A Rough-Set-Based Incremental Approach for Updating Approximations under Dynamic Maintenance Environments

Hongmei Chen, Tianrui Li, *Senior Member*, *IEEE*, Da Ruan, *Member*, *IEEE*, Jianhui Lin, and Chengxiang Hu

Abstract—Approximations of a concept by a variable precision rough-set model (VPRS) usually vary under a dynamic information system environment. It is thus effective to carry out incremental updating approximations by utilizing previous data structures. This paper focuses on a new incremental method for updating approximations of VPRS while objects in the information system dynamically alter. It discusses properties of information granulation and approximations under the dynamic environment while objects in the universe evolve over time. The variation of an attribute's domain is also considered to perform incremental updating for approximations under VPRS. Finally, an extensive experimental evaluation validates the efficiency of the proposed method for dynamic maintenance of VPRS approximations.

Index Terms—Variable precision rough-set model, knowledge discovery, granular computing, information systems, incremental updating

1 Introduction

Granular computing (GrC) based on Zadeh's "information granularity" [1], [2] describes and processes uncertain, vague, incomplete, and mass information. By using granules and relationships between granules, the focus on certain granularity, and transfer freely between different granularities are realized. Problems can be solved at different granularities' levels of GrC. Yao et al. described basic issues and methods of GrC [3], [4], [5], [6], [7], [8]. The frameworks, models, methodologies, and techniques of GrC were studied in [9], [10], [11], and [12]. By now GrC has been successfully used in knowledge discovery [13], [14], [15]. Rough Set Theory (RST), one of the leading special cases of GrC approaches [16], [17], is a formal mathematical theory that models knowledge about the domain of interest in terms of a collection of equivalence relations [18], [19]. Each equivalent class of RST may be viewed as a granule since the elements in the same equivalence class are indistinguishable. For any subset of the universe is described approximately by the equivalence classes. The granulation structure is thus induced by the equivalence relation in RST.

- *H. Chen and T. Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China. E-mail: {hmchen, trli}@swjtu.edu.cn.*
- *D. Ruan is deceased.*
- *J. Lin is with the Traction Power State Key Laboratory, Southwest Jiaotong University, Chengdu 610031, China. E-mail: linjhyz@126.com.*
- *C. Hu is with the School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China. E-mail: chengxiang0550@163.com.*

Manuscript received 5 June 2010; revised 18 Sept. 2011; accepted 4 Oct. 2011; published online 14 Oct. 2011.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-06-0321.
Digital Object Identifier no. 10.1109/TKDE.2011.220.

In this paper, we focus on the variation of granularity in RST. With the variation of an information system, the equivalence classes in the information system may vary over time while the granularity of the information system also evolves over time. In RST, $S = (U, A, V, f)$ describes an information system. The variation of an information system means the four elements, U, A, V, f, may vary over time. Those are the coarsening and refining of an object set, the coarsening and refining of an attribute set, the coarsening and refining of attribute values, and the compound variation of different factors. When the information system is changed, it is highly time consuming to reprocess the whole operations for getting new results. Therefore, it is important to avoid unnecessary computations by utilizing the previous data structures or results. Generally, incremental updating is an effective method to maintain knowledge dynamically and it has been used to data analysis in the real-time applications, e.g., steam data, interactive application, multiple learning agents, and the applications with a limited memory or computation ability [20], [21], [22]. For example, Kang et al. proposed a novel algorithm for incremental and general evaluation of continuous reverse nearest neighbor queries [23]. Altiparmak et al. proposed an algorithm to update a synopsis over a sliding window of most recent entries dynamically in constant time [24].

There has been much research on dealing with incremental updating in RST. When the object set remains unchanged and a single attribute is added into or deleted from the system, An et al. proposed an algorithm for incremental updating approximations and a method for extracting rules [25]. Li et al. presented an algorithm for incremental updating approximations and extracting rules incrementally while multiple attributes are deleted from or added into the system simultaneously [26]. When the attribute set keeps unchanged and a single object is inserted into or deleted from the system, Shan and Ziarko proposed an incremental algorithm for obtaining rules [27]. In this algorithm, it needs a new object that is consistent with the original decision table, and new decision classes do not appear. Bang and Zeungnam [28] proposed an incremental algorithm to find a minimal set of rules from a decision table when new instances are added. Tong and An developed a δ-decision matrix algorithm in [29]. When a new object is added to the decision table, they analyzed the changes to condition classes and decision classes and then proposed methods for incremental updating δ-decision matrix and measures of the rule [29]. An incremental algorithm for extracting rules based on the improved discernibility matrix was given by Liu et al. in [30]. Guo et al. proposed an incremental method for extracting rules based on the search tree in [31] of which it does not need to create the discernibility matrix. Furthermore, an effective incremental algorithm, named as RRIA, for knowledge acquisition based on the rule tree was proposed by Zheng and Wang [32]. The algorithm for updating rules is based on the original rule set. It avoids recalculation and then improves the efficiency. Based on the Rule-extraction algorithm (REA) of Tseng [33], Fan et al. proposed an incremental approach for rule induction as new data adding to the information system [34]. The updating data are divided into four cases. That is, the original rules cannot cover all instances, updating data causes a contradiction in the original rules, updating data does not cause any contradiction in original rules, updating data does not cause any contradiction and the original REA rules cannot dominate the new data set but it can be dominated by original reducts. Therefore, Fan et al. designed different algorithms for different cases to compute the reducts or strength index (SI) influenced by a new object set. The rule sets are updated by modifying original reducts partially. In [35], Chen et al. discussed algorithms for incremental updating approximations while attribute values coarsening and refining.

One limitation of Traditional Rough Sets model (TRS) is that the classification must be completely correct or certain. It only considers complete "including" or "nonincluding," but not "including" in a certain degree. In real-life applications, the imprecise and missing data are common because of the errors of measurement, misapprehension, and restriction on access to data or negligence on the registration data. This motivates many researchers to incorporate probabilistic approaches into TRS [36]. Yao and Wong proposed a Decision-Theoretic Rough Set model (DTRS) to require threshold parameters for defining approximations [37]. In DTRS, the parameters are calculated based on a loss function through the Bayesian decision procedure. It gives guidelines for the estimation of different parameters in probabilistic rough set approximations. The Variable Precision Rough Set model (VPRS) proposed by Ziarko gives a classification strategy in which the error rate is less than a given threshold [38], [39]. The VPRS has been solving noise data problems with great success in many applications [40], [41], [42] although a difficulty with VPRS is the estimation of the threshold. The difficulty has been well addressed by DTRS since the threshold can be obtained by the loss function in DTRS. When the information system evolves with time, there are a few works on knowledge updating in VPRS. Wang et al. discussed the relation between a new record and the existing equivalence classes on the condition attributes and effects on the rule set caused by the new record in VPRS. They further proposed an incremental rule acquisition algorithm based on VPRS [43]. In [44], Liu et al. constructed an accuracy matrix and a coverage matrix under VPRS. Then they proposed an incremental approach to update the accuracy matrix and coverage matrix to obtain interesting knowledge w.r.t. the immigration or emigration of objects.

Generally, the computation of approximations is a necessary step in knowledge representation and reduction based on rough sets. Approximation may further be applied to data mining related work. To our best knowledge, incremental updating of approximations under VPRS has not yet been discussed so far. In this paper, we investigate approaches for incremental updating approximations in terms of the coarsening and refining of the object set. When an information system is changed by inserting or deleting objects, there is no need to recalculate the whole data. It is efficient to carry out the step of the incremental learning for the new data [45]. However, the problems of incremental updating in the information system have only been discussed in the case that objects increase in most existing literatures. In this paper, we outline the principle of incremental updating approximations when objects change (objects may increase or decrease and the concept induced by the decision attribute may also alter) dynamically in information systems under VPRS. We also consider the variation of the value domain of an attribute when inserting or deleting an object into the universe. We further discuss the variation of granulation of knowledge due to the evolution of objects.

