Knowledge-Based Systems 73 (2015) 81–96

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Incremental updating approximations in probabilistic rough sets under the variation of attributes

Dun Liu [a,*], Tianrui Li [b], Junbo Zhang [b]

[a] School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China
[b] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

ARTICLE INFO

Article history:
Received 29 May 2014
Received in revised form 5 August 2014
Accepted 15 September 2014
Available online 2 October 2014

Keywords:
Rough sets theory
Probabilistic rough sets
Incremental learning
Updating approximations
Knowledge discovery

ABSTRACT

The attribute set in an information system evolves in time when new information arrives. Both lower and upper approximations of a concept will change dynamically when attributes vary. Inspired by the former incremental algorithm in Pawlak rough sets, this paper focuses on new strategies of dynamically updating approximations in probabilistic rough sets and investigates four propositions of updating approximations under probabilistic rough sets. Two incremental algorithms based on adding attributes and deleting attributes under probabilistic rough sets are proposed, respectively. The experiments on five data sets from UCI and a genome data with thousand attributes validate the feasibility of the proposed incremental approaches.

1. Introduction

With today's databases increase at an unprecedented rate, information systems evolve over time. Various approaches for updating knowledge incrementally are getting more and more popular. The algorithms or strategies for updating knowledge from information systems are vital in inductive machine learning and data mining.

An information system is composed by the objects, the attributes and the domain of attributes' values [30]. The literature for updating knowledge from information systems are mainly on three aspects, namely, variation of objects (instances) [2,7,19,22,23,29,31,43,46,57,58,60,62], variation of attributes (features) [4,9,20,21,26,32,59,64] and variation of attributes' values [5,6,8,24,25,27]. All these studies help decision makers to update knowledge with different viewpoints from different kinds of information systems.

By considering many databases with thousands of attributes (features), we face lots of massive challenges in real technical applications. The updating strategies and incremental algorithms under the variation of attributes are significantly affecting the knowledge updating, both in quantitative and qualitative aspects [21]. In this paper, we mainly focus on discussing scenarios of the variation of attributes.

In view of granular computing, the variation of attributes affects the granularity of the knowledge space. Yao pointed out that the two simple and commonly operators named "Zooming-in" and "Zooming-out" can describe the dynamic character while systems vary [51]. The Zooming-in operator refines the granules of the universe, like decomposing a granule into many granules; The Zooming-out operator coarsens the granules of the universe by omitting some details of the problem, such as combining many granules to form a new granule [51]. The granularity becomes refining when adding the attributes. On the contrary, the granularity becomes coarsening when deleting the attributes. Qian et al. proposed a multi-granulation rough set model and discussed several important measures in both complete information systems [40] and incomplete information systems [41]. Furthermore, Hu et al. unitized the granularity to the neighborhood rough set for heterogeneous feature subset selection. They further provided useful properties and prefect experimental results on variation of feature numbers [14,15].

Rough set theory (RST) is one special case of granular computing [34]. In RST, the variation of attributes, includes adding and deleting of attributes, affects reducts and approximations of a concept in information systems. For incremental attribute reduction approaches, Qian et al. investigated an accelerator for attribute reduction in positive approximation [42]. Hu et al. proposed an incremental attribute reduction based on elementary sets [13]. For incremental approaches of updating approximations in RST, Chan discussed an incremental approach for updating approximations of a concept when adding or deleting an attribute in a

* Corresponding author.
E-mail addresses: newton83@163.com (D. Liu), trli@swjtu.cn (T. Li), JunboZhang86@163.com (J. Zhang).

http://dx.doi.org/10.1016/j.knosys.2014.09.008
0950-7051/

complete information system by using the lower and upper boundaries [3,4]. Li et al. presented a method for updating approximations of a concept in an incomplete information system under the characteristic relation when the attribute set varies over time [21]. Cheng proposed an incremental model for fast computing the rough fuzzy approximations [9]. Li et al. investigated an approach for incremental updating approximations in dominance-based rough sets under the variation of the attribute sets [19]. Luo et al. discussed an approach for incremental updating approximations in set-valued ordered decision systems under the attribute generalization [32]. Zhang et al. proposed a rough sets based matrix approaches to compute the incremental updating approximations [58], and further considered a parallel method for computing rough set approximations for massive data analysis [60,61,63]. Davide summarized the classification of dynamics in rough sets with synchronic cases and diachronic cases. He further discussed dynamic in information systems, approximation spaces and coverings, which provided a basic framework on dynamic studies of rough sets [10,11].

As a generalized model of RST, probabilistic rough sets (PRS) allow a flexible approximation boundary region by using the threshold parameters with a better tolerance ability for inconsistent data [65,67]. The model of PRS generalizes the restrictive definition of the lower and upper approximations by allowing certain acceptable levels of errors. A pair of threshold parameters define the lower and upper approximations [52,53]. By considering the PRS leads to a new direction of research and applications on one hand and some confusions and inconsistencies on the other [53], this paper focuses on development of the incremental approach for updating approximations in PRS under the variation of attributes.

The rest of the paper is organized as follows: Section 2 provides basic concepts of Pawlak rough sets and PRS. The related propositions, strategies and algorithms for incremental learning knowledge in PRS are presented when attributes vary in Section 3. Section 4 illustrates the proposed approach with experiments under five data sets from UCI and a genome data with thousand attributes. The paper ends with conclusions and further research topics in Section 5.

2. Preliminaries

Basic concepts, notations and results of rough sets as well as their extensions are outlined in this section [1,16,18,33,34,36,38,39,45,47–50,52–55,65,66].

2.1. Pawlak rough sets

An approximation space $apr = (U, R)$ is defined by a universe U and a binary relation R. Let U be a finite and non-empty set and R be an binary relation on U. The pair $apr = (U, R)$ is defined as Pawlak approximation space when R is an equivalence relation. In Pawlak rough sets, the equivalence relation R induces a partition of U, denoted by $[x]_R$ or $[x]$. For a subset $X \subseteq U$, its lower and upper approximations are defined respectively by:

$$\begin{aligned} &\underline{R}(X) = \{x \in U \mid [x] \subseteq X\};\\ &\overline{R}(X) = \{x \in U \mid [x] \cap X \neq \emptyset\}. \end{aligned} \tag{1}$$

Intuitively, these two approximations divide the universe U into three disjoint regions: the positive region $\mathrm{POS}_R(X)$, the negative region $\mathrm{NEG}_R(X)$ and the boundary region $\mathrm{BND}_R(X)$.

$$\begin{aligned} &\mathrm{POS}_R(X) = \underline{R}(X);\\ &\mathrm{BND}_R(X) = \overline{R}(X) - \underline{R}(X);\\ &\mathrm{NEG}_R(X) = U - \overline{R}(X). \end{aligned} \tag{2}$$

The positive region $\mathrm{POS}(X)$ is defined by the lower approximation, the negative region $\mathrm{NEG}(X)$ is defined by the complement of the upper approximation, and the boundary region $\mathrm{BND}(X)$ is defined by the difference between the upper and lower approximations. If $\underline{R}(X) = X = \overline{R}(X)$ holds, a subset X of U is definable in (U, R). Otherwise, X is indefinable in (U, R).

2.2. Probabilistic rough sets

To introduce the PRS, Pawlak and Skowron suggested using a rough membership function to redefine the two approximations [37]. The rough membership function μ_A is defined by:

$$\mu_A(x) = Pr(X|[x]) = \frac{|[x] \cap X|}{|[x]|}, \tag{3}$$

where $|\cdot|$ stands for the cardinal number of objects in sets. $Pr(X|[x])$ denotes the conditional probability of the classification. The degree of the overlap between an equivalence class $[x]$ and a set X is calculated as the conditional probability $Pr(X|[x])$ of the set given the equivalence class $[x]$.

From a rough membership function, the Pawlak approximations in Eq. (1) can be equivalently defined respectively as follows:

$$\begin{aligned} &\underline{R}(X) = \{x \in U \mid Pr(X|[x]) = 1\};\\ &\overline{R}(X) = \{x \in U \mid Pr(X|[x]) > 0\}. \end{aligned} \tag{4}$$

In Eq. (4), two extreme values, 1 and 0, are utilized to define the two approximations by rough membership function. The classifications in the lower approximation must be absolutely consistent. However, the definition in Eq. (4) is too strict because the magnitude of the value $Pr(X|[x])$ is not taken into account [44,49,65]. A main result in PRS is parameterized probabilistic approximations, which is similar to the notion of α-cuts of fuzzy sets [53]. This can be done by replacing the values 1 and 0 in Eq. (5) by a pair of parameters α and β. The (α, β)-lower approximation and (α, β)-upper approximation are defined respectively as follows.

$$\begin{aligned} &\underline{R}_{(\alpha,\beta)}(X) = \{x \in U \mid Pr(X|[x]) \geqslant \alpha\},\\ &\overline{R}_{(\alpha,\beta)}(X) = \{x \in U \mid Pr(X|[x]) > \beta\}. \end{aligned} \tag{5}$$

In (5), an equivalence class $[x]$ is a part of the lower approximation if the conditional probability $Pr(X|[x])$ is above α, and it is a part of the upper approximation if the conditional probability $Pr(X|[x])$ is above β. The (α, β)-probabilistic positive, boundary and negative regions can be defined by the (α, β)-probabilistic lower and upper approximations.

$$\begin{aligned} &\mathrm{POS}_{(\alpha,\beta)}(X) = \{x \in U \mid Pr(X|[x]) \geqslant \alpha\},\\ &\mathrm{BND}_{(\alpha,\beta)}(X) = \{x \in U \mid \beta < Pr(X|[x]) < \alpha\},\\ &\mathrm{NEG}_{(\alpha,\beta)}(X) = \{x \in U \mid Pr(X|[x]) \leqslant \beta\}. \end{aligned} \tag{6}$$

The parameters α and β allow certain acceptable levele of errors, and the PRS makes the process of decision making more reasonable. Specially, when we set $\alpha = \beta = 0.5$, it becomes the 0.5-probabilistic rough set model [35]; when we set $\alpha + \beta = 1$ and $\alpha > 0.5$, it becomes the symmetric variable precision rough set model [65]; when we set $0 \leqslant \beta < \alpha \leqslant 1$, it becomes the asymmetric variable precision rough set model [17]. In addition, the values of α and β can be automatically computed by using the bayesion minimum conditional risk criterion in decision-theoretic rough set models [28,49]. In our following discussions, we focus on investigating the incremental updating approximations approaches in PRS.

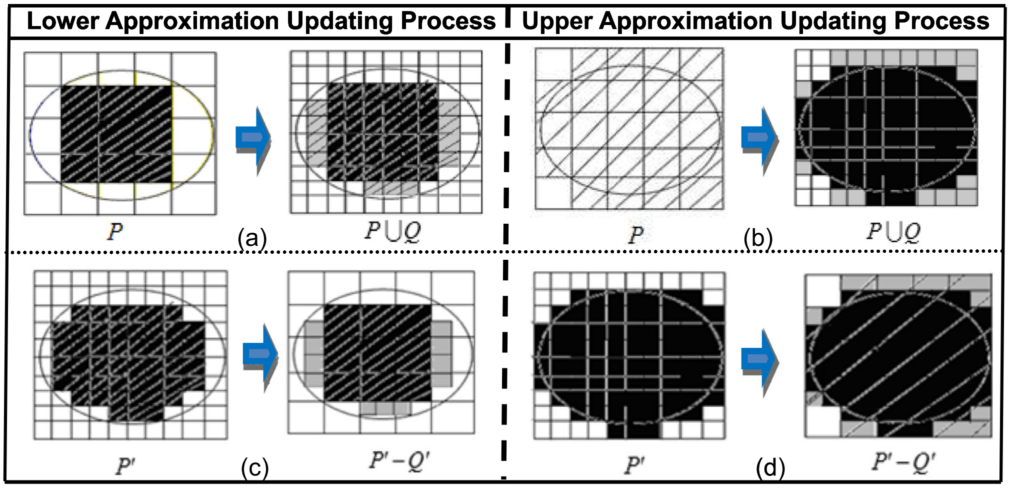

Fig. 1. The lower and upper approximations updating process. *Note:* the block in the each sub-figure denotes an equivalence class.

3. Approaches for incremental updating approximations under the variation of attributes

The variation of attributes in an information system has the following two scenarios: the adding of the attributes and the deleting of the attributes [4,21]. In the first scenario, the lower approximation may expand and the upper approximation may contract with the specialization of the system, and the granularity of the system becomes refining. The opposite situations may happen in the second scenario with the generalization of the system. Since the lower approximation generates certain decision rules, and the upper approximation generates possible decision rules [12], we focus on the studies of the updating process of the two approximations when attributes vary.

To clearly illuminate this idea, we review the approaches for dynamic attribute generalization in Pawlak rough sets in Section 3.1. The differences of the strategies for updating approximations between Pawlak rough sets and PRS are carefully investigated in Section 3.2. We present a new incremental learning method under PRS and discuss its propositions in Section 3.3. Two incremental algorithms for updating approximations by adding and deleting attributes are proposed for updating approximations in Section 3.4.

3.1. Approaches for dynamic attribute generalization in Pawlak rough sets

An approximation space $apr=(U,R)$ is characterized by an information system $S=(U,A,V,f)$ with $A=C\bigcup D$ and $C\bigcap D=\emptyset$, where C denotes the condition attribute set and D denotes the decision attribute set. $V=\bigcup_{a\in A}V_a$, V_a is a domain of the attribute a. $f:U\times A\to V$ is an information function such that $f(x,a)\in V_a$ for every $x\in U, a\in A$ [36].

To update approximations incrementally, Chan firstly investigated a method for updating approximations by considering adding and deleting one attribute at a time. The key point of his algorithm is updating approximations by deleting the original attribute first, followed by inserting the generalized one using information of current approximations [4]. To achieve the goals, Chan introduced a concept of boundary sets. He suggested to divide the boundary region $\mathrm{BND}(X)$ into two parts, namely, lower boundary sets $\underline{\Delta}_R(X)$ and upper boundary sets $\overline{\Delta}_R(X)$ [4].

$$\begin{aligned}
&\mathrm{BND}_R(X)=\overline{\Delta}_R(X)\cup\underline{\Delta}_R(X);\\
&\underline{\Delta}_R(X)=X-\underline{R}(X); \qquad (7)\\
&\overline{\Delta}_R(X)=\overline{R}(X)-X.
\end{aligned}$$

According to Eq. (7), we have $\overline{R}(X)-\underline{R}(X)=\overline{\Delta}(X)+\underline{\Delta}(X)$. If $\underline{\Delta}(X)=\overline{\Delta}(X)=\emptyset$ holds, a subset X of U is definable in (U,R). Otherwise, X is indefinable in (U,R). The boundary sets are utilized as a tool for learning rules from examples and for updating approximations of a subset X incrementally.

