viewed as the 1st incremental dataset to be added into the original dataset, resulting in an updated original dataset, or the current dataset; the second part is regarded as the 2nd incremental dataset to be added into the updated original dataset, resulting in another updated original dataset; and so on.

B. Comparison of ASS-IAR and rough-set-based-methods

In this section, we compare ASS-IAR with IAR and GIARC. We first point out the main differences among the three

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

TABLE II
DATASET DESCRIPTION

Data	Data type	Samples	Ori_samples	Parts	Feature	Class
Soybean	Symbolic	683	342	5	35	19
Yeast	Mixed	1484	742	5	8	10
Contraceptive Method Choice (Cmc)	Real	1473	737	5	9	3
Sick	Symbolic	2800	1400	5	28	2
Kr-vs-kp	Symbolic	3196	1598	5	36	2
Spambase (Spam)	Real	4601	2301	5	57	2
Magic	Real	78823	39412	5	10	2
Letter-recognition (Letter)	Symbolic	20000	5000	5	16	26
Poker-hand (Poker)	Real	1025010	25010	10	10	10

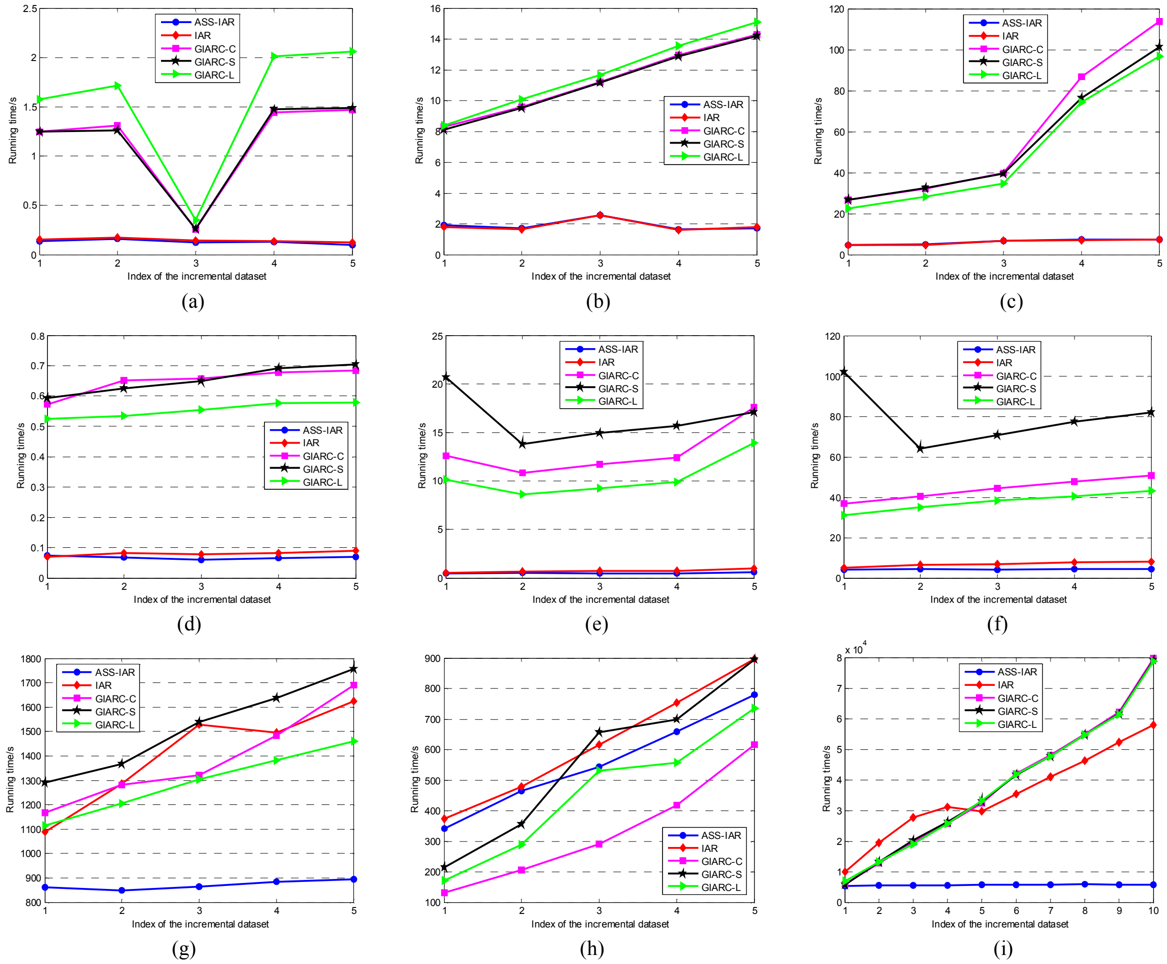

Fig. 4. Running time of ASS-IAR, IAR and GIARC with the arrival of an incremental dataset. (a) Cmc, (b) Soybean, (c) Kr-vs-kp, (d) Yeast, (e) Sick, (f) Spam, (g) Magic, (h) Letter, (i) Poker.

methods. One is that IAR and GIARC have no the scheme of active sample selection, whereas ASS-IAR has. The other is that GIARC has no an insight into the attribute reduction process that reveals which attributes should be added into and deleted from a current reduct, whereas IAR and ASS-IAR have.

The experimental results are summarized in Table III, Table IV and Fig. 4. Fig. 4 shows the runtime of ASS-IAR, IAR and GIARC with the arrival of each incremental dataset. In view of paper length, the results of the three methods on each dataset are shown in one figure. The *x*-coordinate is the index of each incremental dataset, while the *y*-coordinate is the runtime of each method. In each subfigure, GIARC-L, GIARC-S and GIARC-C refer to the GIARC algorithm based on Liang's entropy, Shannon's entropy and Combination entropy, respectively.