The paper is organized as follows: In Section 2, we review basic concepts of rough sets and VPRS. In Section 3, we describe the principle of dynamically updating equivalence classes. In Section 4, we propose an incremental method for updating approximations under VPRS when the object set varies over time. In Section 5, we illustrate how to update equivalence classes and approximations under VPRS incrementally when an object is inserted into or deleted from the system. In Section 6, we show all performance evaluations. And in Section 7, we conclude the paper and outline our future research directions.

2 Preliminaries

We cite several concepts of rough sets and VPRS from [18], [19], [38], [39].

Definition 2.1. *A quadruple $S = (U, A, V, f)$ is an information system, where U is a nonempty finite set of objects, called the universe. A is a nonempty finite set of attributes, $A = C \cup D$, $C \cap D = \emptyset$, where C and D denote the sets of condition attributes and decision attributes, respectively. $V = \bigcup_{a \in A} V_a$, V_a is a domain of attribute a. $f : U \times A \to V$ is an information function, which gives values to every object on each attribute, namely, $\forall a \in A$, $x \in U$, $f(x, a) \in V_a$.*

The pair (U, R) is called an *approximation space*, where U is a nonempty finite set of objects, R is an equivalence relation on U, $U/R = \{E_1, E_2, \dots, E_n\}$ is a set of equivalence classes induced by R. The equivalence class containing x is denoted by $[x]_R = \{y | (x, y) \in R\}$.

Definition 2.2. *Let X and Y be nonempty subsets of the universe U. Let*

$$c(X, Y) = \begin{cases} 1 - |X \cap Y|/|X|, \ |X| > 0 \\ 0, \ |X| = 0, \end{cases} \tag{1}$$

where $|X|$ denotes the cardinality of set X, $c(X, Y)$ is called the relative degree of misclassification.

Definition 2.3. *Let $0 \le \beta < 0.5$. The majority inclusion relation is defined as*

$$Y \overset{\beta}{\supseteq} X \Leftrightarrow c(X, Y) \le \beta \tag{2}$$

"Majority" requires that the number of common elements between X and Y is larger than half of the number of elements in X.

Definition 2.4. *$\forall X \subseteq U$, β lower approximation, β upper approximation, β boundary region are defined, respectively, as follows:*

$$\underline{R}_\beta(X) = \cup\{E \in U/R | c(E, X) \le \beta\}; \tag{3}$$

$$\overline{R}_\beta(X) = \cup\{E \in U/R | c(E, X) < 1 - \beta\}; \tag{4}$$

$$bnr_\beta(X) = \cup\{E \in U/R | \beta < c(E, X) < 1 - \beta\}. \tag{5}$$

$\underline{R}_\beta(X)$ is also called β positive region, denoted by $posr_\beta(X)$. β negative region of X is defined as

$$negr_\beta(X) = \cup\{E \in U/R | c(E, X) \ge 1 - \beta\}, \tag{6}$$

Definition 2.5. *An approximation space (U, R) can be regarded as a knowledge base about U. $U/C = \{C_1, C_2, \ldots, C_n\}$ is a partition of U. Let*

$$\hat{R}(U) = \{\{x\} | x \in U\}, \overset{\vee}{R}(U) = \{U\}. \tag{7}$$

That is, $\hat{R}(U)$ is the discrete partition, $\overset{\vee}{R}(U)$ is the indiscrete partition [46].

Definition 2.6. *Let $S = (U, A, V, f)$ be an information system. $A = C \cup D$. Suppose $U/C = \{C_1, C_2, \ldots, C_n\}$ is a partition of U. A granulation of knowledge C, denoted by $GK(C)$, is defined as follows:*

$$GK(C) = \frac{1}{|U|^2} \sum_{i=1}^{m} |C_i|^2. \tag{8}$$

If $C = \hat{R}$, then a granulation of knowledge R achieves the minimum value $|U|./|U|^2 = 1/|U|$.

If $C = \overset{\vee}{R}$, then a granulation of knowledge R achieves the maximum value $|U|^2/|U|^2 = 1$.

Obviously, we have $1/|U| \le GK(C) \le 1$. The discernibility ability of knowledge is represented by the knowledge granulation. The smaller $GK(C)$ is, the stronger its discernibility ability [46].

Definition 2.7. *Let (U, R) be an approximate space. Suppose $U/R = \{C_1, C_2, \ldots, C_n\}$ is a partition of U. A measure of granularity for a partition is defined as follows:*

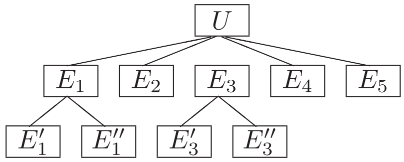

Fig. 1. The knowledge hierarchical tree.

$$G(\mathrm{U/R}) = \sum_{i=1}^{n} \frac{|C_i|}{|U|} \log_2 |C_i|, \tag{9}$$

where $|C_i|/|U|$ represents the probability of equivalence class C_i within the universe U, $\log_2 |C_i|$ is commonly known as the Hartley measure of information of the set C_i [47].

3 The Principle of Updating Equivalence Classes Dynamically

In RST, the universe is composed of objects. Different partitions are formed by different attribute sets. A is an equivalence relation on U. Then $U/A = \{[x]_A | x \in U\}$. Suppose there exist $A_1 \subset A_2 \subset A_3 \cdots \subset A_{m-1} \subset A$, where $A_i = \{a_1, a_2, \ldots, a_i\}$, then $[x]_A \subseteq [x]_{A_{m-1}} \subseteq \cdots \subseteq [x]_{A_1}$. That is, the granulation of knowledge becomes smaller while more attributes taking part in forming equivalence classes. The knowledge levels are formed when different attribute sets are utilized to partition the universe. The root of the knowledge hierarchical tree is composed of all elements of the universe U. Sub node is composed of equivalence classes, which are formed via partitions by the attribute set A_i. Each equivalence class is divided into smaller equivalence classes by adding more attributes. Suppose partitions E_1, E_2, E_3, E_4, E_5 (see Fig. 1) are formed by the attribute set A_1 on U. Every equivalence class is further divided by the attribute set A_2, then it forms partitions based on A_1 and A_2. E_1 is further divided into E_1' and E_1'', and E_3 is further divided into E_3' and E_3'', and so on. In TRS, information granules are induced by equivalence classes. It is time consuming to calculate equivalence classes. Therefore, it is important to avoid reconstruct equivalence classes in the dynamic change of information systems. Using incremental methods for updating knowledge can generally increase the efficiency of knowledge discovery in a dynamic environment. Incremental updating classes do not require recalculation of equivalence classes. New equivalence classes can be constructed according to existing equivalence classes structure and updating information. Therefore, we consider how to dynamically construct the knowledge hierarchy tree for the information system. When adding an attribute, then some equivalence classes of the current knowledge hierarchy tree are partitioned to subcategories, as a new knowledge level of the tree. When a new object x_{n+1} is added to the information system, we can traverse the knowledge hierarchical tree. If x_{n+1} does not belong to any equivalence class, then build a new equivalence class. If x_{n+1} belongs to a certain equivalence class, then x_{n+1} is emerged to that equivalence class with a bigger cardinality [8], [13]. Most parts of the knowledge hierarchy tree will keep unchanged when the information system is changed by adding or deleting an object. For example, when adding an object, a new node will be added to a knowledge hierarchy tree or nodes of the tree will be enlarged. Therefore, the knowledge hierarchy tree can be updated incrementally through incremental updating of equivalence classes w.r.t. evolution of objects.

On the other hand, the granularity of knowledge will be changed when a new object is inserted into or deleted from the universe. Let $U/R = \{E_1, E_2, \ldots, E_l\}(1 \le l \le |U|)$ be a partition of U. Assume a new object x_{n+1} is inserted into U. Let $U' = U \cup \{x_{n+1}\}$. If $x_{n+1} \in E_i$, then $E'_i = E_i \cup \{x_{n+1}\}$, $U'/R = \{E_1, E_2, \ldots, E'_i, \ldots, E_l\}$. If $x_{n+1} \notin E_i$, then $E_{l+1} = \{x_{n+1}\}$, $U'/R = \{E_1, E_2, \ldots, E_l, E_{l+1}\}$. Let $GK_i(C)$ and $G_i(\mathrm{U}'/R)$ denote the *knowledge granulation* and the *measure of granularity for a partition*, respectively, after inserting the object x_{n+1} into U. The subscript i means x_{n+1} joins E_i to be a new equivalence class. Then, the following lemmas hold.

