Dynamic Variable Precision Rough Set Approach for Probabilistic Set-valued Information Systems ☆

Yanyong Huang[a], Tianrui Li[a,*], Chuan Luo[b], Hamido Fujita[c], Shi-jinn Horng[a]

[a] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*
[b] *College of Computer Science, Sichuan University, Chengdu 610065, China*
[c] *Faculty of Software and Information Science, Iwate Prefectural University, 020-0693, Iwate, Japan*

Abstract

Set-valued information systems are important type of data tables in many real applications, where the attribute values are described by sets to characterize uncertain and incomplete information. However, in some real situations, set-values may be depicted by probability distributions, which results in that the traditional tolerance relation based on intersection operation could not reasonably describe the indiscernibility relation of objects. To address this issue, we introduce the concept of probabilistic set-valued information systems (PSvIS), and present the extended variable precision rough set model (VPRS) based on the λ-tolerance relation in terms of Bhattacharyya distance. Considering the features of information systems will evolve over time in a dynamic data environment, it will lead to the change of information granulation and approximation structures. A matrix representation of rough approximation is presented based on two matrix operators and two vector functions in PSvIS. Then incremental mechanisms by the utilization of previously learned approximation results and region relation matrices for updating rough approximations are proposed, and the corresponding algorithms are developed and analyzed. Experimental results show that the proposed algorithms outperform the static algorithms and related incremental algorithms while inserting into or removing from attributes in PSvIS.

Keywords: Set-valued information systems, Incremental learning, Rough sets, Matrix operators.

1. Introduction

Rough set theory (RST), originated by Pawlak, has become an effective mathematical tool for dealing with uncertain or inconsistent information [1]. Since the equivalence relation in RST is too restrictive to be employed in practical applications, various extended rough set models based on different binary relations have been developed in recent two decades [2, 3, 4, 5, 6, 7]. Nowadays, RST has been widely applied in machine learning, data mining, pattern recognition and knowledge discovery [8, 9, 10, 11, 12].

Set-valued information systems (SvIS) are important generalized models of single-valued information systems, in which sets are used to characterize the imprecise and missing information. Orlowska et al. studied SvIS based on non-deterministic information and introduced the concept of non-deterministic information system [13]. Yao et al. explicitly introduced SvIS and presented some set-based operations for set-valued data [14]. Guan et al. presented three different relative reducts via the maximal tolerance relation in SvIS [15]. Dai et al. constructed a fuzzy rough set model for set-valued data and presented corresponding method for attribute reduction [16]. Wei et al. presented two different fuzzy rough set models based on the fuzzy similarity class and the fuzzy similar degree, respectively [17]. Considering the queuing problems in the presence of multiple criteria, Qian et al. presented two set-valued ordered

☆This is an extended version of the paper presented at the 2016 International Joint Conference on Rough Sets (IJCRS 2016), Santiago 2016, Chile.

*Corresponding author.

Email addresses: yyhswjtu@163.com (Yanyong Huang), trli@swjtu.edu.cn (Tianrui Li), cluo@scu.edu.cn (Chuan Luo), HFujita-799@acm.org (Hamido Fujita), horngsj@yahoo.com.tw (Shi-jinn Horng)

Preprint submitted to Elsevier *January 22, 2017*

information systems in terms of disjunctive and conjunctive semantics and constructed two extended dominance-based rough set models [18]. Zhang et al. combined quantitative rough sets and dominance-based rough sets for dealing with feature selection and approximate reasoning in large-scale set-valued decision systems [19]. Although SvIS have been investigated extensively, no previous study has specifically focused on that set values of SvIS are described by probability distributions. There are many research work on interval-valued data with continuous distribution, which could not be directly applied in the set-valued data with probability distribution due to its distribution is discrete [20, 21, 22]. Set-valued data with discrete distribution exist in many real world situations. For example, in a language-test information system, a set value {German, Polish} under the test attribute of spoken language indicates that a candidate can speak German and Polish in terms of conjunctive semantic [18]. However, in reality, the candidate may speak German fluently, but a little Polish. In order to describe this phenomenon more accurately, we distinguish the ability of spoken language by characterizing the set value with a discrete probability distribution, such as $\frac{\{\text{German,Polish}\}}{\{0.99,0.01\}}$. The information systems with such probability distribution of data are suggested as Probabilistic Set-valued Information Systems (PSvIS) in our study. Moreover, the traditional tolerance relation based on intersection operation in SvIS could not be applied directly in PSvIS. For instance, two candidates with set values $\frac{\{\text{German,Polish}\}}{\{0.99,0.01\}}$ and $\frac{\{\text{German,Polish}\}}{\{0.01,0.99\}}$ under the attribute "speaking a language" are indiscernible according to the tolerance relation. However, it is not reasonable in terms of the probability distributions, i.e., two people where one speaks well in German and only a little Polish and the other is reverse are in the same tolerance class. Furthermore, considering that the classical rough set model is sensitive to misclassified and noisy data, Ziarko presented a robust model, e.g., variable precision rough set model (VPRS), which allows some degree of partial classification by introducing the majority inclusion degree [23]. VPRS has been widely applied in various fields, such as water demand prediction [23], economic and financial prediction [24, 25], medical decision making [26, 27, 28], and so on. Motivated by these considerations, this paper presents the λ-tolerance relation based on Bhattacharyya distance and the extended VPRS approach for PSvIS.

Another important issue inspiring this work is that the data will continuously update due to the new data are added and the outdated data are discarded in a dynamic information system. This paper focus on the variation of attributes, which is a common case in a dynamic data environment. For example, patient symptoms in clinical trials [29], texture features in image processing [30], geographic attributes in environmental monitoring [31]. For the emergence of a novel attribute set or the disappearance of an outdated attribute set in an information system, the static learning approach needs to retrain the whole model on the entire updated data, which is time-consuming or even infeasible for real-time decision making. To improve the computational efficiency with the variations of attributes, incremental learning incorporating with RST has been explored by utilizing the accumulated knowledge for analyzing the newly updated data [32, 33, 34]. Li et al. presented incremental strategies for computing approximations with respect to the characteristic relation in the incomplete information systems [35]. Luo et al. developed matrix-based incremental approach for updating approximations in set-valued ordered information systems [36]. Yang et al. proposed incremental algorithms for maintaining multigranulation rough approximations under dynamic granulation [37]. Wang et al. investigated incremental mechanisms based on three different information entropies for attribution reduction [38]. Shu et al. introduced incremental approaches of calculating the positive region and tolerance classes for positive region-based attribute reduction [39]. Since calculation of approximations is a necessary step for attribute reduction and knowledge discovery in RST [40, 41], this study aims at investigation of the incremental mechanisms to speed up the calculation of rough approximations when evolving features in PSvIS.

Since the matrix form is benefit for intuitive description, simplifying calculation and easy maintainability, it has been widely employed for rough data analysis [42, 43, 44]. Zhang et al. presented four cut matrices for constructing rough approximations and developed incremental algorithms to update approximations under attribute generalization in SvIS [45]. Wang et al. introduced Boolean matrices for representing covering approximation operators and developed a corresponding decomposition algorithm [46]. Tan et al. presented matrix-based approaches for calculating set approximations and reducts in a covering decision information system [47]. Luo et al. investigated the matrix representation of probabilistic rough approximations and designed dynamic algorithms for updating approximations when adding or deleting objects [48]. Inspired by these advantages of matrix, this paper presents a matrix-based method for dynamic maintenance approximations in PSvIS. Firstly, we present a matrix characterization of rough approximations based on the relation matrix and two vector functions associated with multiplication and dot divide operators. Considering that attributes will evolve over time in a dynamic PSvIS, which will result in the changes of information granulation and approximation structures. Hence, we develop incremental mechanisms for maintenance of approximations by utilizing previously accumulated approximation results and region relation matrices. Finally, a series of comparative experiments are conducted to demonstrate the effectiveness of proposed incremental methods.

The main contributions include: (i) we proposed an extension of SvIS, viz., PSvIS, which depict set-valued objects with probability distribution. (ii) Bhattacharyya distance is adopted to measure the similarity degree of objects in PSvIS, and VPRS is extended by the introduction of $\lambda-$tolerance relation. (iii) two matrix operators and two vector functions are designed to characterize the matrix representation of rough approximations. (iv) two incremental mechanisms of updating approximations are presented under the variation of attributes in PSvIS. Theoretical and experimental results demonstrated the efficiency of the proposed method compared with the static and existing incremental approaches. The rest of this paper is organized as follows. In Section 2, some preliminary concepts of RST and VPRS are briefly reviewed. Then the concept of PSvIS and the extended VPRS based on λ-tolerance relation is presented. In Section 3, we construct rough approximations based on matrix approaches and discuss the related properties. In Section 4, by introducing the concept of region matrices, incremental mechanisms for maintaining rough approximations are presented with respect to the variation of attributes. Incremental algorithms are presented for updating rough approximations in Section 5. Section 6 reports experimental results, and the conclusions are presented in Section 7.

2. Preliminaries

In this section, we review some basic concepts and notations of classical rough set model and VPRS. More details can be found in [1, 23]. Then we introduce the basic concept of PSvIS and present λ-tolerance relation in terms of Bhattacharyya distance for extending VPRS model in PSvIS.

Definition 2.1. *[1] Let $S = \{U, AT = C \bigcup D, V, f\}$ be an information system, where U is a non-empty finite set of objects, called the universe; AT is a non-empty finite set of attributes including condition attributes C and decision attributes D; $V = \bigcup_{a \in AT} V_a$ and V_a is a domain of attribute a; $f : U \times AT \to V$ is an information function such that $f(x, a) \in V_a$ for every $a \in AT$, $x \in U$. $\forall X \subseteq U$ and $B \subseteq A$, the lower and upper approximations of X with respect to the equivalence relation R_B are respectively defined as:*

$$\underline{R_B}(X) = \{x | [x]_{R_B} \subseteq X\} \tag{1}$$

$$\overline{R_B}(X) = \{x | [x]_{R_B} \cap X \neq \emptyset\} \tag{2}$$

where $[x]_{R_B} = \{y | (x, y) \in R_B\}$ is the equivalence class determined by the equivalence relation $R_B = \{(x, y) \in U \times U | f(x, b) = f(y, b), \forall b \in B\}$.

According to the lower and upper approximations, the positive, negative and boundary regions of X are easy to obtain as follows.

$$\begin{cases} \text{The positive region: } POS_B(X) = \underline{R_B}(X) \\ \text{The negative region: } NEG_B(X) = U - \overline{R_B}(X) \\ \text{The boundary region: } BND_B(X) = \overline{R_B}(X) - \underline{R_B}(X) \end{cases} \tag{3}$$

Although classical rough set model has been applied in various fields, it is not robust for dealing with the data with misclassification and noise in real applications. To efficiently address this issue, Ziarko presented the VPRS by introducing a threshold parameter β for controlling the degree of misclassification [23].

Definition 2.2. *[23] Let $S = \{U, AT = A \bigcup D, V, f\}$ be an information system. The parameter with respect to the proportion of correct classification is denoted as β and $\beta \in (0.5, 1]$. $\forall X \subseteq U$ and $B \subseteq A$, the lower and upper approximations of VPRS model are respectively defined as:*

$$\underline{R_B}^{\beta}(X) = \{x | P(X | [x]_{R_B}) \geq \beta\} \tag{4}$$

$$\overline{R_B}^{\beta}(X) = \{x | P(X | [x]_{R_B}) > 1 - \beta\} \tag{5}$$

where $P(X | [x]_{R_B}) = \frac{|X \bigcap [x]_{R_B}|}{|[x]_{R_B}|}$, where $| \bullet |$ denotes the cardinality of a set.

