Int. J. Mach. Learn. & Cyber. (2017) 8:849–864
DOI 10.1007/s13042-015-0473-z

ORIGINAL ARTICLE

Incremental knowledge discovering in interval-valued decision information system with the dynamic data

Jianhang Yu[1] · **Weihua Xu**[1,2,3]

Received: 12 February 2015 / Accepted: 6 December 2015 / Published online: 19 December 2015

Abstract With the development of information technology and the rapid updating of data sets, the object sets in an information system may evolve in time when new information arrives and redundancy information leaves in real life. Interval-valued decision information systems are important type of data decision tables and generalized models of single-valued information systems. Fast updating the lower and upper approximations is the core technology of knowledge discovery that is based rough set theory in dynamic data environment. Consequently, in this paper we focus on incremental approaches updating approximations with dynamic data sets in interval-valued decision system. Firstly, we define an interval similarity degree by which a binary relation can be constructed, followed a rough set model be established. Then, incremental approaches for updating approximations are proposed and the incremental algorithms are shown. At last, comparative experiments on several UCI data sets show the proposed incremental updating methods are efficient and effective for dynamic data sets, namely, these approaches significantly outperform the classical methods with a dramatic reduction in the computational time.

Keywords Dynamic data sets · Incremental knowledge discovering · Interval similarity degree · Interval-valued decision information system

✉ Weihua Xu
chxuwh@gmail.com

Jianhang Yu
yujh2013@foxmail.com

1 School of Mathematics and Statistics, Chongqing University of Technology, 400054 Chongqing, People's Republic of China

2 Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, Nanjing University of Science and Technology, 210094 Nanjing, People's Republic of China

3 Key Laboratory of Oceanographic Big Data Mining and Application of Zhejiang Province, Zhejiang Ocean University, 316022 Zhoushan, Zhejiang, People's Republic of China

1 Introduction

Rough set theory (RST), which was first proposed by Pawlak in 1980s [23–25], is one of the effective mathematical tool for discovering knowledge in the information systems and decision systems. The theory has been demonstrated to be useful in the fields of data mining, pattern recognition, conflict analysis, decision support and so on.

However, the data collected from practical problem are usually real numbers and generally include errors due to the human cognitive uncertainty and interference of some random factors in real-life application. Therefor, an information system with its attribute values being interval-valued is perhaps more appropriate for describing such data because one of the useful ways for characterizing the values of a variable with uncertainty is to use the interval-valued specified by the properly defined lower and upper limits of the values that this variable possibly takes. In recent years, some studies have been investigated in the context in interval-valued information system. As a counterpart of the interval-number algebra, Yao introduced an interval-set algebra for representing qualitative information [35, 37]. Yao and Li compared rough set and interval set models and showed that these two models provide different and complementary extensions of the set theory [36].

Zhang et al. presented a monotonic inclusion measure approach to rank two intervals [39]. Li et al. proposed a novel method for extracting rules from an incomplete decision system by using an interval set model [9]. Leung et al. presented a rough set method for the discovery of the α-classification rules in interval-valued decision systems [8]. Miao et al. developed a new framework of knowledge reduction in interval-valued information systems based on the maximal consistent blocks [21]. Dai et al. studied the uncertainty measurement problem in interval-valued decision systems based on an extended conditional entropy and the notion of the possible degree between intervals [6]. Qian et al. introduced a dominance relation to interval-valued information systems, provided an object-ranking method using the whole dominance degree of each object, and proposed an attribute reduction procedure to extract compact dominance rules [26]. Yang et al. developed a data complement method to transform an incomplete interval-valued information system with a dominance relation into a complete one and proposed six types of relative reduces according to the different requirements of simplifying the dominance rules supported by an object [33, 34]. Gong et al. proposed the rough set theory for the interval-valued fuzzy information systems [7]. Sun et al. presented a rough approximation of an interval-valued fuzzy set and investigated the issue of attribute reduction in the interval-valued fuzzy information system [28]. Decision information system is a kind of special information system which have decision attribute. Li et al. studied the knowledge reduction in decision formal contexts [18] and real decision formal contexts [19], respectively. Yang et al. discussed the rule acquisition and attribute reduction in real decision formal contexts [38]. Make decision is the core part of management activity and knowledge discovery. Hence, we will try doing some discussion about knowledge discovering in interval-valued decision information system (IvDIS), in this talk.

With the development of the information society and the science and technology continue to improve, the number of data in the databases has increased not only in quantities but also in scales. The technology of data mining emerges as the times require and has been successfully applied on many areas. The incremental learning technique is an important approach to solve the issue of dynamic data changing in many fields and this approach have already received much attention in recent years [1]. Many researchers have proposed incremental algorithms for knowledge discovery by RST when the information system varies with time. Luo et al. point out the present study focused on the following three cases [17]: The first is the object set varies with time in the information systems. Shan and Ziarko presented an RST-based incremental methodology for finding all maximally generalized rules and adaptive modification of these rules when new data become available [27]. Liu et al. defined a new concept of interesting knowledge based on both accuracy and coverage, and proposed an incremental model for inducing interesting knowledge when the object set varies over time [12, 13]. Li investigate dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set [16]. Zhang et al. proposed a method of incrementally updating approximations based on neighborhood rough sets in dynamic data mining [40]. Luo discussed incremental approaches for updating approximations in set-valued ordered information systems [17]. Liu studied a rough set-based incremental approach for learning knowledge in dynamic incomplete information systems [14]. The second is the attribute set varies with time in the information systems. Chan presented an incremental method for updating approximations by using the concept of lower and upper boundary sets based on traditional rough sets [2]. Li et al. discussed an attribute generalization and its relation to feature selection and feature extraction, and then proposed an incremental approach for updating approximations under the characteristic relation based rough sets [10, 11]. Liu studied incremental updating approximations in probabilistic rough sets under the variation of attributes [15]. The third is the attribute values varies with time in the information systems. Chen et al. defined the attribute values coarsening and refining in information systems as the semantic level changes, and then proposed an incremental algorithm for updating the approximations of a concept when coarsening or refining attribute values [3]. Uncertainty processing plays a key role in incremental knowledge discovery [31, 32]. It is found that, in a knowledge discovering system, the modeling of fuzziness and roughness can significantly improves the system' performance [20, 29, 30]. Furthermore, in incomplete ordered decision systems, Chen et al. also presented a method to dynamically maintain approximations of upward and downward unions when attribute values changes [4, 5]. All these studies help decision makers to update knowledge with different viewpoints from different kinds of information systems.

The motivation of this paper is that try to propose an incremental approaches for knowledge discovering in dynamic interval-valued decision information system, namely, incremental updating the upper and lower approximations with dynamic data sets. In order to guarantee the proposed incremental updating methods are efficient and effective for dynamic data sets. In this paper, we define an interval similarity degree based on intervals' intersection and union operations and according the interval similarity degree construct a binary relation in interval-valued decision information system at first. Followed, we construct the lower and upper approximations based the relation which we have proposed and some properties are investigated. Then, the theories of the incremental knowledge discovering with the dynamic object set in interval-valued decision information system and two incremental algorithms are proposed when the objects are deleted or inserted, respectively. At last, the performances of two incremental algorithms are evaluated on serval variety of UCI data sets.

The remainder of this paper is organized as follows. In Sect. 2, some basic concepts of RST and interval-valued information systems are simply introduced. In Sect. 3, we define a interval similarity degree between two intervals and a new binary relation be generated based it, followed, we construct the rough set model and some properties are shown. The principles for incremental updating approximations with the variation of object set are presented, in Sect. 4. We designed the incremental algorithms for computing approximations based on the previous updating principles, in Sect. 5. In Sect. 6, performance evaluations are illustrated and the experiment results have exhibited. The paper ends with conclusions shown in Sect. 7.

2 Preliminaries

In this section, the basic concepts of RST and some important properties of interval-valued information system are reviewed [8, 22, 26, 41, 42].

Given a quadruple $I=(U,C\cup\{d\},V,f)$ be an information system. U is a finite non-empty set of objects, called the universe. C is a non-empty finite set of condition attributes and d is a decision attribute, respectively. And $C\cap\{d\}=\emptyset$. With every attribute $a\in C$, a set of its values V_A is associated. $f: U\times C\rightarrow V$ is a total function such that $f(x,a)\subseteq V_a$ for every $a\in C, x\in U$. The class of all subsets of U is denoted by $P(U)$. For $X\in P(U)$, the equivalence relation R in a Pawlak approximation space (U, R) partitions the universe U into disjoint subsets. Such a partition of the universe is a quotient set of U and is denoted by

$$U/R=\{[x]_R : x\in U\}.$$

where $[x]_R=\{y\in U:(x,y)\in R\}$ is the equivalence class containing x with respect to R. In the view of granular computing, equivalence classes are the basic building blocks for the representation and approximation of any subset of the universe of discourse. Each equivalence class may be viewed as a granule consisting of indistinguishable elements.

Definition 2.1 (see [23, 24]) Let $I=(U,C\cup\{d\},V,f)$ be an information system, and $X\in P(U)$ is an basic concept, one can characterize X by a pair of upper and lower approximations which are

$$\underline{R}(X)=\{x\in U:[x]_R\subseteq X\},$$
$$\overline{R}(X)=\{x\in U:[x]_R\cap X\neq\emptyset\}.$$

Here, $pos(X)=\underline{R}(X), neg(X)=\sim\overline{R}(X), bn(X)=\overline{R}(X)-\underline{R}(X)$ are called the positive region, negative region and boundary region of X, respectively.

Definition 2.2 (see [22]) Let $I_1=[u_1,v_1]$ and $I_2=[u_2,v_2]$ are two intervals. The intersection and union operation of I_1 and I_2 are defined as follows

$$I_1\cap I_2=[max\{u_1,u_2\},min\{v_1,v_2\}],$$
$$I_1\cup I_2=[min\{u_1,u_2\},max\{v_1,v_2\}].$$

Definition 2.3 (see [22]) The length of a nonempty interval $I=[u,v]$ can be defined as $\rho(I)$,

$$\rho(I)=v-u$$

If $u=v$ the interval just be a single point and the length of I is $\rho(I)=0$. Especially, if $I=[,]$ is an empty interval, the length of I is ruled as $\rho(I)=0$. It's obvious that the for any two intervals I_1 and I_2, if $I_1\cap I_2$ is nonempty, then $\rho(I_1\cap I_2)\geq 0$; otherwise, $\rho(I_1\cap I_2)=0$.