Lemma 3.1. *If* $|E_i| \le |E_j|$, *then* $GK_i(C) \le GK_j(C)$.

Proof. $\because |E_i| \le |E_j|$, $|E'_j| = |E_j| + 1$, $|E'_i| = |E_i| + 1$, $\therefore |E'_i|^2 \le |E'_j|^2$. By Definition 2.6, we have $GK_i(C) \le GK_j(C)$. □

Lemma 3.2. *If* $|E_i| \le |E_j|$, *then* $G_i(\mathrm{U}'/R) \le G_j(\mathrm{U}'/R)$.

Proof.

$$\because |E_i| \le |E_j|, |E'_j| = |E_j| + 1, |E'_i| = |E_i| + 1, |U'| = |U| + 1,$$
$$\therefore \frac{|E'_i|}{|U| + 1} \log_2 |E'_i| \le \frac{|E'_j|}{|U| + 1} \log_2 |E'_j|.$$

By Definition 2.7, we have $G_i(\mathrm{U}'/R) \le G_j(\mathrm{U}'/R)$. □

When the object x_k is deleted from U, $\forall x_k \in U$, let $U' = U - \{x_k\}$, $E'_i = E_i - \{x_k\}$. Then $U'/R = \{E_1, E_2, \ldots, E'_i, \ldots, E_l\}$. Let $GK^i(C)$ and $G^i(\mathrm{U}'/R)$ denote the *knowledge granulation* and the *measure of granularity for a partition*, respectively, after deleting the object x_k from U. The superscript i means x_k is deleted from E_i. Then, the following lemmas hold.

Lemma 3.3. *If* $|E_i| \le |E_j|$, *then* $GK^i(C) \ge GK^j(C)$.

Proof. It is similar to the proof of Lemma 3.1. □

Lemma 3.4. *If* $|E_i| \le |E_j|$, *then* $G^i(\mathrm{U}'/R) \ge G^j(\mathrm{U}'/R)$.

Proof. It is similar to the proof of Lemma 3.1. □

Remark 1. When inserting/deleting an object into/from the universe, the *knowledge granulation* and the *measure of granularity for a partition* will change differently due to the different cardinality of the equivalence class in which the object belongs to.

4 An Incremental Method for Updating Approximations under VPRS When Objects Vary over Time

Given an information system $S = (U, C \cup D, V, f)$, X is a subset of U. Equivalence classes generated by the set of attributes A are denoted by $[x]_R = \{E_1, E_2, E_3, \ldots, E_j\}$, $1 \le j \le |U|$. The objects in U may increase or decrease. We aim to develop an approach for incremental learning rough set approximations without recalculating the old data set. In other words, on the basis of existing equivalence classes, β upper and lower approximations of the concept X are updated. Here, we only discuss the case that a single object varies. The change of multiple objects can be seen as the cumulative change of a single object. β upper and lower approximations of multiple objects can be updated step by step using the incremental updating principle of the a single object.

For convenience, let X' denote X after updating. Then β upper and lower approximations of X' are denoted as $\overline{R}_\beta(X')$ and $\underline{R}_\beta(X')$, respectively. The equivalence class E_i after updating is E'_i, and the new equivalence class is E_{j+1}.

4.1 Deletion of an Object

When object x_i is deleted from U, two cases may occur, that is, $x_i \notin X$ and $x_i \in X$.

Case 1. $x_i \notin X$. *Since* $x_i \notin X$, *then* $X' = X$. *It means that deletion of an object certainly deletes the object from a certain equivalence class* E_i, *then* $E'_i = E_i - \{x_i\}$. *In this case, the following lemma holds.*

Lemma 4.1. *If* $U' = U - x_i$, $x_i \notin X$, *then* $c(E'_i, X') < c(E_i, X)$.

Proof.

$$\because E'_i = E_i - \{x_i\}, x_i \notin X, \quad \therefore \quad c(E'_i, X')$$
$$= 1 - \frac{|E'_i \cap X'|}{|E'_i|} = 1 - \frac{|E_i \cap X|}{|E_i| - 1} < 1 - \frac{|E_i \cap X|}{|E_i|}.$$

That is, $c(E'_i, X') < c(E_i, X)$. □

The propositions for updating β upper and lower approximations are discussed respectively as follows:

Proposition 4.1. $\forall \beta (0 \le \beta < 0.5)$, *for* $\underline{R}_\beta(X')$, *we have:*

1. *If* $E_i \subseteq \underline{R}_\beta(X)$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) - \{x_i\}$.
2. *If* $E_i \not\subset \underline{R}_\beta(X)$, *then*
   a. *If* $c(E'_i, X') \le \beta$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup E'_i$;
   b. *Otherwise,* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$.

Proof.

1. $\because E_i \subseteq \underline{R}_\beta(X), \therefore c(E_i, X) \le \beta$. By Lemma 4.1,
$$c(E'_i, X') < c(E_i, X), \therefore c(E'_i, X') \le \beta,$$
$$\therefore E'_i \subseteq \underline{R}_\beta(X), \because E'_i = E_i - \{x_i\},$$
$$\therefore \underline{R}_\beta(X') = \underline{R}_\beta(X) - \{x_i\}.$$
2.
   a. $\because E_i \not\subset \underline{R}_\beta(X)$, $\therefore c(E_i, X) > \beta$. By Lemma 4.1, $c(E'_i, X') < c(E_i, X)$, $\therefore$ If $c(E'_i, X') \le \beta$, then $E'_i \in \underline{R}_\beta(X')$, $\therefore \underline{R}_\beta(X') = \underline{R}_\beta(X) \cup E'_i$;
   b. $\because E_i \not\subset \underline{R}_\beta(X), c(E'_i, X') > \beta$, $\therefore \underline{R}_\beta(X') = \underline{R}_\beta(X)$. □

Proposition 4.2. $\forall \beta (0 \le \beta < 0.5)$, *for* $\overline{R}_\beta(X')$, *we have:*

1. *If* $E_i \subseteq \overline{R}_\beta(X)$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X) - \{x_i\}$.
2. *If* $E_i \not\subset \overline{R}_\beta(X)$, *then*
   a. *If* $c(E'_i, X') < 1 - \beta$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup E'_i$;
   b. *Otherwise,* $\overline{R}_\beta(X') = \overline{R}_\beta(X)$.

Proof.

1. $\because E_i \subseteq \overline{R}_\beta(X), \therefore c(E_i, X) \leq 1-\beta$. By Lemma 4.1,

$$c(E_i', X') < c(E_i, X),\ \therefore c(E_i', X') \leq 1-\beta,$$
$$\therefore E_i' \subseteq \overline{R}_\beta(X).\ \because E_i' = E_i - \{x_i\},$$
$$\therefore \overline{R}_\beta(X') = \overline{R}_\beta(X) - \{x_i\}.$$

2.

   a. $\because E_i \not\subset \underline{R}_\beta(X), \therefore c(E_i, X) \geq 1-\beta$. By Lemma 4.1, $c(E_i', X') < c(E_i, X)$, $\therefore$ If $c(E_i', X') < 1-\beta$, then $E_i' \subseteq \overline{R}_\beta(X')$,

$$\therefore \overline{R}_\beta(X') = \overline{R}_\beta(X) \cup E_i';$$

   b.

$$\because E_i \not\subset \overline{R}_\beta(X), c(E_i', X') \geq 1-\beta,$$
$$\therefore \overline{R}_\beta(X') = \overline{R}_\beta(X).$$

□

Case 2. $x_i \in X$. *Since* $x_i \in X$, *then* $X' = X - \{x_i\}$, $E_i' = E_i - \{x_i\}$.

In this case, the following lemma holds.

Lemma 4.2. *If* $U' = U - x_i$, $x_i \in X$, *then* $c(E_i', X') > c(E_i, X)$.