Fig. 4 displays that ASS-IAR is faster than GIARC on most datasets. Moreover, with the arrival of each incremental dataset, the runtime of GIARC basically grows monotonically. The reason is that GIARC adds and removes attributes by computing incremental entropies of all possible feature subsets at each loop. The incremental entropy is obtained by considering the combinative cases of equivalence classes and decision classes between the current dataset and the incremental dataset. With the increment of samples, it takes

TABLE III
TOTAL RUNNING TIME OF ASS-IAR, IAR AND GIARC

Data	ASS-IAR/s	IAR/s	GIARC-L/s	GIARC-S/s	GIARC-C/s
Soybean	10.6745	**10.5722**	57.5454	57.0801	59.9627
Yeast	**0.7014**	0.7666	3.6113	3.6314	3.1360
Cmc	**1.1083**	1.1896	6.1806	6.1913	8.1761
Sick	**6.5829**	7.614	69.2331	86.1752	55.7862
Kr-vs-kp	**36.9302**	36.5188	305.4245	282.4433	262.4014
Spam	**41.9085**	54.4849	240.8804	416.5303	242.2943
Magic	16537.83	19204.92	19129.06	19773.00	18650.0
Letter	3143.03	3472.26	2020.25	3178.15	2638.16
Poker	**57774.12**	352437.76	383827.56	384963.24	385089.78
Average	**8616.987**	41691.79	45073.31	45418.49	45223.34

TABLE IV
TOTAL NUMBER OF USELESS SAMPLES

Data	Remaining samples	Filtered samples	Ratio
Soybean	341	46	0.1349
Yeast	742	716	0.96
Cmc	736	617	0.8383
Sick	1400	1212	0.8657
Kr-vs-kp	1598	0	0
Spam	2300	2013	0.8752
Magic	39411	38644	0.9805
Letter	15000	1191	0.0794
Poker	1000000	811945	0.8119

much time to find the incremental entropy, which results in the increasing of the runtime of GIARC. In contrast, the runtime of ASS-IAR is basically stable, since it can discard useless samples to save the memory space and improve the time efficiency of updating reduct.

Table III shows the total runtime of the three methods. Here, the total runtime of each method is obtained by the sum of the runtime on the original dataset and each incremental dataset. It can be observed that the average runtime of ASS-IAR (8616.987 seconds) is much less than that of IAR (41691.79 seconds) and GIARC (GIARC-L: 45073.31 seconds; GIARC-S: 45418.49 seconds; GIARC-C: 45223.34 seconds). Moreover, ASS-IAR is much more efficient than IAR and GIARC on large-scale datasets. For example, on 'Poker', the runtime of ASS-IAR is 57774.12 seconds, which is much lower than that of IAR (352437.76 seconds) and GIARC (GIARC-L: 383827.56 seconds; GIARC-S: 384963.24 seconds; GIARC-C: 385089.78 seconds). The facts imply the time efficiency of ASS-IAR when finding a reduct.