Definition 2.4 (see [42]) Let $I_1=[u_1,v_1]$ and $I_2=[u_2,v_2]$ are two intervals. The interval-inclusion degree of I_1 in I_2 is defined by

$$\delta_{12}=\frac{\rho(I_1\cap I_2)}{\rho(I_1)}$$

The interval-inclusion of I_1 include in I_2 is δ_{12} and denote this inclusion relation by $I_1\subseteq_{\delta_{12}} I_2$. It's obvious that $0\leq\delta_{12}\leq 1$ for any two nonempty intervals I_1 and I_2, but $\delta_{12}\neq\delta_{21}$ in general. They constructed a new binary relation based on this interval-inclusion degree by following method. Let (U, C, V, f) is an interval-valued information system with $U=\{x_1,x_2,\ldots,x_n\}$, $B\subseteq C$ and given an interval-inclusion threshold $\delta\in(0,1]$ then, the new binary relation with respect to the attribute subset B is $R_B^\delta=\{(x_i,x_j)\in U\times U: B(x_j\subseteq_{\delta_{ji}^B} B(x_i),\delta_{ji}^B\geq\delta)$. The δ-interval neighborhood of the object x_i with respect to B is denoted by $[x_i]_B^\delta=\{x_j\in U:(x_i,x_j)\in R_B^\delta\}$. It is a cluster in which, each object has the common knowledge to x_i at the δ-interval inclusion level with respect to B. The relation R_B^δ which based on the δ is reflexive but is neither symmetric nor transitive. Consequently, in our following study we define a interval similarity degree and construct a new binary relation in a similar way.

Definition 2.5 (see [8]) An interval-valued information system is a quadruple as $I=(U,C,V,f)$, where U is a finite non-empty set of objects and C is a finite non-empty set of attributes, $V=\cup_{a\in C}V_a$ and V_a is a domain of attribute a, $f:U\times C\rightarrow V$ is a total function such that

$f(x,a) \in V_a$ for every $a \in C$,$x \in U$, called an information function, where V_a is a set of interval numbers. Denoted by

$$f(x,a) = [a^L(x), a^U(x)] = \{p : a^L(x) \leq p \leq a^U(x), \times a^L(x),^U(x) \in \mathbf{R}\},$$

we call it as interval number where x under the attribute a, and $\mathbf{R}$ is the set of real number. Furthermore, if $I = (U, C \cup \{d\}, V, f)$ and d is a decision attribute where $C \cap \{d\} = \emptyset$, then I is an interval-valued decision information system. In particular, $f(x, a)$ would degenerate into a real number if $a^L(x) = a^U(x)$. Under this consideration, we regard a single-valued information system as a special form of interval information system.

3 Rough sets in interval-valued information system

In this section, we define a similarity degree between two intervals and a new binary relation be generated based it in interval-valued decision information system. Based on the new binary relation we construct the rough set model and some properties of it are discussed.

Definition 3.1 Let I_1 and I_2 are two intervals, the interval similarity degree of I_1 and I_2 is defined as follows

$$\delta_{12} = \frac{\rho(I_1 \cap I_2)}{\rho(I_1 \cup I_2)}$$

where $\rho(I)$ is the length of interval I, $\cap$ and $\cup$ are intersection and union operation, respectively.

It's obvious that $0 \leq \delta_{12} \leq 1$ for any two nonempty intervals I_1 and I_2, and $\delta_{12} = \delta_{21}$. One can know from this definition that if $I_1 \subseteq I_2$, namely, I_1 is completely included in I_2. That is for any real number $u_1 \leq r \leq v_1$, we have $u_2 \leq r \leq v_2$, in this case that $\delta_{12} = \frac{\rho(I_1)}{\rho(I_2)} = \delta_{21}$. Especially, if the I_1 or I_2 is a real number we ruled $\delta_{12} = 0$. If $I_1 = I_2$, then we can get that $\delta_{12} = \delta_{21} = 1$, that is the highest similarity degree between two intervals, and it means δ is reflexive.

Definition 3.2 Let $I = (U, C \cup \{d\}, V, f)$ be an interval-valued decision information system with $U = \{x_1, x_2, \ldots, x_n\}$ and for any $A \subseteq C$. We say $A(x_i) \subseteq_{\delta_{ij}^A} A(x_j)$ if $a(x_i) \subseteq_{\delta_{ij}^{\{a\}}} a(x_j)$ for any $a \in A$, where $\delta_{ij}^A = \min_{a \in A} \{\delta_{ij}^{\{a\}} : \delta_{ij}^{\{a\}} = \rho(a(x_i) \cap a(x_j)) / \rho(a(x_i) \cup a(x_j))\}$. Given an interval similarity threshold $\delta \in (0, 1]$ then, we can define a new binary relation with respect to the attribute subset A as follows

$$R_A^\delta = \{(x_i, x_j) \in U \times U : A(x_j) \subseteq_{\delta_{ij}^A} A(x_i), \delta_{ij}^A \geq \delta\},$$

and based the relation the δ-interval neighborhood of the object x_i with respect to A are defined by

$$[x_i]_A^\delta = \{x_j \in U : (x_i, x_j) \in R_A^\delta\}.$$

The δ-interval neighborhood $[x_i]_A^\delta$ is an object set in which each object x_j has the relation R_A^δ with x_i, namely, $(x_i, x_i) \in R_A^\delta$. It can be deemed as a cluster in which with respect to attribute set A, each object has the common knowledge to x_i at the δ-interval similarity level. Given a permissible threshold δ of the interval similarity degree, the knowledge implied by each object x_j in $[x_i]_A^\delta$ can be used to characterize the knowledge of the object x_i with respect to A when the interval similarity degree is δ. Therefore, the δ-interval neighborhood of each object with respect to some attribute subset can be considered as a basic knowledge granule in which the parameter δ is allowed to be adjustable according to the different perspectives of knowledge acquisition in the interval-valued decision information system. Furthermore, the larger the value of δ is, the smaller the granule $[x_i]_A^\delta$ is and the finer the knowledge represented by $[x_i]_A^\delta$ is. Suppose an interval-valued information system $I = (U, C \cup \{d\}, V, f)$, let $A \subseteq C, \delta \in (0.1]$, one can get that the constructed binary relation R_A^δ is reflexive and symmetric but not transitive. It can be obtained straight from the Definitions 3.1 and 3.2.

Definition 3.3 Let $I = (U, C \cup \{d\}, V, f)$ be an interval-valued information system with $U = \{x_1, x_2, \ldots, x_n\}$ and for any $A \subseteq C \text{and} X \subseteq U$. The lower and upper approximations of X based on the relation R_A^δ are respectively defined as

$$\underline{R_A^\delta}(X) = \{x \in U : [x]_A^\delta \subseteq X\},$$
$$\overline{R_A^\delta}(X) = \{x \in U : [x]_A^\delta \cap X \neq \emptyset\}.$$

Similar to the lower and upper approximations operators in the classical rough set theory, the following properties hold for $\underline{R_A^\delta}(X)$ and $\overline{R_A^\delta}(X)$.

(1) $\underline{R_A^\delta}(\emptyset) = \overline{R_A^\delta}(\emptyset) = \emptyset, \underline{R_A^\delta}(U) = \overline{R_A^\delta}(U) = U;$

(2) $\underline{R_A^\delta}(X) \subseteq X \subseteq \overline{R_A^\delta}(X);$

(3) $\underline{R_A^\delta}(\underline{R_A^\delta}(X)) = \underline{R_A^\delta}(X), \overline{R_A^\delta}(\overline{R_A^\delta}(X)) = \overline{R_A^\delta}(X);$

(4) $\underline{R_A^\delta}(X) = \sim \overline{R_A^\delta}(\sim X), \overline{R_A^\delta}(X) = \sim \underline{R_A^\delta}(\sim X);$

(5) $\underline{R_A^\delta}(X) \subseteq \underline{R_C^\delta}(X), \overline{R_A^\delta}(X) \supseteq \overline{R_C^\delta}(X), \text{and} Bn_C(X) \subseteq Bn_A(X);$

(6) If $X \subseteq Y$, then $\underline{R_A^\delta}(X) \subseteq \underline{R_A^\delta}(Y), \overline{R_A^\delta}(X) \subseteq \overline{R_A^\delta}(Y)$;

(7) $\underline{R_A^\delta}(X \cap Y) = \underline{R_A^\delta}(X) \cap \underline{R_A^\delta}(Y), \overline{R_A^\delta}(X \cup Y) = \overline{R_A^\delta}(X) \cup \overline{R_A^\delta}(Y)$;

(8) $\underline{R_A^\delta}(X \cup Y) \supseteq \underline{R_A^\delta}(X) \cup \underline{R_A^\delta}(Y), \overline{R_A^\delta}(X \cap Y) \subseteq \overline{R_A^\delta}(X) \cap \overline{R_A^\delta}(Y)$.

Example 3.1 An interval-valued decision information system be presented in Table 1. It is a case about the diagnosis of myocardial infarction, where $U = \{x_1, x_2, \ldots, x_{10}\}$ representatives of ten different patients and $C = \{a_1, a_2, \ldots, a_5\}$ representatives of several enzymes related to the diagnosis of myocardial infarction. Where a_1 represents Aspartate amino transferase (AST), a_2 represents Lactate dehydrogenase (LDH) and isoenzyme, a_3 represents Alfa hydroxybutyrate dehydrogenase (α−HBDH) , a_4 represents Creatine Kinase (CK), a_5 represents Creatine Kinase isoenzymes (CKMB). And the different decision attribute values mean different diagnosis results.