Definition 2.3. *A PSvIS is a sextuple $(U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$, where $U = \{x_i | i \in \{1, 2, \cdots, n\}\}$ is a non-empty finite set of objects, called the universe. AT is a non-empty finite set of attributes, where A is a non-empty finite set of condition attributes and D is a decision attribute set with $A \bigcap D = \emptyset$. $V = V_A \bigcup V_D$ is the domain of attributes set AT, where V_A is the set of condition attribute values, V_D is the set of decision attribute values. $f : U \times A \to 2^{V_A}$ is a set-valued mapping, and $f : U \times D \to V_D$ is an information function such that $f(x, d) \in V_d$ for every $d \in D$, $x \in U$. σ is a sigma-algebra in V_A, which satisfies three properties in [49]. P is the probability distribution defined on σ, such that $P(f_i(x, a)) \geq 0$ and $\sum\limits_i P(f_i(x, a)) = 1$, where $f_i(x, a) \in f(x, a), \forall x \in U$ and $\forall a \in A$.*

Example 2.1. *Table 1 shows a PSvIS $S = (U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$* w.r.t. *the election information, where $U = \{x_i | i \in \{1, 2, \cdots, 14\}\}$ denotes fourteen different districts, $A = \{$Economic construction, Social construction, Cultural construction$\}$ indicates three different measure indexes of the candidate of governing capability, D is a decision attribute, $V_A = \{$Dissatisfaction, Neutrality, Satisfaction$\} = \{-1, 0, 1\}$ and $V_D = \{$Yes, No$\} = \{Y, N\}$. In Table 1, $f(x_1, a_1) = \frac{\{-1,0,1\}}{(0.23,0.45,0.32)}$ denotes the attribute value of the object x_1 under a_1 is set-valued {-1,0,1} with probability distribution {0.23,0.45,0.32}. Other notations are defined in a similar way.*

Table 1: A probabilistic set-valued information system

U	a_1	a_2	a_3	D	U	a_1	a_2	a_3	D
x_1	$\frac{\{-1,0,1\}}{(0.23,0.45,0.32)}$	$\frac{\{-1,0,1\}}{(0.10,0.40,0.50)}$	$\frac{\{0,1\}}{(0.40,0.60)}$	Y	x_8	$\frac{\{-1,0,1\}}{(0.81,0.14,0.05)}$	$\frac{\{-1,0,1\}}{(0.03,0.77,0.20)}$	$\frac{\{-1,0,1\}}{(0.82,0.11,0.07)}$	Y
x_2	$\frac{\{-1,0,1\}}{(0.20,0.43,0.37)}$	$\frac{\{-1,0,1\}}{(0.12,0.38,0.5)}$	$\frac{\{0,1\}}{(0.43,0.57)}$	Y	x_9	$\frac{\{0,1\}}{(0.32,0.68)}$	$\frac{\{0,1\}}{(0.33,0.67)}$	$\frac{\{-1,0,1\}}{(0.44,0.32,0.24)}$	N
x_3	$\frac{\{-1,0,1\}}{(0.25,0.42,0.33)}$	$\frac{\{-1,0,1\}}{(0.13,0.39,0.48)}$	$\frac{\{0,1\}}{(0.44,0.56)}$	N	x_{10}	$\frac{\{0,1\}}{(0.34,0.66)}$	$\frac{\{0,1\}}{(0.34,0.66)}$	$\frac{\{-1,0,1\}}{(0.43,0.33,0.24)}$	N
x_4	$\frac{\{-1,0,1\}}{(0.24,0.44,0.32)}$	$\frac{\{-1,0,1\}}{(0.12,0.41,0.47)}$	$\frac{\{-1,0,1\}}{(0.38,0.52,0.10)}$	Y	x_{11}	$\frac{\{-1,0,1\}}{(0.82,0.12,0.06)}$	$\frac{\{-1,0,1\}}{(0.02,0.78,0.20)}$	$\frac{\{-1,0\}}{(0.90,0.10)}$	N
x_5	$\frac{\{-1,0,1\}}{(0.22,0.41,0.37)}$	$\frac{\{-1,0,1\}}{(0.11,0.42,0.47)}$	$\frac{\{-1,0,1\}}{(0.41,0.53,0.06)}$	N	x_{12}	$\frac{\{0,1\}}{(0.34,0.66)}$	$\frac{\{0,1\}}{(0.35,0.65)}$	$\frac{\{1\}}{(1)}$	Y
x_6	$\frac{\{-1,0,1\}}{(0.24,0.42,0.34)}$	$\frac{\{-1,0,1\}}{(0.10,0.44,0.46)}$	$\frac{\{-1,0,1\}}{(0.40,0.52,0.08)}$	N	x_{13}	$\frac{\{-1,0\}}{(0.80,0.20)}$	$\frac{\{-1,0,1\}}{(0.24,0.29,0.47)}$	$\frac{\{-1,0,1\}}{(0.47,0.52,0.01)}$	Y
x_7	$\frac{\{-1,0,1\}}{(0.82,0.12,0.06)}$	$\frac{\{-1,0,1\}}{(0.02,0.76,0.22)}$	$\frac{\{-1,0,1\}}{(0.81,0.10,0.09)}$	Y	x_{14}	$\frac{\{1\}}{(1)}$	$\frac{\{-1,0,1\}}{(0.30,0.54,0.16)}$	$\frac{\{0,1\}}{(0.30,0.70)}$	N

Since the equivalence relation is limited to categorical data, it could not be employed to PSvIS. In addition, the traditional tolerance relation $T_a = \{(x, y) | f(x, a) \bigcap f(y, a) \neq \emptyset, a \in A\}$ in SvIS [15] aims to discriminate the objects according to whether there are the same attribute set-value, which does not take the distribution of set values into account. For example, the samples x_7 and x_{14} are indiscernible in terms of the traditional tolerance relation with respect to the condition attribute a_1 in Table 1. However, it is unreasonable that the objects x_7 and x_{14} belong to the same tolerance class according to the corresponding probability distribution. Hence, the traditional tolerance relation also could not be applied in PSvIS directly.

To more appropriately characterize the relation of objects in PSvIS, in what follows, we present the λ-tolerance relation based on Bhattacharyya distance which can quantify the distance of objects with probability distribution and extended VPRS model.

Definition 2.4. *Let $S = (U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$ be a PSvIS and the threshold $\lambda_a \geq 0$. The $\lambda-$tolerance relation BD_a^λ with regard to the attribute $a \in A$ can be defined as follows.*

$$BD_a^\lambda = \{(x, y) \in U \times U | BD_a(x, y) \leq \lambda_a\} \tag{6}$$

where $BD_a(x, y) = -\ln\left(\sum\limits_{k=1}^{K} \sqrt{P(f_k(x, a))P(f_k(y, a))}\right)$ is the Bhattacharyya distance which measures the similarity of two discrete probability distributions [50], and $P(f_k(x, a))$ denotes the probability distribution of x under the attribute a. Then $\forall B \subseteq A$, the $\lambda-$tolerance relation BD_B^λ is defined by

$$BD_B^\lambda = \{(x, y) \in U \times U | BD_b(x, y) \leq \lambda_b, \forall b \in B\} = \bigcap_{b \in B} BD_b^\lambda \tag{7}$$

The parameter λ_b with respect to the attribute b ranges from 0 to $+\infty$, and the lower λ_b indicates the more similarity between two objects. According to the prior knowledge or the specific requirements of each attribute distribution, we can set different parameter values of λ_b in different practical applications. In the present paper, we only discuss the fixed parameter for all condition attributes.

Proposition 2.1. *λ-tolerance relation is reflexive and symmetric, but not transitive.*

Proposition 2.2. *Let $B_1 \subseteq B_2 \subseteq A$, then $BD^{\lambda}_{B_2} \subseteq BD^{\lambda}_{B_1}$.*

Proposition 2.3. *For $\lambda_1 \leq \lambda_2$, $BD^{\lambda_1}_{B} \subseteq BD^{\lambda_2}_{B}$.*

Proof. For $\lambda_1 \leq \lambda_2$, it is obvious that $BD^{\lambda_1}_{b} \subseteq BD^{\lambda_2}_{b}$ by Equation 6. Then according to Equation 7, we have $BD^{\lambda_1}_{B} \subseteq BD^{\lambda_2}_{B}$.

Note that PSvIS degenerate to disjunctive SvIS when the probability distribution of set values are zero-one distribution, i.e., $P(f_i(x,a)) = 1$ and $P(f_j(x,a)) = 0 (i \neq j)(i, j \in \{1, 2, \ldots, K\})$. Furthermore, if set $\lambda = 0$, the λ-tolerance relation will generalize to the traditional tolerance relation in SvIS.

Definition 2.5. *Given a PSvIS $S = (U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$, $\forall X \subseteq U$ and $B \subseteq A$, then β lower and upper approximations with regard to the $\lambda-$tolerance relation BD^{λ}_{B} are defined as follows, respectively.*

$$\underline{R_B}^{(\beta,\lambda)}(X) = \{x | P(X | [x]_{BD^{\lambda}_{B}}) \geq \beta\} \tag{8}$$

$$\overline{R_B}^{(\beta,\lambda)}(X) = \{x | P(X | [x]_{BD^{\lambda}_{B}}) > 1 - \beta\} \tag{9}$$

where $[x]_{BD^{\lambda}_{B}} = \{y | (x, y) \in BD^{\lambda}_{B}\}$, $\beta \in (0.5, 1]$ and $\lambda \geq 0$.

Then the positive region $POS^{(\beta,\lambda)}_{B}(X) = \underline{R_B}^{(\beta,\lambda)}(X)$, negative region $NEG^{(\beta,\lambda)}_{B}(X) = U - \overline{R_B}^{(\beta,\lambda)}(X)$ and boundary region $BND^{(\beta,\lambda)}_{B}(X) = \overline{R_B}^{(\beta,\lambda)}(X) - \underline{R_B}^{(\beta,\lambda)}(X)$ can be obtained according to the lower and upper approximations, respectively.

Example 2.2. *(Continuation of Example 2.1) Let $\beta = 0.6$, $\lambda = 0.55$, $B = \{a_1, a_2\}$ and $X = \{x_1, x_2, x_4, x_7, x_8, x_{12}, x_{13}\}$. Then we have the lower and upper approximations in terms of the λ-tolerance relation BD^{λ}_{B}: $\underline{R_B}^{(\beta,\lambda)}(X) = \{x_7, x_8, x_{11}, x_{13}\}$, $\overline{R_B}^{(\beta,\lambda)}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{11}, x_{13}\}$.*

3. Matrix-based representation of approximations in PSvIS

In this section, we present λ-tolerance relation matrix and two vector functions for constructing the lower and upper approximations by two matrix operators in PSvIS. In addition, we discuss the related properties of matrix, which are benefit for dynamic updating approximations in PSvIS.

Definition 3.1. *Let $S = (U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$ be a PSvIS, where $U = \{x_1, x_2, \cdots, x_n\}$. Let $B \subseteq A$ and BD^{λ}_{B} be a λ-tolerance relation on U. Then the λ-tolerance relation matrix $M^{BD^{\lambda}_{B}} = (m_{ij})_{n \times n}$ with regard to BD^{λ}_{B} is defined as follows:*

$$m_{ij} = \begin{cases} 1, & (x_i, x_j) \in BD^{\lambda}_{B} \\ 0, & otherwise \end{cases} \tag{10}$$

Proposition 3.1. *The λ-tolerance relation matrix $M^{BD^{\lambda}_{B}}$ is symmetric, and the elements of primary diagonal $m_{ii} = 1 (i = 1, \ldots, n)$.*

Example 3.1. *(Continuation of Example 2.1) Let $B = \{a_1, a_2\}$ and $\lambda = 0.55$. We can compute the λ-tolerance relation matrix $M^{BD_B^\lambda}$ with respect to BD_B^λ according to Definition 3.1 as follows:*

$$M^{BD_B^\lambda} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Obviously, the λ-tolerance relation matrix $M^{BD_B^\lambda}$ is symmetric and $m_{ii} = 1(i = 1, \ldots, n)$ from Example 3.1.

Definition 3.2. *$\forall X \subseteq U$, the characteristic function $G(X)$ with respect to X in PSvIS is defined as:*

$$G(X) = \begin{pmatrix} g_1, g_2, \ldots, g_n \end{pmatrix}^T, \textit{where } g_i = \begin{cases} 1, & x_i \in X \\ 0, & x_i \notin X \end{cases} \tag{11}$$

where "T" denotes the transpose operation.

Example 3.2. *(Continuation of Example 2.1) Let $X = \{x_1, x_2, x_4, x_7, x_8, x_{12}, x_{13}\}$. Then the characteristic function $G(X)$ can be obtained by Definition 3.2: $G(X) = (1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0)^T$.*

Definition 3.3. *Let $Y = (y_1, y_2, \cdots, y_n)^T$ be a column vector. The vector piece-wise function $l_\beta(Y)$ with respect to β is defined as follows:*

$$l_\beta(Y) = \begin{pmatrix} l_\beta(y_1) \\ l_\beta(y_2) \\ \cdots \\ l_\beta(y_n) \end{pmatrix}, \textit{where } l_\beta(y_i) = \begin{cases} 1, & y_i \geq \beta \\ 0, & 1-\beta < y_i < \beta \\ -1, & y_i \leq 1-\beta \end{cases} \tag{12}$$

where $\beta \in (0.5, 1]$.