(**I**) The scenario of adding attributes.
Let $P,Q\subseteq C, \forall X\subseteq U$. We have: $\underline{P}(X)\subseteq\underline{P\bigcup Q}(X)$, $\underline{Q}(X)\subseteq\underline{P\bigcup Q}(X)$; $\overline{P\bigcup Q}(X)\subseteq\overline{P}(X)$, $\overline{P\bigcup Q}(X)\subseteq\overline{Q}(X)$. According to the definitions of boundary sets in Eq. (7), we obtain: $\underline{\Delta}_{P\bigcup Q}(X)\subseteq\underline{\Delta}_P(X), \underline{\Delta}_{P\bigcup Q}(X)\subseteq\underline{\Delta}_Q(X); \overline{\Delta}_{P\bigcup Q}(X)\subseteq\overline{\Delta}_P(X)$, $\overline{\Delta}_{P\bigcup Q}(X)\subseteq\overline{\Delta}_Q(X)$.

(**II**) The scenario of deleting attributes.
Let $Q'\subset P'\subseteq C, \forall X\subseteq U$. We have: $\underline{P'-Q'}(X)\subseteq\underline{P'}(X); \overline{P'}(X)\subseteq\overline{P'-Q'}(X)$. According to the definitions of boundary sets in Eq. (7), we obtain: $\underline{\Delta}_{P'}(X)\subseteq\underline{\Delta}_{P'-Q'}(X)$; $\overline{\Delta}_{P'}(X)\subseteq\overline{\Delta}_{P'-Q'}(X)$.

Fig. 1 clearly outlines lower approximation and upper approximation updating process with the above two scenarios.

Fig. 1(a) and (b) show the approximations updating process when Q is added into P. In Fig. 1(a), the black parts stand for the lower approximations under P, the gray parts stand for the immigration areas when Q is added to P; in Fig. 1(b), the shadows stand for the upper approximations under P, the gray parts stand for the emigration areas when Q is added to P.

Similarly, Fig. 1(c) and (d) show the approximations updating process when Q' is deleted from P'. In Fig. 1(c), the black parts stand for the lower approximations under P', the gray parts stand for the emigration areas when Q' is deleted from P'; in Fig. 1(d), the black areas stand for the upper approximations under P', the gray parts stand for the immigration areas when Q' is deleted from P'.

The propositions for updating approximations when attributes are added to or deleted from the original system are displayed as follows [4,21].

Proposition 1. *Let* $P,Q\subseteq C, P\bigcap Q=\emptyset$ *and* $\mathcal{A}=P\bigcup Q$. *Then,* $\underline{\mathcal{A}}(X)=\underline{P}(X)\bigcup\underline{Q}(X)\bigcup Y$, *where* $Y=\{x\in\underline{\Delta}_P(X)\bigcap\underline{\Delta}_Q(X)|\bigcap_{a\in\mathcal{A}}[x]_a\subseteq X\}$.

Proposition 2. *Let* $Q'\subset P'\subseteq C$ *and* $\mathcal{B}=P'-Q'$. *Then,* $\underline{\mathcal{B}}(X)=\underline{P'}(X)-\underline{\Delta}_{\mathcal{B}}(X)$, *where* $\underline{\Delta}_{\mathcal{B}}(X)=\{x\in\bigcap_{b\in\mathcal{B}}\underline{\Delta}_b(X)|\bigcap_{b\in\mathcal{B}}[x]_b\neg\subset X\}$.

Proposition 3. *Let* $P,Q\subseteq C, P\bigcap Q=\emptyset$ *and* $\mathcal{A}=P\bigcup Q$. *Then,* $\overline{\mathcal{A}}(X)=X\cup(\overline{\Delta}_P(X)-Z)$, *where* $Z=\{x\in\bigcap_{a\in P}\overline{\Delta}_a(X)|\bigcap_{b\in\mathcal{A}}[x]_b\subseteq\bigcap_{a\in P}\overline{\Delta}_a(X)\}$.

Proposition 4. *Let* $Q' \subset P' \subseteq C$ *and* $\mathcal{B} = P' - Q'$. *Then,* $\overline{\mathcal{B}}(X) = X \bigcup \overline{\Delta}_{P'}(X) \bigcup Z'$, *where* $Z' = \{x \in \bigcap_{b \in \mathcal{B}} \overline{\Delta}_b(X) | \bigcap_{b \in \mathcal{B}} \overline{\Delta}_b(X) \neg \subset \bigcap_{a \in P'} [x]_a\}$.

In addition, Li et al. utilized the characteristic relation instead of the equivalence relation and discovered similar propositions in incomplete information systems [21]. Li et al. considered the information with preference-ordered and discussed the similar propositions in dominance-based rough sets [19]. These researches provide some insightful ideas to design effective algorithms for updating approximations when attributes vary.

3.2. The differences of the strategies for updating approximations between Pawlak rough sets and PRS

In this subsection, we introduce PRS to dynamic maintenance of approximations. By considering the variation of boundary sets is the key step to updating approximations when attributes vary [3,4,21,59], we adopt Chan's suggestions in [4] and focus on investigating the variation of boundary region $BND_{(\alpha,\beta)}(X)$.

Let X be a subset of U. Given the acceptable level parameters α and β with $0 \leqslant \beta < \alpha \leqslant 1$. Ideally, followed by Chan's work in Eq. (7), an intuitive strategy for updating approximations is to define the concepts of (α,β)-boundary sets. Here, we simply define the concept of (α,β)-lower boundary sets and (α,β)-upper boundary sets as:

$$\begin{aligned} &\underline{\Delta}_R^{(\alpha,\beta)}(X) = X - \underline{R}_{(\alpha,\beta)}(X); \\ &\overline{\Delta}_R^{(\alpha,\beta)}(X) = \overline{R}_{(\alpha,\beta)}(X) - X. \end{aligned} \tag{8}$$

For $\alpha \neq 1$ and $\beta \neq 0$, we have:

$$\underline{R}_{(\alpha,\beta)}(X) \nsubseteq X;$$
$$X \nsubseteq \overline{R}_{(\alpha,\beta)}(X).$$

That is,

$$BND_{(\alpha,\beta)}(X) = \overline{R}^{(\alpha,\beta)}(X) - \underline{R}^{(\alpha,\beta)}(X) \neq \overline{\Delta}_R^{(\alpha,\beta)}(X) - \underline{\Delta}_R^{(\alpha,\beta)}(X).$$

The relations between $\underline{R}_{(\alpha,\beta)}(X)$ and X; X and $\overline{R}_{(\alpha,\beta)}(X)$ are absolutely uncertain when $\alpha \neq 1$ and $\beta \neq 0$. Therefore, Chan's approach cannot be directly applied to PRS. The definitions of (α,β)-boundary sets in Eq. (8) do not make sense to the new problem in PRS. We should find another way to solve this puzzle, and the incremental strategies and algorithms for dynamic maintenance of approximations under PRS need to further investigate.

Fig. 2 clearly outlines (α,β)-lower and upper approximations updating process with $\alpha = 0.5$ and $\beta = 0.2$.

Fig. 2(a) and (b) show the approximations updating process when Q is added into P. In Fig. 2(a), the black parts stand for the (α,β)-lower approximations under P, the gray parts stand for the variation areas when Q is added to P; in Fig. 2(b), the shadows stand for the (α,β)-upper approximations under P, the gray parts stand for the variation areas when Q is added to P.

Similarly, Fig. 2(c) and (d) show the approximations updating process when Q' is deleted from P'. In Fig. 2(c), the black parts stand for the (α,β)-lower approximations under P' and $P' - Q'$, respectively; in Fig. 2(d), the black areas stand for the (α,β)-upper approximations under P', the gray parts stand for the variation areas when Q' is deleted from P'.

With the insightful gain from Figs. 1 and 2, it is obvious that we can obtain the updated approximations by the previous results without beginning from scratch, and the variety of boundary region is the key technique to set up an algorithm for updating approximations. Some differences on dynamic maintenance between Pawlak rough sets and PRS are summarized as following two scenarios.

(1) Given an information system $S = (U, A, V, f)$, $P, Q \subseteq C$ and $P \bigcap Q = \emptyset$. $\forall X \subseteq U$, since $\underline{P}(X) \subseteq \underline{P \bigcup Q}(X) \subseteq \overline{P \bigcup Q}(X) \subseteq \overline{P}(X)$, we have $BND_{P \bigcup Q}(X) \subseteq BND_P(X)$, $\underline{\Delta}_{P \bigcup Q}(X) \subseteq \underline{\Delta}_P(X)$ and $\overline{\Delta}_{P \bigcup Q}(X) \subseteq \overline{\Delta}_P(X)$. Intuitively, Consider $P = \{R_1, R_2, \ldots, R_n\}$ a family of attribute sets with $R_1 \succeq R_2 \succeq \cdots \succeq R_n (R_n \in 2^C)$, we have $BND_{R_1}(X) \subseteq BND_{R_2}(X) \subseteq \cdots BND_{R_n}(X), \underline{\Delta}_{R_1}(X) \subseteq \underline{\Delta}_{R_2}(X) \subseteq \cdots \underline{\Delta}_{R_n}(X)$ and $\overline{\Delta}_{R_1}(X) \subseteq \overline{\Delta}_{R_2}(X) \subseteq \cdots \overline{\Delta}_{R_n}(X)$, that is, the boundary region tends to become smaller when we monotonically add the attributes.
On the contrary, the opposite situations happen when we monotonically delete attributes. Given an information system $S = (U, A, V, f)$, $Q' \subset P' \subseteq C$. $\forall X \subseteq U$, since $\underline{P' - Q'}(X) \subseteq \underline{P'}(X) \subseteq \overline{P'}(X) \subseteq \overline{P' - Q'}(X)$, we have $BND_{P'}(X) \subseteq BND_{P'-Q'}(X)$, $\underline{\Delta}_{P'}(X) \subseteq \underline{\Delta}_{P'-Q'}(X)$ and $\overline{\Delta}_{P'}(X) \subseteq \overline{\Delta}_{P'-Q'}(X)$. Intuitively, Consider $P' = \{R'_1, R'_2, \ldots, R'_n\}$ a family of attribute sets with

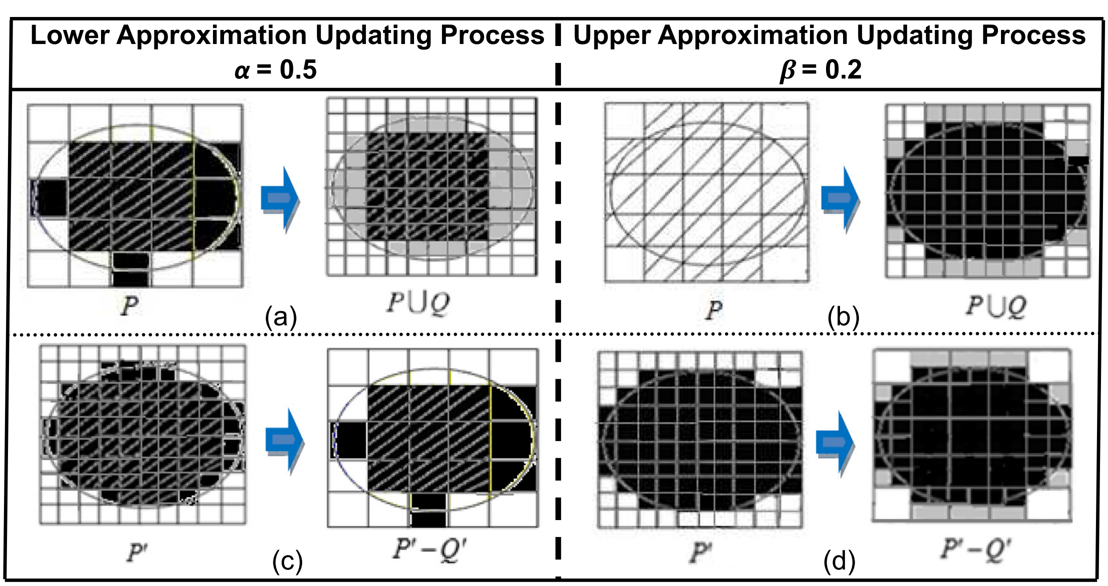

Fig. 2. The (α,β)-lower and upper approximations updating process with $\alpha = 0.5$ and $\beta = 0.2$.

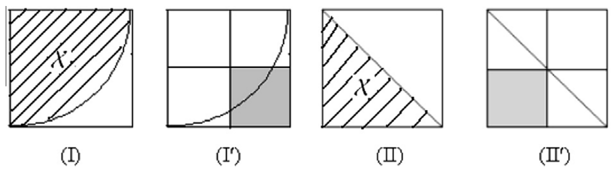

Fig. 3. The variation of (α,β)-lower approximations. *Notes:* The light gray parts stand for the (α,β)-lower approximations, and the dark gray parts stand for the variation areas.

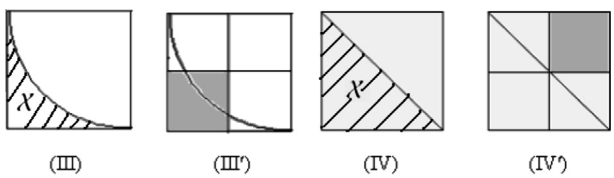

Fig. 4. The variation of (α,β)-upper approximations. The light gray parts stand for the (α,β)-upper approximations, and the dark gray parts stand for the variation areas.

$R'_1 \preceq R'_2 \preceq \cdots \preceq R'_n (R'_n \in 2^C)$, we have $BND_{R'_1}(X) \supseteq BND_{R'_2}(X) \supseteq \cdots BND_{R'_n}(X)$, $\underline{\Delta}_{R'_1}(X) \supseteq \underline{\Delta}_{R'_2}(X) \supseteq \cdots \underline{\Delta}_{R'_n}(X)$ and $\overline{\Delta}_{R'_1}(X) \supseteq \overline{\Delta}_{R'_2}(X) \supseteq \cdots \overline{\Delta}_{R'_n}(X)$, that is, the boundary region tends to become bigger when we monotonically delete the attributes.