Let $A = C$, we can obtain a δ_{ij}^A-matrix according to Definition 3.2 as

$$M_A^\delta = \begin{pmatrix} 1 & 0.65 & 0.8 & 0 & 0.83 & 0.67 & 0.54 & 0.68 & 0.23 & 0.53 \\ 0.65 & 1 & 0.51 & 0 & 0.56 & 0.67 & 0.34 & 0.50 & 0.20 & 0.75 \\ 0.80 & 0.51 & 1 & 0 & 0.82 & 0.51 & 0.60 & 0.64 & 0.22 & 0.48 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0.15 \\ 0.83 & 0.56 & 0.82 & 0 & 1 & 0.56 & 0.66 & 0.75 & 0.27 & 0.46 \\ 0.67 & 0.67 & 0.51 & 0 & 0.56 & 1 & 0.34 & 0.47 & 0.20 & 0.65 \\ 0.54 & 0.34 & 0.60 & 0 & 0.66 & 0.34 & 1 & 0.70 & 0.36 & 0.37 \\ 0.68 & 0.5 & 0.64 & 0 & 0.75 & 0.47 & 0.70 & 1 & 0.33 & 0.38 \\ 0.23 & 0.2 & 0.22 & 0 & 0.27 & 0.20 & 0.36 & 0.33 & 1 & 0.17 \\ 0.53 & 0.75 & 0.48 & 0.15 & 0.46 & 0.65 & 0.37 & 0.38 & 0.17 & 1 \end{pmatrix}$$

Consequently, the δ-interval neighborhoods of each object x_i with $\delta = 0.6$ are

$[x_1]_A^{0.6} = \{x_1, x_2, x_3, x_5, x_6, x_8\}; [x_2]_A^{0.6} = \{x_1, x_2, x_6, x_{10}\}$;
$[x_3]_A^{0.6} = \{x_1, x_3, x_5, x_7, x_8\}$;
$[x_4]_A^{0.6} = \{x_4\}; [x_5]_A^{0.6} = \{x_1, x_3, x_5, x_7, x_8\}$;
$[x_6]_A^{0.6} = \{x_1, x_2, x_{10}\}; [x_7]_A^{0.6} = \{x_3, x_5, x_7, x_8\}$;
$[x_8]_A^{0.6} = \{x_1, x_3, x_5, x_7, x_8\}; [x_9]_A^{0.6} = \{x_9\}; [x_{10}]_A^{0.6} = \{x_2, x_{10}\}$.

We can find that the binary relation R_A^δ do not constitute a partition of U in general, but constitute a covering of U. Let a set $X = \{x_1, x_2, x_3, x_4, x_5\}$, based on above results and according the Definition 3.3. we can compute the lower and upper approximations of X as follows. $\underline{R_A^{0.6}}(X) = \{x_4\}$, and $\overline{R_A^{0.6}}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{10}\}$. Hence, the positive region are $pos(X) = \{x_4\}$, negative region are $neg(X) = \{x_9\}$, and boundary region are $bn(X) = \{x_1, x_2, x_3, x_5, x_6, x_7, x_8, x_{10}\}$, respectively.

4 Approaches for incremental knowledge discovering with dynamic data in IvDIS

With the rapid growth of data sets nowadays, the object sets in an information system may evolve in time when new information arrives and redundancy information be deleted. Incremental learning is an efficient technique for knowledge discovery in a dynamic database, which enables acquiring additional knowledge from new data without forgetting prior knowledge. The technology of data mining emerges as the times require and is successfully applied on many domains. In the field of data mining, the incremental learning technique is an important way to solve the problem of dynamic data changing. In this section, we investigate the variation of approximations of the dynamic interval-valued information system when the object set evolves over time while the attribute set remains constant. We assume the process for incremental update the knowledge lasts two stages, namely, from time t to time $t+1$. By considering the objects may enter into or get out of an information system at time $t+1$ and we denote a dynamic interval-valued decision information system at

Table 1 An interval-valued information system

U	AST	LDH	$\alpha-HBDH$	CK	$CKMB$	d
x_1	[10,40]	[100,240]	[105,195]	[5 ,195]	[0 ,24]	2
x_2	[10,30]	[80 ,210]	[80 ,180]	[10,190]	[0 ,24]	1
x_3	[12,45]	[105,248]	[100,210]	[7 ,203]	[0 ,23]	2
x_4	[5 ,30]	[60 ,80]	[90 ,160]	[0 ,180]	[0 ,10]	1
x_5	[10,46]	[110,246]	[105,195]	[6 ,198]	[0 ,26]	2
x_6	[10,30]	[90 ,200]	[96 ,206]	[5 ,195]	[3 ,24]	2
x_7	[13,60]	[100,240]	[115,200]	[20,260]	[5 ,30]	3
x_8	[10,50]	[120,260]	[115,210]	[8 , 196]	[5 ,28]	2
x_9	[16,80]	[140,260]	[102,300]	[40, 320]	[10,60]	3
x_{10}	[8 ,32]	[60 ,196]	[80 ,178]	[6 , 160]	[2 ,20]	1

time t as $I=(U,C\cup\{d\},V,f)$, and at time $t+1$ the original information system change into $I^{'}=(U^{'},C\cup\{d\},V,f)$ after insertion or deletion of objects.

Here, we only investigate the incremental approach for knowledge discovering in the cases that a single object enter and get out of the interval-valued information system. The change of multiple objects can be seen as the cumulative change of a single object can be updated step by step through the updating principles repeat a single object varies.

4.1 Approaches for incremental knowledge discovering with the deletion of an object in an IvDIS

Given an interval-valued decision information system $I=(U,C\cup\{d\},V,f)$ at time t, After deleting the object $x^-\in U$ the original information system and information granules $[x]_A^\delta(x\in U\text{and}A\subseteq C)$ and the equivalence decision classes $D_i(i\in\{1,\ldots,r\}$, r is the number of decision classes) will be changed. Then the approximations of D_i should be changed, too. Here, we investigate the approaches for updating approximations of D_i as two cases: $x^-\in D_i$ or $x^-\notin D_i$.

Case 1. The deleted object x^- belongs to D_i, namely, $x^-\in D_i$.

Proposition 4.1 *Let $I=(U,C\cup\{d\},V,f)$ be an IvDIS and any $A\subseteq C$. When $x^-\in D_i(i\in\{1,\ldots,r\})$ be deleted from U, the following properties hold for $\underline{R_A^\delta}(D_i)^{'}$ and $\overline{R_A^\delta}(D_i)^{'}$.*

(1) If $x^-\in\underline{R_A^\delta}(D_i)$, then $\underline{R_A^\delta}(D_i)^{'}=\underline{R_A^\delta}(D_i)-\{x^-\}$; Otherwise $\underline{R_A^\delta}(D_i)^{'}=\underline{R_A^\delta}(D_i)$.

(2) $\overline{R_A^\delta}(D_i)^{'}=(\overline{R_A^\delta}(D_i)-[x^-]_A^\delta)\cup\Delta_1^-$, *where* $\Delta_1^-=\{x:x\in[x^-]_A^\delta\cap\Delta_2^-\}$ *and* $\Delta_2^-=\cup_{x\in D_i-\{x^-\}}[x]_A^\delta$.

Proof (1) Let $x^-\in D_i$ be deleted from the universe U, then, the U be changed into $U^{'}=U-\{x^-\}$ and D_i be changed into $D_i^{'}=D_i-\{x^-\}$. So for any $x\in U^{'}$, we have $([x]_A^\delta)^{'}=[x]_A^\delta-\{x^-\}$. Because $x^-\in D_i$, so if $[x]_A^\delta\subseteq D_i$, then $([x]_A^\delta)^{'}\subseteq D_i^{'}$. Analogously, if $[x]_A^\delta\nsubseteq D_i$, then $([x]_A^\delta)^{'}\nsubseteq D_i^{'}$. Then, according the Definition 3.3, it's obvious for any $x\in U^{'}$, if $x\in\underline{R_A^\delta}(D_i)$, then $x\in\underline{R_A^\delta}(D_i)^{'}$ and if $x\notin\underline{R_A^\delta}(D_i)$ then $x\notin\underline{R_A^\delta}(D_i)^{'}$. Hence, we can obtain if $x^-\in\underline{R_A^\delta}(D_i)$, then $\underline{R_A^\delta}(D_i)^{'}=\underline{R_A^\delta}(D_i)-\{x^-\}$. Otherwise, the lower approximation of D_i should be remain constant, *i.e.* $\underline{R_A^\delta}(D_i)^{'}=\underline{R_A^\delta}(D_i)$.

(2) According to the Definition 3.3, we have the $\overline{R_A^\delta}(D_i)=\{x:[x]_A^\delta\cap D_i\neq\emptyset\}$. Thus, when the object $x^-\in D_i$ be deleted from U, the set $[x^-]_A^\delta$ should be removed from the upper approximation $\overline{R_A^\delta}(D_i)$ at most. In this case, it's mean $D_i\cap[x^-]_A^\delta=\{x^-\}$. So, $\overline{R_A^\delta}(D_i)^{'}=\overline{R_A^\delta}(D_i)-[x^-]_A^\delta$. But, there maybe are $y\in(D_i-\{x^-\})$ satisfies that $\Delta_1^-=[y]_A^\delta\cap[x^-]_A^\delta\neq\emptyset$ and the object which $x\in[y]_A^\delta$,where $y\in(D_i-\{x^-\})$ should not be removed from $\overline{R_A^\delta}(D_i)$. Therefore, we can obtain $\overline{R_A^\delta}(D_i)^{'}=(\overline{R_A^\delta}(D_i)-[x^-]_A^\delta)\cup\Delta_1^-$, where $\Delta_1^-=\{x:x\in[x^-]_A^\delta\cap\Delta_2^-\}$ and $\Delta_2^-=\cup_{x\in D_i-\{x^-\}}[x]_A^\delta$.

Thus, the proof is fulfilled. □

Case 2. The deleted object x^- does not belongs to D_i, namely, $x^-\notin D_i$.

Proposition 4.2 *Let $I=(U,C\cup\{d\},V,f)$ be an IvDIS and any $A\subseteq C$. When $x^-\notin D_i(i\in\{1,\ldots,r\})$ be deleted from U, the following properties hold for $\underline{R_A^\delta}(D_i)^{'}$ and $\overline{R_A^\delta}(D_i)^{'}$.*

(1) $\underline{R_A^\delta}(D_i)^{'}=\underline{R_A^\delta}(D_i)\cup\Delta_1^-$. *Where* $\Delta_1^-=\{x:x\in(D_i-\underline{R_A^\delta}(D_i)),\ ([x]_A^\delta)^{'}\subseteq D\}$, *where, if* $x^-\in[x]_A^\delta$ *then* $([x]_A^\delta)^{'}=[x]_A^\delta-\{x^-\}$, *otherwise* $([x]_A^\delta)^{'}=[x]_A^\delta$.