Proposition 3.2. *Let $Q_1 \triangleq M^{BD_B^\lambda} \times G(X)$ and $Q_2 \triangleq M^{BD_B^\lambda} \times I$, where "$\times$" represents matrix multiplication and $I = (1, 1, \cdots, 1)^T$. Then we have $Q_1(i) = |[x_i]_{BD_B^\lambda} \bigcap X|$ and $Q_2(i) = |[x_i]_{BD_B^\lambda}|$, where $Q_1(i)$ and $Q_2(i)$ denote the* i*th element of Q_1 and Q_2, respectively.*

Example 3.3. *(Continuation of Examples 3.1 and 3.2) According to the results of Examples 3.1 and 3.2, we can calculate the intermediate vectors Q_1 and Q_2 by Proposition 3.2: $Q_1 = M^{BD_B^\lambda} \times G(X) = (3, 3, 3, 3, 3, 3, 2, 2, 1, 1, 2, 1, 1, 0)^T$, $Q_2 = M^{BD_B^\lambda} \times I = (6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 1, 1)^T$.*

Theorem 3.1. *Given a PSvIS $S = (U, AT = A \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$, $U = \{x_1, x_2, \cdots, x_n\}$. $\forall X \subseteq U$, let $Q_3 \triangleq Q_1 /. Q_2$, where "$/.$" denotes matrix dot divide. Then the positive, negative and boundary regions with respect to $B \subseteq A$ can be obtained from $l_\beta(Q_3)$ as follows:*

$$\begin{aligned} POS_B^{(\beta,\lambda)}(X) &= \{x_i | l_\beta(Q_3(i)) = 1\} \\ NEG_B^{(\beta,\lambda)}(X) &= \{x_i | l_\beta(Q_3(i)) = -1\} \\ BND_B^{(\beta,\lambda)}(X) &= \{x_i | l_\beta(Q_3(i)) = 0\} \end{aligned} \tag{13}$$

Then the lower approximation $\underline{R_B}^{(\beta,\lambda)}(X) = POS_B^{(\beta,\lambda)}(X)$ *and the upper approximation* $\overline{R_B}^{(\beta,\lambda)}(X) = U - NEG_B^{(\beta,\lambda)}(X)$.

Proof. According to Definition 3.3 and Proposition 3.2, if $l_\beta(Q_3(i)) = 1$, namely, $\frac{|[x_i]_{BD_B^\lambda} \bigcap X|}{|[x_i]_{BD_B^\lambda}|} \geq \beta = P(X|[x]_{BD_B^\lambda}) \geq \beta$, then $x_i \in POS_B^{(\beta,\lambda)}(X)$. The negative and boundary regions can be obtained in a similar way.

Example 3.4. *(Continuation of Example 3.3) Let* $\beta = 0.6$. *Then we have* $Q_3 = Q_1 /_. Q_2 = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{1}{3}, 1, 0)^T$ *based on the dot divide operator and* $l_\beta(Q_3) = (0, 0, 0, 0, 0, 0, 1, 1, -1, -1, 1, -1, 1, -1)^T$ *in terms of Definition 3.3. According to Theorem 3.1, the positive, negative and boundary regions can be obtained as follows:* $POS_B^{(\beta,\lambda)}(X) = \{x_7, x_8, x_{11}, x_{13}\}$, $NEG_B^{(\beta,\lambda)}(X) = \{x_9, x_{10}, x_{12}, x_{14}\}$ *as well as* $BND_B^{(\beta,\lambda)}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. *Furthermore we have the lower approximation* $\underline{R_B}^{(\beta,\lambda)}(X) = \{x_7, x_8, x_{11}, x_{13}\}$ *and upper approximation* $\overline{R_B}^{(\beta,\lambda)}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{11}, x_{13}\}$.

4. Matrix-based approach for incremental updating approximations under the variation of attributes in PSvIS

The variation of attributes, including the addition and deletion of attributes, are incurred by the demand of real applications, which will lead to the variation of knowledge structure in dynamic PSvIS. To address this issue, we present incremental mechanisms based on matrix for computing the lower and upper approximations. More specifically, the incremental mechanisms mainly include three aspects: i) region matrices are constructed, i.e., the positive, negative and boundary relation matrices are extracted from the whole λ-tolerance relation matrix; ii) these three region relation matrices are updated by the previous matrix information; and iii) the lower and upper approximations are computed according to the accumulated approximations results and the updated region matrices.

4.1. Dynamic updating approximations with the addition of attributes

In this section, we introduce the approach for dynamic update of approximations in a PSvIS with the addition of attributes. Let $S^t = (U, AT = A^t \bigcup D, V = V_{A^t} \bigcup V_D, f, \sigma, P)$ be a PSvIS at time t, where $U = \{x_1, x_2, \cdots, x_n\}$. At time $t+1$, the new attribute set ΔA is added to the conditional attribute set A^t, i.e., the PSvIS S^t is altered as $S^{t+1} = (U, AT^{t+1} = A^{t+1} \bigcup D, V^{t+1} = V_A^{t+1} \bigcup V_D, f^{t+1}, \sigma^{t+1}, P^{t+1})$, where $A^{t+1} = A^t \bigcup \Delta A$. To utilize the accumulated information, we denote the positive, negative and boundary regions of Pawlak rough set model as $POS_{A^t}(X)$, $NEG_{A^t}(X)$ and $BND_{A^t}(X)$ at time t, respectively.

Definition 4.1. *Let* $M^{BD_{A^t}^\lambda}$ *denote the* λ*-tolerance relation matrix with respect to* A^t *in the PSvIS* S *at time* t. *The positive region relation matrix* $M_{POS} = (m_{ij}^{POS})$, *the negative region relation matrix* $M_{NEG} = (m_{ij}^{NEG})$ *and the boundary region relation matrix* $M_{BND} = (m_{ij}^{BND})$ *are defined as follows, respectively.*

$$m_{ij}^{POS} = \begin{cases} 1, & (x_i, x_j) \in BD_{A^t}^\lambda, x_i \in POS_{A^t}(X), x_j \in U \\ 0, & otherwise \end{cases} \tag{14}$$

$$m_{ij}^{NEG} = \begin{cases} 1, & (x_i, x_j) \in BD_{A^t}^\lambda, x_i \in NEG_{A^t}(X), x_j \in U \\ 0, & otherwise \end{cases} \tag{15}$$

$$m_{ij}^{BND} = \begin{cases} 1, & (x_i, x_j) \in BD_{A^t}^\lambda, x_i \in BND_{A^t}(X), x_j \in U \\ 0, & otherwise \end{cases} \tag{16}$$

Example 4.1. *(Continuation of Examples 2.2 and 3.1) According to Definition 2.1, we have the positive, negative and boundary regions of* X *with respect to the classical rough set model as follows:* $POS_B(X) = \{x_{13}\}$, $NEG_B(X) = \{x_{14}\}$, $BND_B(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$. *Then the positive, negative and boundary region relation matrices* M_{POS}, M_{NEG} *and* M_{BND} *can be obtained in terms of the result of Example 3.1 as follows.*

$$M_{POS} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad M_{NEG} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_{BND}=\begin{pmatrix}1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&1&0&0&1&0&0&0\\0&0&0&0&0&0&1&1&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&1&0&1&0&0\\0&0&0&0&0&0&0&0&1&1&0&1&0&0\\0&0&0&0&0&0&1&1&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&1&0&1&0&0\end{pmatrix}$$

According to Definition 4.1, we only need to update the region relation matrix, rather than the whole relation matrix under the variation of attributes. More precisely, the boundary relation matrix is updated when adding attributes, and the positive and negative relation matrices are updated when deleting attributes. It will improve the efficiency of our incremental method since the size of region relation matrix is smaller than that of whole relation matrix.

Theorem 4.1. *Given a PSvIS $S=(U, AT=A^t \bigcup D, V=V_{A^t}\bigcup V_D, f, \sigma, P)$, $U=\{x_1, x_2, \cdots, x_n\}$. Let $M_{BND'}=(m_{ij}^{BND'})$ denote the boundary relation matrix when the attribute set ΔA is added to A^t at time $t+1$. Then it can be updated by the following mechanisms.*

(1) *If $m_{ij}^{BND}=0$, then $m_{ij}^{BND'}=m_{ij}^{BND}$;*

(2) *If $m_{ij}^{BND}=1$ and $x_i \in [x_j]_{BD_{\Delta A}^{\lambda}}$ then $m_{ij}^{BND'}=m_{ij}^{BND}$;*

(3) *If $m_{ij}^{BND}=1$ and $x_i \notin [x_j]_{BD_{\Delta A}^{\lambda}}$ then $m_{ij}^{BND'}=0$.*

Proof. If $m_{ij}^{BND}=0$, namely, $x_i \notin [x_j]_{BD_{A^t}^{\lambda}}$, it is clear that $x_i \notin [x_j]_{BD_{A^t \bigcup \Delta A}^{\lambda}}$ when adding the attribute set ΔA to A^t. Then we have $m_{ij}^{BND'}=m_{ij}^{BND}=0$. If $m_{ij}^{BND}=1$, that is, $x_i \in [x_j]_{BD_{A^t}^{\lambda}}$. In addition, if $x_i \in [x_j]_{BD_{\Delta A^t}^{\lambda}}$ when ΔA is added to A^t, it is obvious that $x_i \in [x_j]_{BD_{A^t \bigcup \Delta A}^{\lambda}}$, i.e., $m_{ij}^{BND'}=m_{ij}^{BND}=1$; otherwise, $x_i \notin [x_j]_{BD_{A^t \bigcup \Delta A}^{\lambda}}$, i.e., $m_{ij}^{BND'}=0$.

It can be seen from Theorem 4.1 that we only update the elements of boundary relation matrix according to the case 3, which can enhance the performance of updating procedures.

Theorem 4.2. *Assume intermediate vectors $Q_1 \triangleq M_{BND}\times G(X)$ and $Q_2 \triangleq M_{BND}\times I$ at time t. Let $Q_1^{'}=M_{BND'}\times G(X)$ and $Q_2^{'}=M_{BND'}\times I$ when the boundary relation matrix $M_{BND'}$ has been updated at time $t+1$. Then we have $Q_1^{'}(i)=Q_1(i)-\sum\limits_{j=1}^{n}\left(1-m_{ij}^{BND'}\right)m_{ij}^{BND}G(j)$ and $Q_2^{'}(i)=Q_2(i)-\sum\limits_{j=1}^{n}\left(1-m_{ij}^{BND'}\right)m_{ij}^{BND}$.*

Proof. According to the definition of matrix multiplication, we have $Q_1(i)=\sum\limits_{j=1}^{n}m_{ij}^{BND}G(j)$. Obviously, the element m_{ij}^{BND} is updated according to Theorem 4.1, thus $Q_1^{'}(i)=Q_1(i)$ when m_{ij}^{BND} keeps unchanged, otherwise, $Q_1^{'}(i)=Q_1(i)-\sum\limits_{j=1}^{n}\left(1-m_{ij}^{BND'}\right)m_{ij}^{BND}G(j)$ when m_{ij}^{BND} updates from 1 to 0. The proof of $Q_2^{'}(i)$ is similar.

Theorem 4.3. *Let $Q_1^{'}=M_{BND'}\times G(X)$, $Q_2^{'}=M_{BND'}\times I$, $Q_3^{'}=Q_1^{'}/_{.}Q_2^{'}$ and $l_{\beta}^{'}=l_{\beta}(Q_3^{'})$. Then $\forall X \subseteq U$, the positive region $POS_{A^{t+1}}^{(\beta,\lambda)}(X)$ and the negative region $NEG_{A^{t+1}}^{(\beta,\lambda)}(X)$ at time $t+1$ are updated as follows.*

(1) *$POS_{A^{t+1}}^{(\beta,\lambda)}(X)=POS_{A^t}(X)\bigcup\{x_i|l_{\beta}^{'}(Q_3^{'}(i))=1\}$;*

(2) *$NEG_{A^{t+1}}^{(\beta,\lambda)}(X)=NEG_{A^t}(X)\bigcup\{x_i|l_{\beta}^{'}(Q_3^{'}(i))=-1\}$.*

Then we can compute the lower approximation $\underline{R_{A^{t+1}}}^{(\beta,\lambda)}(X)=POS_{A^{t+1}}^{(\beta,\lambda)}(X)$ and the upper approximations $\overline{R_{A^{t+1}}}^{(\beta,\lambda)}(X)=U-NEG_{A^{t+1}}^{(\beta,\lambda)}(X)$.