Proof. $\because X' = X - \{x_i\}$, $E_i' = E_i - \{x_i\}$, $\therefore c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|} = 1 - \frac{|E_i \cap X| - 1}{|E_i| - 1} > 1 - \frac{|E_i \cap X|}{|E_i|}$, that is, $c(E_i', X') > c(E_i, X)$. □

Then, the propositions for updating β upper and lower approximations are given, respectively, as follows:

Proposition 4.3. $\forall \beta (0 \leq \beta < 0.5)$, *for* $\underline{R}_\beta(X')$, *we have*

1. *If* $E_i \subseteq \underline{R}_\beta(X)$,
   a. *If* $c(E_i', X') \leq \beta$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) - \{x_i\}$;
   b. *Otherwise,* $\underline{R}_\beta(X') = \underline{R}_\beta(X) - E_i$.
2. *If* $E_i \not\subset \underline{R}_\beta(X)$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$.

Proof. It is similar to the proof of Proposition 4.1. □

Proposition 4.4. $\forall \beta (0 \leq \beta < 0.5)$, *for* $\overline{R}_\beta(X')$, *we have*

1. *If* $E_i \subseteq \overline{R}_\beta(X)$, *then*
   a. *If* $c(E_i', X') < 1-\beta$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X) - \{x_i\}$;
   b. *Otherwise,* $\overline{R}_\beta(X') = \overline{R_\beta}(X) - E_i$.
2. *If* $E_i \not\subset \underline{R}_\beta(X)$, *then* $\overline{R}_\beta(X') = \overline{R_\beta}(X)$.

Proof. It is similar to the proof of Proposition 4.2. □

According to the principle of deleting an object from the information system, we will present the following algorithm for updating approximations of a concept under VPRS.

Algorithm 4.1. An incremental algorithm for updating upper and lower approximations of a concept under VPRS when deleting an object from the universe.

Input: $U/R = \{E_1, E_2, E_3, \ldots, E_j\}, 1 \leq j \leq |U|, \underline{R_\beta}(X), \overline{R_\beta}(X), V_a, \beta, x_i$.

Output: $\underline{R_\beta}(X'), \overline{R_\beta}(X')$.

if $x_i \in E_i$ **then**
  $E_i' = E_i - \{x_i\}$;
end if
if $E_i = \emptyset$ **then**
  update V_a;
end if
if $f(x_i, d) \neq f(x_k, d), \forall x_k \in X, \forall d \in D$ **then**
  if $E_i \subseteq \underline{R}_\beta(X)$ **then**
    $\underline{R}_\beta(X') = \underline{R}_\beta(X) - \{x_i\}, \overline{R}_\beta(X') = \overline{R}_\beta(X) - \{x_i\}$;
  else
    $c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|}$;
    if $c(E_i', X') \leq \beta$ **then**
      $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup E_i'$;
    else
      $\underline{R}_\beta(X') = \underline{R}_\beta(X)$;
    end if
    if $c(E_i', X') < 1-\beta$ **then**
      $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup E_i'$;
    else
      $\overline{R}_\beta(X') = \overline{R}_\beta(X)$;
    end if
  end if
else
  $X' = X - \{x_i\}$;
  if $E_i \not\subset \underline{R}_\beta(X)$ **then**
    $\underline{R}_\beta(X') = \underline{R}_\beta(X), \overline{R}_\beta(X') = \overline{R_\beta}(X)$;
  else
    $c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|}$;
    if $c(E_i', X') \leq \beta$ **then**
      $\underline{R}_\beta(X') = \underline{R}_\beta(X) - \{x_i\}$;
    else
      $\underline{R}_\beta(X') = \underline{R}_\beta(X) - E_i$;
    end if
    if $c(E_i', X') < 1-\beta$ **then**
      $\overline{R}_\beta(X') = \overline{R}_\beta(X) - \{x_i\}$;
    else
      $\overline{R}_\beta(X') = \overline{R_\beta}(X) - E_i$;
    end if
  end if
end if
return $\underline{R_\beta}(X'), \overline{R_\beta}(X')$;

4.2 Insertion of a New Object

Let $U' = U \cup \{x_{n+1}\}$ denote the universe after insertion of a new object x_{n+1}. Let V' denote the domain of U'. When inserting a new object x_{n+1} into U, two cases may occur w.r.t. the change of the domain. That is, $V' \supset V$ and $V' = V$. In each case, two cases may occur, that is, $x_{n+1} \notin X'$ (It means that $\forall x_k \in X, \forall d \in D, f(x_{n+1}, d) \neq f(x_k, d), 1 \leq k \leq n$) and $x_{n+1} \in X'$ (It means that $\forall x_k \in X, \forall d \in D, f(x_{n+1}, d) = f(x_k, d), 1 \leq k \leq n$).

Case 1. $V' \supset V$. *In this case, since* $f(x_{n+1}, a) \notin V_a (\forall a \in C), x_{n+1}$ *does not belong to any equivalence classes. It forms a new equivalence class. Two cases, that is,* $x_{n+1} \notin X'$ *and* $x_{n+1} \in X'$ *will occur.*

Proposition 4.5. *If* $f(x_{n+1}, d) \neq f(x_k, d)$, *for* $\underline{R}_\beta(X')$ *and* $\overline{R}_\beta(X')$, *we have:*

1. *If* $x_{n+1} \notin X'$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$, $\overline{R}_\beta(X') = \overline{R}_\beta(X)$.
2. *If* $x_{n+1} \in X'$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$, $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$.

Proof. It follows directly by Definition 2.4. □

Case 2. $V' = V$. *In this case, we also consider the following two subcases.*

1) $x_{n+1} \notin X'$, $X' = X$.

If x_{n+1} *does not belong to any equivalence classes, it becomes a new equivalence class. Because it is not included in* X, *it cannot affect* β *upper and lower approximations. That is,* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$, $\overline{R}_\beta(X') = \overline{R}_\beta(X)$. *If* x_{n+1} *joins a certain equivalence class, we assume that* $E_i' = E_i \cup \{x_{n+1}\}$. *Then, the following lemma holds.*

Lemma 4.3. *If* $V' = V, x_{n+1} \notin X', E_i' = E_i \cup \{x_{n+1}\}$, *then* $c(E_i', X') > c(E_i, X)$.

Proof. $\because |E_i'| > |E_i|$, $|E_i' \cap X'| = |E_i \cap X|$, $\therefore c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|} = 1 - \frac{|E_i \cap X|}{|E_i| + 1} > 1 - \frac{|E_i \cap X|}{|E_i|} = c(E_i, X)$. □

That is, the *relative degree of misclassification* increases.

The following two propositions are to update β upper and lower approximations.

Proposition 4.6. $\forall\beta(0 \le \beta < 0.5)$, *for* $\underline{R}_\beta(X')$, *we have*

1. *If* $E_i \subseteq \underline{R}_\beta(X)$, *then*
   a. *If* $c(E_i', X') \le \beta$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$;
   b. *Otherwise,* $\underline{R}_\beta(X') = \underline{R}_\beta(X) - E_i$.
2. *If* $E_i \not\subset \underline{R}_\beta(X)$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$.

Proof. It is similar to the proof of Proposition 4.1. □

Proposition 4.7. $\forall\beta(0 \le \beta < 0.5)$, *for* $\overline{R}_\beta(X')$, *we have*

1. *If* $E_i \subseteq \overline{R}_\beta(X)$, *then*
   a. *If* $c(E_i', X') < 1 - \beta$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$;
   b. *Otherwise,* $\overline{R}_\beta(X') = \overline{R}_\beta(X) - E_i$.
2. *If* $E_i \not\subset \overline{R}_\beta(X)$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X)$.

Proof. It is similar to the proof of Proposition 4.2. □

2) $x_{n+1} \in X'$, $X' = X \cup \{x_{n+1}\}$.

If a new equivalence class E_i' is generated because of the insertion of x_{n+1}, then $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$, $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$.

If the cardinality of a certain equivalence class is enlarged because of the insertion of x_{n+1}, namely, $E_i' = E_i \cup \{x_{n+1}\}$, then the following lemma holds.