(2) If $x^-\in\overline{R_A^\delta}(D_i)$ then $\overline{R_A^\delta}(D_i)^{'}=\overline{R_A^\delta}(D_i)-\{x^-\}$, otherwise $\overline{R_A^\delta}(D_i)^{'}=\overline{R_A^\delta}(D_i)$.

Proof (1) When the object $x^-\notin D_i$ be deleted from the universe U, we have that $U^{'}=U-\{x^-\}$ and $D_i^{'}=D_i$. So, for any $x\in U^{'}$, $([x]_A^\delta)^{'}=[x]_A^\delta-\{x^-\}$ and it is easy to get that if $[x]_A^\delta\subseteq D_i$ then $([x]_A^\delta)^{'}\subseteq D_i^{'}$. According to definition of lower approximation in Definition 3.3, for any $x\in D_i$, if $x\in\underline{R_A^\delta}(D_i)$ then $([x]_A^\delta)^{'}\subseteq[x]_A^\delta\subseteq D_i=D_i^{'}$, namely, $([x]_A^\delta)^{'}\subseteq D_i^{'}$. Thus, for any $x\in\underline{R_A^\delta}(D_i)\Rightarrow x\in\underline{R_A^\delta}(D_i)^{'}$. On the other hand, for for any $x\in(D_i-\underline{R_A^\delta}(D_i))$, we can get

$[x]_A^\delta \nsubseteq D_i$. However, it maybe exist $x^- \in [x]_A^\delta$ such that $([x]_A^\delta)' \subseteq D_i$ after the deletion of x^-. Then the remain x should be union to $\underline{R_A^\delta}(D_i)'$, namely, $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) \cup \{x\}and([x]_A^\delta)' \subseteq D_i$. Therefore, we have $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) \cup \Delta_1^-$, where $\Delta_1^- = \{x : x \in (D_i - \underline{R_A^\delta}(D_i)),\ ([x]_A^\delta)' \subseteq D_i\}and([x]_A^\delta)' = [x]_A^\delta - \{x^-\}$. (2) Based on Definition 3.3, if $x^- \in \overline{R_A^\delta}(D_i)$ and $x^- \notin D_i$ then, there exists an object $x \in D_i$ such that $x^- \in [x]_A^\delta$. The object x^- be deleted, so $([x]_A^\delta)' = [x]_A^\delta - \{x^-\}$ and $([x]_A^\delta)' \cap D_i \neq \emptyset$. Therefore, the upper approximation of D_i become to $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) - \{x^-\}$. On the other hand, if $x^- \notin \overline{R_A^\delta}(D_i)$, we have for any $x \in D_i$, $x^- \notin [x]_A^\delta$. Hence, the upper approximation of D_i will remain constant, namely, $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)$.

Thus, the Proposition 4.2 is proved. □

4.2 Approaches for incremental knowledge discovering with the insertion of a new object in an IvDIS

Given an interval-valued decision information system $I = (U, C \cup \{d\}, V, f)$ at time t, when the information system is updated by inserting a new object into the universe U at time $t+1$, where denotes the inserted object as x^+. There are three situations may occur: (1) The object x^+ be inserted and exist $x \in D_i$, such that $f(x,d) = f(x^+,d)$, then, $D_i = D_i \cup \{x^+\}$; (2) for any $x \in D_i$, $f(x,d) \neq f(x^+,d)$, then, $D_i = D_i$; (3) There exist a special situation, for any $x \in U$, $f(x,d) \neq f(x^+,d)$, namely, insert x^+ generates a new decision class $D_{r+1} = \{x^+\}$.

Case 1. The object x^+ be inserted and exist $x \in D_i$, such that $f(x,d) = f(x^+,d)$, then, $D_i' = D_i \cup \{x^+\}$.

Proposition 4.3 *Let $I = (U, C \cup \{d\}, V, f)$ be an IvDIS and for any $A \subseteq C$. When the object x^+ be inserted into U and exist $x \in D_i$ $(i \in \{1,2,\ldots,r\})$, such that $f(x,d) = f(x^+,d)$. The following properties hold for $\underline{R_A^\delta}(D_i)'$ and $\overline{R_A^\delta}(D_i)'$.*

(1) If $[x^+]_A^\delta \subseteq D_i'$, where $D_i' = D_i \cup \{x^+\}$then $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) \cup \{x^+\}$, otherwise, $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$.

(2) $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) \cup [x^+]_A^\delta$.

Proof (1) When the object x^+ be inserted into U and exist $x \in D_i$, suchthat$f(x,d) = f(x^+,d)$ then $D_i' = D_i \cup \{x^+\}$ and $U' = U \cup \{x^+\}$. According to definition, we have for any $x \in D_i$, if $[x]_A^\delta \subseteq D_i$ then $x \in \underline{R_A^\delta}(D_i)$. Thus, for any $x \in D_i'$, if $x^+ \in [x]_A^\delta$ then $([x]_A^\delta)' = [x]_A^\delta \cup \{x^+\}$. That is, if $[x]_A^\delta \subseteq D_i$ then $([x]_A^\delta)' \subseteq D_i'$. Analogously, if $[x]_A^\delta \nsubseteq D_i$ then $([x]_A^\delta)' \nsubseteq D_i'$. It follows that if $x \in \underline{R_A^\delta}(D_i)$ then $x \in \underline{R_A^\delta}(D_i)'$. If $x \notin \underline{R_A^\delta}(D_i)$, then $x \notin \underline{R_A^\delta}(D_i)'$. Therefore, if $[x^-]_A^\delta \subseteq D_i'$, we have $x^+ \in \underline{R_A^\delta}(D_i)'$ and $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) \cup \{x^+\}$. Otherwise, $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$.

(2) According to definition, we have $\overline{R_A^\delta}(D_i)' = \{x : ([x]_A^\delta)' \subseteq D_i'\}$. Since $D_i' = D_i \cup \{x^+\}$ then we have $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)' \cup [x^+]_A^\delta$. Because for any $x \in U$ there $([x]_A^\delta)' = [x]_A^\delta \cup \{x^+\}$ or $([x]_A^\delta)' = [x]_A^\delta$. We can obtain that $\overline{R_A^\delta}(D_i)' = \{x \in U' : [x]_A^\delta \subseteq D_i'\} = \bigcup_{x \in D_i} [x]_A^\delta \cup [x^+]_A^\delta = \overline{R_A^\delta}(D_i) \cup [x^+]_A^\delta$.

Thus, the proof is accomplished. □

Case 2. The object x^+ be inserted and for any $x \in D_i$, $f(x,d) \neq f(x^+,d)$, then, $D_i' = D_i$.

Proposition 4.4 *Let $I = (U, C \cup \{d\}, V, f)$ be an IvDIS and for any $A \subseteq C$. When the object x^+ be inserted into U and for any $x \in D_i, f(x,d) \neq f(x^+,d)$, $i \in \{1,2,\ldots,r\}$. We have the following properties about $\underline{R_A^\delta}(D_i)'$ and $\overline{R_A^\delta}(D_i)'$.*

(1) $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) - \Delta_1^+$, *where* $\Delta_1^+ = \{x : x \in \underline{R_A^\delta}(D_i), x^+ \in ([x]_A^\delta)'\}$.

(2) If there exists $x \in D_i$ such that $x^+ \in [x]_A^\delta$, then $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) \cup \{x^+\}$; otherwise $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)$.

Proof (1) When the object x^+ be inserted into U, since $f(x,d) \neq f(x^+,d)$ $(x \in D_i)$ we have $U' = U \cup \{x^+\}$ and $D_i' = D_i$. For any $x \in D_i'$, there $([x]_A^\delta)' = [x]_A^\delta$ or $([x]_A^\delta)' = [x]_A^\delta \cup \{x^+\}$. We have if $[x]_A^\delta \nsubseteq D_i$ then $[x]_A^\delta \nsubseteq D_i'$. That is, if $x \notin \underline{R_A^\delta}(D_i)$ then $x \notin \underline{R_A^\delta}(D_i)'$. Hence, we only consider the object $x \in \underline{R_A^\delta}(D_i)$, namely, $[x]_A^\delta \subseteq D_i$. After the object x^+ be inserted into universe U, there maybe exist $x \in \underline{R_A^\delta}(D_i)$ and $([x]_A^\delta)' = [x]_A^\delta \cup \{x^+\}$ then $([x]_A^\delta)' \nsubseteq D_i' = D_i$, namely, $x \notin \underline{R_A^\delta}(D_i)'$. Therefore, we have $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) - \Delta_1^+$, where $\Delta_1^+ = \{x : x \in \underline{R_A^\delta}(D_i), x^+ \in ([x]_A^\delta)'\}$.

(2) For any $x \in D_i' = D_i$, if $x^+ \in ([x]_A^\delta)'$ where $([x]_A^\delta)' = [x]_A^\delta \cup \{x^+\}$ then we have $x^+ \in \overline{R_A^\delta}(D_i)'$, that is , $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) \cup \{x^+\}$. Otherwise, if for any $x \in D_i$, $x^+ \notin ([x]_A^\delta)'$, that is, $([x]_A^\delta)' = [x]_A^\delta$. Then, we can get $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)$.

Thus, the Proposition 4.4. have been proved. □

Case 3. The object x^+ be inserted and for any $x \in U$, $f(x,d) \neq f(x^+,d)$ then generates a new decision class $D_{r+1} = \{x^+\}$.

Proposition 4.5 *Let $I = (U, C \cup \{d\}, V, f)$ be an IvDIS and for any $A \subseteq C$. When the object x^+ be inserted into U and for any $x \in U$, $f(x,d) \neq f(x^+,d)$, $i \in \{1,2,\ldots,r\}$. Then, the approximations of the new decision class D_{r+1}, $\underline{R_A^\delta}(D_{r+1})$ and $\overline{R_A^\delta}(D_{r+1})$ can be got as follows.*

(1) If $[x^+]_A^\delta = \{x^+\}$ then $\underline{R_A^\delta}(D_{r+1}) = \{x^+\}$, Otherwise $\underline{R_A^\delta}(D_{r+1}) = \emptyset$.

(2) $\overline{R_A^\delta}(D_{r+1}) = [x^+]_A^\delta$.

Proof Because the decision class D_{r+1} is a new class so there are no previously knowledge about D_{r+1}. Hence, we should use the definition of approximations compute the results.