To sum up, the variation of attributes leads to a one-way translation, namely, increasing or decreasing of objects in Pawlak boundary sets.

(2) As we stated in Fig. 2, the variation of attributes leads to two-way translations of objects in (α,β)-probabilistic boundary sets, namely, immigration and emigration of objects, simultaneously. In order to clearly illustrate our ideas, we choose two representative examples to explain the variation process for updating lower and upper approximations, which is shown in Figs. 3 and 4, respectively.

In Fig. 3, let $\mathcal{X}$ (the shadow parts) be a concept and $\mathcal{X} \subseteq X \subset U$. Suppose (I) and (II) are two blocks (equivalence classes) in the boundary region of original system, (I) will change into (I') and (II) will change into (II') when adding attributes to the system. For simplicity and clarity, we divide the four pictures into two groups. The first group includes the left two pictures (see Fig. 3 (I) and (I')), and the second group includes the right two pictures (see Fig. 3 (II) and (II')). If $0 \leqslant \beta < 0.5 < \alpha \leqslant 1$, the variation of (α,β)-lower approximations can be outlined as follows by adding the attributes.

- The bottom right corner block in (I') will go out of (I) because of $Pr(\mathcal{X}|[x]) < \alpha$.
- The bottom left corner block in (II') will enter into (II) because of $Pr(\mathcal{X}|[x]) = 1 \geqslant \alpha$.

Furthermore, we obtain the opposite conclusions by simply changing the place of (I) and (I'), (II) and (II') when deleting attributes from the system, namely, the bottom right corner block in (I') will immigrate to (I), and the bottom left corner block in (II') will emigrate from (II).

Similarly, in Fig. 4, let $\mathcal{X}$ (the shadow parts) be a concept and $\mathcal{X} \subseteq X \subset U$. Suppose (III) and (IV) are two blocks (equivalence classes) in the boundary region of original system, (III) will change into (III') and (IV) will change into (IV') when deleting attributes in the system. For simplicity and clarity, we also divide the four pictures into two groups as ((III), (III')) and ((IV), (IV')). If $0 \leqslant \beta < 0.5 < \alpha \leqslant 1$, the variation of (α,β)-upper approximations can be outlined as follows by deleting the attributes.

- The bottom left corner block in (III') will enter into (III) because of $Pr(\mathcal{X}|[x]) > \beta$.
- The top right corner block in (IV') will go out of (IV) because of $Pr(\mathcal{X}|[x]) = 0 \leqslant \beta$.

Furthermore, we obtain the opposite conclusions by simply changing the place of (III) and (III'), (IV) and (IV') when deleting attributes from the system, namely, the bottom left corner part in (III') will emigrate from (III), and the top right corner part in (IV') will immigrate to (IV).

As we stated above, the differences for updating approximations between Pawlak rough sets and PRS can be summarized in Table 1.

Table 1 gives an intuitive explanation of the differences for updating approximations between the two rough set models. The key technology for updating approximations under PRS is to find a quick way to handle both immigration and emigration of objects when adding or deleting attributes. In the next subsection, we begin to investigate the strategy and algorithm on dynamic maintenance of approximations under PRS.

3.3. Dynamic maintenance of approximations under PRS

Followed by Table 1 and Figs. 2–4, the objects in (α,β)-lower and upper approximations may simultaneously immigrate to or emigrate from the original system in PRS when attributes vary. In the following, we investigate two scenarios, adding and deleting attributes, for the incremental strategies for updating approximations under PRS.

3.3.1. Dynamic maintenance of approximations under PRS when adding attributes

3.3.1.1. Lower approximation updating when adding attributes. Given an information system $S = (U, A, V, f), P, Q \subseteq C$ and $P \bigcap Q = \emptyset$. The variation of lower approximations when adding Q to P is outlined in Fig. 5.

For simplicity, we divide the variations into six parts, which are labeled in Table 2.

Table 1
The differences for updating approximations between Pawlak rough sets and PRS.

Indexes	Pawlak rough sets		Probabilistic rough sets	
	Deleting attributes	Adding attributes	Deleting attributes	Adding attributes
Lower approximation	Decrease	Increase	Uncertain	Uncertain
Upper approximation	Increase	Decrease	Uncertain	Uncertain
Boundary region	Increase	Decrease	Uncertain	Uncertain
Lower boundary sets	Increase	Decrease	N/A	N/A
Upper boundary sets	Increase	Decrease	N/A	N/A
Variation of boundary region	One-way translation		Two-way translations	
Relations	$\overline{R}(X) - \underline{R}(X) = \overline{\Delta}(X) + \underline{\Delta}(X)$		N/A	

Notes: N/A represents "not applicable".

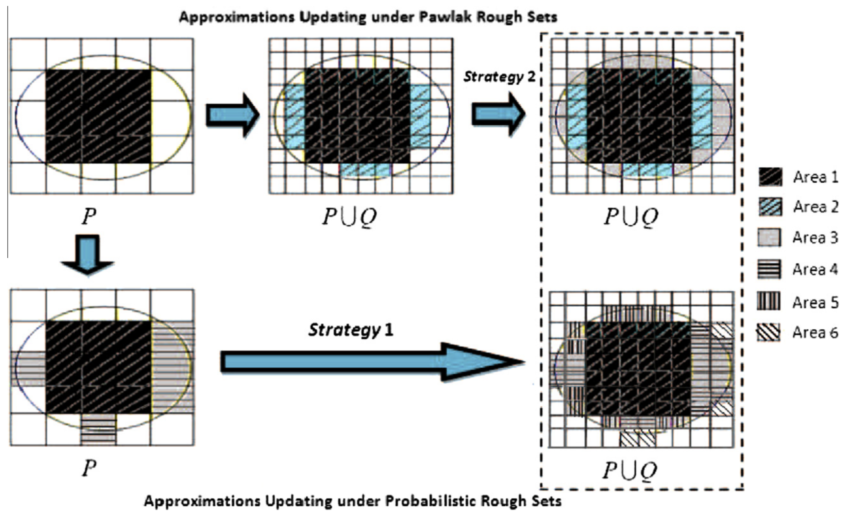

Fig. 5. Dynamic maintenance of lower approximation when adding attributes.

Table 2
The variation areas of lower approximation when adding attributes.

Areas	Labels
Area 1	The lower approximation in Pawlak rough sets under P
Area 2	The adding parts of lower approximation when Q is added to P in Pawlak rough sets
Area 3	The adding parts of lower approximation between Pawlak rough sets and PRS when Q is added to P
Area 4	The adding parts of lower approximation between Pawlak rough sets and PRS under P
Area 5	The adding parts of (α,β)-lower approximation when Q is added to P in PRS
Area 6	The deleting parts of (α,β)-lower approximation when Q is added to P in PRS

Intuitively, according to Fig. 5, there are two strategies for updating lower approximation incrementally.

Strategy 1: Updating based on $\underline{P}_{(\alpha,\beta)}(X)$.
As we stated in Section 3.2, the variation when Q is added to P leads to two-way translations. For simplicity, we denote $IN(X)=\{x\in U|Pr(X|[x]_{P\bigcup Q})\geqslant\alpha\}$ as the immigration sets (See Area 5 in Fig. 5) and $OUT(X)=\{x\in U|Pr(X|[x]_{P\bigcup Q})<\alpha\}$ as the emigration sets (See Area 6 in Fig. 5), respectively. Obviously, $\underline{P\bigcup Q}_{(\alpha,\beta)}(X)=\underline{P}_{(\alpha,\beta)}(X)\bigcup IN(X)-OUT(X)$.

Strategy 2: Updating based on $\underline{P\bigcup Q}(X)$.
According to Eqs. (4) and (5), we obtain: $\underline{R}(X)\subseteq\underline{R}_{(\alpha,\beta)}(X)$. Furthermore, the difference between $\underline{R}(X)$ and $\underline{R}_{(\alpha,\beta)}(X)$ can be defined as (α,β)-lower difference set $\underline{\Phi}_R^{(\alpha,\beta)}(X)$:

$$\underline{\Phi}_R^{(\alpha,\beta)}(X)=\underline{R}_{(\alpha,\beta)}(X)-\underline{R}(X)=\{x\in BND_R(X)|\alpha\leqslant Pr(X|[x]_R)<1\}. \quad (9)$$

Hence, let $R=P\bigcup Q$, we have $\underline{P\bigcup Q}_{(\alpha,\beta)}(X)=\underline{P\bigcup Q}(X)\bigcup\underline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$. Obviously, $\underline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)\subseteq BND_{P\bigcup Q}(X)$. A smart strategy to update lower approximation in PRS when Q is added to P includes two parts: one is the variation of Pawlak lower approximation $\underline{P\bigcup Q}(X)$, the other one is the variation of $\underline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$ (See Area 3 in Fig. 5).

3.3.1.2. Upper approximation updating under PRS when adding attributes. Similarly, there have two strategies for updating upper approximation when adding Q to P.

Strategy 1: Updating based on $\overline{P}_{(\alpha,\beta)}(X)$.
For simplicity, we denote $IN'(X)=\{x\in U|Pr(X|[x]_{P\bigcup Q})>\beta\}$ as the immigration sets and $OUT'(X)=\{x\in U|Pr(X|[x]_{P\bigcup Q})\leqslant\beta\}$ as the emigration sets, respectively. Obviously, $\overline{P\bigcup Q}_{(\alpha,\beta)}(X)=\overline{P}_{(\alpha,\beta)}(X)\bigcup IN\prime(X)-OUT'(X)$.

Strategy 2: Updating based on $\overline{P\bigcup Q}(X)$.
According to Eqs. (4) and (5), we obtain: $\overline{R}_{(\alpha,\beta)}(X)\subseteq\overline{R}(X)$. Furthermore, the difference between $\overline{R}(X)$ and $\overline{R}_{(\alpha,\beta)}(X)$ can be defined as (α,β)-upper difference set $\overline{\Phi}_R^{(\alpha,\beta)}(X)$:

$$\overline{\Phi}_R^{(\alpha,\beta)}(X)=\overline{R}(X)-\overline{R}_{(\alpha,\beta)}(X)=\{x\in BND_R(X)|0<Pr(X|[x]_R)\leqslant\beta\}. \quad (10)$$

Hence, let $R=P\bigcup Q$, we have $\overline{P\bigcup Q}_{(\alpha,\beta)}(X)=\overline{P\bigcup Q}(X)-\overline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$. Obviously, $\overline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)\subseteq BND_{P\bigcup Q}(X)$. A smart strategy to update upper approximation in PRS when Q is added to P includes two parts: one is the variation of Pawlak upper approximation $\overline{P\bigcup Q}(X)$, the other one is the variation of $\overline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$.

3.3.2. Dynamic maintenance of approximations under PRS when deleting attributes

3.3.2.1. Lower approximation updating when deleting attributes. Given an information system $S=(U,A,V,f)$, $P',Q'\subseteq C$ and $Q'\subset P'\subseteq C$. The variation of lower approximations when deleting Q' from P' is outlined in Fig. 6.

For simplicity, we divide the variations into six parts, which are labeled in Table 3.

Intuitively, there are two strategies for updating lower approximation incrementally.

Strategy 1: Updating based on $\underline{P'}_{(\alpha,\beta)}(X)$.
As we stated in Section 3.2, the variation when Q' is deleted from P' leads to two-way translations. For

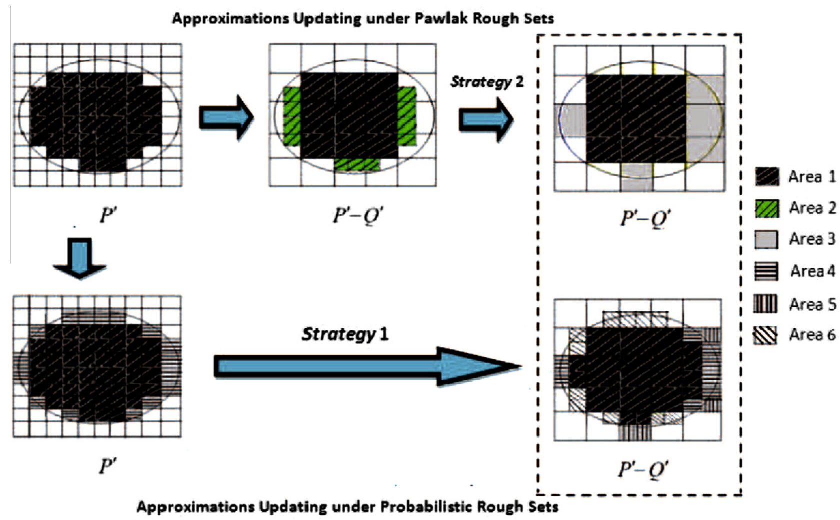

Fig. 6. Dynamic maintenance of lower approximation under PRS when deleting attributes.

Table 3
The variation areas of lower approximation when deleting attributes.

Areas	Labels
Area 1	The lower approximation in Pawlak rough sets under P'
Area 2	The deleting parts of lower approximation when Q' is deleted from P' in Pawlak rough sets
Area 3	The adding parts of lower approximation between Pawlak rough sets and PRS when Q' is deleted from P'
Area 4	The adding parts of lower approximation between Pawlak rough sets and PRS under P'
Area 5	The adding parts of (α,β)-lower approximation when Q' is deleted from P' in PRS
Area 6	The deleting parts of (α,β)-lower approximation when Q' is deleted from P' in PRS

simplicity, we denote $IN(X) = \{x \in U | Pr(X|[x]_{P'-Q'}) \geqslant \alpha\}$ as the immigration sets (See Area 5 in Fig. 6) and $OUT(X) = \{x \in U | Pr(X|[x]_{P'-Q'}) < \alpha\}$ as the emigration sets (See Area 6 in Fig. 6), respectively. Obviously, $\underline{P'-Q'}_{(\alpha,\beta)}(X) = \underline{P'}_{(\alpha,\beta)}(X) \bigcup IN(X) - OUT(X)$.