Lemma 4.4. *If* $V' = V, x_{n+1} \in X', E_i' = E_i \cup \{x_{n+1}\}$, *then* $c(E_i', X') < c(E_i, X)$.

Proof. $\because x_{n+1} \in X'$, $E_i' = E_i \cup \{x_{n+1}\}$, $\therefore c(E_i', X') = (1 - \frac{|E_i' \cap X'|}{|E_i'|}) = 1 - \frac{|E_i \cap X| + 1}{|E_i| + 1} < 1 - \frac{|E_i \cap X|}{|E_i|} = c(E_i, X)$. □

That is, a *relative degree of misclassification* decreases.

Proposition 4.8. $\forall\beta(0 \le \beta < 0.5)$, *for* $\underline{R}_\beta(X')$ *and* $\overline{R}_\beta(X')$, *we have:*

1. *If* $E_i \subseteq \underline{R}_\beta(X)$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$, $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$.
2. *If* $E_i \not\subset \underline{R}_\beta(X)$, *then*
   a. *If* $c(E_i', X') \le \beta$, *then* $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup E_i'$; *Otherwise,* $\underline{R}_\beta(X') = \underline{R}_\beta(X)$.
   b. *If* $c(E_i', X') < 1 - \beta$, *then* $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup E_i'$; *Otherwise,* $\overline{R}_\beta(X') = \overline{R}_\beta(X)$.

Proof. It is similar to the proof of Propositions 4.1 and 4.2. □

Obviously, these propositions can be used to update β upper and lower approximations. Next, we will present an algorithm for updating approximations of a concept under VRPS on the case of insertion of an object into the universe.

Algorithm 4.2. An incremental algorithm for updating upper and lower approximations of a concept under VPRS when inserting an object into the universe.

Input: $U/R = \{E_1, E_2, E_3, \ldots, E_j\}, 1 \le j \le |U|$, $\underline{R_\beta}(X), \overline{R_\beta}(X), V_a(\forall a \in A), \beta, x_{n+1}$.

Output: $\underline{R_\beta}(X'), \overline{R_\beta}(X')$.

if $f(x_{n+1}, a) \notin V_a(\forall a \in C)$ **then**
  $V_a = V_a \cup \{f(x_{n+1}, a)\}$; $E_{j+1} = \{x_{n+1}\}$;
  if $f(x_{n+1}, d) \ne f(x_k, d), \forall x_k \in X, \forall d \in D$ **then**
    $\underline{R}_\beta(X') = \underline{R}_\beta(X), \overline{R}_\beta(X') = \overline{R}_\beta(X)$;
  else
    $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$;
    $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$;
  end if
else
  if $f(x_{n+1}, d) \ne f(x_k, d)$, $\forall x_k \in X$, $\forall d \in D$ **then**
    if $x_{n+1} \notin E_j$, $1 \le j \le |U|$ **then**
      $E_{j+1} = \{x_{n+1}\}$;
      $\underline{R}_\beta(X') = \underline{R}_\beta(X)$;
      $\overline{R}_\beta(X') = \overline{R}_\beta(X)$
    else
      $E_i = E_i \cup \{x_{n+1}\}$; $c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|}$;
      if $E_i \subseteq \underline{R}_\beta(X)$ **then**
        if $c(E_i, X') \le \beta$ **then**
          $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$;
        else
          $\underline{R}_\beta(X') = \underline{R}_\beta(X) - E_i$;
        end if
      else
        $\underline{R}_\beta(X') = \underline{R}_\beta(X)$;
      end if
      if $E_i \subseteq \overline{R}_\beta(X)$ **then**
        if $c(E_i', X') < 1 - \beta$ **then**
          $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$;
        else
          $\overline{R}_\beta(X') = \overline{R}_\beta(X) - E_i$;
        end if
      else
        $\overline{R}_\beta(X') = \overline{R}_\beta(X)$;
      end if
    end if
  else
    if $x_{n+1} \notin E_j$, $1 \le j \le |U|$ **then**
      $E_{j+1} = \{x_{n+1}\}$;
      $\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$;
      $\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$;

TABLE 1
An Information System

U	a_1	a_2	a_3	a_4	d
x_1	1	0	1	2	0
x_2	1	0	0	1	0
x_3	1	0	1	2	1
x_4	2	3	3	1	0
x_5	1	0	1	2	0
x_6	1	0	0	1	0
x_7	2	3	3	1	1
x_8	1	0	0	1	1
x_9	1	1	0	1	1

else
$E_i = E_i \cup \{x_{n+1}\}$;
if $E_i \subseteq \underline{R}_\beta(X)$ **then**
$\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup \{x_{n+1}\}$;
$\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup \{x_{n+1}\}$;
else
$c(E_i', X') = 1 - \frac{|E_i' \cap X'|}{|E_i'|}$;
if $c(E_i', X') \leq \beta$ **then**
$\underline{R}_\beta(X') = \underline{R}_\beta(X) \cup E_i'$;
else
$\underline{R}_\beta(X') = \underline{R}_\beta(X)$;
end if
if $c(E_i, X') < 1 - \beta$ **then**
$\overline{R}_\beta(X') = \overline{R}_\beta(X) \cup E_i'$;
else
$\overline{R}_\beta(X') = \overline{R}_\beta(X)$;
end if
end if
end if
end if
end if
return $\underline{R}_\beta(X')$, $\overline{R}_\beta(X')$.

5 A Numerical Example

Given an information system $S = (U, A, V, f)$ in Table 1, where $U = \{x_1, x_2, \ldots, x_9\}$, $A = C \cup D$, $C = \{a_1, a_2, a_3, a_4\}$, $D = \{d\}$. $V_{a_1} = \{1, 2\}, V_{a_2} = V_{a_3} = \{0, 1, 3\}, V_{a_4} = \{1, 2\}$. Then, $U/D = \{E_{d1}, E_{d2}\}$, $E_{d1} = \{x_1, x_2, x_4, x_5, x_6\}(f(x_k, d) = 0)$, $E_{d2} = \{x_3, x_7, x_8, x_9\}(f(x_k, d) = 1)$. Let $X = E_{d1}$, $\beta = 0.4$.

Now, we consider the following two cases:

1. An object x_{10} is inserted into the information system, see Table 2.
2. The object x_6 is deleted from information system, see Table 3.

For the information system before updating, equivalence classes on R and β approximations are listed as follows:

$U/R = \{E_1, E_2, E_3\}$, $E_1 = \{x_1, x_3, x_5\}$, $E_2 = \{x_2, x_6, x_8\}$, $E_3 = \{x_4, x_7\}$, $E_4 = \{x_9\}$, $\underline{R}_{0.4}(X) = \{E_1, E_2\}$, $\overline{R}_{0.4}(X) = \{E_3, E_1, E_2\}$. Then, we consider the following two cases.

1. When x_{10} is inserted into the information system, the equivalence classes and β upper and lower approximations of the concept X are updated as follows:

TABLE 2
The Information System after Insertion of an Object

U	a_1	a_2	a_3	a_4	d
x_1	1	0	1	2	0
x_2	1	0	0	1	0
x_3	1	0	1	2	1
x_4	2	3	3	1	0
x_5	1	0	1	2	0
x_6	1	0	0	1	0
x_7	2	3	3	1	1
x_8	1	0	0	1	1
x_9	1	1	0	1	1
x_{10}	1	0	1	2	1

a. $\because f(x_{10}, d) \neq f(x_k, d), \forall x_k \in X, \therefore X' = X$.
b. $E_1' = E_1 \cup \{x_{10}\} = \{x_1, x_3, x_5, x_{10}\}$.
c.

$$\because x_{10} \notin X, E_1 \subseteq \underline{R}_{0.4}(X), c(E_1', X') = 1 - \frac{|E_1' \cap X'|}{|E_1'|} = 0.5 > \beta = 0.4, \therefore \underline{R}_{0.4}(X') = \underline{R}_{0.4}(X) - E_1 = E_2.$$

d. $\because c(E_1', X') = 1 - \frac{|E_1' \cap X'|}{|E_1'|} = 0.5 < 1 - \beta = 0.6$, $\therefore \overline{R}_{0.4}(X') = \overline{R}_{0.4}(X) \cup \{x_{10}\} = \{E_3, E_1', E_2\}$.