(1) If for any $x \in U, f(x,d) \neq f(x^+,d)$ then a new decision class be formed as $D_{r+1} = \{x^+\}$. It just have an nonempty subset, that is $\{x^+\}$. So according the Definition 3.3, if $[x^+]_A^\delta = \{x^+\}$, then $\underline{R_A^\delta}(D_{r+1}) = \{x^+\}$; otherwise $\underline{R_A^\delta}(D_{r+1}) = \emptyset$.

(2) According the Definition 3.2, we can get the binary relation R_A^δ is symmetric. So $x \in [x^+]_A^\delta$ be equivalence to $x^+ \in [x]_A^\delta$. Because $D_{r+1} = \{x^+\}$, so $\overline{R_A^\delta}(D_{r+1}) = \{x : [x]_A^\delta \cap \{x^+\} \neq \emptyset\} = \{x : [x]_A^\delta \cap \{x^+\} = \{x^+\} = [x^+]_A^\delta$.

Thus, the proof is finished. □

5 Static and incremental algorithms for computing approximations in an IvDIS with dynamic object set

In this section, we design the static and incremental algorithms on the variation of the object set in interval-valued decision information system, respectively. Their flow-process diagram of Algorithms and computational complexity are shown.

5.1 The statical algorithm for computing approximations in an IvDIS

The given static algorithm (Algorithm 1) is the traditional approach for computing the lower and upper approximations in an IvDIS when the object set in the information system is changed. First, we should calculate all the decision classes $U/d = \{D_1, D_2, \ldots, D_r\}$ and conditional classes $[x]_A^\delta$ for each $x \in U$ based R_A^δ. Then, compute the lower and upper approximations in IvIS based on the Definition 3.3. The computational complexity of Algorithm 1, as shown in Table 2.

In steps 1–3, we compute all decision classes, and the set of condition classes for each $x \in U$ based on R_A^δ. Steps 4–6, initialize all lower and upper approximations as empty set. Steps 8–16, calculate the lower and upper approximations in IvIS based on the Definition 3.3. At last, return the results.

Table 2 The computational complexity of Algorithm 1

Steps 1–3	$O(\lvert U\rvert^2 + \lvert U\rvert^2)$
Steps 4–6	$O(r)$
Steps 8–15	$O(\lvert U\rvert \times (\lvert U\rvert \times \lvert D_i + \lvert D_i\rvert\rvert))$
Total	$O(U\rvert^2 + \sum_{i=1}^{r} \lvert D_i\rvert \times (\lvert U\rvert^2 + \lvert U\rvert))$

Algorithm 1: An static algorithm for computing approximations in an IvDIS

Input : An decision interval-valued information system $I = (U, C \cup \{d\}, V, f)$;
Output : The lower and upper approximations of D_i in IvDIS.

1 **begin**
  compute: $U/d = \{D_1, D_2, \cdots, D_r\}$; // the r is the cardinal number of the U/d ;
2   **for** *each* $x \in U$ **do**
      compute: $[x]_A^\delta$; // compute all $[x]_A^\delta$, for any $x \in U$;
3   **end**
4   **for** $i = 1 : r$ **do**
5   **let:** $\underline{R_A^\delta}(D_i) \leftarrow \emptyset$, $\overline{R_A^\delta}(D_i) \leftarrow \emptyset$ // initialize all approximations as empty set ;
6   **end**
7   **for** $i = 1, \cdots, r$ **do**
8   **for** *each* $x \in U$ **do**
9   **if** $[x]_A^\delta \subseteq D_i$ **then**
10        $\underline{R_A^\delta}(D_i) = \underline{R_A^\delta}(D_i) \cup \{x\}$ // compute the lower approximation of D_i by definition ;
11   **end**
12   **if** $x \in D_i$ **then**
13        $\overline{R_A^\delta}(D_i) = \overline{R_A^\delta}(D_i) \cup [x]_A^\delta$ // compute the upper approximation of D_i by definition. ;
14   **end**
15   **end**
16 **end**
17 **return**: $\underline{R_A^\delta}(D_i)$, $\overline{R_A^\delta}(D_i)$.
18 **end**

5.2 The incremental algorithm for updating approximations in an IvDIS when deleting an object

Based on the discussion in previous section, Algorithm 2 was proposed. The given Algorithm 2 is an incremental algorithm for updating the lower and upper approximations in an IvIS when the object set be deleted from the universe U in the interval-valued decision information system.

In Steps 3–16 update the lower and upper approximations of the decision classed D_i, when the deleted object x^- belongs to the decision classes D_i. Among them, the steps 4–8 update the lower approximations of D_i by Proposition 4.1, steps 9–16 update the upper approximations of D_i by Proposition 4.1. Steps 18–32 update the approximations of the decision classes D_i, where the deleted object x^- does not belong to the decision classes D_i. Among them, the steps 18–26 compute the lower approximations of D_i by Proposition 4.2, steps 27–32 compute the upper approximations of D_i by Proposition 3.2. At last, return the result of approximations after deleting the object x^-. The computational complexity of Algorithm 2, as shown in Table 3. The flow-process diagram of Algorithm 2 as shown in Fig. 1.

Algorithm 2: An incremental algorithm for updating approximations in an IvDIS when deleting an object from the universe

Input :
(1)The original interval-valued information system at time t : $I = (U, C \cup \{d\}, V, f)$, where $A \subseteq C$;
(2)The sets $[x]_A^\delta$ at time t for each $x \in U$ and $U/d = \{D_1, D_2, \cdots, D_r\}$;
(3)The original lower and upper approximations at time t : $\underline{R_A^\delta}(D_i)$, $\overline{R_A^\delta}(D_i)$, $i = 1, \cdots, r$;
(4)The object will be deleted from U: x^-.
Output : The lower and upper approximations in an IvDIS after deleting x^-: $\underline{R_A^\delta}(D_i)'$, $\overline{R_A^\delta}(D_i)'$.

1 **begin**
2 **for** $i = 1, \cdots, r$ **do**
3 **if** $x^- \in D_i$ **then**
4 **if** $x^- \in \underline{R_A^\delta}(D_i)$ **then**
5 $\underline{R_A^\delta}(\overline{D_i})' = \underline{R_A^\delta}(D_i) - \{x^-\}$; // update the lower approximation by Proposition 4.1;
6 **else**
7 $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$;
8 **end**
9 $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) - [x^-]_A^\delta$; //update the upper approximation by Proposition 4.1;
10 **for** *each* $x \in [x^-]_A^\delta$ **do**
11 **for** *each* $x' \in D_i - \{x^-\}$ **do**
12 **if** $x \in [x']_A^\delta$ **then**
13 $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)' \cup \{x\}$;
14 **end**
15 **end**
16 **end**
17 **else**
18 **let:** $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$; //update the lower approximation by Proposition 4.2;
19 **for** *each* $x \in (D_i) - \underline{R_A^\delta}(D_i)$ **do**
20 **if** $x^- \in [x]_A^\delta$ **then**
21 $[x]_A^\delta = [x]_A^\delta - \{x^-\}$;
22 **end**
23 **if** $[x]_A^\delta \subseteq D_i$ **then**
24 $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)' \cup \{x\}$;
25 **end**
26 **end**
27 **if** $x^- \in \overline{R_A^\delta}(D_i)$ **then**
28 $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) - \{x^-\}$; //update the upper approximation by Proposition 4.2;
29 **else**
30 $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)$;
31 **end**
32 **end**
33 **end**
34 **return**: $\underline{R_A^\delta}(D_i)'$, $\overline{R_A^\delta}(D_i)'$.
35 **end**

Table 3 The computational complexity of Algorithm 2

Steps 4–9	$O(\lvert\underline{R_A^\delta}(D_i)\rvert + \lvert U\rvert)$
Steps 10–16	$O(\lvert U\rvert \times \lvert D_i\rvert \times \lvert U\rvert)$
Steps 19–26	$O(\lvert D_i\rvert \times (\lvert U\rvert + \lvert U\rvert \times \lvert D_i\rvert))$
Steps 27–31	$O(\lvert\overline{R_A^\delta}(D_i)\rvert)$
Total	$O(\sum_{i=1}^{r} \lvert D_i\rvert(\lvert\underline{R_A^\delta}(D_i)\rvert + \lvert D_i\rvert^2 \times \lvert U\rvert + \lvert D_i\rvert \times \lvert U\rvert^2 + \lvert\overline{R_A^\delta}(D_i)\rvert))$

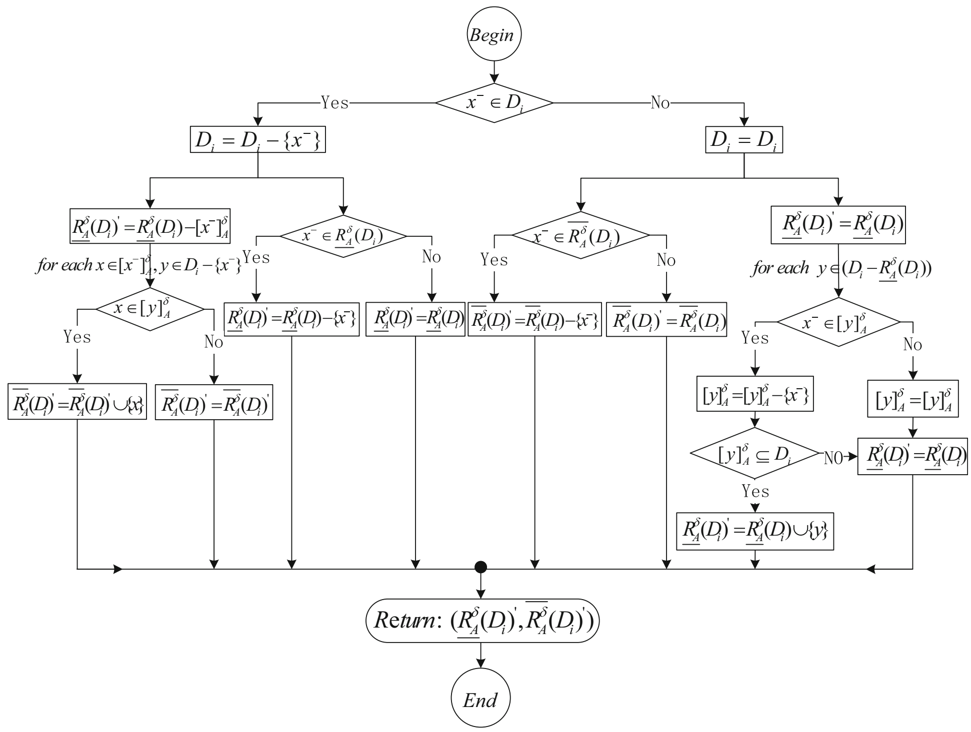

Fig. 1 The flow-process diagram of Algorithm 2

5.3 The incremental algorithm for updating approximations in an IvDIS when inserting an object

According to discussion in Sect. 4.1. Algorithm 3 was proposed. The given Algorithm 3 is an incremental algorithm for updating the lower and upper approximations in an IvDIS when the object set be inserted into the universe U in the interval-valued decision information system.