Strategy 2: Updating based on $\underline{P'-Q'}(X)$.
According to Eq. (9), $\underline{P'-Q'}_{(\alpha,\beta)}(X) = \underline{P'-Q'}(X) \bigcup \underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$, where, $\underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) = \{x \in BND_{P'-Q'} |Pr(X|[x]) \geqslant \alpha\}$. Obviously, $\underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) \subseteq BND_{P'-Q'}(X)$. A smart strategy to update lower approximation in PRS when Q' is deleted from P' includes two parts: one is the variation of Pawlak lower approximation $\underline{P'-Q'}(X)$, the other one is the variation of $\underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$ (See Area 3 in Fig. 6).

3.3.2.2. Upper approximation updating when deleting attributes. Similarly, there have two strategies for updating upper approximation when deleting Q' from P'.

Strategy 1: Updating based on $\overline{P'}_{(\alpha,\beta)}(X)$.
For simplicity, we denote $IN'(X) = \{x \in U | \beta \leqslant Pr(X|[x]_{P'-Q'}) < \alpha\}$ as the immigration sets and $OUT'(X) = \{x \in U | Pr(X|[x]_{P'-Q'}) < \beta\}$ as the emigration sets, respectively. Obviously, $\overline{P'-Q'}_{(\alpha,\beta)}(X) = \overline{P'}_{(\alpha,\beta)}(X) \bigcup IN'(X) - OUT'(X)$.

Strategy 2: Updating based on $\overline{P'-Q'}(X)$.
According to Eq. (10), let $R = P'-Q'$, we have $\overline{P'-Q'}_{(\alpha,\beta)}(X) = \overline{P'-Q'}(X) - \overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$. Where, $\overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) = \{x \in BND_{P'-Q'} | Pr(X|[x]_{P'-Q'}) \leqslant \beta\}$. Obviously, $\overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) \subseteq BND_{P'-Q'}(X)$. A smart strategy to update upper approximation in PRS when Q' is deleted from P' includes two parts: one is the variation of Pawlak upper approximation $\overline{P'-Q'}(X)$, the other one is the variation of $\overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$.
Compared with Strategy 1 and Strategy 2, the latter one is based on the updating approximations under Pawlak rough sets, it mainly focuses on the variation of a part of the boundary region (i.e., $\underline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$, $\overline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X)$ $\underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$, $\overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X)$) and considers only one-way translation for approximations updating under PRS. Specially, if $\underline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X) = \overline{\Phi}^{(\alpha,\beta)}_{P\bigcup Q}(X) = \underline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) = \overline{\Phi}^{(\alpha,\beta)}_{P'-Q'}(X) = \emptyset$, the process of updating approximations under PRS becomes the process of updating approximations under Pawlak rough sets. In our following discussions, we adopt the "Strategy 2" to design the incremental algorithms.

3.4. Incremental algorithms for updating approximations under PRS

In this subsection, we begin to design the incremental algorithm for updating approximations in PRS. As Zhang et al. stated in [59], an effective incremental algorithms can apply an incremental updating scheme, which is an effective method to maintain knowledge dynamically, to avoid unnecessary computations by utilizing the previous data structures or results. In addition, Zhang et al. further suggested a block based strategy for updating approximations of a concept. The idea to design the incremental algorithm is based on that the objects in the universe are divided into serval equivalence classes [59]. The experimental evaluations proved that the block based strategy can handle dynamic attribute generalization effectively [22,23,59]. In the light of the "Strategy 2" in Section 3.3, the methods for updating approximations in PRS are mainly decided by the variation of boundary sets. Therefore, we divide the boundary sets into serval equivalence classes at first, then the incremental algorithms for updating approximations in PRS are equal to update the boundary equivalence classes.

Suppose the process of updating approximation lasts for two different periods: at time t and at time $t+1$. There are two possible scenarios when attributes vary at time $t+1$, namely, adding and deleting of attributes. The processes of updating approximations with the two scenarios are discussed as follows.

3.4.1. The incremental algorithm for updating approximations when adding attributes

First, we consider the scenario of adding attributes. Suppose $S=(U,P,V,f)$ at time t, the attributes (their set is denoted as Q) enter into the attribute set P and the system will change as $S'=(U,P\bigcup Q,V,f)$ at time $t+1$, $\mathcal{A}=P\bigcup Q$. $P\subseteq\mathcal{A}, \forall X\subseteq U$, we have: $\underline{P}(X)\subseteq\underline{\mathcal{A}}(X), \overline{\mathcal{A}}(X)\subseteq\overline{P}(X), BND_{\mathcal{A}}(X)\subseteq BND_P(X)$. An effective strategy of updating approximation is to calculate the variation of boundary sets. For simplicity and clarity, we denote $BND=BND_P(X)$ and $BND/P=\{BND_1,BND_2,\ldots,BND_m\}$. $\forall i\in m$, BND_i is divided into some small parts when Q is added into P, and the criteria to judge whether BND_i refines or not are shown as follows.

(1) $\forall x,y\in BND_i$, if $\forall a\in Q$, $f(x,a)=f(y,a)$, the BND_i keeps constant;
(2) $\forall x,y\in BND_i$, if $\exists a\in Q, f(x,a)\neq f(y,a)$, the BND_i refines.

Suppose $BND_i=BND_{i1}\bigcup BND_{i2}\bigcup\cdots\bigcup BND_{in_i}$ at time $t+1$, $\forall i\in m$, we have $|BND_i|=|BND_{i1}|+|BND_{i2}|+\cdots+|BND_{in_i}|$ and $|BND_P(X)|=\sum_{i=1}^{m}\sum_{j=1}^{n_i}|BND_{ij}|$, namely, m equivalence classes in boundary sets at time t refine as $(n_1+n_2+\cdots+n_m)$ equivalence classes at time $t+1$. BND_{ij} is used to calculate $Pr(BND_{ij},X)$. BND_{ij} is a part of the lower approximation if $Pr(BND_{ij},X)\geqslant\alpha$, and it is not a part of the upper approximation if $Pr(BND_{ij},X)\leqslant\beta$. Therefore, $\underline{\mathcal{A}}_{(\alpha,\beta)}(X)$ and $\overline{\mathcal{A}}_{(\alpha,\beta)}(X)$ can be calculated as follows.

$$\underline{\mathcal{A}}_{(\alpha,\beta)}(X)=\underline{P}(X)\bigcup\{BND_{ij}|Pr(BND_{ij},X)\geqslant\alpha\},\ \text{where } i=1,2,\ldots,m; j=1,2,\ldots,n_i;$$
$$\overline{\mathcal{A}}_{(\alpha,\beta)}(X)=\overline{P}(X)-\{BND_{ij}|Pr(BND_{ij},X)\leqslant\beta\},\ \text{where } i=1,2,\ldots,m; j=1,2,\ldots,n_i.$$

We outline the updating process when adding attributes in Algorithm 1 as follows.

Algorithm 1. The algorithm for updating approximations when adding attributes.

Input:
(1) $S=(U,P,V,f)$ at time t, the thresholds α and β;
(2) The approximations $\underline{P}(X)$ and $\overline{P}(X)$ at time t; the classifications with boundary sets: $BND_P(X)/P=\{BND_1,BND_2,\cdots,BND_m\}$ at time t;
(3) The adding attribute set Q.
Output:
The updated approximations at time $t+1$ as $\underline{\mathcal{A}}_{(\alpha,\beta)}(X), \overline{\mathcal{A}}_{(\alpha,\beta)}(X)$.

1 **begin**
2 $\quad\mathcal{A}\leftarrow P\bigcup Q$;
3 $\quad$**for** $i=1$ *to* m **do**
4 $\quad\quad$**for** *each* BND_i, $\forall x,y\in BND_i$ **do** ;
5 $\quad\quad$**if** $\forall a\in Q, f(x,a)=f(y,a)$ **then** ;
6 $\quad\quad BND_i$ keeps constant;
7 $\quad\quad$**else** $BND_i=BND_{i1}\bigcup BND_{i2}\bigcup\cdots\bigcup BND_{in_i}$;
8 $\quad$**end**
9 $\quad$**Recompute** $BND_P(X)=\bigcup_{i=1,j=1}^{m,n_i}BND_{ij}$;
10 $\quad$**for** $i=1$ *to* m; $j=1$ *to* n_i **do**
11 $\quad\quad\underline{\mathcal{A}}_{(\alpha,\beta)}(X)=\underline{P}(X)\bigcup\{BND_{ij}\mid Pr(BND_{ij},X)\geq\alpha\}$;
12 $\quad\quad\overline{\mathcal{A}}_{(\alpha,\beta)}(X)=\overline{P}(X)-\{BND_{ij}\mid Pr(BND_{ij},X)\leq\beta\}$.
13 $\quad$**end**
14 **end**

The computational complexity of Algorithm 1 is outlined in Table 4. There are three key steps to calculate its time complexity.

Table 4
The computational complexity of Algorithm 1.

Key steps	Computational complexity of Algorithm 1
Steps 1–7	$O(\Sigma_{i=1}^{m}\lvert BND_i\rvert\lvert Q\rvert)$
Step 8	$O(\Sigma_{i=1}^{m}n_i)$
Steps 9–12	$O(\sum_{i=1}^{m}\sum_{j=1}^{n_i}(\lvert BND_{ij}\rvert\lvert X\rvert+2))=O(\sum_{i=1}^{m}\sum_{j=1}^{n_i}\lvert BND_{ij}\rvert\lvert X\rvert)$
Total	$O(\Sigma_{i=1}^{m}(\lvert BND_i\rvert\lvert Q\rvert+n_i+\sum_{j=1}^{n_i}\lvert BND_{ij}\rvert\lvert X\rvert))$

3.4.2. An incremental algorithm for updating approximations when deleting attributes

Second, we consider the scenario of deleting attributes. Suppose $S=(U,P',V,f)$ at time t, the attributes (their set is denoted as Q') go out of the attribute set P' and the system will change as $S''=(U,P'-Q',V,f)$ at time $t+1$, $\mathcal{B}=P'-Q'$. $\mathcal{B}\subseteq P'$, $\forall X\subseteq U$, we have: $\underline{\mathcal{B}}(X)\subseteq\underline{P'}(X), \overline{P'}(X)\subseteq\overline{\mathcal{B}}(X), BND_{P'}(X)\subseteq BND_{\mathcal{B}}(X)$ and $NEG_{\mathcal{B}}(X)\subseteq NEG_{P'}(X)$. The deletion of the attributes Q' may lead to a part of sets in $\underline{P'}(X)$ and $NEG_{P'}(X)$ become $BND_{\mathcal{B}}(X)$. For simplicity and clarity, we divide U into three pair-wise disjoint parts: $U'=POS_{P'}(X)$, $U''=BND_{P'}(X)$ and $U'''=NEG_{P'}(X)$ at time t, where, $U'=\underline{P'}(X)$, $U'/P'=\{P'_1,P'_2,\ldots,P'_u\}$; $U''=\overline{P'}(X)-\underline{P'}(X)$, $U''/P'=\{P''_1,P''_2,\ldots,P''_v\}$; $U'''=U-\overline{P\prime}(X)$, $U'''/P\prime=\{P'''_1,P'''_2,\ldots,P'''_w\}$.

By the discussions in Sections 3.1 and 3.2, an effective strategy for updating approximation is to calculate $\underline{\mathcal{B}}(X)$ and $BND_{\mathcal{B}}(X)$, which can be divided into two steps as follows.

Step 1: Finding $\underline{\mathcal{B}}(X)$.
$\forall P_i\subseteq U', \forall P_j\subseteq U''\bigcup U'''$, there exist two situations.
(1) $\forall x\in P_i, \forall y\in P_j$, if $\exists a\in\mathcal{B}, f(x,a)\neq f(y,a)$, then $P_i\bigcap P_j=\emptyset$. If $\forall P_j\subseteq U''\bigcup U''', P_i\bigcap P_j=\emptyset$ hold, $P_i\subseteq\underline{\mathcal{B}}(X)$ because of $P_i\subseteq X$. We denote these sets $\underline{P'_1}(X)\subseteq\underline{P'}(X)$ satisfy this criterion as Θ_1, and $\Theta_1=\underline{P'_1}(X)$.
(2) $\forall x\in P_i, \forall y\in P_j$, if $\forall a\in\mathcal{B}, f(x,a)=f(y,a)$, then P_i and P_j are merging to one set. $P_i\subseteq X$, $P_j\neg\subseteq X$, so $P_i\bigcup P_j\subseteq BND_{\mathcal{B}}(X)$. We denote $\Theta_2=\bigcup[x]_{\mathcal{B}}$ with $x\in P_i$ which satisfy this criterion. Moreover, suppose $\underline{P'_2}(X)\subseteq\underline{P'}(X)$ satisfy this criterion, and $\underline{P'_2}(X)$ are the emigrations from $\underline{P'}(X)$ to $BND_{\mathcal{B}}(X)$ and $\underline{P'_2}(X)\subseteq\Theta_2$.
Obviously, $U'=\underline{P'}(X)=\underline{P'_1}(X)\bigcup\underline{P'_2}(X)$, $\underline{\mathcal{B}}(X)=\underline{P'_1}(X)=\Theta_1$ and $\underline{P'_2}(X)\subseteq BND_{\mathcal{B}}(X)$.