Proof. For the first case, according to Definition 2.5, the positive region $POS^{(\beta,\lambda)}_{A^{t+1}}(X) = POS_{A^{t+1}}(X) \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^{t+1}}(X)\}$. When the attribute set ΔA is appended to A, the positive region $POS_{A^{t+1}}(X)$ of Pawlak rough sets is divided into two parts, namely, one is $POS_{A^t}(X)$, the other is $\{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \subseteq X, x_i \in BND_{A^t}(X)\}$. Furthermore, we have $BND_{A^{t+1}}(X) \subseteq BND_{A^t}(X)$ when $A^t \subseteq A^{t+1}$. Evidently, we have $POS^{(\beta,\lambda)}_{A^{t+1}}(X) = POS_{A^t}(X) \bigcup \{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \subseteq X, x_i \in BND_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^{t+1}}(X)\} = POS_{A^t}(X) \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^t}(X)\}$. Finally, according to Definition 4.1 and Theorem 3.1, we have $POS^{(\beta,\lambda)}_{A^{t+1}}(X) = POS_{A^t}(X) \bigcup \{x_i | l'_{\beta}(Q'_3(i)) = 1\}$. The second case can be obtained in a similar way.

Example 4.2. *(Continuation of Example 2.1) Let $A^t = B = \{a_1, a_2\}$ at time t and $\Delta A = \{a_3\}$ is appended to A^t at time $t+1$. Based on the result of Example 4.1, we can update the boundary relation matrix $M_{BND'}$ according to Theorem 4.1.*

$$M_{BND'} = \begin{pmatrix}
1 & 1 & 1 & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \underline{\boldsymbol{0}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \underline{\boldsymbol{0}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \underline{\boldsymbol{0}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \underline{\boldsymbol{0}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} & 0 & 1 & 0 & 0
\end{pmatrix},$$

where only the underlined elements are updated in M_{BND}.

Then, according to Theorem 4.2, we have $Q'_1 = (2,2,2,1,1,1,2,2,0,0,0,1)^T$, $Q'_2 = (3,3,3,3,3,3,2,2,2,2,1,1)^T$ and $Q'_3 = (\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1, 1, 0, 0, 0, 1)^T$. Finally, the positive and negative regions can be updated by Theorem 4.3 as follows: $POS^{(\beta,\lambda)}_{A^{t+1}}(X) = \{x_{13}\} \bigcup \{x_1, x_2, x_3, x_7, x_8, x_{12}\} = \{x_1, x_2, x_3, x_7, x_8, x_{12}, x_{13}\}$ and $NEG^{(\beta,\lambda)}_{A^{t+1}}(X) = \{x_{14}\} \bigcup \{x_4, x_5, x_6, x_9, x_{10}, x_{11}\} = \{x_4, x_5, x_6, x_9, x_{10}, x_{11}, x_{14}\}$. Hence, the lower approximation $\underline{R_{A^{t+1}}}^{(\beta,\lambda)}(X) = POS^{(\beta,\lambda)}_{A^{t+1}}(X)$ and the upper approximation $\overline{R_{A^{t+1}}}^{(\beta,\lambda)}(X) = U - NEG^{(\beta,\lambda)}_{A^{t+1}}(X) = \{x_1, x_2, x_3, x_7, x_8, x_{12}, x_{13}\}$.

Obviously, the computing overhead of incremental updating approximations can be efficiently reduced by locally updating boundary relation matrix and utilizing the accumulated positive and negative regions information, not rather updating the whole relation matrix and computing the approximations from scratch.

4.2. Dynamic updating approximations with the deletion of attributes

In this section, we introduce the approach for dynamic update of approximations in a PSvIS with the deletion of attributes. Analogously, let $S^t = (U, AT = A^t \bigcup D, V = V_A \bigcup V_D, f, \sigma, P)$ be a PSvIS at time t, where $A^t = A \bigcup \Delta A$. At time $t+1$, the attribute set ΔA is deleted from the condition attribute set A^t, i.e., the PSvIS S^t is updated as $S^{t+1} = (U, AT^{t+1} = A^{t+1} \bigcup D, V^{t+1} = V^{t+1}_A \bigcup V_D, f^{t+1}, \sigma^{t+1}, P^{t+1})$, where $A^{t+1} = A$.

Theorem 4.4. *Let $M_R = (m^R_{ij})$, $M_{R'} = (m^{R'}_{ij})$ denote the region relation matrix at time t and $t+1$, respectively. When the attribute set ΔA is deleted from A^t, the region relation matrix $M_{R'} = (m^{R'}_{ij})$ is updated as follows, where "R" indicates "POS" or "NEG", respectively.*

(1) *If $m^R_{ij} = 1$, then $m^{R'}_{ij} = m^R_{ij}$;*

(2) *If $m^R_{ij} = 0$ and $x_i \notin [x_j]_{BD^{\lambda}_A}$ then $m^{R'}_{ij} = m^R_{ij}$;*

(3) *If $m^R_{ij} = 0$ and $x_i \in [x_j]_{BD^{\lambda}_A}$ then $m^{R'}_{ij} = 1$.*

Proof. Observe that $[x_j]_{BD^{\lambda}_{A^t}} \subseteq [x_j]_{BD^{\lambda}_{A^{t+1}}}$ when $A^{t+1} \subseteq A^t$. Then if $x_i \in [x_j]_{BD^{\lambda}_{A^t}}$, namely, $m^R_{ij} = 1$, we have $x_i \in [x_j]_{BD^{\lambda}_{A^{t+1}}}$, that is, $m^{R'}_{ij} = m^R_{ij} = 1$. If $x_i \notin [x_j]_{BD^{\lambda}_{A^t}}$, viz., $m^R_{ij} = 0$, it is clear that there are two scenarios after deleting ΔA from A^t, if $x_i \notin [x_j]_{BD^{\lambda}_A}$, we have $m^{R'}_{ij} = m^R_{ij} = 0$; otherwise, $m^{R'}_{ij} = 1$.

It can be seen from Theorem 4.4 that we only update the elements of positive and negative relation matrices according to the case 3, which can enhance the performance of updating procedures.

Considering that the positive and negative regions of Pawlak rough sets will decrease, but the boundary region will increase when removing the attribute set ΔA from A^t, we set $POS_{A^t}(BND) \triangleq \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^t}(X)\}$ and $NEG_{A^t}(BND) \triangleq \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \leq 1 - \beta, x_i \in BND_{A^t}(X)\}$ for saving part of boundary region information. $\forall X \subseteq U$, let $Q1_{POS} = M_{POS} \times G(X)$, $Q2_{POS} = M_{POS} \times I$, $Q3_{POS} = Q1_{POS} / .Q2_{POS}$, $Q1_{NEG} = M_{NEG} \times G(X)$, $Q2_{NEG} = M_{NEG} \times I$ and $Q3_{NEG} = Q1_{NEG} / .Q2_{NEG}$. Then, the intermediate vectors are updated as follows.

Theorem 4.5. *Suppose* $Q'_{1_{POS}} \triangleq M_{POS'} \times G(X)$, $Q'_{2_{POS}} \triangleq M_{POS'} \times I$, $Q'_{1_{NEG}} \triangleq M_{NEG'} \times G(X)$ *and* $Q'_{2_{NEG}} \triangleq M_{NEG'} \times I$ *when the attribute set* ΔA *is removed from* A^t *at time* $t + 1$. *Then we have*

(1) $Q'_{1_{POS}}(i) = Q1_{POS}(i) + \sum_{j=1}^{n} \left(1 - m_{ij}^{POS}\right) m_{ij}^{POS'} G(j)$ *and* $Q'_{2_{POS}}(i) = Q2_{POS}(i) + \sum_{j=1}^{n} \left(1 - m_{ij}^{POS}\right) m_{ij}^{POS'}$;

(2) $Q'_{1_{NEG}}(i) = Q1_{NEG}(i) + \sum_{j=1}^{n} \left(1 - m_{ij}^{NEG}\right) m_{ij}^{NEG'} G(j)$ *and* $Q'_{2_{NEG}}(i) = Q2_{NEG}(i) + \sum_{j=1}^{n} \left(1 - m_{ij}^{NEG}\right) m_{ij}^{NEG'}$.

Proof. This proof is similar to Theorem 4.2.

Theorem 4.6. *Let* $Q'_{3_{POS}} = Q'_{1_{POS}} / .Q'_{2_{POS}}$, $l_{\beta}^{POS} = l_{\beta}(Q'_{3_{POS}})$, $Q'_{3_{NEG}} = Q'_{1_{NEG}} / .Q'_{2_{NEG}}$ *and* $l_{\beta}^{NEG} = l_{\beta}(Q'_{3_{NEG}})$. *Then, the positive region* $POS_{A^{t+1}}^{(\beta,\lambda)}(X)$ *and the negative region* $NEG_{A^{t+1}}^{(\beta,\lambda)}(X)$ *at time* $t + 1$ *are updated as follows.*

(1) $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = POS_{A^t}(BND) \bigcup \{x_i | l_{\beta}^{POS}(i) = 1\} \bigcup \{x_i | l_{\beta}^{NEG}(i) = 1\}$;

(2) $NEG_{A^{t+1}}^{(\beta,\lambda)}(X) = NEG_{A^t}(BND) \bigcup \{x_i | l_{\beta}^{NEG}(i) = -1\} \bigcup \{x_i | l_{\beta}^{POS}(i) = -1\}$.

Proof. Notice that $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = POS_{A^{t+1}}(X) \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^{t+1}}(X)\}$. After removing ΔA from A^t, the positive region $POS_{A^{t+1}}(X)$ will decrease and the boundary region $BND_{A^{t+1}}(X)$ will increase. Thus $POS_{A^{t+1}}(X) = \{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \subseteq X, x_i \in POS_{A^t}(X)\}$ and $BND_{A^{t+1}}(X) = BND_{A^t}(X) \bigcup \{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \bigcap X \neq \emptyset \text{ and } [x_i]_{BD^{\lambda}_{A^{t+1}}} \nsubseteq X, x_i \in POS_{A^t}(X)\} \bigcup \{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \bigcap X \neq \emptyset \text{ and } [x_i]_{BD^{\lambda}_{A^{t+1}}} \nsubseteq X, x_i \in NEG_{A^t}(X)\}$. Then we have $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = \{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \subseteq X, x_i \in POS_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in POS_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in NEG_{A^t}(X)\}$. Since $\{x_i | [x_i]_{BD^{\lambda}_{A^{t+1}}} \subseteq X, x_i \in POS_{A^t}(X)\} \subseteq \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in POS_{A^t}(X)\}$, thus $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in BND_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in POS_{A^t}(X)\} \bigcup \{x_i | P(X|[x_i]_{BD^{\lambda}_{A^{t+1}}}) \geq \beta, x_i \in NEG_{A^t}(X)\}$. Finally, according to Definition 4.1 and Theorem 3.1, we have $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = POS_{A^t}(BND) \bigcup \{x_i | l_{\beta}^{POS}(i) = 1\} \bigcup \{x_i | l_{\beta}^{NEG}(i) = 1\}$. The proof of second case is similar.

Example 4.3. *(Continuation of Example 2.1) Let* $\Delta A = \{a_3\}$ *is deleted from* $A^t = \{a_1, a_2, a_3\}$, *i.e.,* $A^{t+1} = \{a_1, a_2\}$ *at time* $t + 1$. *Set* $\beta = 0.6$ *and* $\lambda = 0.55$. *Then we have* $POS_{A^t}(X) = \{x_7, x_8, x_{12}, x_{13}\}$, $NEG_{A^t}(X) = \{x_9, x_{10}, x_{11}, x_{14}\}$ *and* $BND_{A^t}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. *Firstly, we compute* $POS_{A^t}(BND) = \emptyset$ *and* $NEG_{A^t}(BND) = \emptyset$. *Then* $M_{POS'}$ *and* $M_{NEG'}$ *can be obtained as follows according to Theorem 4.4.*

$$M_{POS} = \begin{pmatrix} 0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix} \quad M_{POS'} = \begin{pmatrix} 0&0&0&0&0&0&1&1&0&0&\underline{\boldsymbol{1}}&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&\underline{\boldsymbol{1}}&0&0&0 \\ 0&0&0&0&0&0&0&0&\underline{\boldsymbol{1}}&\underline{\boldsymbol{1}}&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix}$$

$$M_{NEG} = \begin{pmatrix} 0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \quad M_{NEG'} = \begin{pmatrix} 0&0&0&0&0&0&0&0&1&1&0&\underline{\boldsymbol{1}}&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&\underline{\boldsymbol{1}}&0&0 \\ 0&0&0&0&0&0&\underline{\boldsymbol{1}}&\underline{\boldsymbol{1}}&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix}$$

where only the underlined elements are updated in M_{POS} *and* M_{NEG}.