In step 1, compute the set $[x^+]_A^\delta$ with respect to the inserted object x^+ based on the relation R_A^δ. Steps 3–23, update the lower and upper approximations of the decision classes D_i when the inserted object x^+ will belong to the decision classes D_i or $D_j, i \neq j$. Steps 3–10, update the approximations of D_i by Proposition 4.3. Steps 12–22, update the approximations of decision class D_i where the inserted object x^+ not belong to D_i by Proposition 4.4. Steps 24–31, generate a new decision class and compute the approximations of the new decision class D_{r+1} according the Proposition 4.5. The computational complexity of Algorithm 3 as shown in Table 4. The flow-process diagram of Algorithm 3 as shown in Fig. 2.

Algorithm 3: An incremental algorithm for updating approximations in an IvDIS when inserting an object into the universe U.

Input :
(1)The original interval-valued information system at time t : $I = (U, AT \cup \{d\}, V, f)$;
(2)The sets $[x]_A^\delta$ at time t for each $x \in U$ and $U/d = \{D_1, D_2, \cdots, D_r\}$;
(3)The original lower and upper approximations at time t : $\underline{R_A^\delta}(D_i), \overline{R_A^\delta}(D_i), i = 1, \cdots, r$;
(4)The object will be inserted into U: x^+.
Output : The approximations in an IvDIS after inserting x^+: $\underline{R_A^\delta}(D_i)', \overline{R_A^\delta}(D_i)', i \in \{1, 2, \cdots, r, r+1\}$.

begin
  compute: the set $[x^+]_A^\delta$ with respect to R_A^δ;
  for $i = 1, \cdots, r$ **do**
    if $x^+ \in D_i$ **then**
      $D_i = D_i \cup \{x^+\}$;
      if $[x^+]_A^\delta \subseteq D_i$ **then**
        $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i) \cup \{x^+\}$; // update the lower approximation by Proposition 4.3;
      else
        $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$;
      end
      $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i) \cup [x^+]_A^\delta$; // update the upper approximation by Proposition 4.3;
    else
      let: $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)$;
      for *each* $x \in \underline{R_A^\delta}(D_i)'$ **do**
        if $x^+ \in [x]_A^\delta$ **then**
          $\underline{R_A^\delta}(D_i)' = \underline{R_A^\delta}(D_i)' - \{x\}$; // update the lower approximation by Proposition 4.4;
        end
      end
      for $x \in D_i$ **do**
        if $x^+ \in [x]_A^\delta$ **then**
          $\overline{R_A^\delta}(D_i)' = \overline{R_A^\delta}(D_i)' \cup \{x^+\}$; // update the upper approximation by Proposition 4.4;
        end
      end
    end
    if $\forall x \in U, f(x, d) \neq f(x^+, d)$ **then**
      generate: a new decision class D_{r+1};
      if $[x^+]_A^\delta \subseteq D_{r+1}$ **then**
        $\underline{R_A^\delta}(D_{r+1}) = \{x^+\}$; // update the lower approximation by Proposition 4.5;
      else
        $\underline{R_A^\delta}(D_{r+1}) = \emptyset$;
      end
      $\overline{R_A^\delta}(D_{r+1}) = [x^+]_A^\delta$; // update the upper approximation by Proposition 4.5;
    end
  end
  return: $\underline{R_A^\delta}(D_i)', \overline{R_A^\delta}(D_i)', i \in \{1, 2, \cdots, r, r+1\}$
end

6 Experiment evaluations

In this section, to evaluate the performance of the proposed incremental algorithms, we conduct a series of experiments to compare the computational time between the statical algorithm and incremental algorithms for updating approximations based on standard data sets where download from the machine learning data repository, University of California at Irvine (http://archive.ics.uci.edu/ml/datasets.html), They are named "Energy efficiency", "Airfoil Self-Noise", "Wine Quality-red", "Wine Quality-white", "Letter Recognition", "Spoken Arabic Digit" and the basic information of data sets is outlined in Table 5.

We must point out that most data sets that we can download are the single-valued attributes characteristics. Hence, we should construct the interval-valued information data by utilizing multiply error precision α based on the download data sets. Let $I = (U, C \cup \{d\}, V, f)$ be a single-valued decision information system, for any $x_i \in U$and$a_j \in C$, the attribute value of x_i at a_j is $v = a_j(x_i)$. Then, an interval number can be generated as followed way.

Table 4 The computational complexity of Algorithm 3

Step 1	$O(\lvert U \rvert)$
Steps 5–10	$O(\lvert D_i \rvert \times \lvert U \rvert + \lvert U \rvert)$
Steps 13–17	$O(\lvert U \rvert \times \lvert \underline{R}_A^{\delta}(D_i) \rvert)$
Steps 18–22	$O(\lvert D_i \rvert \times \lvert U \rvert)$
Steps 24–31	$O(\lvert U \rvert \times (\lvert U \rvert + \lvert U \rvert))$
Total	$O(\lvert U \rvert + \sum_{i=1}^{r} \lvert D_i \rvert \lvert U \rvert (\lvert D_i \rvert + \lvert \underline{R}_A^{\delta}(D_i) \rvert) + \lvert U \rvert^2)$

$$I_{ij} = [(1-\alpha) \times v, (1+\alpha) \times v].$$

In this paper, we set the error precision $\alpha = 0.05$ and given an interval similarity threshold $\delta = 0.65$. In different fields one can use different error precisions. After completing this step the interval-valued decision information system is obtained. This experimental computing program is running on a personal computer with following hardware and software as Table 6.

The objective of the following experiments is to show the time efficiency of the proposed incremental algorithms for updating lower and upper approximations while the object set varies with time and the attribute set keep constant. In order to distinguish the computational times between static and incremental algorithms, we let the one data set as the basic data set at time t, and choose another data set as the test data set. The deletion and insertion experiments are set as follows, respectively.

- In deletion experiment, let the whole original data set as the basic data at time t. Then, each experiment choose one part of the original set (the ratio from 5 to 50 %) as the immigrating objects which will be deleted from the information system at time $t+1$.
- In insertion experiment, we choose the 60 % of the original data sets as the primary data set at time t, and the remaining parts as the test data. Each test choose a part of the test data (the ratio from 10 to 100 %) enter into the information system at time $t+1$.

We compute the lower and upper approximations in dynamic object set by static and incremental approaches, respectively, and record time for each experiment.

6.1 A comparison of computational efficiency between static and incremental algorithm with the deletion of the object set

To compare the efficiency of non-incremental algorithm (Algorithm 1) and incremental algorithm (Algorithm 2) for computing lower and upper approximations when deleting the objects from the data sets. We compute the time of the two algorithms on the given datasets in Table 5 with the different deleting ratio (from 5 to 50 %), we show the

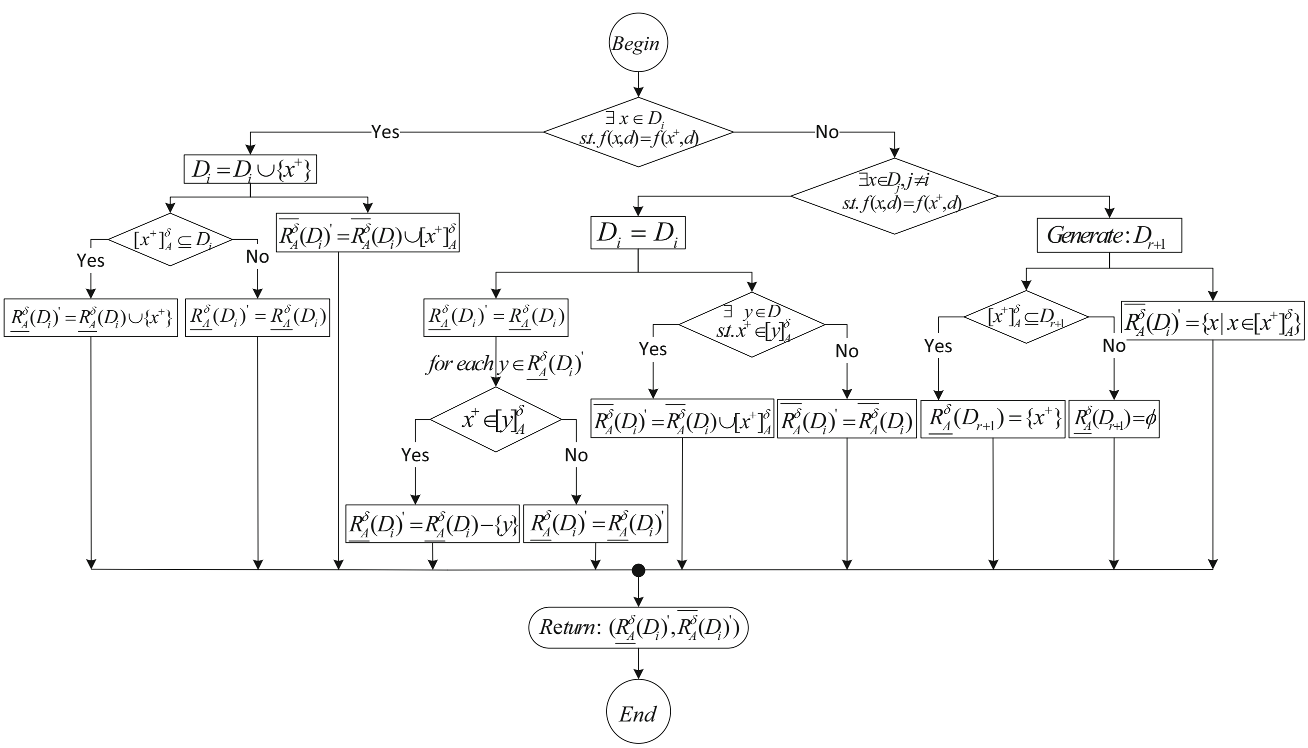

Fig. 2 The flow-process diagram of Algorithm 3

Table 5 The basic information of data sets

No.	Data set name	Abbreviation	Objects	Attributes	Decision classes
1	Energy efficiency	EE	768	8	3
2	Airfoil self-noise	AS	1503	6	5
3	Wine quality-red	WQ-r	1599	11	6
4	Wine quality-white	WQ-w	4898	11	7
5	Letter recognition	LR	8084	16	14
6	Spoken arabic digit	SAD	8800	13	3

Table 6 The description of experiment environment

Name	Model	Parameters
CPU	Intel i3-370	2.40GHz
Memory	Samsung DDR3	2GB, 1067MHz
Hard disk	West data	500GB
System	Windows 7	32 bit
Platform	VC++	6.0

experimental results in Table 7, and the unit for these numbers are seconds.