Step 2: Finding $BND_{\mathcal{B}}(X)$.
We find the $BND_{\mathcal{B}}(X)$ with excluding Θ_2. $\forall P_i\subseteq U'', \forall P_j\subseteq U'''$, there are two scenarios.
(1) $\forall x\in P_i, \forall y\in P_j$, if $\forall a\in\mathcal{B}, f(x,a)=f(y,a)$, then P_i and P_j are merging to one set. $P_i\subseteq BND_{P'}(X)\subseteq BND_{\mathcal{B}}(X), P_i\bigcup P_j\subseteq BND_{\mathcal{B}}(X)$. We denote these sets $NEG_{P'}(X)$ satisfy this criterion as Θ_3, and Θ_3 are the immigrations from $NEG_{P'}(X)$ to $BND_{\mathcal{B}}(X)$.
(2) $\forall x\in P_i, \forall y\in P_j$, if $\exists a\in\mathcal{B}, f(x,a)\neq f(y,a)$, then $P_i\bigcap P_j=\emptyset$. So $P_j\subseteq NEG_{\mathcal{B}}(X)$.
Intuitively, we have $BND_{\mathcal{B}}(X)=BND_{P'}(X)\bigcup\Theta_2\bigcup\Theta_3, \overline{\mathcal{B}}(X)=BND_{P'}(X)\bigcup\Theta_1\bigcup\Theta_2\bigcup\Theta_3$. For simplicity and clarity, we denote $BND'=BND_{\mathcal{B}}(X)$ and $BND'/\mathcal{B}=\{BND'_1,BND'_2,\ldots,BND'_l\}$. $\underline{\mathcal{B}}_{(\alpha,\beta)}(X)$ and $\overline{\mathcal{B}}_{(\alpha,\beta)}(X)$ can be calculated as follows.
$\underline{\mathcal{B}}_{(\alpha,\beta)}(X)=\underline{\mathcal{B}}(X)\bigcup\{BND_i|Pr(BND_i,X)\geqslant\alpha\}$, where, $(i=1,2,\ldots,l)$;
$\overline{\mathcal{B}}_{(\alpha,\beta)}(X)=\overline{\mathcal{B}}(X)-\{BND_i|Pr(BND_i,X)\leqslant\beta\}$. where, $(i=1,2,\ldots,l)$.
We outline the updating process when deleting attributes in Algorithm 2 as follows.

Algorithm 2. The algorithm for updating approximations when deleting attributes.

Input:
(1) $S = (U, P', V, f)$ at time t, the thresholds α and β;
(2) The approximations $\underline{P'}(X)$ and $\overline{P'}(X)$ at time t.
The classifications with positive regions, boundary regions and negative regions at time t as:
$U'/P' = \{P'_1, P'_2, \cdots, P'_u\}$;
$U''/P' = \{P''_1, P''_2, \cdots, P''_v\}$;
$U'''/P' = \{P'''_1, P'''_2, \cdots, P'''_w\}$;
(3) The deleting attribute set Q'.
Output:
The updated approximations at time $t+1$ as $\underline{\mathcal{B}}_{(\alpha,\beta)}(X)$, $\overline{\mathcal{B}}_{(\alpha,\beta)}(X)$.

1 **begin**
2 $\quad \mathcal{B} \leftarrow P' - Q'$;
3 $\quad$ **for** $i = 1$ *to* u, $j = 1$ *to* $v + w$, $\Theta_1 = \emptyset$, $\Theta_2 = \emptyset$ **do**
4 $\quad\quad$ **for** $\forall x \in P_i, P_i \subseteq U'; \forall y \in P_j, P_j \subseteq U'' \bigcup U'''$ **do** ;
5 $\quad\quad$ **if** $\exists a \in \mathcal{B}, f(x,a) \neq f(y,a)$ **then** ;
6 $\quad\quad$ $\Theta_1 \leftarrow \Theta_1 \bigcup P_i$;
7 $\quad\quad$ **else** $\Theta_2 \leftarrow \Theta_2 \bigcup (P_i \bigcup P_j)$;
8 $\quad$ **end**
9 $\quad$ Output $\Theta_1, \Theta_2, \underline{\mathcal{B}}(X) = \Theta_1$.
10 **end**
11 **for** $i = 1$ *to* v, $j = 1$ *to* w, $\Theta_3 = \emptyset$ **do**
12 $\quad$ **for** $\forall x \in P_i - \Theta_2, P_i \subseteq U''; \forall y \in P_j - \Theta_2, P_j \subseteq U'''$ **do** ;
13 $\quad$ **if** $\forall a \in \mathcal{B}, f(x,a) = f(y,a)$ **then** ;
14 $\quad$ $\Theta_3 \leftarrow \Theta_3 \bigcup P_j$;
15 $\quad$ **else** $P_j \in NEG_{\mathcal{B}}(X)$;
16 **end**
17 Output Θ_3.
18 **Recompute** $BND_{\mathcal{B}}(X) = BND_{P'}(X) \bigcup \Theta_2 \bigcup \Theta_3$, $\overline{\mathcal{B}}(X) = BND_{P'}(X) \bigcup \Theta_1 \bigcup \Theta_2 \bigcup \Theta_3$ and we denote $BND_{\mathcal{B}}(X)/\mathcal{B} = \{BND'_1, BND'_2, \cdots, BND'_l\}$;
19 **for** $i = 1$ *to* l **do**
20 $\quad \underline{\mathcal{B}}_{(\alpha,\beta)}(X) = \underline{\mathcal{B}}(X) \bigcup \{BND_i \mid Pr(BND_i, X) \geq \alpha\}$;
21 $\quad \overline{\mathcal{B}}_{(\alpha,\beta)}(X) = \overline{\mathcal{B}}(X) - \{BND_i \mid Pr(BND_i, X) \leq \beta\}$.
22 **end**

The computational complexity of Algorithm 2 is outlined in Table 5. There are five key steps to calculate its time complexity.

Finally, we consider the above two incremental algorithms for updating approximations in PRS with the non-incremental strategies. The non-incremental algorithm retrains the system from scratch when the information system varies. Suppose $U/R = \{R_1, R_2, \ldots, R_k\}$ at time $t+1$, the computational complexity of compute equivalence classes at time $t+1$ is $O(|U||R|)$, the computational complexity of compute approximations is $O(\sum_{i=1}^{k}|R_i||X|)$. So the total computational complexity of non-incremental algorithm is $O(|U||R| + \sum_{i=1}^{k}|R_i||X|)$, where, $R = \mathcal{A}$ under the scenario of adding attributes and $R = \mathcal{B}$ under the scenario of deleting attributes. Compared the incremental algorithm with the non-incremental algorithm, the former one only focuses on the variation of boundary equivalence classes, but the latter one needs to compute all the universe, which is very costly or even intractable, especially the data set is massive.

4. An illustration and experimental evaluations

In this section, we demonstrate our incremental algorithms by using an example first. We then compare our algorithms with the non-incremental algorithm by using 5 data sets from UCI and a genome data with thousand attributes, and finally present the experimental results.

Table 5
The computational complexity of Algorithm 2.

Key steps	Computational complexity of Algorithm 2
Steps 1–9	$O(\lvert B\rvert\sum_{i=1}^{v}\sum_{j=1}^{v+w}\lvert P_i\rvert\lvert P_j\rvert)$
Steps 10–14	$O(\lvert B\rvert\sum_{i=1}^{v}\sum_{j=1}^{w}\lvert P_i - \Theta_2\rvert\lvert P_j - \Theta_2\rvert)$
Steps 15	$O(1)$
Steps 16	$O(\lvert BND_B(X)\rvert\lvert B\rvert)$
Steps 17–20	$O(\sum_{i=1}^{l}(\lvert BND_i\rvert\lvert X\rvert + 2)) = O(\sum_{i=1}^{l}(\lvert BND_i\rvert\lvert X\rvert)$
Total	$O(\lvert B\rvert\sum_{i=1}^{v}(\sum_{j=1}^{v+w}\lvert P_i\rvert\lvert P_j\rvert + \sum_{j=1}^{w}\lvert P_i - \Theta_2\rvert\lvert P_j - \Theta_2\rvert) + \lvert BND_B(X)\rvert\lvert B\rvert + \sum_{i=1}^{l}\lvert BND_i\rvert\lvert X\rvert)$

4.1. An illustration

We illustrate how to utilize the incremental algorithms in Sub-Sections 3.4.1 and 3.4.2 to maintain approximations in PRS dynamically. In the complete information system given in Table 6, $U = \{x_1, x_2, \ldots, x_{12}\}$. Let $P = \{a_1, a_2\}$, $Q = \{a_3, a_4\}$. $\mathcal{A} = P \bigcup Q$, $P' = \{a_1, a_2, a_3, a_4\}$, $Q' = \{a_1, a_4\}$, $\mathcal{B} = P' - Q'$. We set $X = \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}$, $\alpha = 0.6$ and $\beta = 0.4$.

From Table 6, we have $U/P = \{\{x_1\}, \{x_2, x_3, x_4, x_5\}, \{x_6, x_7\}, \{x_8, x_9, x_{10}\}, \{x_{11}, x_{12}\}\}$. $\underline{P}(X) = \{x_1\}$, $\overline{P}(X) = U$. So,
$BND_P(X) = \overline{P}(X) - \underline{P}(X) = \{x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$.
$BND_P(X)/P = \{BND_1, BND_2, BND_3, BND_4\} = \{\{x_2, x_3, x_4, x_5\}, \{x_6, x_7\}, \{x_8, x_9, x_{10}\}, \{x_{11}, x_{12}\}\}$.
According to Algorithm 1,

$$BND_1/Q = \{BND_{11}, BND_{12}\} = \{\{x_2, x_3, x_4\}, \{x_5\}\}.$$
$$BND_2/Q = \{BND_{21}, BND_{22}\} = \{\{x_6\}, \{x_7\}\}.$$
$$BND_3/Q = \{BND_{31}\} = \{x_8, x_9, x_{10}\}.$$
$$BND_4/Q = \{BND_{41}\} = \{x_{11}, x_{12}\}.$$

We obtain:

$$Pr(BND_{11}, X) = \frac{|\{x_2, x_3, x_4\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_2, x_3, x_4\}|} = 0.33,$$
$$Pr(BND_{12}, X) = \frac{|\{x_5\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_5\}|} = 1,$$
$$Pr(BND_{21}, X) = \frac{|\{x_6\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_6\}|} = 1,$$
$$Pr(BND_{22}, X) = \frac{|\{x_7\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_7\}|} = 0,$$
$$Pr(BND_{31}, X) = \frac{|\{x_8, x_9, x_{10}\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_8, x_9, x_{10}\}|} = 0.67,$$
$$Pr(BND_{41}, X) = \frac{|\{x_{11}, x_{12}\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_{11}, x_{12}\}|} = 0.5.$$

$\underline{\mathcal{A}}_{(0.6,0.4)}(X)$ and $\overline{\mathcal{A}}_{(0.6,0.4)}(X)$ can be calculated as:

$$\underline{\mathcal{A}}_{(0.6,0.4)}(X) = \underline{P}(X) \bigcup \{BND_{ij} | Pr(BND_{ij}, X) \geqslant 0.6\}$$
$$= \{x_1\} \bigcup BND_{12} \bigcup BND_{21} \bigcup BND_{31} = \{x_1, x_5, x_6, x_8, x_9, x_{10}\}.$$
$$\overline{\mathcal{A}}_{(0.6,0.4)}(X) = \overline{P}(X) - \{BND_{ij} | Pr(BND_{ij}, X) \leqslant 0.4\}$$
$$= U - BND_{11} - BND_{22} = \{x_1, x_5, x_6, x_8, x_9, x_{10}, x_{11}, x_{12}\}.$$

Similarly, we have $U/P' = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8, x_9, x_{10}\}, \{x_{11}, x_{12}\}\}$, $\underline{P'}(X) = \{x_1, x_5, x_6\}$, $\overline{P'}(X) = U - \{x_7\}$, $BND_{P'} = \overline{P'}(X) - \underline{P'}(X) = \{x_2, x_3, x_4, x_8, x_9, x_{10}, x_{11}, x_{12}\}$ and $NEG_{P'} = \{x_7\}$. According to Algorithm 2, $\forall x \in \underline{P'}(X)$, we have $\{x_1\} \in \Theta_1$, $\{x_5, x_6\} \in \Theta_2$. Since $[x_5]_{\mathcal{B}} = \{x_5, x_7\}, [x_6]_{\mathcal{B}} = \{x_2, x_3, x_4, x_6\}, \Theta_2 = [x_5]_{\mathcal{B}} \bigcup [x_6]_{\mathcal{B}} = \{x_2, x_3, x_4, x_5, x_6, x_7\}$. So, $\underline{\mathcal{B}}(X) = \Theta_1 = \{x_1\}$.

However, $BND_{P'}(X) - \Theta_2 = \{x_8, x_9, x_{10}, x_{11}, x_{12}\}$, $NEG_{P'}(X) - \Theta_2 = \emptyset$, that is, $\Theta_3 = \emptyset$. Hence, $BND_{\mathcal{B}}(X) = BND_{P'}(X) \bigcup \Theta_2 \bigcup \Theta_3 = \{x_2,$

Table 6
A complete information system.

U	a_1	a_2	a_3	a_4	U	a_1	a_2	a_3	a_4
x_1	1	1	1	0	x_7	2	2	2	1
x_2	1	2	1	0	x_8	2	3	3	1
x_3	1	2	1	0	x_9	2	3	3	1
x_4	1	2	1	0	x_{10}	2	3	3	1
x_5	1	2	2	1	x_{11}	3	3	3	2
x_6	2	2	1	1	x_{12}	3	3	3	2

$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}, \overline{\mathcal{B}}(X) = BND_{P'}(X) \bigcup \Theta_1 \bigcup \Theta_2 \bigcup \Theta_3 = U$ and $BND_{\mathcal{B}}(X)/\mathcal{B} = \{BND'_1, BND'_2, BND'_3\} = \{\{x_2, x_3, x_4, x_6\}, \{x_5, x_7\}, \{x_8, x_9, x_{10}, x_{11}, x_{12}\}\}$. We obtain:

$Pr(BND'_1, X) = \frac{|\{x_2, x_3, x_4, x_6\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_2, x_3, x_4, x_6\}|} = 0.5,$

$Pr(BND'_2, X) = \frac{|\{x_5, x_7\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_5, x_7\}|} = 0.5,$

$Pr(BND'_3, X) = \frac{|\{x_8, x_9, x_{10}, x_{11}, x_{12}\} \bigcap \{x_1, x_4, x_5, x_6, x_8, x_9, x_{11}\}|}{|\{x_8, x_9, x_{10}, x_{11}, x_{12}\}|} = 0.6.$

$\underline{\mathcal{B}}_{(0.6,0.4)}(X)$ and $\overline{\mathcal{B}}_{(0.6,0.4)}(X)$ can be calculated as:

$\underline{\mathcal{B}}_{(0.6,0.4)}(X) = \underline{\mathcal{B}}(X) \bigcup \{BND'_i | Pr(BND'_i, X) \geqslant 0.6\} = \{x_1\} \bigcup BND'_3 = \{x_1, x_8, x_9, x_{10}, x_{11}, x_{12}\}$;

$\overline{\mathcal{B}}_{(0.6,0.4)}(X) = \overline{\mathcal{B}}(X) - \{BND'_i | Pr(BND'_i, X) \leqslant 0.4\} = U - \emptyset = U.$

4.2. The experimental evaluations

4.2.1. The basic descriptions of the experimental environment

Experiments were performed on a computer with Inter(R) Core(TM) i7-2670QM @2.20 GHz (8 cores in all) and 8G of memory, running Microsoft Windows 7. Algorithms of incremental updating proposed in this paper and the non-incremental updating were developed in VC++. Followed by the suggestions in [9,21], we choose the data set "Mushroom" [68], publicly available from the UCI Machine Learning Database Repository, as a benchmark data set for the performance tests. The data set "Mushroom" consists of 8124 instances (objects). The number of attributes, all having been nominalized, is 22, namely, cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, spore-print-color, population and habitat. For simplicity, we denoted them as $A = \{a_1, a_2, \ldots, a_{22}\}$, respectively. There are two classes, namely, edible (including 4208 instances) and poisonous (including 3916 instances) [68]. By considering these 2480 instances with missing values (denoted by "?") in a_{11} (stalk-root), we simply define the sign "?" as a character but not a missing value, and data set "Mushroom" converts to a complete information system. The basic statistical information of the data set "Mushroom" is briefly summarized in Table 7.