Based on Definition 3.3, we can compute $l_{\beta}^{POS} = (1, 1, -1, 1)^T$ *and* $l_{\beta}^{NEG} = (-1, -1, 1, -1)^T$. *Finally, we have* $POS_{A^{t+1}}^{(\beta,\lambda)}(X) = \{x_7, x_8, x_{11}, x_{13}\}$ *and* $NEG_{A^{t+1}}^{(\beta,\lambda)}(X) = \{x_9, x_{10}, x_{12}, x_{14}\}$ *according to Theorem 4.6. Furthermore, the lower and upper approximations can be obtained by Equation (3).*

It is evident that the incremental strategies for computing approximations can reduce the computational cost by partly updating the region relation matrices rather than updating the whole relation matrice when attributes are removed.

5. Dynamic algorithms for updating approximations in PSvIS under the variation of attributes

In this section, according to dynamic updating mechanisms for maintenance of approximations in PSvIS, we develop dynamic algorithms based on matrix for computing approximations.

5.1. The naive algorithm for updating approximations in PSvIS

Algorithm 1: Matrix-based static algorithm for updating approximations in PSvIS (MSAUA)

```
Input:
    1. A PSvIS $S = (U, AT = C \bigcup D, V = V_C \bigcup V_D, f, \sigma, P)$;
    2. The parameters $\lambda$ and $\beta$;
    3. An attribute set $A^{t+1} \subseteq C$.
Output: The lower and upper approximations of each decision class in PSvIS.
1  begin
2      for $1 \leq i \leq n$ do    // Compute the $\lambda$-tolerance relation matrix $M^{BD^{\lambda}_{A^{t+1}}}$ by Definition 3.1;
3          for $1 \leq j \leq n$ do
4              if $x_i \in [x]_{BD^{\lambda}_{A^{t+1}}}$ then
5                  $m_{ij} = 1$;
6              else
7                  $m_{ij} = 0$;
8              end
9          end
10     end
11     Construct the characteristic vector $G(d_k)$ for each decision class $d_k$.
12     for $1 \leq i \leq n$ do    // Compute the intermediate vectors $Q_1$, $Q_2$, $Q_3$ and $l_\beta(Q_3)$;
13         $Q_1(i) = \sum_{j=1}^{n} m_{ij} G(j)$;
14         $Q_2(i) = \sum_{j=1}^{n} m_{ij}$;
15         $Q_3(i) = \frac{Q_1(i)}{Q_2(i)}$;
16         $l_\beta(i) = l_\beta(Q_3(i))$;
17     end
18     Set $POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) = \emptyset$ and $NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) = \emptyset$;
19     for $1 \leq i \leq n$ do    // Compute the positive, negative and boundary regions according to Theorem 3.1;
20         if $l_\beta(i) == 1$ then
21             $POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) = POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) \bigcup \{x_i\}$;
22         end
23         if $l_\beta(i) == -1$ then
24             $NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) = NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) \bigcup \{x_i\}$;
25         end
26     end
27     return $\underline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = POS^{(\beta,\lambda)}_{A^{t+1}}(d_k)$, $\overline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = U - NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k)$.
28 end
```

Algorithm 1, which is abbreviated as MSAUA for convenience in this paper, is a naive (non-incremental) algorithm based on matrix for computing approximations. To conveniently analyze the time complexity of MSAUA, let $|U| = n$ and $|A^{t+1}| = m$. Steps 2-14 are to construct the λ-tolerance relation matrix $M^{BD^{\lambda}_{A^{t+1}}}$ according to Definition 3.1 and Proposition 3.1, whose time complexity is $O(n^2 m)$. Step 15 is to compute the character vector according to Definition 3.2, whose time complexity is $O(n)$. Steps 16-21 are to compute the intermediate vectors Q_1, Q_2, Q_3 and $l_\beta(Q_3)$ according to Proposition 3.2 and Definition 3.3, whose time complexity is $O(n^2)$. Steps 23-30 are to compute the positive and negative regions according to Theorem 3.1, whose time complexity is $O(n)$. Hence, the total time complexity is $O(n^2 m + n + n^2 + n) = O(n^2 m + n)$.

Algorithm 2: Matrix-based dynamic algorithm for updating approximations in PSvIS when adding an attribute set (MDAUAA)

```
   Input:
     1. A PSvIS $S = (U, AT = A^t \bigcup D, V = V_{A^t} \bigcup V_D, f, \sigma, P)$;
     2. The parameters $\lambda$ and $\beta$;
     3. The new added attribute set $\Delta A$ and $\Delta A \bigcap A^t = \emptyset$;
     4. The characteristic vector $G(d_k)$ of each decision class $d_k$;
     5. The positive, negative and boundary regions of Pawlak's rough set: $POS_{A^t}(d_k)$, $NEG_{A^t}(d_k)$ and $BND_{A^t}(d_k)$;
     6. Initial results: the boundary region relation matrix $M_{BND} = (m_{ij}^{BND})$ with respect to $A^t$ at time $t$, $Q_1 = M_{BND} \times G(d_k)$, $Q_2 = M_{BND} \times I$,
        $Q_3 = Q_1 ./ Q_2$ and $l_\beta = l_\beta(Q_3)$.
   Output: The lower and upper approximations of each decision class in PSvIS.
1  begin
2      for $1 \le i \le |BND_{A^t}(d_k)|$ do                // Updating the boundary relation matrix $M_{BND} = (m_{ij}^{BND})$;
3          for $1 \le j \le n$ do
4              if $m_{ij}^{BND} == 0$ then         // Computing the boundary relation matrix $M_{BND} = (m_{ij}^{BND})$ by Theorem 4.1;
5                  $m_{ij}^{BND} = 0$ are constant;
6              else
7                  if $x_i \in [x]_{BD^{\lambda}_{\Delta A}}$ then
8                      $m_{ij}^{BND} = 1$ are constant;
9                  else
10                     $m_{ij}^{BND} = 0$;
11                     $Q_1(i) = Q_1(i) - G(j)$;      // Dynamic updating $Q_1$, $Q_2$, $Q_3$ and $l_\beta(Q_3)$ according to Theorem 4.2;
12                     $Q_2(i) = Q_2(i) - 1$;
13                     $Q_3(i) = Q_1(i)/Q_2(i)$;
14                     $l_\beta(i) = l_\beta(Q_3(i))$;
15                 end
16             end
17         end
18     end
19     Set $POS_{A^{t+1}}^{(\beta,\lambda)}(d_k) = POS_{A^t}(d_k)$ and $NEG_{A^{t+1}}^{(\beta,\lambda)}(d_k) = NEG_{A^t}(d_k)$;
20     for $1 \le i \le |BND_{A^t}(d_k)|$ do          // Updating the positive and negative regions according to Theorem 4.3;
21         if $l_\beta(i) == 1$ then
22             $POS_{A^{t+1}}^{(\beta,\lambda)}(d_k) = POS_{A^{t+1}}^{(\beta,\lambda)}(d_k) \bigcup \{x_i\}$;
23         end
24         if $l_\beta(i) == -1$ then
25             $NEG_{A^{t+1}}^{(\beta,\lambda)}(d_k) = NEG_{A^{t+1}}^{(\beta,\lambda)}(d_k) \bigcup \{x_i\}$;
26         end
27     end
28     return $\underline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = POS_{A^{t+1}}^{(\beta,\lambda)}(d_k)$, $\overline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = U - NEG_{A^{t+1}}^{(\beta,\lambda)}(d_k)$.
29 end
```

5.2. The dynamic algorithm for updating approximations in PSvIS when adding an attribute set

Algorithm 2, which is abbreviated as MDAUAA, is a dynamic algorithm based on matrix for computing approximations under the addition of attributes. Steps 2-18 are to update the boundary relation matrix according to Theorem 4.1, whose time complexity is $O(I_{BND1}|\Delta A|)$, where I_{BND1} is the numbers of "1" in the boundary relation matrix M_{BND} and $I_{BND1} \le |BND_{A^t}(d_k)| \cdot n \le n^2$, where $|BND_{A^t}(d_k)|$ denotes the numbers of boundary region $BND_{A^t}(d_k)$ at time t. Steps 20-27 are to update the positive and negative regions according to Theorem 4.3, whose time complexity is $O(|BND_{A^t}(d_k)|)$. Thus the total time complexity is $O(I_{BND1}|\Delta A| + |BND_{A^t}(d_k)|)$. Obviously, the time complexity of algorithm MDAUAA is better than that of the static algorithm MSAUA.

5.3. The dynamic algorithm for updating approximations in PSvIS when deleting an attribute set

Algorithm 3, which is abbreviated as MDAUAD, is a dynamic algorithm based on matrix for computing approximations under the deletion of attributes. Steps 3-10 are to compute $POS_{A^t}(BND)$ and $NEG_{A^t}(BND)$ according to Definition 2.5, whose time complexity is $O(|BND_{A^t}(d_k)|n(m-|\Delta A|))$. Steps 12-28 are to update the positive relation matrix $M_{POS} = (m_{ij}^{POS})$ according to Theorem 4.4, whose time complexity is $O(I_{POS0}(m - |\Delta A|))$, where I_{POS0} is the numbers of "0" in the positive relation matrix M_{POS} and $I_{POS0} \le |POS_{A^t}(d_k)| \cdot n$, where $|POS_{A^t}(d_k)|$ denotes the numbers of positive region $POS_{A^t}(d_k)$ at time t. Steps 29-45 are to update the negative relation matrix $M_{NEG} = (m_{ij}^{NEG})$ according to Theorem 4.4, whose time complexity is $O(I_{NEG0}(m-|\Delta A|))$, where I_{NEG0} is the numbers of "0" in the negative relation matrix M_{NEG} and $I_{NEG0} \le |NEG_{A^t}(d_k)| \cdot n$, where $|NEG_{A^t}(d_k)|$ denotes the numbers of negative region $NEG_{A^t}(d_k)$

Algorithm 3: Matrix-based dynamic algorithm for updating approximations in PSvIS when deleting an attribute set (MDAUAD)

Input:

1. A PSvIS $S = (U, AT = A^t \bigcup D, V = V_{A^t} \bigcup V_D, f, \sigma, P)$;
2. The parameters λ and β;
3. The attribute set ΔA is removed from A^t and let $A^{t+1} = A^t - \Delta A$;
4. The characteristic vector $G(d_k)$ of each decision class d_k;
5. The positive, negative and boundary regions of Pawlak's rough set: $POS_{A^t}(d_k)$, $NEG_{A^t}(d_k)$ and $BND_{A^t}(d_k)$;
6. Initial results: the positive region relation matrix $M_{POS} = (m_{ij}^{POS})$ and the negative region relation matrix $M_{NEG} = (m_{ij}^{NEG})$ with respect to A^t at time t, $Q_{1POS} = M_{POS} \times G(d_k)$, $Q_{2POS} = M_{POS} \times I$, $Q_{3POS} = Q_{1POS} /_{.} Q_{2POS}$, $l_\beta^{POS} = l_\beta(Q_{3POS})$, $Q_{1NEG} = M_{NEG} \times G(d_k)$, $Q_{2NEG} = M_{NEG} \times I$, $Q_{3NEG} = Q_{1NEG} /_{.} Q_{2NEG}$ and $l_\beta^{NEG} = l_\beta(Q_{3NEG})$.