More detailed changing trend-line of each of Algorithms 1 and 2 are illustrated in Fig. 3.

In each sub-figure (a)–(f) of Fig. 3, the *x*-coordinate pertains to the ratio of the numbers of the deleting data and original data, while the *y*-coordinate concerns the computational time. According to the experimental results in Table 7 and Fig. 3, we can find, for the non-incremental algorithm the computational time for computing approximations with deletion of the objects from the universe U is decreasing monotonically along with the increase of ratios. On the contrary, for the incremental algorithm, we can see that the computational efficiency for computing approximations is changing smoothly along with the increase of deleting ratios. It's easy to get the incremental algorithm always performs faster than the non-incremental algorithm for computing approximations. In sub-figure (a) of Fig. 3, the two trend-line of two algorithms are very closed when the ratio is 50 %, that is, when the deletion objects is equal or bigger than remain objects. It must be note that there is a threshold depending on the data set. Different data sets have different thresholds. Once the delete ratio over the threshold, namely, the deleted data set is bigger than the remaining data set maybe the incremental algorithm is slower than the non-incremental. In other five sub-fingers, the Incremental algorithms' commotional time are smaller than static algorithm. So, the incremental algorithm is very efficiency especially when need to delete the data set is far smaller than the original data set.

6.2 A comparison of computational efficiency between static and incremental algorithm with the insertion of the object set

Similar to the experiment schemes for comparing the efficiencies between non-incremental and incremental algorithms when deleting the objects from the universe U, we also adopt such schemes to compare the performance of algorithms on the case of inserting the objects into the universe U. We compute the two algorithms (Algorithms 1 and 3) on the six UCI data sets in Table 5 with the

Table 7 A comparison of computational time between Algorithms 1 and 2 versus different updating rates when deleting objects (unit is second)

Del. (%)	EE (s)		AS (s)		WQ-r (s)		WQ-w (s)		LR (s)		SAD (s)	
	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.
5	0.826	0.002	2.059	0.008	4.274	0.010	39.983	0.109	234.585	0.632	164.577	0.460
10	0.699	0.009	1.872	0.020	3.844	0.050	35.485	0.478	210.723	2.655	147.547	1.707
15	0.672	0.020	1.661	0.050	3.442	0.101	31.731	1.007	188.346	5.825	130.570	3.807
20	0.592	0.028	1.466	0.073	3.026	0.190	27.956	1.772	165.930	8.322	115.759	6.712
25	0.523	0.050	1.341	0.150	2.616	0.220	24.612	2.538	145.877	14.116	102.631	10.458
30	0.462	0.090	1.127	0.184	2.308	0.362	21.622	3.415	127.016	17.094	89.037	15.048
35	0.359	0.120	1.005	0.259	1.995	0.509	18.618	4.699	109.581	25.422	76.576	20.477
40	0.335	0.140	0.838	0.301	1.724	0.631	15.694	5.907	93.552	34.974	65.435	26.709
45	0.279	0.171	0.745	0.382	1.414	0.798	13.180	7.269	78.695	41.918	54.861	33.762
50	0.236	0.221	0.578	0.426	1.190	0.919	10.942	8.759	64.818	54.083	46.340	41.685

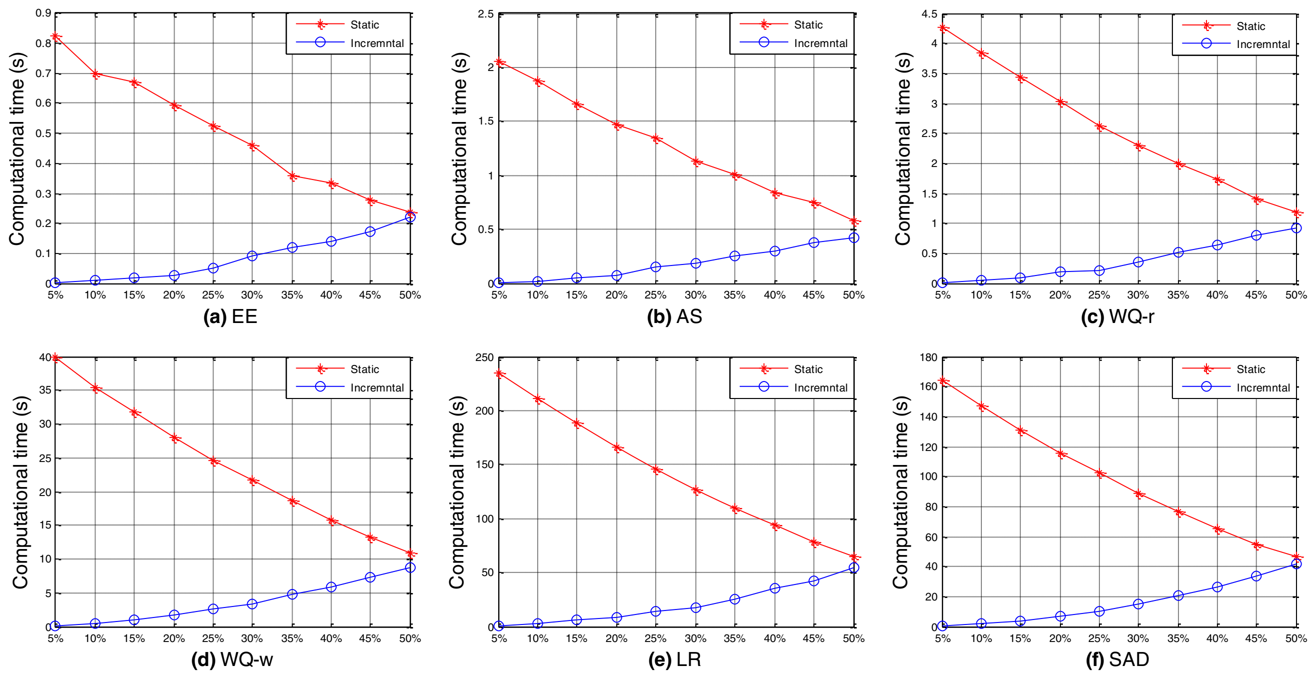

Fig. 3 A comparison of non-incremental (Algorithm 1) and incremental (Algorithm 2) Algorithms versus the deleting ratio of data

Table 8 A comparison of computational time between Algorithms 1 and 3 versus different updating rates when inserting objects (unit is second)

Ins. (%)	EE (s)		AS (s)		WQ-r (s)		WQ-w (s)		LR (s)		SAD (s)	
	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.	Static	Inc.
10	0.369	0.007	0.936	0.015	1.950	0.010	17.877	0.070	106.236	0.404	68.203	0.310
20	0.418	0.012	1.107	0.020	2.202	0.040	20.529	0.310	119.949	1.644	78.034	1.121
30	0.521	0.030	1.201	0.040	2.465	0.070	23.698	0.627	134.391	3.741	86.640	2.447
40	0.539	0.060	1.341	0.060	2.768	0.130	27.078	1.114	149.742	6.572	96.938	4.316
50	0.592	0.089	1.466	0.090	3.042	0.200	29.641	1.720	165.947	10.296	107.142	6.699
60	0.654	0.152	1.637	0.140	3.354	0.310	32.572	2.498	182.948	14.831	119.743	9.621
70	0.698	0.210	1.793	0.210	3.679	0.390	35.395	3.407	200.819	20.112	141.846	13.117
80	0.764	0.226	1.989	0.302	4.013	0.504	38.960	4.443	219.678	26.214	153.966	17.080
90	0.829	0.312	2.178	0.365	4.360	0.631	42.255	5.582	238.985	33.206	168.124	21.644
100	0.891	0.374	2.285	0.429	4.751	0.782	46.264	6.886	259.343	40.974	182.621	26.680

changing of updating ratios for each data sets. The experimental results are shown in Table 8, and the units for these numbers are seconds. More detailed changing trend-line of each of Algorithms 2 and 3 are illustrated in Fig. 4.

In each sub-figure (a)–(f) of Fig. 4, the x-coordinate pertains to the ratio of the numbers of the inserted objects and test data, while the y-coordinate concerns the computational time. According to the experimental results in Table 8 and Fig. 4, we can see, for the non-incremental algorithm, the computational time for computing approximations with insertion of the objects into the universe U is increasing monotonically along with the increase of ratios. On the contrary, for the incremental algorithm, we can see that the computational efficiency for computing approximations is changing smoothly along with the increase of inserting ratios. It's easy to get the incremental algorithm always performs faster than the non-incremental algorithm for computing approximations. So, the incremental algorithm is efficiency when the objects insert into the universe, especially the original data set is an big data set and when the changing data set relatively small is very efficiency.