2. When x_6 is deleted from the information system, the equivalence classes and β upper and lower approximations of the concept X are computed as follows:

a. $E_2' = E_2 - \{x_6\} = \{x_2, x_8\}$.
b. $\because f(x_6, d) = f(x_k, d)(\forall x_k \in X)$,

$$\therefore X' = X - \{x_6\} = \{x_1, x_2, x_4, x_5\}.$$

c.

$$\because E_2 \subseteq \underline{R}_{0.4}(X), c(E_2', X') = 1 - \frac{|E_2' \cap X'|}{|E_2'|} = 0.5 > \beta = 0.4, \therefore \underline{R}_{0.4}(X') = \underline{R}_{0.4}(X) - E_2 = E_1.$$

TABLE 3
The Information System after Deletion of an Object

U	a_1	a_2	a_3	a_4	d
x_1	1	0	1	2	0
x_2	1	0	0	1	0
x_3	1	0	1	2	1
x_4	2	3	3	1	0
x_5	1	0	1	2	0
x_7	2	3	3	1	1
x_8	1	0	0	1	1
x_9	1	1	0	1	1

TABLE 4
Data Sets

ID	Data sets	Number of rows	Number of attributes
1	Zoo	101	18
2	Hayes	132	6
3	Wine	178	14
4	Flags	194	30
5	Computer Hardware	209	10
6	Liver Disorders	345	7
7	Meta	528	22
8	balance	625	5
9	Solar Flare	1389	13
10	Contraceptive Method Choice	1473	10
11	Car Evaluation	1728	7

d.

$$\because E_2 \subseteq \underline{R}_{0.4}(X), c(E_2', X') = 1 - \frac{|E_2' \cap X'|}{|E_2'|} = 0.5 < 1 - \beta = 0.6, \therefore \overline{R}_{0.4}(X) = \overline{R}_{0.4}(X') - \{x_6\} = \{E_3, E_1, E_2'\}.$$

6 Experimental Evaluation

Extensive experiments have been carried out to verify the effectiveness of the algorithms. We have selected data sets from UCI in Table 4, publicly available from the UC Irvine Machine Learning Database Repository (www.ics.uci.edu/~mlearn/MLRepository.html), as benchmark databases for performance tests.

The algorithms were developed in $C\#$. Experiments were performed on a computer with 1.66 GHz CPU, 1.0 GB of memory, running Microsoft Windows XP Professional and Microsoft SQL Server 2000. Since the proposed incremental updating methods are limited in the complete information system, we deleted data with null attribute value from data sets in Table 4. The object is selected randomly from data sets when deleting the object from the system. The values of the object are randomly selected when inserting the object into the system.

The computational times are simple arithmetic average computational time of all steps of insertion or deletion. The experimental results are depicted in Figs. 2 and 3, respectively.

In Figs. 2 and 3, the numbers on the horizontal axis are the serial number of the different data sources listed in Table 4. The vertical axis is the computational time of different data sets under different conditions. As shown in Table 4, the objects and attributes of different data sets are different. The objects include in each data set increase with the raise of the serial number. So from left to right the serial number of data on the horizontal axis raises imply the number of objects in data source increases.

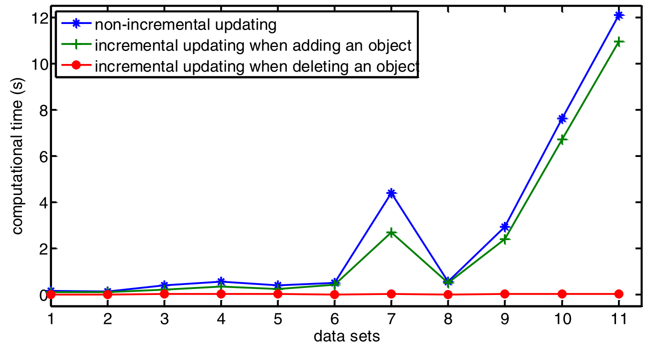

Fig. 2. A comparison of incremental updating approximations and nonincremental updating approximaitons when an object inserting into and deleting from the universe.

In Fig. 2, we compare the computational time between incremental updating and nonincremental updating methods under different conditions for different data sets. The star lines denote the computational time of nonincremental updating methods on different data sets; the plus lines denote the computational time of incremental updating methods on different data sets when inserting an object into the universe; the dot lines denote the computational time of incremental updating method on different data sets when deleting an object from the universe. From Fig. 2, the incremental updating method outperforms that of nonincremental updating method both in the case of insertion and deletion of an object into the universe. Moreover, the computational time of incremental updating approximations increases with an increasing number of rows. In addition, the computational time of data sets 4 and 7 (especially data set 7) are longer than that of neighbor data sets. This is because the attribute numbers in data sets 4 and 7 are much bigger than those of other data sets from Table 4. The computational time of incremental updating method on the case of deletion of an object from the universe is less than that of insertion an object.

In Fig. 3, we compare the computational time of incremental updating when inserting an object with a new value with inserting an object with a value within the original domain of the attribute. When inserting a new object with a new value into the universe, the object becomes a new equivalence class. So it does not need to

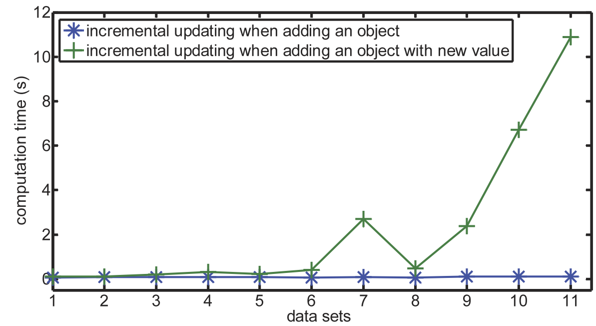

Fig. 3. A comparison of incremental updating approximations computation time between inserting an object into the universe with a new value and the case with no new value.

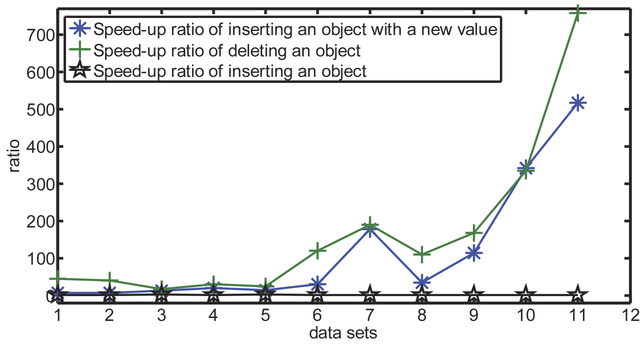

Fig. 4. Speed-up ratios of incremental updating approximation in different conditions.

find which equivalence class it belongs to or to compute a relative error rate. Consequently, its computational time is far less than that of inserting the new object for which the value is within the attribute domain. Fig. 4 shows the speed-up ratios of incremental updating methods in different conditions (i.e., deleting an object, inserting an object without changing the domain of the attribute, and inserting an object with a new value).

The time of computing approximations includes two parts, namely, the time of computing equivalence classes and a comparison of equivalence classes with X. Suppose $S = (U, A, V, f)$ is the information system. $|U| = n$, $|A| = m$, then the computational complexity of computing equivalence classes is $O(m * n)$. If $|U/R| = k$, then the computational complexity of computing approximations is $O(m * n + k)$. The computational time of incremental updating approximations includes two parts when inserting an object into the universe, i.e., the time needed to update equivalence classes and the time used to update approximations. The computational complexity of updating equivalence classes is $O(k * m)$. The total computational complexity of incremental updating is $O(k * m + 1)$. The computational time of incremental updating approximations also includes two parts when deleting an object from the universe, i.e., the time needed to update equivalence class and the time used to update approximations too. The total computational complexity of incremental updating is $O(k + 1)$. Therefore, the computational time of incremental updating methods is less than that of nonincremental updating methods. The time of incremental updating for inserting of an object is more than that of deleting of an object since the computational complexity of inserting of an object is more than that of deleting of an object.