Output: The lower and upper approximations of each decision class in PSvIS.

```
begin
    Let $POS_{A^t}(BND) = \emptyset$ and $NEG_{A^t}(BND) = \emptyset$;
    for $1 \le i \le |BND_{A^t}(d_k)|$ do        // Computing $POS_{A^t}(BND)$ and $NEG_{A^t}(BND)$ according to Definition 2.5;
        if $P(d_k|[x_i]_{BD^\lambda_{A^{t+1}}}) \ge \beta$ then
            $POS_{A^t}(BND) = POS_{A^t}(BND) \bigcup \{x_i\}$;
        end
        if $P(d_k|[x_i]_{BD^\lambda_{A^{t+1}}}) \le 1 - \beta$ then
            $NEG_{A^t}(BND) = NEG_{A^t}(BND) \bigcup \{x_i\}$;
        end
    end
    Set $POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) = POS_{A^t}(BND)$ and $NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) = NEG_{A^t}(BND)$;
    for $1 \le i \le |POS_{A^t}(d_k)|$ do        // Updating the positive relation matrix $M_{POS} = (m_{ij}^{POS})$ according to Theorem 4.4;
        for $1 \le j \le n$ do
            if $m_{ij}^{POS} == 1$ then
                $m_{ij}^{POS} = 1$ are constant;
            else
                if $x_i \notin [x]_{BD^\lambda_{A^{t+1}}}$ then
                    $m_{ij}^{POS} = 0$ are constant;
                else
                    $m_{ij}^{POS} = 1$;        // Updating $Q_{1POS}$, $Q_{2POS}$, $Q_{3POS}$ and $l_\beta^{POS}$ according to Theorem 4.5;
                    $Q_{1POS}(i) = Q_{1POS}(i) + G(j)$;
                    $Q_{2POS}(i) = Q_{2POS}(i) + 1$;
                    $Q_{3POS}(i) = Q_{1POS}(i)/Q_{2POS}(i)$;
                    $l_\beta^{POS}(i) = l_\beta(Q_{3POS}(i))$;
                end
            end
        end
    end
    for $1 \le i \le |NEG_{A^t}(d_k)|$ do    // Updating the negative relation matrix $M_{NEG} = (m_{ij}^{NEG})$ according to Theorem 4.4 ;
        for $1 \le j \le n$ do
            if $m_{ij}^{NEG} == 1$ then
                $m_{ij}^{NEG} = 1$ are constant;
            else
                if $x_i \notin [x]_{BD^\lambda_{A^{t+1}}}$ then
                    $m_{ij}^{NEG} = 0$ are constant;
                else
                    $m_{ij}^{NEG} = 1$;        // Updating $Q_{1NEG}$, $Q_{2NEG}$, $Q_{3NEG}$ and $l_\beta^{NEG}$ according to Theorem 4.5;
                    $Q_{1NEG}(i) = Q_{1NEG}(i) + G(j)$;
                    $Q_{2NEG}(i) = Q_{2NEG}(i) + 1$;
                    $Q_{3NEG}(i) = Q_{1NEG}(i)/Q_{2NEG}(i)$;
                    $l_\beta^{NEG}(i) = l_\beta(Q_{3NEG}(i))$;
                end
            end
        end
    end
    for $1 \le i \le max(|POS_{A^t}(d_k)|, |NEG_{A^t}(d_k)|$ do    // Updating positive and negative regions according to Theorem 4.6;
        if $l_\beta^{POS}(i) == 1$ or $l_\beta^{NEG}(i) == 1$ then
            $POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) = POS^{(\beta,\lambda)}_{A^{t+1}}(d_k) \bigcup \{x_i\}$;
        end
        if $l_\beta^{POS}(i) == -1$ or $l_\beta^{NEG}(i) == -1$ then
            $NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) = NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k) \bigcup \{x_i\}$;
        end
    end
    return $\underline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = POS^{(\beta,\lambda)}_{A^{t+1}}(d_k), \overline{R_{A^{t+1}}}^{(\beta,\lambda)}(d_k) = U - NEG^{(\beta,\lambda)}_{A^{t+1}}(d_k)$.
end
```

at time t. Steps 46-53 are to update positive and negative regions according to Theorem 4.6, whose time complexity is $O(max(|POS_{A^t}(d_k)|, |NEG_{A^t}(d_k)|))$, where $max(,)$ denotes the maximum operation. Thus the total time complexity is $O((|BND_{A^t}(d_k)|n + I_{POS0} + I_{NEG0})(m - |\Delta A|) + max(|POS_{A^t}(d_k)|, |NEG_{A^t}(d_k)|))$. Since $I_{POS0} \leq |POS_{A^t}(d_k)| \cdot n$ and $I_{NEG0} \leq |NEG_{A^t}(d_k)| \cdot n$, it follows $(|BND_{A^t}(d_k)|n + I_{POS0} + I_{NEG0})(m - |\Delta A|) \leq n^2(m - |\Delta A|) \leq n^2 m$. Hence, the time complexity of algorithm MDAUAD is better than that of the static algorithm MSAUA.

6. Experimental evaluations

Since the proposed method aims at the case that the attribute values are set-valued with probability distribution, there are not available in any public data repositories. In addition, the purpose of our study is to demonstrate the performance of dynamical algorithms for addressing the probability set-valued data, not rather for analyzing the data for one particular application. Hence, four data sets with missing values from the UCI Repository of Machine Learning are generated as the probability set-valued. More specially, the missing values are filled with the set of all possible values of each attribute, and the corresponding probability distributions are constructed by the frequencies of each single attribute value under each attribute. In addition, the probability distribution of each single value is one-point distribution, i.e., the accordant probability is one. Moreover, two artificial data sets are generated for validating the performance of dynamic algorithms. A detail description of six data sets can be founded in Table 2.

Table 2: A detail description of data sets

	Data sets	Abbreviation	Samples	Attributes	Classes	Source
1	Audiology (Standardized)	Audiology	226	69	24	UCI
2	Dermatology	Dermatology	336	34	6	UCI
3	Congressional voting records	CVR	435	16	2	UCI
4	Mushroom	Mushroom	8124	22	2	UCI
5	Artificial data 1	AD1	1000	100	4	Data generator
6	Artificial data 2	AD2	10000	2000	10	Data generator

All concerned experiments are performed with Matlab 2012a on a personal computer with Intel Core i5-4200U CPU 1.60GHZ, 4.0 GB of memory.

To show the time efficiency of dynamical algorithms, each of data sets is divided into ten parts of equal size. Moreover, the first part is regarded as the 1st test set, the combination of the first and second parts is considered as the 2nd test set, the combination of the 2nd test set and the third part is viewed as the 3rd test set, ..., the whole data set is regarded as the 10th data set. In what follows, these test sets are employed to compare the running time between the dynamic algorithm and the static algorithm, and the Zhang's dynamic method [45] under the variation of attributes.

6.1. The performance comparison between static and dynamic algorithms under the variation of attributes

In this subsection, the computational times of static and dynamic algorithms are compared on six data sets shown in Table 2 under the variation of attributes. In the following, two experimental results are shown by the addition of attributes and the deletion of attributes, respectively. In addition, we set the parameters $\lambda = 1.5$ and $\beta = 0.6$ in the experiments.

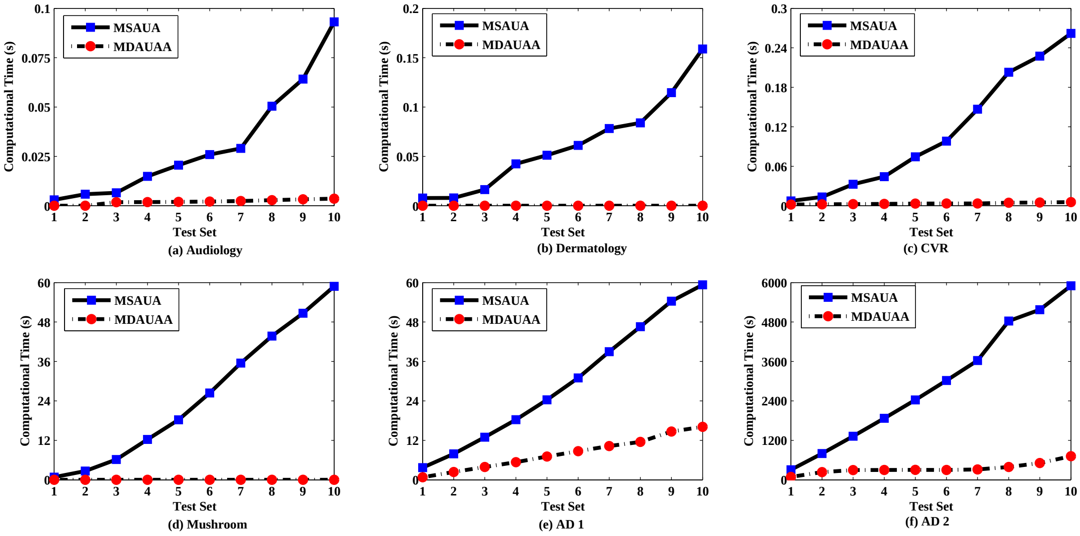
6.1.1. A comparison of MSAUA and MDAUAA under the addition of an attribute set

In this subsection, we compare Algorithm MSAUA with Algorithm MDAUAA when adding an attribute set. We take out the original and appended attribute sets from each data set, which are shown in Table 3.

The experimental results are depicted in Fig. 1, where the x-coordinate pertains to the test sets, while y-coordinate concerns the computing time of updating approximations. As shown in Fig. 1, the running times of MSAUA and MDAUAA grow up with the increase of the size of data. In addition, Algorithm MDAUAA is much faster than Algorithm MSAUA, and the difference between them are becoming larger while the size of data increases.

Table 3: A description of adding attribute set

	Data sets	Attributes	Original attribute set	Adding attribute set
1	Audiology	69	$\{a_1, a_2, \cdots, a_{48}\}$	$\{a_{49}, a_{50}, \cdots, a_{69}\}$
2	Dermatology	34	$\{a_1, a_2, \cdots, a_{24}\}$	$\{a_{25}, a_{26}, \cdots, a_{34}\}$
3	CVR	16	$\{a_1, a_2, \cdots, a_{11}\}$	$\{a_{12}, a_{13}, \cdots, a_{16}\}$
4	Mushroom	22	$\{a_1, a_2, \cdots, a_{14}\}$	$\{a_{15}, a_{16}, \cdots, a_{22}\}$
5	AD1	100	$\{a_1, a_2, \cdots, a_{70}\}$	$\{a_{71}, a_{72}, \cdots, a_{100}\}$
6	AD2	2000	$\{a_1, a_2, \cdots, a_{1200}\}$	$\{a_{1201}, a_{1202}, \cdots, a_{2000}\}$



(a) Audiology

(b) Dermatology

(c) CVR

(d) Mushroom

(e) AD 1

(f) AD 2

Figure 1: A comparison of computational times between MSAUA and MDAUAA versus the different test sets when adding an attribute set.

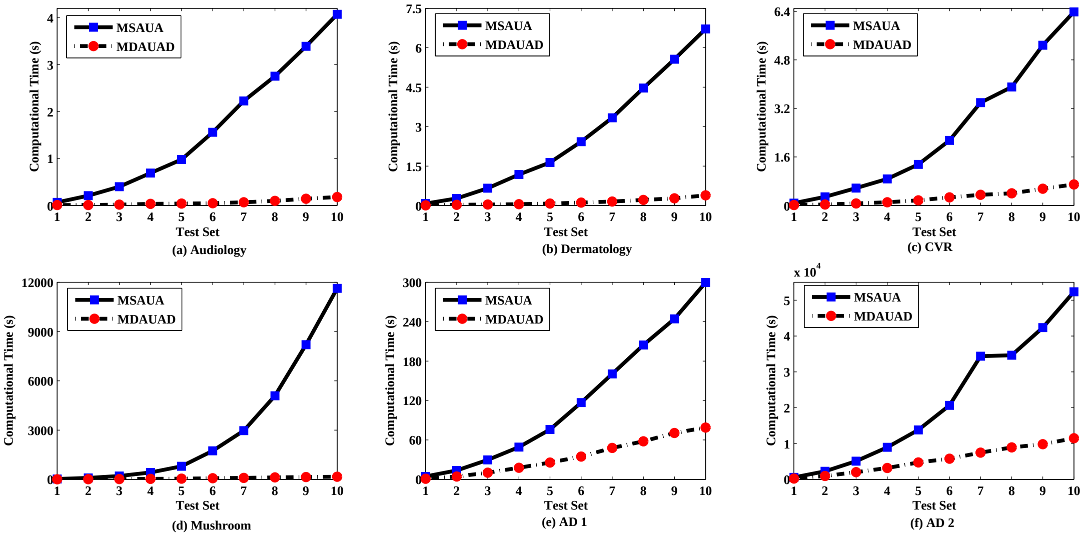
6.1.2. *A comparison of MSAUA and MDAUAD with the deletion of an attribute set*

In this subsection, we compare Algorithm MSAUA with Algorithm MDAUAD when deleting an attribute set. We take out the original and removed attribute sets, which are shown in Table 4.