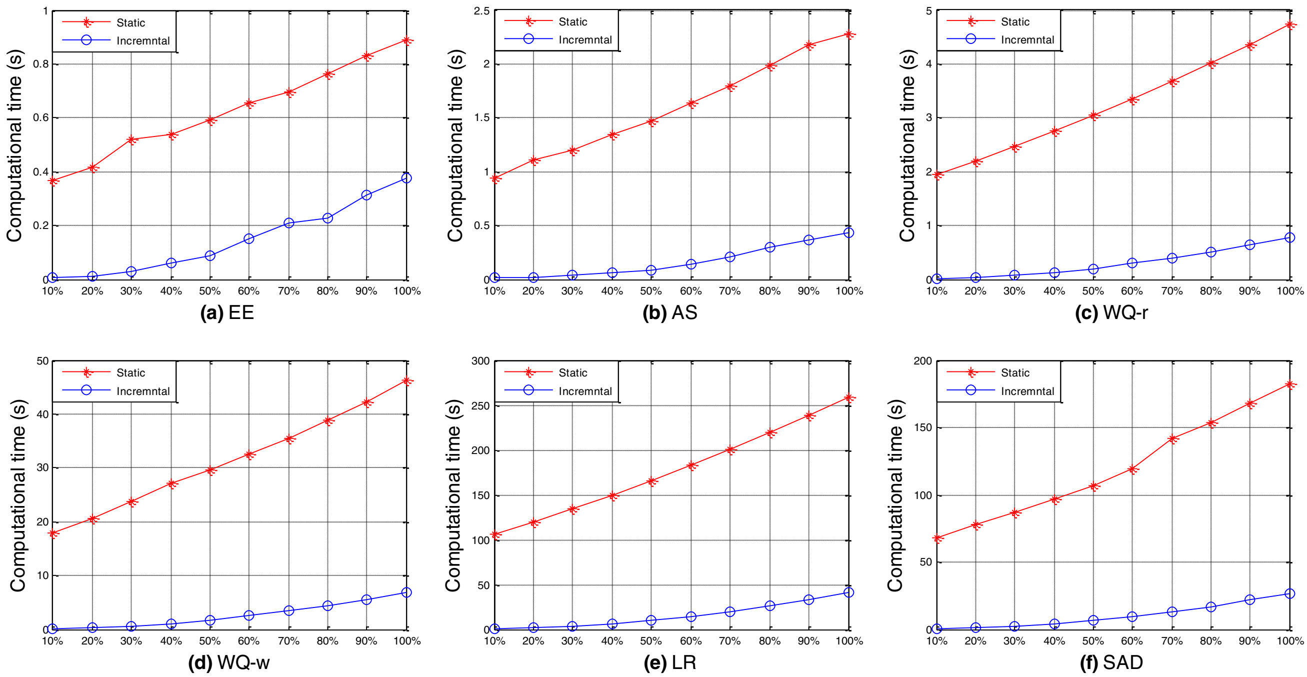

Fig. 4 A comparison of non-incremental (Algorithm 1) and incremental (Algorithm 3) Algorithms versus the inserting ratio of data

7 Conclusions

An information system evolves with time in the dynamic data environment. The approximations may vary according to the variation of the information system and dynamically maintain approximations are crucial to the efficiency of knowledge discovery. Therefore, how to effectively exploit the prior data structure and knowledge to achieve the dynamic maintenance of knowledge is vital to RST-based data analysis in the real applications. In this paper, according to the extension of the classical single-valued information system to the interval-valued decision information system under the binary relation R_A^δ that we constructed from the interval similarity degree. Then, we presented the dynamic maintenance strategies of approximations in the IvDIS by adding or removing some objects, respectively. Two incremental algorithms to update approximations under the dynamic data set were introduced. Experimental studies pertaining to six UCI data sets showed that the incremental algorithms can improve the computational efficiency for updating approximations when the object set in the information system varies over time and the data is bigger the more effective.

The incremental technique is an effective way to maintain knowledge in the dynamic circumstance. We just have discussed the incremental methods for updating approximations in IvDIS when the information system is updated by inserting or deleting objects in this talk. In our further work, we will extend the proposed approaches to handle the problems of updating approximations when the object attributes' adding or deleting or attributes' values vary with time in interval-valued decision information system. Furthermore, the studies of incremental updating knowledge on fuzzy information system and other generalized information systems based on rough set theory need to further promote.

Acknowledgments The author would like to thank the valuable suggestions from the anonymous referees and the editor in chief for improving the quality of the paper. This work is supported by Natural Science Foundation of China (No. 61105041, No. 61472463, No. 61402064), National Natural Science Foundation of CQ CSTC (No. cstc 2013jcyjA40051, cstc2015jcyjA40053), Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information (Nanjing University of Science and Technology), Ministry of Education (No. 30920140122006), Key Laboratory of Oceanographic Big Data Mining and Application of Zhejiang Province (No.OBDMA201503), and Graduate Innovation Foundation of Chongqing University of Technology (No.YCX2015227, YCX2014236), and Graduate Innovation Foundation of CQ (No.CYS15223).

References

1. Bang WC, Bien Z (1999) New incremental learning algorithm in the framework of rough set theory. Int J Fuzzy Syst 1:25–36
2. Chan CC (1998) A rough set approach to attribute generalization in data mining. Inf Sci 107(1–4):169–176
3. Chen HM, Li TR, Qiao SJ, Ruan D (2010) A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values. Int J Intell Syst 25(10):1005–1026

4. Chen HM, Li TR, Ruan D (2012) Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining. Knowl Based Syst 31:140–161
5. Chen HM, Li TR, Ruan D, Lin JH, Hu CX (2013) A rough-set based incremental approach for updating approximations under dynamic maintenance environments. IEEE Trans Knowl Data Eng 25:274–284
6. Dai JH, Wang WT, Xu Q, Tian HW (2012) Uncertainty measurement for interval-valued decision systems based on extended conditional entropy. Knowl Based Syst 27:443–450
7. Gong ZT, Sun BZ, Chen DG (2008) Rough set theory for the interval-valued fuzzy information systems. Inf Sci 178:1968–1985
8. Leung Y, Fischer MM, Wu WZ, Mi JS (2008) A rough set approach for the discovery of classification rules in interval-valued information systems. Int J Approx Reason 47:233–246
9. Li HX, Wang MH, Zhou XZ, Zhao JB (2012) An interval set model for learning rules from incomplete information table. Int J Approx Reason 53:24–37
10. Li TR, Ruan D, Geert W (2007) A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowl Based Syst 20:485–494
11. Li TR, Ruan D, Song J (2007) Dynamic maintenance of decision rules with rough set under characteristic relation. In: Wireless communications, networking and mobile computing, pp 3713–3716
12. Liu D, Li TR, Ruan D, Zou WL (2009) An incremental approach for inducing knowledge from dynamic information systems. Fundam Inf 94:245–260
13. Liu D, Li TR, Ruan D, Zhang JB (2011) Incremental learning optimization on knowledge discovery in dynamic business intelligent systems. J Global Optim 51:325–344
14. Liu D, Li TR, Zhang JB (2014) A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems. Int J Approx Reason 55:1764–1786
15. Liu D, Li TR, Zhang JB (2015) Incremental updating approximations in probabilistic rough sets under the variation of attributes. Int J Knowl Based Syst 73:81–96
16. Li SY, Li TR, Liu D (2013) Dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set. Int J Intell Syst 28:729–751
17. Luo C, Li TR, Chen HM, Liu D (2013) Incremental approaches for updating approximations in set-valued ordered information systems. Int J Knowl Based Syst 50:218–233
18. Li JH, Mei CL, Lv YJ (2011) Knowledge reduction in decision formal contexts. Knowl Based Syst 24:709–715
19. Li JH, Mei CL, Lv YJ (2012) Knowledge reduction in real decision formal contexts. Inf Sci 189:191–207
20. Lu SX, Wang XZ, Zhang GQ, Zhoua X (2015) Effective algorithms of the Moore–Penrose inverse matrices for extreme learning machine. Intell Data Anal 19(4):743–760
21. Miao DQ, Zhang N, Yue XD (2009) Knowledge reduction in interval-valued information systems. In: Proceedings of the 8th IEEE international conference on cognitive informatics, pp 320–327
22. Moore RE (1966) Interval analysis. Prentice Hall, New Jersey
23. Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11:341–356
24. Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht
25. Pawlak Z (1998) Rough set theory and its applications to data analysis. Cybern Syst 29:661–688
26. Qian YH, Liang JY, Dang CY (2008) Interval ordered information systems. Comput Math Appl 56:1994–2009
27. Shan N, Ziarko W (1995) Data-based acquisition and incremental modification of classification rules. Comput Intell 11(2):357–370
28. Sun BZ, Gong ZT, Chen DG (2008) Fuzzy rough set theory for the interval-valued fuzzy information systems. Inf Sci 178:2794–2815
29. Wang XZ, Aamir R, Fu AM (2015) Fuzziness based sample categorization for classifier performance improvement. J Intell Fuzzy Syst 29:1185–1196
30. Wang R, Kwon S, Wang XZ, Jiang QS (2015) Segment based decision tree induction with continuous valued attributes. IEEE Trans Cybern 45(7):1262–1275
31. Wang XZ, Xing HJ, Li Y et al (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst 23(5):1638–1654
32. Wang XZ (2015) Uncertainty in learning from big data-editorial. J Intell Fuzzy Syst 28(5):2329–2330
33. Yang XB, Yu DJ, Yang JY, Wei LH (2009) Dominance-based rough set approach to incomplete interval-valued information system. Data Knowl Eng 68:1331–1347
34. Yang XB, Qi Y, Yu DJ, Yu HL, Yang JY (2015) α-Dominance relation and rough sets in interval-valued information systems. Inf Sci 294:334–347
35. Yao YY (1993) Interval-set algebra for qualitative knowledge representation. In: Proceedings of the 5th international conference on computing and information, pp 370–375
36. Yao YY, Li XN (1996) Comparison of rough-set and interval-set models for uncertain reasoning. Fundam Inf 27:289–298
37. Yao YY (2009) Interval sets and interval-set algebras. In: Proceedings of the 8th IEEE international conference on cognitive informatics, pp 307–314
38. Yang HZ, Yee L, Shao MW (2011) Rule acquisition and attribute reduction in real decision formal contexts. Soft Comput 15:1115–1128
39. Zhang HY, Su YJ (2007) A ranking approach with inclusion measure in multiple-attribute interval-valued decision making. In: RSFDGrC, LNAI, vol 4482, pp 411–418
40. Zhang JB, Li TR, Ruan D, Liu D (2012) Neighborhood rough sets for dynamic data mining. Int J Intell Syst 27(4):317–342
41. Zhang WX, Wu WZ, Liang JY, Li DY (2001) Theory and method of rough sets. Science Press, Beijing
42. Zhang X, Mei CL, Chen DG, Li JH (2014) Multi-confidence rule acquisition and confidence-preserved attribute reduction in interval-valued decision systems. Int J Approx Reason 55:1787–1804