Table IV shows the number of useless samples. Here, 'Remaining samples' represents the number of all incremental samples, 'Filtered samples' is the number of useless samples in all incremental samples and 'Ratio' denotes the ratio of Filtered samples and Remaining samples. It is clear from Table IV that the number of useless samples is very big, even over 80% in some cases. For example, on 'Magic', useless samples account for 98.05% of all incremental samples; on 'Poker', the percentage of all useless samples is 81.19%. These facts suggest useless samples do take up a huge amount of memory space. From Table III, we can also see that filtering out useless samples can indeed improve the time efficiency of updating one reduct. Therefore it is worthwhile to design a scheme of active sample selection to filter out useless samples while retaining useful samples to perform the incremental computation.

To sum up, our ASS-IAR can update a reduct in much less time by comparing with IAR and GIARC. The reason is twofold. The first one is ASS-IAR has an insight into the attribute reduction process that efficiently guides how to add and delete attributes, while GIARC does not have. The second one is that ASS-IAR has the scheme of active sample selection which actively selects useful samples to update the reduct and discards useless samples to save the space memory, while IAR and GIARC do not have. Therefore, it is highly efficient to employ our ASS-IAR to deal with dynamic datasets with successive samples.

C. *Comparison of ASS-IAR and non-rough-set-based feature selection methods*

In this section, three state-of-the-art non-rough-set-based methods, i.e., CONSISTENCY, RELIEFF and SCRAP, are compared with our ASS-IAR. Here, RELIEFF and SCRAP can be considered as feature selection methods including active sample selection mechanisms, since they obtain a feature subset by conducting search not only in the feature space but also in the sample space.

CONSISTENCY is a consistency-based feature selection method of searching the minimal subset that separates classes as consistently as the full set can under best first search strategy. It is similar to rough-set-based attribute reduction method. So, it is necessary to compare ASS-IAR with CONSISTENCY.

RELIEFF weighs each feature according to how well their values distinguish between the instances that are near to each other. Given a randomly selected sample, RELIEFF searches the whole dataset for its two nearest samples: one from the same class called nearest hit, and the other from a different class called nearest miss. It then updates the weight of each attribute depending on the two nearest samples. The process is repeated m times, where m is specified as the number of samples on the first eight datasets and m is set as 5000 on 'Poker'. RELIEFF depends on randomly selecting samples, which implies RELIEFF can be considered as a feature selection method including sample selection. Hence, it is compared with ASS-IAR.

SCRAP is a conservative filtering scheme that tries to identify the features changing at two consecutive Points of Class Change and include them in the feature subset. It searches

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2016.2581186, IEEE Transactions on Fuzzy Systems

TABLE V
RUNTIME OF ASS-IAR, CONSISTENCY, RELIEFF AND SCARP.

Data	ASS-IAR	CONSISTENCY		RELIEFF		SCARP	
	Time/s	Time/s	*p*-val	Time/s	*p*-val	Time/s	*p*-val
Soybean	14.5759	66.5532	0.00 -	14.5451	0.9738	11.7844	0.0071 +
Yeast	0.5510	0.3734	0.00 +	1.5760	0.00 -	1.0559	0.00 -
Cmc	0.9671	1.8448	0.00 -	1.5075	0.00 -	4.8602	0.00 -
Sick	3.0932	17.2767	0.00 -	5.8923	0.00 -	4.1962	0.00 -
Kr-vs-kp	61.6507	251.7640	0.00 -	10.4290	0.00 +	7.8638	0.00 +
Spam	10.5399	273.5061	0.00 -	36.0602	0.00 -	41.1449	0.00 -
Magic	2467.78	36789.58	0.00 -	12726.19	0.00 -	5893.14	0.0453 -
Letter	2839.69	3434.99	0.00 -	2638.11	0.0018 +	226.4985	0.00 +
Poker	55588.23	64191.18	0.0034 -	103576.01	0.00 -	162299.68	0.00 -
Average	**6776.342**	11669.67		13223.37		18721.14	
L/W/T		8/1/0		6/2/1		6/3/0	

for the next Point of Class Change after discarding the samples in the neighborhood of the current Point of Class Change, which implies SCRAP has a scheme of filtering out samples. Therefore, SCRAP is compared with ASS-IAR.