In Table 7, the first and second columns denote the attributes (including 1 decision class) and their explanations, respectively. The third column illustrates whether it exists missing value for each attribute. The forth column shows the number of classifications for each attribute. Moreover, the letter in the fifth column denotes an abbreviation of a class name, and (•) in the fifth column stands for the cardinal number of a class. In our following discussion, we ignore the true classifications (edible and poisonous) for the 8124 instances.

4.2.2. Performance evaluation of dynamic maintenance of PRS approximations with "Mushroom"

The effectiveness of incremental updating methods under Pawlak rough sets and its extensions have been validated in the literatures [4,9,21]. Here, we mainly discuss the incremental updating model based on PRS when attributes vary. For simplicity, we denote Algorithm 1 and Algorithm 2 as the incremental algorithms when adding and deleting attributes, respectively. To validate our proposed algorithms, we introduce the non-incremental updating algorithm to our research. The non-incremental algorithm means we treat the updating data by variation of attributes as absolutely new data without using any incremental strategy and directly compute the rough set approximations [5,23]. We compare the average computational times between incremental algorithm proposed in this paper and non-incremental algorithm under the data set "Mushroom".

Firstly, we consider the scenario when adding attributes. In the data set "Mushroom", we divide the 22 attributes into two groups. We randomly choose a part of attributes and denote them as P, and the other attributes are denoted as $A - P$. Suppose the data generate by P is the original data set at time t. At time $t+1$, we randomly choose some attributes Q in $A - P$ and add them to P as $\mathcal{A} = P \bigcup Q$. Obviously, $P \bigcap Q = \emptyset$. We repeat the selection processes 4 times and generate the corresponding P_i and $Q_i (i = 1, 2, 3, 4)$ as follows.

P_1 {cap-color, odor, gill-color, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, ring-type, spore-print-color, population, habitat}.

Table 7
The basic statistical information of the data set "Mushroom".

Attributes	Label	Miss data	Classifications	Attribute's value distributions
a_1	cap-shape	No	6	b(452), c(4), f(3152), k(828), s(32), x(3656)
a_2	cap-surface	No	4	f(2320), g(4), s(2556), y(3244)
a_3	cap-color	No	10	b(168), c(44), e(1500), g(1840), n(2284), p(144), r(16), u(16), w(1040), y(1072)
a_4	bruises	No	2	f(4748), y(3376)
a_5	odor	No	9	a(400), c(192), f(2160), l(400), m(36), n(3528), p(236), s(576), y(576)
a_6	gill-attachment	No	4	a(210), f(7914), d(0), n(0)
a_7	gill-spacing	No	3	c(6812), w(1312), d(0)
a_8	gill-size	No	2	b(5612), n(2512)
a_9	gill-color	No	12	b(1728), e(96), g(752), h(732), k(408), n(1048), o(64), p(1492), r(24), u(492),w(1202), y(86)
a_{10}	stalk-shape	No	2	e(3516), t(4608)
a_{11}	stalk-root	Yes	6	b(3776), c(556), e(1120), r(192), u(0), z(0), miss value (2480)
a_{12}	stalk-surface-above-ring	No	4	f(552), k(2372), s(5176), y(24)
a_{13}	stalk-surface-below-ring	No	4	f(600), k(2304), s(4936), y(284)
a_{14}	stalk-color-above-ring	No	9	b(432), c(36), e(96), g(576), n(448), o(192), p(1872), w(4464), y(8)
a_{15}	stalk-color-below-ring	No	9	b(432), c(36), e(96), g(576), n(512), o(192), p(1872), w(4384), y(24)
a_{16}	veil-type	No	2	p(8124), u(0)
a_{17}	veil-color	No	4	n(96), o(96), w(7924), y(8)
a_{18}	ring-number	No	3	n(36), o(7748), t(600)
a_{19}	ring-type	No	8	e(2776), f(48), l(1296),n(36), p(3968), c(0), s(0), z(0)
a_{20}	spore-print-color	No	9	b(48), h(1632), k(1872), n(1968), o(48), r(72), u(48), w(2388), y(48)
a_{21}	population	No	6	a(384), c(340), n(400), s(1248), v(4040), y(1712)
a_{22}	habitat	No	7	d(3148), g(2148), l(832), m(292), p(1144), u(368), w(192)
D	class	No	2	Edible (4208), Poisonous (3916)

Q_1 {bruises, gill-attachment, gill-spacing, veil-type,ring-number}.
P_2 {cap-shape, cap-color, bruises, gill-spacing, gill-color, stalk-root, stalk-surface-below-ring, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, population, habitat}.
Q_2 {cap-surface, gill-attachment, gill-size, stalk-surface-above-ring, stalk-color-above-ring}.
P_3 {cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, veil-color}.
Q_3 {ring-number, ring-type, spore-print-color, population, habitat}.
P_4 {cap-shape, cap-surface, cap-color, odor, gill-attachment, gill-spacing, gill-color, stalk-shape, stalk-root, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-color, ring-number, ring-type, population, habitat}.
Q_4 {bruises, gill-size, stalk-surface-above-ring, veil-type, spore-print-color}.

$P_i \bigcup Q_i$ ($i = 1, 2, 3, 4$) means that the attribute set Q_i is merged with P_i.

In the following, we fix the threshold values with $\alpha = 0.7, \beta = 0.3$ and randomly choose 10% data of 8124 instances as the subset X. Furthermore, we calculate the average elapsed times by repeating the computing process for 100 times. The experimental results between Algorithm 1 and Non-incremental algorithm for calculating the approximations when adding attributes are shown in Table 8.

In Table 8, it is clear that the Algorithm 1 is more effective than the Non-incremental algorithm when adding attributes.

Secondly, we consider the scenario when deleting attributes. Similarly, in the data set "Mushroom", we randomly choose a subset of the attributes and denote it as P' at time t. At time $t+1$, we randomly choose some attributes Q' in $P'(Q' \subset P')$ and delete them from original data set as $\mathcal{B} = P' - Q'$. We repeat the selection processes 4 times and generate the corresponding P'_i and $Q'_i(i = 1, 2, 3, 4)$ as follows.

P'_1 {cap-surface, cap-color, bruises, gill-attachment, gill-spacing, gill-size, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, ring-number, ring-type, spore-print-color, habitat}.
Q'_1 {cap-surface, gill-size, stalk-surface-above-ring, habitat}.
P'_2 {cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-below-ring, stalk-color-below-ring, veil-color, ring-number, ring-type, spore-print-color, population, habitat}.
Q'_2 {cap-shape, cap-surface, spore-print-color, population, habitat}.
P'_3 {cap-shape, cap-surface, cap-color, odor, gill-attachment, gill-spacing, gill-color, stalk-shape, stalk-root, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-color, ring-number, ring-type, spore-print-color, population, habitat}.
Q'_3 {cap-color, gill-attachment, gill-color, stalk-color-below-ring}.
P'_4 {cap-shape, cap-surface, odor, gill-attachment, gill-spacing, gill-size, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, veil-color, ring-number, ring-type, spore-print-color, population, habitat}.
Q'_4 {odor, gill-size, stalk-color-below-ring, ring-number, ring-type}.

$P'_i - Q'_i$
($i = 1, 2, 3, 4$) means that the attribute set Q'_i is deleted from P'_i. In the same way, we fix the threshold values with $\alpha = 0.7, \beta = 0.3$ and randomly choose 10% data of 8124 instances as the subset X. Furthermore, we calculate the average elapsed times by repeating the computing process for 100 times. The experimental results between Algorithm 2 and non-incremental algorithm for calculating the approximations when deleting attributes are shown in Table 9.

In Table 9, Algorithm 2 clearly outperforms the Non-incremental algorithm when deleting attributes.

In addition, we consider the effectiveness of the two incremental algorithms when the subset X is changed. For simplicity, we choose the data set "Mushroom" as a benchmark data set for an example and change the rate of $X \subseteq U$ from 10% to 100% with a step size of 10% for the experimental strategy. We fix $P_i, Q_i(i = 1, 2, 3, 4)$ and gradually increase the rate of X to compare the effectiveness between Algorithm 1 and Non-incremental algorithm when adding attributes. The experimental results are shown in Table 10 and Fig. 7.

From Table 10 and Fig. 7, we find Algorithm 1 is effective and plays much faster than the Non-incremental algorithm. The trends of the two algorithms are approximately linear with the increasing of the rate of X.

Following our above discussions, we also fix $P'_1, Q'_1(i = 1, 2, 3, 4)$ and gradually increase the rate of X from 10% to 100% with a step size of 10% to compare the effectiveness between Algorithm 2 and Non-incremental algorithm when deleting attributes. The experimental results are shown in Table 11 and Fig. 8.

In Table 11 and Fig. 8, the trends of the two algorithms are approximately linear. The average elapsed times in Algorithm 2 outperform the non-incremental algorithm with the increasing of the rate of X. Since the attribute set in an information system evolves in time when new information arrives, the proposed approach helps decision makers to make a quicker and better choice in real applications.

Furthermore, we consider the sensibility of our proposed algorithms when α and β change. For simplicity, we set $\alpha \in [0.5, 1)$ and $\beta \in [0, 1)$ with step 0.1, respectively. Here, we randomly choose 10% objects in "Mushroom" as the subset X. When we consider the scenario of adding attributes, we set $P = \{a_1, a_2, \ldots, a_{20}\}$ and

Table 8
The average elapsed times between Algorithm 1 and Non-incremental algorithm when adding attributes (unit: 10^{-3} s).

Data sets	Non-incremental algorithm	Algorithm 1
$P_1 \bigcup Q_1$	12.524	3.855
$P_2 \bigcup Q_2$	12.766	5.215
$P_3 \bigcup Q_3$	19.796	7.779
$P_4 \bigcup Q_4$	19.749	7.801

Table 9
The average elapsed times between Algorithm 2 and Non-incremental algorithm when deleting attributes (unit: 10^{-3} s).

Data sets	Non-incremental algorithm	Algorithm 2
$P'_1 - Q'_1$	11.96	5.706
$P'_2 - Q'_2$	11.916	8.958
$P'_3 - Q'_3$	12.013	6.048
$P'_4 - Q'_4$	11.011	4.507

Table 10
The average elapsed times by adding Q_i into P_i when the rate of X changes (unit: 10^{-3} s).

Rate of X (%)	Non-incremental algorithm	Algorithm 1	Rate of X (%)	Non-incremental algorithm	Algorithm 1
$P_1 \bigcup Q_1$			$P_2 \bigcup Q_2$		
10	12.524	3.855	10	12.766	5.215
20	15.130	4.995	20	16.817	6.026
30	17.260	6.262	30	18.408	6.961
40	18.688	7.368	40	20.811	8.433
50	21.693	8.731	50	22.901	9.523
60	22.487	9.893	60	24.567	10.470
70	25.401	11.015	70	26.053	11.538
80	27.239	11.799	80	29.346	12.957
90	29.313	13.256	90	31.385	13.931
100	31.319	13.355	100	32.816	14.345
$P_3 \bigcup Q_3$			$P_4 \bigcup Q_4$		
10	19.796	7.779	10	19.749	7.801
20	22.630	8.805	20	21.829	8.825
30	23.760	10.010	30	24.842	10.112
40	25.823	10.432	40	26.833	10.819
50	27.347	11.825	50	27.295	11.688
60	27.966	12.802	60	29.899	12.912
70	30.636	13.801	70	32.473	13.869
80	33.186	14.739	80	33.178	14.486
90	35.658	16.012	90	35.999	15.728
100	37.322	16.744	100	37.141	16.823

$Q = \{a_{21}, a_{22}\}$, the calculating results are shown in Table 12. Similarly, when we consider the scenario of deleting attributes, we set $P' = \{a_1, a_2, \ldots, a_{15}\}$ and $Q' = \{a_1, a_2\}$, the calculating results are shown in Table 13.

From Tables 12 and 13, Algorithm 1 and Algorithm 2 perform always faster than the non-incremental algorithm with the increasing of α and β, which indicate the proposed incremental algorithms are robust and sensible.

4.2.3. Performance evaluation of dynamic maintenance of PRS approximations with other data sets

In this subsection, we choose another 4 data sets, from the well-known machine learning web UC Irvine Machine Learning Database Repository (http://archive.ics.uci.edu/ml/datasets.html), named "IRIS", "Balance-scale", "Cup" and "Soybean" as benchmarks for the performance tests. All data in the 4 data sets is marked with numerical or category types. Table 14 shows the basic statistical information of the 4 data sets, the first and second column represent the number and name of the 4 chosen data sets in UCI database. The third and fourth column stand for the cardinal number of the samples (objects) and features (attributes) of each data set.