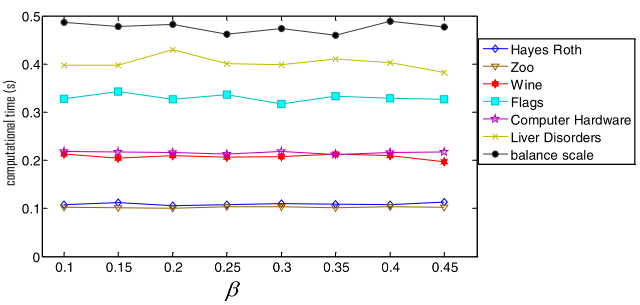

Fig. 5. A comparison of incremental updating approximations time when an object is inserted into the universe with different β.

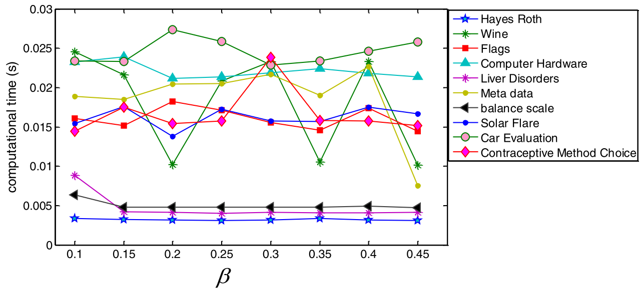

Fig. 6. A comparison of incremental updating approximations time when an object is deleted from the universe with different β.

In addition, since the *relative degree of misclassification* is introduced in $VPRS$, it allows a certain degree of error in the classification. May the variation of β affect the computational time of updating VPRS approximations? We evaluate it on the data sets with different β when deleting an object from the universe or inserting an object into the universe. The evaluation results are shown in Figs. 5 and 6, respectively.

Figs. 5 and 6 depict the variation of computational time of incremental updating when an object is inserted into or deleted from different data sets with different β values, respectively. Clearly, the computational time fluctuates a little with different β when inserting the object into the universe or deleting the object from the universe. There is no rule between the variation of β and the computational time of incremental updating approximations since the complexity of computation is almost as the same as in the case of different β except for some random factors.

7 Conclusions

In this paper, we proposed incremental methods for updating approximations under VPRS when the information system is updated by inserting or deleting an object. The methods are the first effort to efficiently update approximations. After having discussed the principle of updating equivalence classes, we proposed the incremental methods for updating approximations under VPRS in terms of inserting or deleting an object. By considering the change of a concept and the attribute domain we discussed the alteration of knowledge granulation w.r.t. the variation of data sets. At last, we carried out extensive experiments to verify the effectiveness of the proposed algorithms. Experimental results show that the algorithms are effective to maintain knowledge when the object set in the information system varies over time. One of our future study is to consider the objects to be added/deleted as a whole and develop algorithms for updating approximations under VPRS. Another future study will focus on an extension of the algorithms to DTRS, incomplete information systems and other extended rough sets models. The variation of attributes in the information system may further be taken into consideration in terms of incremental updating knowledge.

Acknowledgments

This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117) and the

Fundamental Research Funds for the Central Universities (SWJTU11ZT08). In addition, thanks to Xiong Wu and Xiaolan Ji for their early contributions to this work.

REFERENCES

[1] L.A. Zadeh, "Towards a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic," *Fuzzy Sets and Systems,* vol. 90, no. 2, pp. 111-127, 1997.

[2] L.A. Zadeh, "Fuzzy Logic=Computing with Words," *IEEE Trans. Fuzzy Systems,* vol. 4, no. 1, pp. 103-111, Feb. 1996.

[3] J.T. Yao, "Information Granulation and Granular Relationships," *Proc. IEEE Conf. Granular Computing (GrC),* pp. 326-329, 2005.

[4] J.T. Yao and Y.Y. Yao, "A Granular Computing Approach to Machine Learning," *Proc. First Int'l Conf. Fuzzy Systems and Knowledge Discovery,* pp. 732-736, 2002.

[5] Y.Y. Yao and N. Zhong, "Potential Applications of Granular Computing in Knowledge Discovery and Data Mining," *Proc. World Multiconf. Systemics, Cybernetics and Informatics,* pp. 573-580, 1999.

[6] Y.Y. Yao, "Information Granulation and Rough Set Approximation," *Int'l J. Intelligent Systems,* vol. 16, no. 1, pp. 87-104, 2001.

[7] Y.Y. Yao, "Perspectives of Granular Computing," *Proc. IEEE Int'l Conf. Granular Computing (GrC),* pp. 85-90, 2005.

[8] Y.Y. Yao, "Interpreting Concept Learning in Cognitive Informatics and Granular Computing," *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics,* vol. 39, no. 4, pp. 855-866, Aug. 2009.

[9] T.Y. Lin, "Neighborhood Systems-A Qualitative Theory for Fuzzy and Rough Sets," *Advances in Machine Intelligence and Soft Computing,* vol. 4, pp. 132-155, 1997.

[10] T.Y. Lin, "Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems," *Rough Sets in Knowledge Discovery,* pp. 107-121, Physica-Verlag, 1998.

[11] A. Bargiela and W. Pedrycz, *Granular Computing: An Introduction.* Kluwer Academic Publishers, 2002.

[12] A. Skowron and J. Stepaniuk, "Information Granules: Towards Foundations of Granular Computing," *Int'l J. Intelligent Systems,* vol. 16, pp. 57-85, 2001.

[13] B. Apolloni, W. Pedrycz, S. Bassis, and D. Malchiodi, *The Puzzle of Granular Computing.* Springer, 2008.

[14] W.Z. Wu, Y. Leung, and J.S. Mi, "Granular Computing and Knowledge Reduction in Formal Contexts," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 10, pp. 1461-1474, Oct. 2009.

[15] H.O. Ghaffari, M. Sharifzadeh, K. Shahriar, and W. Pedrycz, "Application of Soft Granulation Theory to Permeability Analysis," *Int'l J. Rock Mechanics and Mining Sciences,* vol. 46, no. 3, pp. 577-589, 2009.

[16] Y.Y. Yao, "Granular Computing: Basic Issues and Possible Solutions," *Proc. Fifth Joint Conf. Information Sciences,* pp. 186-189, 2000.

[17] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of Granular Computing.* Wiley-Interscience, 2008.

[18] Z. Pawlak, "Rough Sets," *Int'l J. Computer and Information Sciences,* vol. 11, pp. 341-356, 1982.

[19] Z. Pawalk, *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, 1991.

[20] W. Pedrycz, "A Dynamic Data Granulation through Adjustable Fuzzy Clustering," *Pattern Recognition Letters,* vol. 29, no. 16, pp. 2059-2066, 2008.

[21] W. Pedrycz and R. Weber, "Special Issue on Soft Computing for Dynamic Data Mining," *Applied Soft Computing,* vol. 8, no. 4, pp. 1281-1282, 2008.

[22] T.C. Chou and M.C. Chen, "Using Incremental PLSI for Threshold-Resilient Online Event Analysis," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 3, pp. 289-299, Mar. 2008.

[23] J.M. Kang, M.F. Mokbel, S.S. Shekhar, T. Xia, and D. Zhang, "Incremental and General Evaluation of Reverse Nearest Neighbors," *IEEE Trans. Knowledge and Data Eng.,* vol. 22, no. 7, pp. 983-998, July 2010.

[24] F. Altiparmak, E. Tuncel, and H. Ferhatosmanoglu, "Incremental Maintenance of Online Summaries over Multiple Streams," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 2, pp. 216-229, Feb. 2008.

[25] A. An, N. Shan, C. Chan, N. Cercone, and W. Ziarko, "Discovering Rules for Water Demand Prediction: An Enhanced Rough-Set Approach," *Eng. Application and Artificial Intelligence,* vol. 9, no. 6, pp. 645-653, 1996.

[26] T.R. Li, D. Ruan, W. Geert, J. Song, and Y. Xu, "A Rough Sets Based Characteristic Relation Approach for Dynamic Attribute Generalization in Data Mining," *Knowledge-Based Systems,* vol. 20, no. 5, pp. 485-494, 2007.