Each method in this section is run 10 times on each selected dataset. The average runtime of each method on each dataset is summarized in Table V. For each dataset, we conduct Student's paired two tailed *t*-Test in order to evaluate the statistical significance of the difference between two runtime: one resulted from ASS-IAR and the other resulted from one of CONSISTECY, RELIEFF and SCRAP. The *p*-value is recorded in Table V to show the probability associated with the *t*-Test. The last row (L/W/T) in Table V summarizes over all selected datasets the losses/wins/ties in the runtime of three feature selection methods over that of ASS-IAR.

From Table V, we can observe that ASS-IAR (6776.342 seconds) achieves the fastest average runtime comparing to the baseline algorithms (CONSISTENCY: 11669.67 seconds; RELIEFF: 13223.37 seconds; SCRAP: 18721.14 seconds). More specially, ASS-IAR is much more efficient than the baseline algorithms on the large-scale datasets. For example, on 'Poker', the runtime of ASS-IAR is 86.6% of that of CONSISTENCY, 53.67% of that of RELIEFF, and 34.25% of that of SCRAP, respectively. Moreover, the L/W/T records show ASS-IAR statistically outperforms CONSISTENCY 8 times, RELIEFF 6 times, and SCRAP 6 times respectively.

To be brief, in comparison with CONSISTENCY, RELIEFF and SCRAP, ASS-IAR also shows the time efficiency, especially for large-scale datasets. The reason is as follows. CONSISTENCY searches for a feature subset by scanning the whole sample space to obtain the inconsistency rate at each loop. Although the time complexity of RELIEFF is proportional to the iteration time *m*, it has to weigh each feature by searching for the whole sample space to obtain two nearest neighbors of a randomly selected sample. Hence, CONSISTENCY and RELIEFF are clearly time-consuming to select a feature subset from datasets with large-scale samples. Besides, SCRAP is very costly to obtain a feature subset, which is because although SCRAP discards the samples in the neighborhood of a current Point of Class Change, it has to reduce the weight for the irrelevant features by scanning the discarded neighborhood. In contrast, ASS-IAR can forever filter out useless samples to compress the sample space, so that the memory space is saved and the time efficiency of finding a feature subset is also improved.

Hence, our ASS-IAR can not only improve the time efficiency of the existing incremental attribute reduction algorithms, but also is highly effective to deal with large-scale datasets.

VI. CONCLUSIONS

In this paper, we present an active sample selection based incremental algorithm for attribute reduction, which actively filters out useless samples and retains useful samples to perform the incremental computation of attribute reduction. Our algorithm has the following two advantages. One advantage is that through active sample selection, our incremental algorithm actively responds only to useful samples rather than passively using all incremental samples in the incremental computation of attribute reduction, so that the memory space usage and computation time are both reduced which is particularly effective in dealing with dynamic, large datasets. The other advantage is that our incremental algorithm exploits deeper insights into the attribute reduction process so that it knows correctly which attributes should be added into or deleted from an existing reduct. Experimental comparisons show that our incremental algorithm is indeed very efficient, especially when dealing with large datasets. Our future work will concentrate on how to extend our idea in this paper to other rough set models such as fuzzy rough sets and covering rough sets.

REFERENCES

[1] W.C. Bang, Z. Bien, "Incremental inductive learning algorithm in the framework of rough set theory and its application," *Int. J. Fuzzy Syst.*, vol. 1, no. 1, pp. 25-36, 1999.

[2] D.G. Chen, S.Y. Zhao, L. Zhang, Y.P. Yang, X. Zhang, "Sample pair selection for attribute reduction with rough set," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2080-2093, 2012.

[3] D.G. Chen, Y.Y. Yang, "Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1325-1334, 2014.

[4] Y. Cheng, "The incremental method for fast computing the rough fuzzy approximations," *Data Knowl. Eng.*, vol. 70, no. 1, pp. 84-100, 2011.