For each data set in Table 14, we divide the attributes into two groups. We choose a part of attributes and denote them as P, and some other attributes are denoted as Q. Suppose the data generated by P is the original data set at time t. At time $t+1$, we

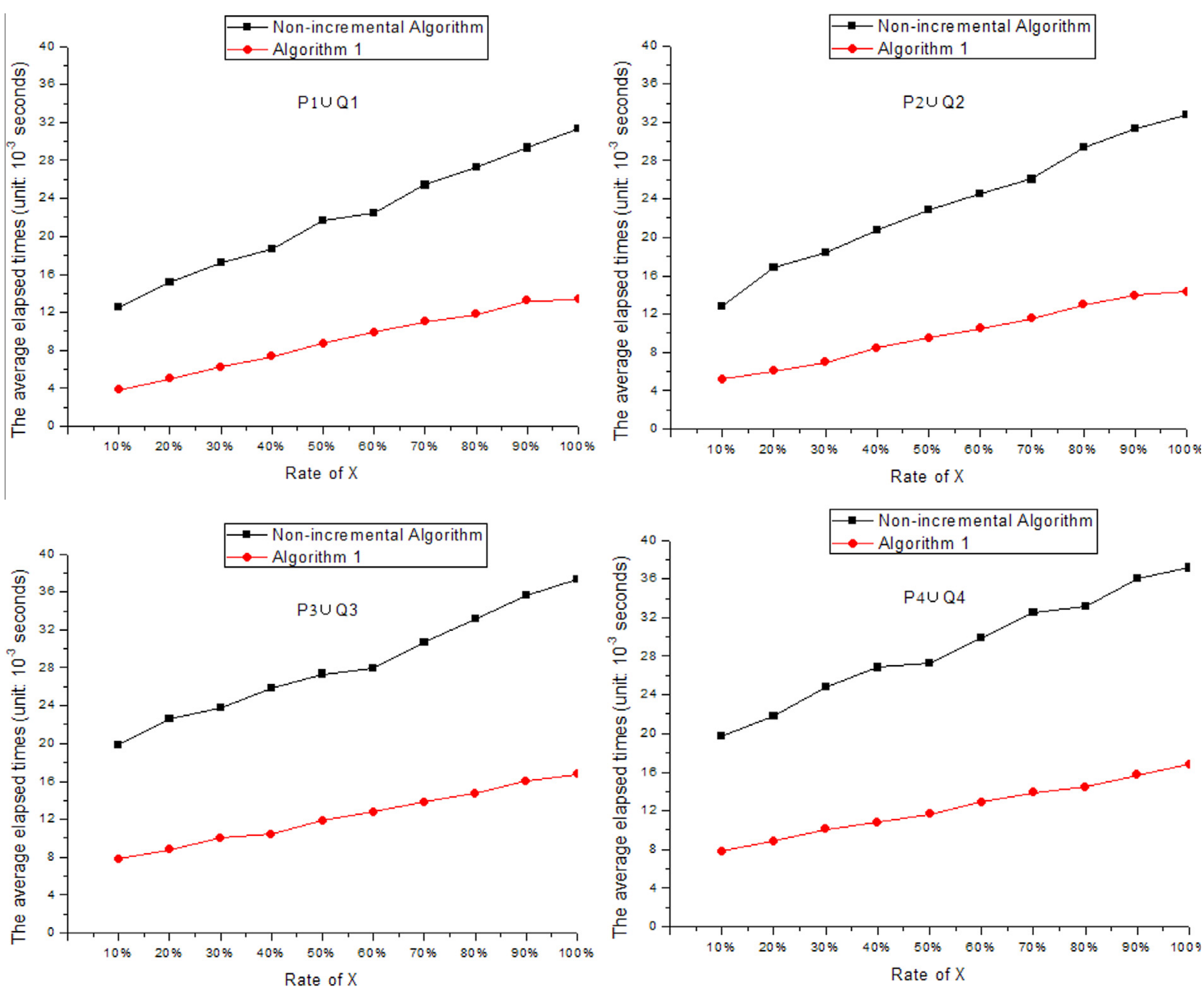

Fig. 7. The average elapsed times adding Q_i into P_i when the rate of X changes (unit: 10^{-3} s).

Table 11
The average elapsed times by deleting Q'_i from P'_i when the rate of X changes (unit: 10^{-3} s).

Rate of X (%)	Non-incremental algorithm	Algorithm 2	Rate of X (%)	Non-incremental algorithm	Algorithm 2
$P'_1 - Q'_1$			$P'_2 - Q'_2$		
10	11.960	5.706	10	11.916	8.958
20	14.040	6.529	20	13.185	9.051
30	16.537	6.553	30	18.737	11.001
40	20.051	7.496	40	19.150	10.159
50	22.940	8.142	50	21.779	10.937
60	23.538	7.736	60	23.488	11.176
70	25.743	8.371	70	26.559	12.197
80	28.368	9.056	80	29.517	12.895
90	29.905	9.255	90	31.680	12.306
100	32.572	9.011	100	33.729	13.204
$P'_3 - Q'_3$			$P'_4 - Q'_4$		
10	12.013	6.048	10	11.011	4.507
20	14.196	6.517	20	13.043	4.647
30	17.342	6.954	30	15.001	4.693
40	19.877	8.258	40	18.638	5.892
50	21.771	7.794	50	19.429	5.796
60	23.286	8.204	60	21.053	6.046
70	24.671	8.689	70	22.973	6.770
80	28.560	9.320	80	23.369	6.524
90	29.294	9.290	90	26.425	7.318
100	31.057	9.288	100	28.644	7.268

randomly choose some attributes in Q and add them to P as $\mathcal{A} = P \bigcup Q$. Then, we fix the threshold values with $\alpha = 0.7, \beta = 0.3$ and randomly choose 10% samples in each data set as the subset X. Furthermore, we calculate the average elapsed times by repeating the computing process for 100 times. The experimental results for the 4 data sets between Algorithm 1 and non-incremental algorithm with the addition of attributes are shown in Table 15.

Similarly, for each data set in Table 14, we denote all the attributes as P' at time t. At time $t+1$, we randomly choose some attributes Q' in $P'(Q' \subset P')$ and delete them from original data set as $\mathcal{B} = P' - Q'$. Then, we fix the threshold values with $\alpha = 0.7, \beta = 0.3$ and randomly choose 10% samples in each data set as the subset X. Furthermore, we calculate the average elapsed times by repeating the computing process for 100 times. The experimental results for the 4 data sets between Algorithm 2 and non-incremental algorithm with the removal of attributes are shown in Table 16.

Finally, we consider the performance of the proposed algorithms in the case of databases with thousand attributes (including 800 samples and 1000 attributes). We use one of the genome data from [56] (Data source: "Case 1 – Disease loci with main effects", http://bioinformatics.ust.hk/BOOST.html). Since our proposed algorithms only deal with categorical attributes, the 1000 numeric attributes are discretized firstly. In our experiments, we fix the threshold values with $\alpha = 0.7, \beta = 0.3$ and randomly

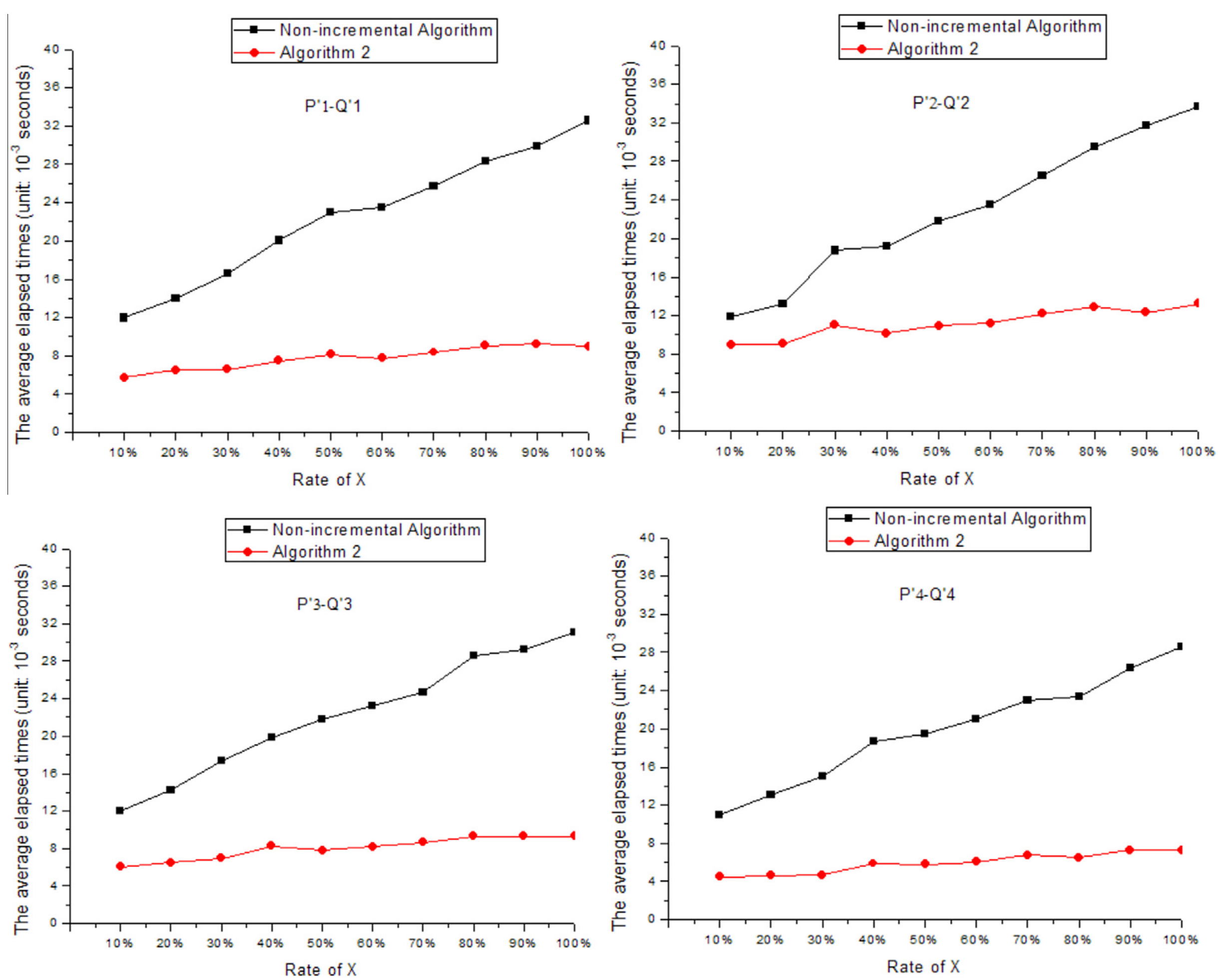

Fig. 8. The average elapsed times deleting Q'_i from P'_i when the rate of X changes (unit: 10^{-3} s).

Table 12
The average elapsed times on adding attributes when α and β change (unit: 10^{-3} s)

$\alpha\downarrow$	$\beta\rightarrow$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.5	Non	46.969	17.469	16.746	16.962	16.812	17.277	16.978	16.461	16.684	16.835	16.745
	Alg.1	7.729	7.986	7.722	7.857	7.349	7.951	7.654	7.034	7.745	7.736	7.910
0.6	Non	17.204	17.435	17.010	17.049	16.754	16.677	16.753	16.561	16.775	16.554	16.663
	Alg.1	7.947	8.125	7.891	7.611	7.442	7.089	7.151	7.106	7.125	6.940	7.056
0.7	Non	17.021	17.302	16.977	17.375	16.556	16.748	16.496	16.569	16.520	16.874	16.692
	Alg.1	7.694	8.082	8.247	7.604	7.818	7.370	7.123	7.150	7.819	7.143	7.049
0.8	Non	17.119	17.132	17.306	16.961	16.561	16.133	16.731	16.634	16.679	16.779	16.627
	Alg.1	7.787	7.714	7.651	7.619	7.133	6.985	7.177	7.327	7.099	7.123	7.303
0.9	Non	16.925	16.957	16.805	17.023	16.645	16.58	16.737	16.539	16.387	16.451	16.477
	Alg.1	8.310	7.545	7.551	8.004	7.616	7.265	7.103	7.238	7.164	7.199	7.226
1.0	Non	17.515	17.003	17.027	16.955	17.135	16.736	16.682	17.355	17.135	16.620	16.672
	Alg.1	7.694	7.910	7.465	8.097	7.234	7.744	7.168	7.206	7.121	6.902	7.474

Table 13
The average elapsed times on deleting attributes when α and β change (unit: 10^{-3} s).

$\alpha\downarrow$	$\beta\rightarrow$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.5	Non	38.511	18.823	10.075	9.773	9.749	9.706	9.734	9.859	9.723	9.737	9.682
	Alg.2	11.697	10.769	5.832	5.757	5.581	5.729	5.747	5.759	5.651	5.732	5.975
0.6	Non	12.940	11.564	10.204	9.769	9.436	9.809	9.783	9.595	10.075	9.838	9.509
	Alg.2	7.091	6.674	5.856	5.717	5.630	5.633	5.728	5.705	5.443	5.648	5.705
0.7	Non	12.991	11.601	10.163	9.852	9.677	9.804	9.831	9.795	9.584	9.793	9.638
	Alg.2	6.858	6.553	5.804	5.700	5.909	5.704	5.632	5.691	5.815	5.460	5.600
0.8	Non	12.47	11.496	10.022	9.745	9.873	9.987	9.773	9.803	9.734	9.651	9.767
	Alg.2	7.367	6.477	5.882	5.829	5.669	5.661	5.656	6.058	5.834	5.703	5.738
0.9	Non	12.611	11.802	9.943	9.549	9.608	9.938	9.895	9.811	9.538	9.725	9.620
	Alg.2	7.116	6.257	5.849	6.153	5.702	5.712	5.779	5.645	5.885	5.778	5.619
1.0	Non	12.578	11.611	10.075	9.557	9.390	9.704	9.913	9.987	9.709	9.820	9.797
	Alg.2	7.068	6.447	5.974	5.697	5.709	5.752	5.752	5.674	5.697	5.866	5.941

Table 14
The basic information of the four databases.

No.	Data sets name	Samples (objects)	Features (attributes)
1	IRIS	150	4
2	Balance-scale	625	4
3	Cup	209	6
4	Soybean	682	35

choose 10% data as the subset X. Then, we randomly select a half attributes (501 attributes) as the original attribute sets. Then we do 5 groups of experiments, namely, randomly add 50, 100, 150, 200, 250 other attributes to the original attribute sets and randomly delete 51, 101, 151, 201, 251 attributes from the original attribute sets, respectively. The experimental results are outlined in Table 17.

According to Tables 15–17, we find Algorithms 1 and 2 are faster than non-incremental algorithm on updating approximations when attributes are added into or deleted from the original attribute set in probabilistic information system, and they are also effective in the databases with thousand attributes.