Table 4: A description of deleting attribute set

	Data sets	Attributes	Original attribute set	Deleting attribute set
1	Audiology	69	$\{a_1, a_2, \cdots, a_{69}\}$	$\{a_{49}, a_{50}, \cdots, a_{69}\}$
2	Dermatology	34	$\{a_1, a_2, \cdots, a_{34}\}$	$\{a_{25}, a_{26}, \cdots, a_{34}\}$
3	CVR	16	$\{a_1, a_2, \cdots, a_{16}\}$	$\{a_{12}, a_{13}, \cdots, a_{16}\}$
4	Mushroom	22	$\{a_1, a_2, \cdots, a_{22}\}$	$\{a_{15}, a_{16}, \cdots, a_{22}\}$
5	AD1	100	$\{a_1, a_2, \cdots, a_{100}\}$	$\{a_{71}, a_{72}, \cdots, a_{100}\}$
6	AD2	2000	$\{a_1, a_2, \cdots, a_{2000}\}$	$\{a_{1201}, a_{1202}, \cdots, a_{2000}\}$

The experimental results are depicted in Fig .2, where the x-coordinate pertains to the test sets, while y-coordinate concerns the computing time of updating approximations. As shown in Fig. 2, the running times of MSAUA and MDAUAD grow up with the increasing size of data. Furthermore, Algorithm MDAUAD is much faster than Algorithm MSAUA, and the difference between them are getting larger while the size of data increases.



(a) Audiology
(b) Dermatology
(c) CVR
(d) Mushroom
(e) AD 1
(f) AD 2

Figure 2: A comparison of computational times between MSAUA and MDAUAD versus the different test sets when deleting an attribute set.

6.2. *Comparative experiments on the variation of attributes*

In this experiment, to demonstrate the performance of our proposed incremental method, a series of comparative experiments are carried out on six data sets under the variation of attributes. However, none of the existing approach for attribute generalization is developed to process the PSvIS. Zhang et al. presented an incremental method for dealing with the SvIS with the variation of attributes [45]. In order to compare with Zhang's method, the tolerance relation is modified as the λ-tolerance relation in Zhang's method for applications with probabilistic set-valued data. The dynamical algorithms in Zhang's method are abbreviated as ZDAUAA and ZDAUAD under the addition of attributes and the deletion of attributes, respectively.

6.2.1. *Comparison between ZDAUAA and MDAUAA when adding an attribute set*

In this subsection, we compare Algorithm ZDAUAA and Algorithm MDAUAA while adding an attribute set. The information of the adding attribute sets and test sets is the same to Subsection 6.1.1 and we set the parameters $\lambda = 1.5$ and $\beta = 0.6$ in this experiment. The experimental results are shown in Table 5 and Fig. 3. Table 5 shows the speed-up ratios of experiments on six data sets with different test sets. From Table 5, it is obviously that Algorithm MDAUAA is more effective than Algorithm ZDAUAA while adding an attribute set. In addition, the average speed-up ratio dramatically fluctuates from 1.435 to 221532.820. In what follows, we will demonstrate that the computational time of Algorithm MDAUAA under the variation of parameter λ. Fig. 3 shows the running time of ZDAUAA and MDAUAA with different test sets. Clearly, the proposed incremental algorithm MDAUAA is better than ZDAUAA. Furthermore, the time complexity of Algorithm ZDAUAA is $O(I_{M1}|\Delta A| + n)$, where I_{M1} is the numbers of "1" in the whole relation matrix $M^{BD^{\lambda}_{A^t}}$. Evidently, the time complexity of Algorithm MDAUAA, namely, $O(I_{BND1}|\Delta A| + |BND_{A^t}(d_k)|)$, is better than that of Algorithm ZDAUAA.

Table 5: The incremental speed-up ratio between ZDAUAA and MDAUAA versus each test set

	Data Set					
Test Set	Audiology	Dermatology	CVR	Mushroom	AD1	AD2
1	53.795	1.435	88.828	12884.864	1.988	2.043
2	90.906	2.917	144.446	39648.841	1.678	1.389
3	1.353	6.258	314.105	81462.002	1.845	1.306
4	2.230	12.073	529.562	133148.153	1.713	1.323
5	3.353	22.393	788.651	184619.555	1.646	1.384
6	4.707	26.862	1266.313	246917.482	1.645	1.427
7	8.326	32.913	1711.497	251633.841	1.601	1.429
8	11.085	32.748	1808.002	341957.712	1.701	1.339
9	12.149	50.092	2638.146	421947.496	1.530	1.388
10	15.764	50.846	2814.184	501108.253	1.585	1.324
average	20.367	23.854	1210.303	221532.820	1.693	1.435

6.2.2. *Comparison between ZDAUAD and MDAUAD when deleting an attribute set*

In this subsection, we compare Algorithm ZDAUAD and Algorithm MDAUAD while removing an attribute set. The information of the deleting attribute sets and test sets is the same to Subsection 6.1.2 and set the parameters $\lambda = 1.5$ and $\beta = 0.6$. Table 6 indicates the speed-up ratios of experiments on six data sets with different test sets. Fig. 4 shows the computing time of ZDAUAD and MDAUAD with different test sets. From the Table 6 and Fig. 4, it demonstrates Algorithm MDAUAD is a little better than Algorithm ZDAUAD. Furthermore, the time complexity of ZDAUAD is $O(nI_{M0}(m - |\Delta A|) + n)$, where I_{M0} is the numbers of "0" in the whole relation matrix $M^{BD^{\lambda}_{A^t}}$. According to $I_{M0} = I_{POS0} + I_{NEG0} + I_{BND0}$ and $I_{BND0} \geq |BND_{A^t}(d_k)|$, then we have $nI_{M0}(m - |\Delta A|) + n \geq (|BND_{A^t}(d_k)|n + I_{POS0} + I_{NEG0})(m - |\Delta A|)$, i.e., the time complexity of Algorithm MDAUAD is better than that of Algorithm ZDAUAD.

6.3. *The performance comparisons on different ratios of attributes*

In this section, in order to evaluate the performance of the proposed dynamical algorithms when updating (adding or deleting) different ratios of attributes, we conduct a series of compared experiments between the static, dynamic and Zhang's algorithms for computing approximations. The parameters of these Algorithms are set $\lambda = 1.5$ and $\beta = 0.6$.

When adding different ratios of attributes, we take out 50% attributes of different data sets as the basic attribute set, and gradually add 10% attributes from the rest of attributes. The comparative results are shown in Fig. 5. In Fig. 5, the x-coordinate pertains to the ratios of added attributes and a point in the y-coordinate is the logarithm value of the running times of Algorithm MSAUA, MDAUAA and ZDAUAA. It is shown that the computational time of Algorithm MDAUAA is lower than those of Algorithms MSAUA and ZDAUAA while inserting different updating ratios of attributes.

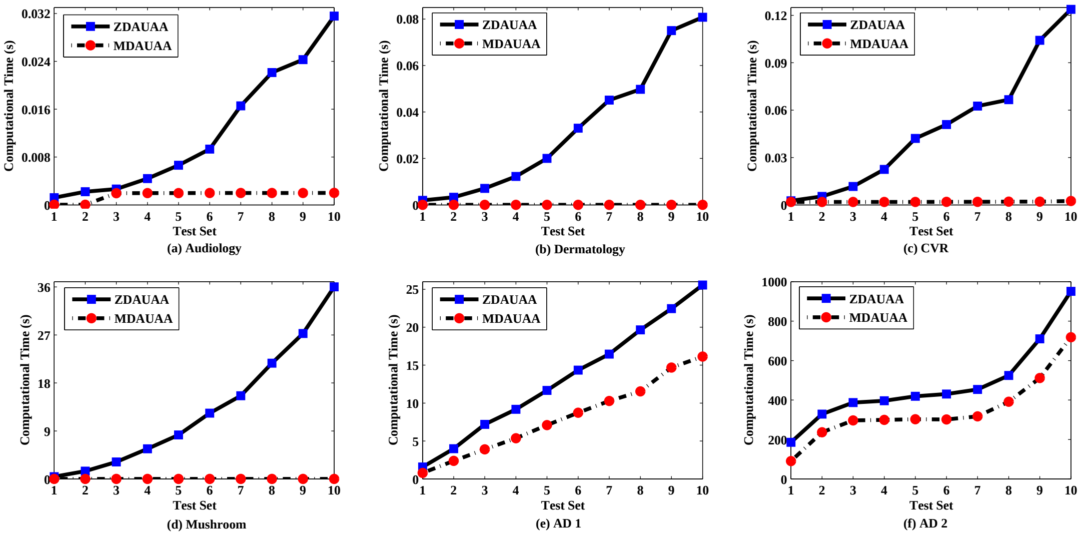
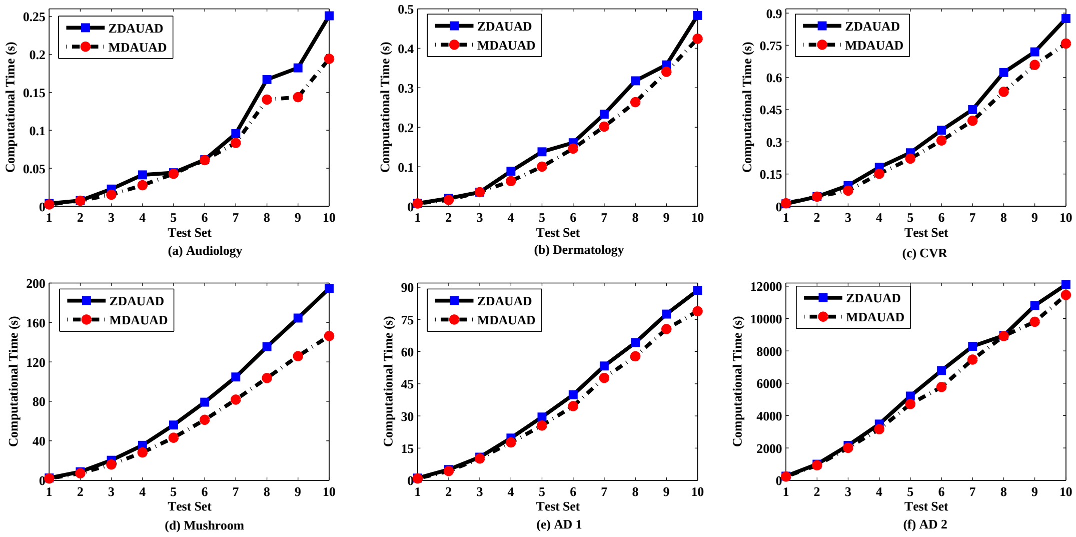


Figure 3: Running times of Algorithm ZDAUAA and Algorithm MDAUAA versus the different test sets when adding an attribute set.

Table 6: The incremental speed-up ratio between ZDAUAD and MDAUAD versus each test set

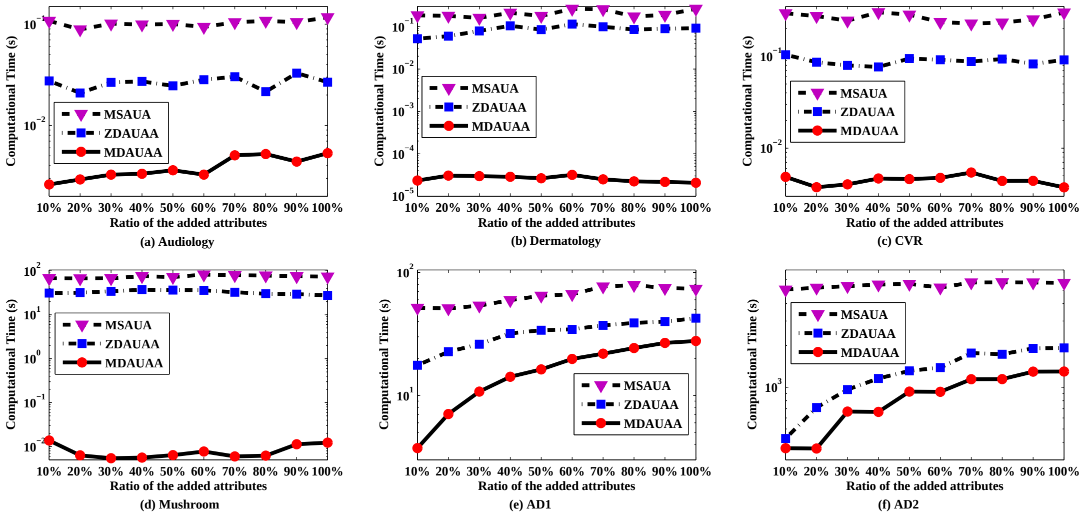
	Data Set					
Test Set	Audiology	Dermatology	CVR	Mushroom	AD1	AD2
1	1.548	1.182	1.024	1.266	1.198	1.122
2	1.010	1.001	1.262	1.222	1.161	1.065
3	1.492	1.330	1.001	1.278	1.064	1.076
4	1.493	1.201	1.394	1.254	1.117	1.100
5	1.031	1.124	1.375	1.301	1.156	1.105
6	1.011	1.157	1.100	1.290	1.152	1.175
7	1.148	1.130	1.156	1.280	1.117	1.109
8	1.189	1.169	1.205	1.305	1.110	1.004
9	1.269	1.093	1.052	1.307	1.099	1.102
10	1.291	1.154	1.139	1.329	1.123	1.056
average	1.248	1.154	1.171	1.283	1.130	1.091



(a) Audiology (b) Dermatology (c) CVR

(d) Mushroom (e) AD 1 (f) AD 2

Figure 4: Running times of Algorithm ZDAUAD and Algorithm MDAUAD versus the different test sets when deleting an attribute set.