[5] S.R. Feng, D.Z. Zhang, "Incremental algorithm for attribute reduction based on improvement of discernibility matrix," *J. Shenzhen University Sci. Eng.*, vol. 29, no. 5, 2012.

[6] L.H. Guan, "An incremental updating algorithm of attribute reduction set in decision tables," in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Dis.*, pp. 421-425, 2009.

[7] F. Hu, G.Y. Wang, H. Huang, Y. Wu, "Incremental attribute reduction based on elementary sets," in *Proc. 10th Int. Conf. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Regina, Canada*, pp. 185-193, 2005.

[8] F. Hu, J.H. Dai, G.Y. Wang, "Incremental algorithms for attribute reduction in decision table," *Control Decis.*, vol. 22, no. 3, pp. 268-272, 2007.

[9] Q.H. Hu, D.R. Yu, Z.X. Xie, X.D. Li, "EROS: ensemble rough subspaces," *Pattern Recognit.*, vol. 40, no. 12, pp. 3728-3739, 2007.

[10] Q.H. Hu, D.R. Yu, J.F. Liu, C.X. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Inf. Sci.*, vol. 178, no. 18, pp. 3577-3594, 2008.

[11] T.R. Li, D. Ruan, W. Geert, "A rough sets based characteristic relation approach for dynamic attribute generalization in data mining," *Knowl.-Based Syst.*, vol. 20, no. 5, pp. 485-494, 2007.

[12] J.Y. Liang, Z.B. Xu, "The algorithm on knowledge reduction in incomplete information systems," *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 1, pp. 95-103, 2002.

[13] J.Y. Liang, F. Wang, C.Y. Dang, Y.H. Qian, "A group incremental approach to feature selection applying rough set technique," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 294-308, 2014.

[14] Z.T. Liu, "An incremental arithmetic for the smallest reduction of attributes," *Acta Electronica Sinica*, vol. 27, no. 11, pp. 96-98, 1999.

[15] C. Luo, T.R. Li, H.M. Chen, D. Liu, "Incremental approaches for updating approximations in set-valued ordered information systems," *Knowl-Based Syst.*, vol. 50, pp. 218-233, 2013.

[16] UCI Machine Learning Repository. (2005). [Online]. Available: http://www.ics.uci.edu/~mlearn//MLRepository.html.

[17] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341-356, 1982.

[18] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data [M]*. Dordrecht, The Netherlands: Kluwer, 1991.

[19] Y.H. Qian, J.Y. Liang, "Combination entropy and combination granulation in rough set theory," *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. 2, pp. 179-193, 2008.

[20] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, "Positive approximation: an accelerator for attribute reduction in rough set theory," *Artificial Intelligence*, vol. 174, no. 9, pp. 597-618, 2010.

[21] N. Shan, W. Ziarko, "Data-based acquisition and incremental modification of classification rules," *Computat. Intell.*, vol. 11. no.2, pp. 357-370, 1995.

[22] W.H. Shu, H. Shen, "A rough-set based incremental approach for updating attribute reduction under dynamic incomplete decision systems," in *Proc. 2013 Int. Conf. Fuzzy Syst.*, pp. 1-7, 2013.

[23] W.H. Shu, H. Shen, "Updating attribute reduction in incomplete decision systems with the variation of attribute set," *Int. J. Approx. Reason.*, vol. 55, no. 3, pp. 867-884, 2014.

[24] W.H. Shu, H. Shen, "Incremental feature selection based on rough set in dynamic incomplete data," *Pattern Recognit*., vol. 47, no. 12, pp. 3890-3906, 2014.

[25] A. Skowron, C. Rauszer, "The discernibility matrices and functions in information systems," *Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory*, R. Slowinski, ed., pp. 331-362, Kluwer Academic Publishers, 1992.

[26] D. Slezak, "Approximate entropy reducts," *Fundam. Inf.*, vol. 53, nos. 3-4, pp. 365-390, 2002.

[27] R. Susmaga, "Experiments in incremental computation of reducts," *Rough Sets in Data Mining and Knowledge Discovery*, A. Skowron and L. Polkowski, eds., Springer-Verlag, 1998.

[28] F. Wang, J.Y. Liang, Y.H. Qian, "Attribute reduction: a dimension incremental strategy," *Knowl-Based Syst.*, vol. 39, pp. 95-108, 2013.