Table 15
The average elapsed times between Algorithm 1 and non-incremental algorithm with the addition of attributes (unit: 10^{-3} s).

Data sets name	P	Q	Non-incremental algorithm	Algorithm 1
IRIS	$\{a_1, a_2, a_3\}$	$\{a_4\}$	0.560	0.227
Balance-scale	$\{a_1, a_2\}$	$\{a_3\}$	1.732	0.767
Cup	$\{a_1, a_2, a_3, a_4\}$	$\{a_5, a_6\}$	0.819	0.424
Soybean	$\{a_6, a_7, \ldots, a_{18}\}$	$\{a_1, a_2, \ldots, a_6\}$	4.332	1.903

Table 16
The average elapsed times between Algorithm 2 and non-incremental algorithm with the removal of attributes (unit: 10^{-3} s).

Data sets name	P'	Q'	Non-incremental algorithm	Algorithm 2
IRIS	$\{a_1, a_2, a_3, a_4\}$	$\{a_4\}$	0.227	0.126
Balance-scale	$\{a_1, a_2, a_3\}$	$\{a_3\}$	1.359	0.808
Cup	$\{a_1, a_2, \ldots, a_6\}$	$\{a_5, a_6\}$	0.378	0.300
Soybean	$\{a_1, a_2, \ldots, a_{18}\}$	$\{a_1, a_2, \ldots, a_5\}$	3.326	2.531

Table 17
Performance evaluation of dynamic maintenance of PRS approximations with a thousand attributes data set (unit: 10^{-3} s).

Original attributes	Add attributes	Non.	Alg.1	Original attributes	Delete attributes	Non.	Alg.2
501	50	13.361	1.880	501	51	6.573	1.408
501	100	9.362	1.241	501	101	6.205	1.478
501	150	8.452	1.310	501	151	5.697	1.611
501	200	9.045	1.463	501	201	5.351	1.837
501	250	9.564	1.582	501	251	4.884	1.666

5. Conclusion

In this paper, we introduce PRS to our research for incremental updating approximations when attributes vary. The variation of attributes under Pawlak rough sets only leads to one-way translation changes for objects, but the variation of attributes under PRS may lead to two-way translation changes for objects. A block based strategy with boundary sets under PRS is presented to design the corresponding algorithms. Two incremental algorithms for updating the lower and upper approximations under PRS are proposed with the addition and removal of attributes, respectively. Finally, we utilize 5 UCI data sets and a genome data set with thousand attributes to validate the effectiveness of the proposed algorithms. Our future research work will try to extend our approach to incomplete information systems, and to explore if the proposed approach can be extended to other generalized rough sets models.

Acknowledgements

This work is partially supported by the National Science Foundation of China (Nos. 71201133, 61175047, 71090402/G1), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630127), the Research Fund for the Doctoral Program of Higher Education of China (No. 20120184120028), the China Postdoctoral Science Foundation (Nos. 2012M520310, 2013T60132) and the Fundamental Research Funds for the Central Universities of China (No. SWJTU12CX117).

References

[1] L.P. An, Y.H. Wu, L.Y. Tong, Rough set approach to incremantal acquisition of rules, J. NaiKai Univ. 36 (2003) 98–103.
[2] W.C. Bang, Z. Bien, New incremental learning algorithm in the framework of rough set theory, Int. J. Fuzzy Syst. 1 (1999) 25–36.
[3] C.C. Chan, Incremental learning of production rules from examples under uncertainty: a rough set approach, Int. J. Softw. Eng. Knowl. Eng. 1 (1991) 439–461.
[4] C.C. Chan, A rough set approach to attribute generalization in data mining, Inform. Sci. 107 (1998) 177–194.
[5] H.M. Chen, T.R. Li, S.J. Qiao, D. Ruan, A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values, Int. J. Intell. Syst. 25 (2010) 1005–1026.
[6] H.M. Chen, T.R. Li, D. Ruan, Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining, Knowl.-Based Syst. 31 (2012) 140–161.
[7] H.M. Chen, T.R. Li, D. Ruan, J.H. Lin, C.X. Hu, A rough-set based incremental approach for updating approximations under dynamic maintenance environments, IEEE Trans. Knowl. Data Eng. 25 (2013) 274–284.
[8] H.M. Chen, T.R. Li, C. Luo, S.J. Horng, G.Y. Wang, A rough set-based method for updating decision rules on attribute values' coarsening and refining, IEEE Trans. Knowl. Data Eng. (2014), http://dx.doi.org/10.1109/TKDE.2014.2320740.
[9] Y. Cheng, The incremental method for fast computing the rough fuzzy approximations, Data Knowl. Eng. 70 (2011) 84–100.
[10] C. Davide, Classification of dynamics in rough sets, in: Proceedings of 7th International Conference on Rough Sets and Current Trends in Computing, LNAI, 6086, 2010, pp. 257–266.
[11] D. Ciucci, Temporal dynamics in information tables, Fundam. Inform. 115 (2012) 57–74.
[12] J.W. Grzymala-Busse, Knowledge acquisition under uncertainty – a rough set approach, J. Intell. Robot. Syst. 1 (1988) 3–16.
[13] F. Hu, G.Y. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets, in: RSFDGrC2005, LNAI, 3641, 2005, pp. 185–193.
[14] Q.H. Hu, D.R. Yu, J.F. Liu, C.X. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inform. Sci. 178 (2008) 3577–3594.
[15] Q.H. Hu, J.F. Liu, D.R. Yu, Mixed feature selection based on granulation and approximation, Knowl.-Based Syst. 21 (2008) 294–304.
[16] X.Y. Jia, L. Shang, J.J. Chen, Incremental versus non-incremental: data and algorithms based on ordering relations, Int. J. Comput. Intell. Syst. 4 (2011) 112–122.
[17] J.D. Katzberg, W. Ziarko, Variable precision rough sets with asymmetric bounds, in: W. Ziarko (Ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer, London, 1994, pp. 167–177.
[18] H. X Li, M.H. Wang, X.Z. Zhou, J.B. Zhao, An interval set model for learning rules from incomplete information table, Int. J. Approx. Reason. 53 (2012) 24–37.
[19] S.Y. Li, T.R. Li, D. Liu, Dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set, Int. J. Intell. Syst. 28 (2013) 729–751.
[20] S.Y. Li, T.R. Li, D. Liu, Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set, Knowl.-Based Syst. 40 (2013) 17–26.
[21] T.R. Li, D. Ruan, W. Geert, J. Song, Y. Xu, A rough sets based characteristic relation approach for dynamic attribute generalization in data mining, Knowl.-Based Syst. 20 (2007) 485–494.
[22] D. Liu, T.R. Li, D. Ruan, W.L. Zou, An incremental approach for inducing knowledge from dynamic information systems, Fundam. Inform. 94 (2009) 245–260.
[23] D. Liu, T.R. Li, D. Ruan, J.B. Zhang, Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, J. Global Optim. 51 (2011) 325–344.
[24] D. Liu, T.R. Li, H.M. Chen, X.L. Ji, Approaches to knowledge incremental learning based on the changes of attribute values, in: Proceedings of the 4th International Conference on Intelligent Systems and Knowledge Engineering, 2009, pp. 94–99.
[25] D. Liu, T.R. Li, G.R. Liu, P. Hu, An approach for inducing interesting incremental knowledge based on the change of attribute values, in: Proceedings of 2009 IEEE International Conference on Granular Computing, 2009, 415–418.
[26] D. Liu, J.B. Zhang, T.R. Li, A probabilistic rough set approach for incremental learning knowledge on the change of attribute, in: Proceedings of 2010 International Conference on Foundations and Applications of Computational Intelligence, 2010, pp. 722–727.
[27] D. Liu, T.R. Li, J.B. Zhang, An incremental approach for rule induction under coarsening and refining of attribute values in e-business systems, in: Proceedings of 2010 International Conference on Electronic-Business Intelligence, 2010, pp. 541–547.
[28] D. Liu, T.R. Li, D. Ruan, Probabilistic model criteria with decision-theoretic rough sets, Inform. Sci. 181 (2011) 3709–3722.
[29] D. Liu, T.R. Li, J.B. Zhang, A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems, Int. J. Approx. Reason. 55 (2014) 1764–1786.
[30] D. Liu, D.C. Liang, Incremental learning researches on rough set theory: status and future, Int. J. Rough Sets Data Anal. 1 (2014) 99–112.
[31] C. Luo, T.R. Li, H.M. Chen, D. Liu, Incremental approaches for updating approximations in set-valued ordered information systems, Knowl.-Based Syst. 50 (2013) 218–233.
[32] C. Luo, T.R. Li, H.M. Chen, Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization, Inform. Sci. 257 (2014) 210–228.
[33] F. Min, H.P. He, Y.H. Qian, W. Zhu, Test-cost-sensitive attribute reduction, Inform. Sci. 181 (2011) 4928–4942.
[34] Z. Pawlak, Rough sets, Int. J. Comput. Inform. Sci. 11 (1982) 341–356.
[35] Z. Pawlak, S.K.M. Wong, W. Ziarko, Rough sets: probabilistic versus deterministic approach, Int. J. Man–Machine Stud. 29 (1988) 81–95.
[36] Z. Pawlak, Rough Wets, Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers Press, Dordrecht, 1991.
[37] Z. Pawlak, A. Skowron, Rough membership functions, in: Advances in the Dempster-Shafer Theory of Evidence, John Wiley and Sons, New York, 1994, pp. 251–271.
[38] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inform. Sci. 177 (2007) 3–27.
[39] Y.H. Qian, C.Y. Dang, J.Y. Liang, H.Y. Zhang, J.M. Ma, On the evaluation of the decision performance of an incomplete decision table, Data Knowl. Eng. 65 (2008) 373–400.
[40] Y.H. Qian, J.Y. Liang, Y.Y. Yao, C.Y. Dang, MGRS: a mulit-granulation rough set, Inform. Sci. 180 (2010) 949–970.
[41] Y.H. Qian, J.Y. Liang, C.Y. Dang, Incomplete multigranulation rough set, IEEE Trans. Syst. Man Cybernet. – Part A 40 (2010) 420–431.

[42] Y.H. Qian, J.Y. Liang, P. Witold, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (2010) 597–618.
[43] L. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, Comput. Intell. 11 (1995) 357–370.
[44] A. Skowron, J. Stepaniuk, Tolerance approximation spaces, Fundam. Inform. 27 (1996) 245–253.
[45] D. Śle zak, Rough sets and bayes factor, LNCS Transactions on Rough Sets III, LNCS 3400, 2005, pp. 202–229.
[46] L.Y. Tong, L.P. An, An incremental learning of decision rules based on rough set theory, in: Proceedings of the World Congress on Intelligent Control and Automation, 2002, pp. 420–425.
[47] S.K.M. Wong, W. Ziarko, Comparison of the probabilistic approximate classification and the fuzzy set model, Fuzzy Sets Syst. 21 (1987) 357–362.
[48] G. Xie, J. Zhang, K.K. Lai, L. Yu, Variable precision rough set for group decision-making: an application, Int. J. Approx. Reason. 49 (2008) 331–343.
[49] Y.Y. Yao, S.K.M. Wong, A decision theoretic framework for approximating concepts, Int. J. Man–Machine Stud. 37 (1992) 793–809.
[50] Y.Y. Yao, Probabilistic approaches to rough sets, Expert Syst. 20 (2003) 287–297.
[51] Y.Y. Yao, A partition model of granular computing, LNCS Transactions on Rough Sets I, LNCS 3100, 2004, pp. 232–253.
[52] Y.Y. Yao, Probabilistic rough set approximations, Int. J. Approx. Reason. 49 (2008) 255–271.
[53] Y.Y. Yao, Two semantic issues in a probabilistic rough set model, Fundam. Inform. 108 (2011) 249–265.
[54] X.B. Yang, D.J. Yu, J.Y. Yang, L.H. Wei, Dominance-based rough set approach to incomplete interval-valued information system, Data Knowl. Eng. 68 (2009) 1331–1347.
[55] X.B. Yang, Y. Qi, H.L. Yu, X.N. Song, J.Y. Yang, Updating multigranulation rough approximations with increasing of granular structures, Knowl.-Based Syst. 64 (2014) 59–69.
[56] X. Wan, C. Yang, Q. Yang, H. Xue, X. Fan, N. Tang, W. Yu, BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies, Am. J. Human Genet. 87 (2010) 325–340.
[57] A.P. Zeng, T.R. Li, J.B. Zhang, H.M. Chen, Incremental maintenance of rough fuzzy set approximations under the variation of object set, Fundam. Inform. 132 (2014) 401–422.
[58] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Neighborhood rough sets for dynamic data mining, Int. J. Intell. Syst. 27 (2012) 317–342.
[59] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, Int. J. Approx. Reason. 53 (2012) 620–635.
[60] J.B. Zhang, T.R. Li, D. Ruan, Z.Z. Gao, C.B. Zhao, A parallel method for computing rough set approximations, Inform. Sci 194 (2012) 209–223.
[61] J.B. Zhang, J.S. Wong, T.R. Li, Y. Pan, A comparison of parallel large-scale knowledge acquisition using rough set theory on different mapReduce runtime systems, Int. J. Approx. Reason. 55 (2014) 896–907.
[62] J.B. Zhang, T.R. Li, H.M. Chen, Composite rough sets for dynamic data mining, Inform. Sci. 257 (2014) 81–100.
[63] J.B. Zhang, J.S. Wong, T.R. Li, Y. Pan, A parallel matrix-based method for computing approximations in incomplete information systems, IEEE Trans. Knowl. Data Eng., http://dx.doi.org/10.1109/TKDE.2014.2330821.
[64] Z. Zheng, G.Y. Wang, RRIA: a rough set and rule tree based incremental knowledge acquisition algorithm, Fundam. Inform. 59 (2004) 299–313.
[65] W. Ziarko, Variable precision rough set model, J. Comput. Syst. Sci. 46 (1993) 39–59.
[66] W. Ziarko, Probabilistic approach to rough sets, Int. J. Approx. Reason. 49 (2008) 272–284.
[67] W. Ziarko, Probabilistic rough sets, in: Proceedings of 10th International Conference of Rough Sets, Fuzzy Sets, Data Mining & Granular Computing, 2005, pp. 283–293.
[68] http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.names.