[27] N. Shan and W. Ziarko, "Data-Based Acquisition and Incremental Modification of Classification Rules," *Computational Intelligence,* vol. 11, no. 2, pp. 357-370, 1995.

[28] W.C. Bang and B. Zeungnam, "New Incremental Learning Algorithm in the Framework of Rough Set Theory," *Int'l J. Fuzzy Systems,* vol. 1, no. 1, pp. 25-36, 1999.

[29] L.Y. Tong and L.P An, "Incremental Learning of Decision Rules Based on Rough Set Theory," *Proc. Fourth World Congress on Intelligent Control and Automation,* pp. 420-425, 2002.

[30] Y. Liu, C.F. Xu, and Y.H. Pan, "A Parallel Approximate Rule Extracting Algorithm Based on the Improved Discernibility Matrix," *Rough Sets and Current Trends in Computing,* vol. 3066, pp. 498-503, Springer, 2004.

[31] S. Guo, Z.Y. Wang, Z.C. Wu, and H.P. Yan, "A Novel Dynamic Incremental Rules Extraction Algorithm Based on Rough Set Theory," *Proc. Fourth Int'l Conf. Machine Learning and Cybernetics,* pp. 1902-1907, 2005.

[32] Z. Zheng and G.Y. Wang, "RRIA: A Rough Set and Rule Tree Based Incremental Knowledge Acquisition Algorithm," *Fundamenta Informaticae,* vol. 59, nos. 2/3, pp. 299-313, 2004.

[33] T.L. Tseng, "Quantitative Approaches for Information Modeling," PhD Dissertation, Univ. of Iowa, 1999.

[34] Y.N. Fan, T.L. Tseng, C.C. Chern, and C.C. Huang, "Rule Induction Based on an Incremental Rough Set," *Expert Systems with Applications,* vol. 36, no. 9, pp. 11439-11450, 2009.

[35] H.M. Chen, T.R. Li, S.J. Qiao, and D. Ruan, "A Rough Sets Based Dynamic Maintenance Approach for Approximations in Coarsening and Refining Attribute Values," *Int'l J. Intelligent Systems,* vol. 25, no. 10, pp. 1005-1026, 2010.

[36] Y.Y. Yao, "Probabilistic Rough Set Approximations," *Int'l J. Approximate Reasoning,* vol. 49, no. 2, pp. 255-271, 2008.

[37] Y.Y. Yao and S.K.M. Wong, "A Decision Theoretic Framework for Approximating Concepts," *Int'l J. Man-Machine Studies,* vol. 37, no. 6, pp. 793-809, 1992.

[38] W. Ziarko, "Variable Precision Rough Set Model," *J. Computer and System Sciences,* vol. 46, no. 1, pp. 39-59, 1993.

[39] W. Ziarko, "Analysis of Uncertain Information in Framework of Variable Precision Rough Set," *Foundation of Computing and Decision Sciences,* vol. 18, pp. 381-396, 1993.

[40] G. Xie, J.L. Zhang, K.K. Lai, and L. Yu, "Variable Precision Rough Set for Group Decision-Making: An Application," *Int'l J. Approximate Reasoning,* vol. 49, no. 2, pp. 331-343, 2008.

[41] M. Ningler, G. Stockmanns, G. Schneider, H.D. Kochs, and E. Kochs, "Adapted Variable Precision Rough Set Approach for EEG Analysis," *Artificial Intelligence in Medicine,* vol. 47, no. 3, pp. 239-261, 2009.

[42] T.P. Hong, T.T. Wang, and S.L. Wang, "Mining Fuzzy β-Certain and β-Possible Rules from Quantitative Data Based on the Variable Precision Rough-Set Model," *Expert Systems with Applications,* vol. 32, no. 1, pp. 223-232, 2007.

[43] L. Wang, Y. Wu, and G.Y. Wang, "An Incremental Rule Acquisition Algorithm Based on Variable Precision Rough Set Model," *J. Chongqing Univ. of Posts and Telecomm. (Natural Science),* pp. 709-713, vol. 17, no. 6, 2005.

[44] D. Liu, T.R. Li, D. Ruan, and W.L. Zou, "An Incremental Approach for Inducing Knowledge from Dynamic Information Systems," *Fundamenta Informaticae,* vol. 94, pp. 245-260, 2009.

[45] H.M. Chen, T.R. Li, C.X. Hu, and X.L. Ji, "An Incremental Updating Principle for Computing Approximations in Information Systems While the Object Set Varies with Time," *Proc. IEEE Int'l Conf. Granular Computing (GrC),* pp. 49-52, 2009.

[46] J.Y. Liang and Z.Z. Shi, "The Information Entropy, Rough Entropy and Knowledge Granulation in Rough Set Theory," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 12, no. 1, pp. 37-46, 2004.

[47] Y.Y. Yao, "Probabilistic Approaches to Rough Sets," *Expert Systems,* vol. 20, no. 5, pp. 287-297, 2003.

Hongmei Chen received the master's degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2000. Currently, she is working toward the PhD degree in the School of Information Science Technology, Southwest Jiaotong University. Her research interests include the areas of data mining, pattern recognition, fuzzy sets, and rough sets.

Tianrui Li received the BS, MS, and PhD degrees from the Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2002, respectively. He was a postdoctoral researcher at the Belgian Nuclear Research Centre (SCK•CEN), Belgium from 2005-2006, a visiting professor at Hasselt University, Belgium, in 2008 and the University of Technology, Sydney, Australia in 2009. Currently, he is a professor and the director of the Key Lab of Cloud Computing and Intelligent Technology, Southwest Jiaotong University, China. His research interests include data mining and knowledge discovery, granular computing and rough sets, cloud computing. Since 2000, he has coedited two textbooks, six proceedings, three special issues of international journal and published more than 90 research papers in refereed journals and conferences. He is the vice chair of IEEE CIS Chengdu Chapter and the area editor of the *International Journal of Computational Intelligence Systems (IJCIS)*. He has served as ISKE2007, ISKE2008, ISKE2009, ISKE2010, ISKE2011 program chairs, IEEE GrC 2009 program vice chair and RSKT2008, FLINS2010 organizing chairs and has been a reviewer for several leading academic journals. He is a senior member of the IEEE.

Da Ruan (M'60-'11) received the PhD degree in mathematics from Ghent University, Belgium, in 1990. He was a principal research scientist at SCK•CEN, was a postdoctoral researcher at SCK•CEN from 1991-1993, a senior researcher, and FLINS project leader at SCK•CEN from 1994-2011. His major research interests were in the areas of mathematical modeling, computational intelligence methods, uncertainty analysis, decision support systems and AI applications to information management, cost/benefit analysis, reactor control, nuclear safety, and security related fields. He served as a regional editor for Europe of the *International Journal of Intelligent Automation and Soft Computing*, coeditor-in-chief of the *International Journal of Nuclear Knowledge Management*, editor-in-chief of the *International Journal of Computational Intelligence Systems*, editor of the book series of Intelligent Information Systems and the proceedings series of Computer Engineering and Information Science, a guest professor in the Department of Applied Mathematics and Computer Science at Ghent University and in the Department of Applied Economics at Hasselt University, respectively, in Belgium, and an adjunct professor in the Faculty of Information Technology at University of Technology, Sydney, Australia. He was a member of the IEEE. He is deceased.

Jianhui Lin received the BS, MS, and PhD degrees from the Southwest Jiaotong University, Chengdu, China, in 1984, 1988, and 1997, respectively. From 1995 to 1996, he was a visiting scholar at the University of Manchester, United Kingdom. Currently, he is a professor in the Traction Power State Key Laboratory of Southwest Jiaotong University, Chengdu, China. His research interests include control technology, fault diagnosis, signal analysis and processing, and virtual instruments. He has published more than 90 articles in journals.

Chengxiang Hu received the MS degree in computer applied technology from Southwest Jiaotong University, Chengdu, China, in 2010. Currently, he is working in the School of Computer and Information Engineering, Chuzhou University, China. His research interests include intelligent information processing, granular computing, and rough sets.