[29] F. Wang, J.Y. Liang, C.Y. Dang, "Attribute reduction for dynamic data sets," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 676-689, 2013.

[30] M. Yang, "An incremental updating algorithm for attributes reduction based on the improved discernibility matrix," *Chinese J. Computers*, vol. 30, no. 5, pp. 815-822, 2007.

[31] X.B. Yang, Y. Qi, H.L. Yu, X.N. Song, J.Y. Yang, "Updating multigranulation rough approximations with increasing of granular structures," *Knowl-Based Syst.*, vol. 64, pp. 59-69, 2014.

[32] Y.Y. Yao, Y. Zhao, "Discernibility matrix simplification for constructing attribute reducts," *Inf. Sci.*, vol. 179, no. 7, pp. 867-882, 2009.

[33] D.R. Yu, Q.H. Hu, W. Bao, "Combining rough set methodology and fuzzy clustering for knowledge discovery from quantitative data," in *Proc. Chin. Soc. Electr. Eng.*, vol. 24, no. 6, pp. 205-210, 2004.

[34] S.Y. Zhao, Eric C.C. Tsang, D.G. Chen, X. Z. Wang, "Building a rule-based classifier—a fuzzy-rough set approach," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 624-638, 2010.

[35] A.P. Zeng, T.R. Li, D. Liu, J.B. Zhang, H.M. Chen, "A fuzzy rough set approach for incremental feature selection on hybrid information systems," *Fuzzy Sets Syst.*, vol. 258, pp. 39-60, 2015.

[36] D. Ślęzak, A. Janusz, "Ensembles of bireducts: towards robust classification and simple representation," *Future generation information technology*, vol. 7105, pp. 64-77, 2011.

[37] S. Stawicki, S. Widz, "Decision bireducts and approximate decision reducts: Comparison of two approaches to attribute subset ensemble construction," in *Proc. Fed. Conf. Computer Science and Information Systems (FedCSIS)*, pp. 331-338, 2012.

[38] S. Stawicki, D. Ślęzak, "Recent advances in decision bireducts: complexity, heuristics and streams," *Rough Sets and Knowledge Technology*, pp. 200-212, 2013.

[39] S.Y. Zhao, H. Chen, C.P. Li, X.Y. Du, H. Sun, "A novel approach to building a robust fuzzy rough classifier," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 769-786, 2015.

[40] L.S. Riza, A. Janusz, C. Bergrmeir, C. Cornelis, F. Herrera, D. Slezak, J.M. Benitez, "Implementing algorithms of rough set theory and fuzzy rough set theory in the R package "RoughSets"," *Inf. Sci.*, vol. 287, pp. 68-89, 2014.

[41] G.Y. Wang, H. Yu, D.C. Yang, "Decision table reduction based on conditional information entropy," *Chinese J. Computer*, vol. 25, no. 7, pp. 759-766, 2002.

[42] A. Janusz, D. Slezak, "Rough set methods for attribute clustering and selection," *Appl. Artif. Intell.*, vol. 28, no. 3, pp. 220-242, 2014.

[43] H.M. Chen, T.R. Li, C. Luo, S.J. Horng, G.Y. Wang, "A rough set-based method for updating decision rules on attribute values' coarsening and refining," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2886-2899, 2014.

[44] H. Liu, H. Motoda, L. Yu, "A selective sampling approach to active feature selection," *Artificial Intelligence*, vol. 159, no. 1, pp. 49-74, 2004.

[45] H. Liu, H. Motoda, L. Yu, "Feature selection with selective sampling," in *Proc. 19th Int. Conf. Mach. Learning*, 2002, pp. 395-402.

[46] K. Kira, L.A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Conf. Mach. Learning*, 1992, pp. 249-256.

[47] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," *Machine Learning: ECML-94*, L. De Raedt and F. Bergadano, eds., pp. 171-182, Springer Verlag, 1994.

[48] B. Raman, T.R. Ioerger, "Instance based filter for feature selection," *J. Machine Learning Research*, vol. 1, no. 3, pp. 1-23, 2002.

[49] M. Dash, H. Liu, "Consistency-based search in feature selection," *Artificial Intelligence*, vol. 151, nos. 1-2, pp. 155-176, 2003.

[50] S.Y. Zhao, E.C.C., Tsang, D.G. Chen, "The model of fuzzy variable precision rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 2, pp. 451-467, 2009.

[51] S.Y. Zhao, H. Chen, C.P. Li, M.Y. Zhai, X.Y. Du, "RFRR: robust fuzzy rough reduction," *IEEE Trans. Fuzzy Syst.*, vol. 21, no.5, pp.825-841, 2013.

[52] Q.H. Hu, L. Zhang, D.G. Chen, W. Pedrycz, D.R. Yu, "Gaussian kernel based fuzzy rough sets: model, uncertainty measures and applications," *Int. J. Approx. Reason.*, vol. 51, no. 4, pp. 453-471, 2010.

[53] Q.H. Hu, D.R. Yu, W. Pedrycz, D.G. Chen, "Kernelized fuzzy rough sets and their applications," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 11, pp.1649-1667, 2011.

[54] M. Orlowska, M. Orlowski, "Maintenance of knowledge in dynamic information systems," *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory*, R. Slowinski, ed., pp. 315-330, Kluwer Academic Publishers, 1992.

[55] A. Skowron, "Boolean reasoning for decision rules generation," in *Proc. Int. Symposium on Methodologies for Intelligent Systems*, pp. 295-305, 1993.

[56] D.Q. Miao, Y, Zhao, Y.Y. Yao, H.X. Li, F.F., Xu, "Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model," *Inf. Sci.*, vol. 179, no.24, pp. 4140-4150, 2009.

[57] X. Z. Wang, H. J. Xing, Y. Li, Q. Hua, C. R. Dong, W. Pedrycz, "A Study on Relationship between Generalization Abilities and Fuzziness of Base

Classifiers in Ensemble Learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1638-1654, 2015.

[58] M. Aggarwal, "Probabilistic variable precision fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 29-39, 2016.

[59] J. T. Yao, N. Azam, "Web-Based Medical Decision Support Systems for Three-Way Medical Decision Making With Game-Theoretic Rough Sets, " *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 3-15, 2015.

[60] D. C. Liang, D. Liu, "A Novel Risk Decision Making Based on Decision-Theoretic Rough Sets Under Hesitant Fuzzy Information," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 2, pp. 237-247. 2015.

[61] H. M. Chen, T. R. Li, C. Luo, S. J. Horng, G. Y. Wang, "A Decision-Theoretic Rough Set Approach for Dynamic Data Mining," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1958-1970, 2015.

[62] Y. Han, P. Shi, S. Chen, "Bipolar-Valued Rough Fuzzy Set and Its Applications to the Decision Information System," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2358-2370, 2015.

Yanyan Yang received the BSc degree from the School of Science, North University of China, Taiyuan, China, in 2010 and the MSc degree from the Department of Mathematics and Physics, North China Electric Power University, Beijing, China, in 2013. She is currently working toward the Ph.D. degree with the School of Control and Computer Engineering, North China Electric Power University. Since October, 2015, she has visited University of Ulster as a Joint Ph.D. student sponsored by China Scholarship Council for one year.

Her current research interests include rough sets, fuzzy sets and machine learning.

Degang Chen received the MSc degree from Northeast North University, Changchun, Jilin, China, in 1994 and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2000.

He was a Postdoctoral Fellow with Xi'an Jiaotong University, Xi'an, China, from 2000 to 2002, and with Tsinghua University, Beijing, China, from 2002 to 2004. Since 2006, he has been a Professor with North China Electric Power University, Beijing, China. He has authored or coauthored more than 140 research publications. His research interests include fuzzy groups, fuzzy algebra, fuzzy analysis, rough sets, machine learning.

Hui Wang received BSc degree in computer science from Jilin University of China in 1985 and then went on a three-year MSc program in artificial intelligence. From 1988 to 1992, he worked at Jilin University as a lecturer. In early 1993, he went to Northern Ireland to begin a doctoral program and completed the DPhil degree in informatics in 1996.

He is currently a Professor in Computer Science and the head of Artificial Intelligence and Applications Research Group, University of Ulster. His research interests are machine learning, logics and reasoning, combinatorial data analytics, and their applications in image, video, spectra and text analysis. He has over 200 publications in these areas.