(a) Audiology (b) Dermatology (c) CVR

(d) Mushroom (e) AD1 (f) AD2

Figure 5: Running times of Algorithms MSAUA, ZDAUAA and MDAUAA when adding different ratios of attributes.

When deleting different ratios of attributes, we gradually remove 10% attributes from the original attribute set. Fig. 6 shows the comparative results. In Fig. 6, the x-coordinate pertains to the ratios of removed attributes and a point in the y-coordinate is the logarithm value of the running times of Algorithm MSAUA, MDAUAD and ZDAUAD. It is shown that the computational time of Algorithm MDAUAD is better than Algorithms MSAUA and ZDAUAD while deleting different updating ratios of attributes.

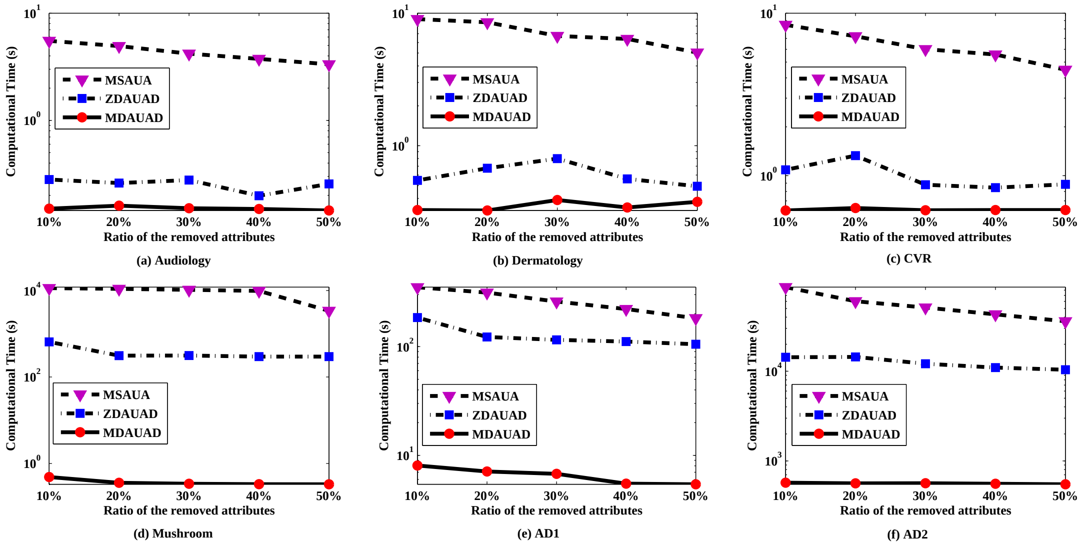


Figure 6: Running times of Algorithms MSAUA, ZDAUAD and MDAUAD when deleting different ratios of attributes.

6.4. *Experiments on the parameters λ and β*

The lower and upper approximations of extended VPRS model are affected by the parameters λ and β. In this subsection, to verify whether the performance of proposed dynamical algorithms are influenced by these two parameters, we carry out experiments to evaluate the efficiency of Algorithm MDAUAA and Algorithm MDAUAD with different λ and β. While adding attributes, 50% of original attributes are token out as the basic attribute set, the rest attributes are appended. While deleting attributes, 30% of attributes are removed from original attribute set.

The parameter λ is increased by 5 times from 0.03 to 58593.75. Fig. 7 shows the experimental results on six data sets with respect to Algorithm MDAUAA when adding attributes. In Fig. 7, the x-coordinate pertains to the parameter λ and the y-coordinate pertains to the logarithm value of the running time of Algorithm MDAUAA. Obviously, the running times of Algorithm MDAUAA in data sets Audiology, Dermatology, CVR and Mushroom fluctuate a little with different λ. However, the running times in data sets AD1 and AD2 raise gradually from $\lambda = 0.03$ to $\lambda = 18.75$, and then achieve a steady fluctuation. According to the complexity of Algorithm MDAUAA in Subsection 5.2, the computational time of Algorithm MDAUAA is mainly affected by the numbers of I_{BND1} and $|BND_{A'}(d_k)|$, which are controlled by the parameter λ. Clearly, the numbers of I_{BND1} and $|BND_{A'}(d_k)|$ increase with the growing λ. In light of Table 7, the numbers of boundary region in data sets Audiology, Dermatology, CVR and Mushroom do not change with different λ. However, in data sets AD1 and AD2, the numbers of boundary region increase from $\lambda = 0.03$ to $\lambda = 18.75$, and then remain unchanged after that. Hence, the change trend is consistent with the variation of the numbers of boundary region in Fig. 7.

Fig. 8 shows the experimental results on six data sets with respect to Algorithm MDAUAD when deleting attributes. In Fig. 8, the x-coordinate pertains to the parameter λ and the y-coordinate pertains to the logarithm value of the running time of Algorithm MDAUAD. According to the complexity of Algorithm MDAUAD in Subsection 5.3, the computational time of Algorithm MDAUAD is mainly affected by the numbers of I_{POS0}, I_{NEG0} and $|BND_{A'}(d_k)|$, which are influenced by the parameter λ. In addition, the numbers of $|BND_{A'}(d_k)|$ increases with the growing λ. In

Fig. 8, the running time of Algorithm MDAUAD in data sets Audiology, Dermatology, CVR and Mushroom fluctuate a little with different λ. It can be interpreted by Table 8, where the number of boundary region keeps invariant in these data sets. In data sets AD1 and AD2, due to the numbers of boundary region increase gradually from $\lambda = 0.03$ to $\lambda = 3.75$ in Table 8, the running times fluctuate dramatically. Moreover, only a little fluctuation is observed when the numbers of boundary region keep constant from $\lambda = 18.75$ to $\lambda = 58593.75$.

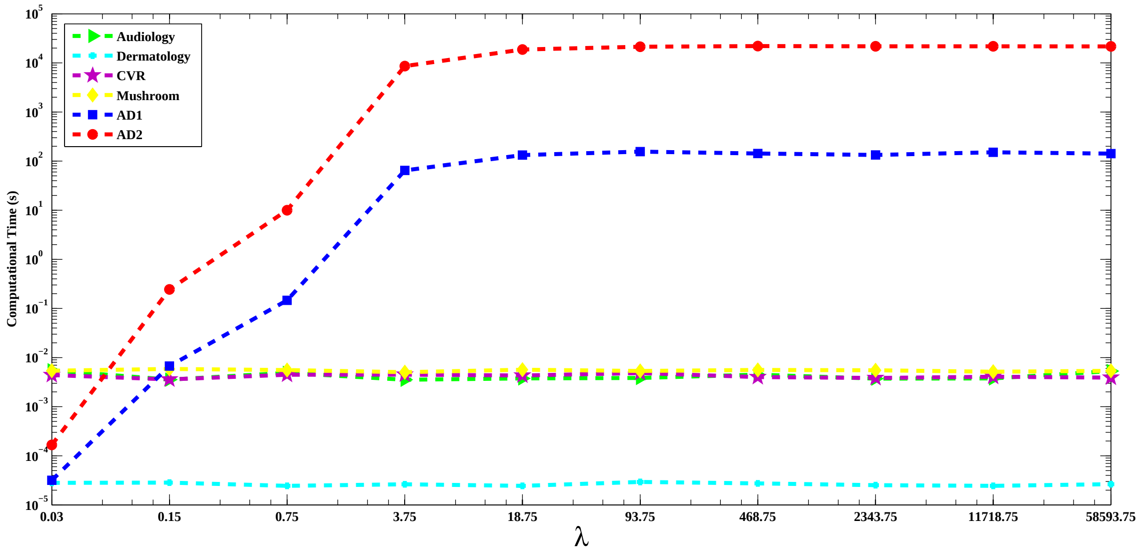


Figure 7: A comparison of Algorithm MDAUAA with different λ.

Table 7: The number of boundary region with different λ in Algorithm MDAUAA

	Data Set					
λ	Audiology	Dermatology	CVR	Mushroom	AD1	AD2
0.03	9	0	106	8	0	0
0.15	9	0	106	8	5	14
0.75	9	0	106	8	39	191
3.75	9	0	106	8	1000	9986
18.75	9	0	106	8	1000	10000
93.75	9	0	106	8	1000	10000
468.75	9	0	106	8	1000	10000
2343.75	9	0	106	8	1000	10000
11718.75	9	0	106	8	1000	10000
58593.75	9	0	106	8	1000	10000

Let the parameter β change from 0.6 to 1. Fig. 9 shows the variation tendency of the computational times of Algorithm MDAUAA and MDAUAD. In Fig. 9, the x-coordinate pertains to the parameter β and the y-coordinate pertains to the logarithm value of the running time of Algorithm MDAUAA in Fig. 9 (a) and Algorithm MDAUAD in Fig. 9 (b), respectively. It is clear that the running times fluctuate a little with different β when adding or deleting attributes, respectively. According to the complexity of Algorithms MDAUAA and MDAUAD in Subsections 5.2 and 5.3, they are almost the same with different β except for some random factors.

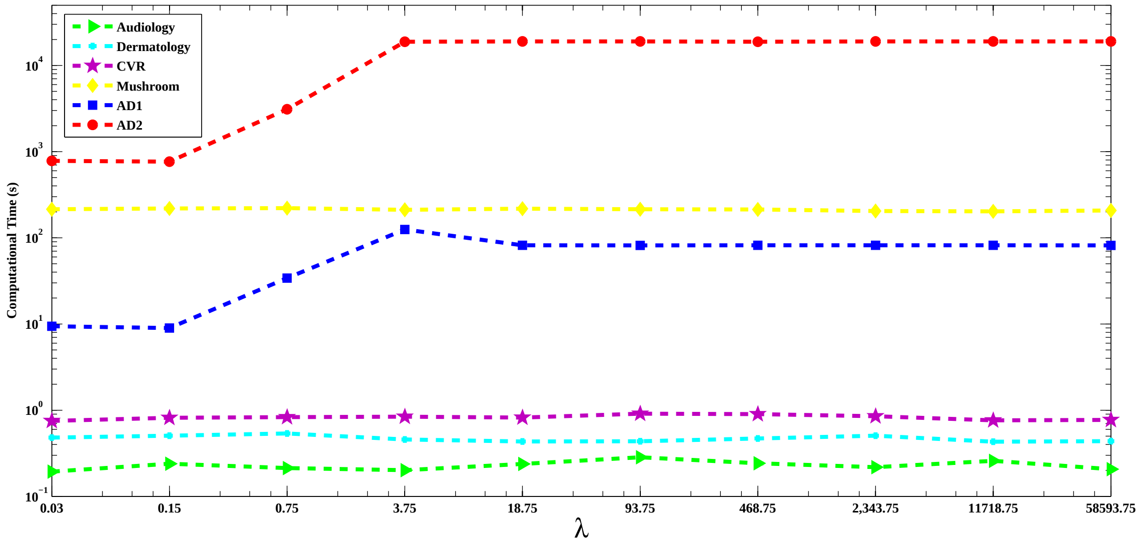


Figure 8: A comparison of Algorithm MDAUAD with different λ.

Table 8: The number of boundary region with different λ in Algorithm MDAUAD

	Data Set					
λ	Audiology	Dermatology	CVR	Mushroom	AD1	AD2
0.03	26	6	5	4	0	0
0.15	26	6	5	4	3	0
0.75	26	6	5	4	20	96
3.75	26	6	5	4	999	10000
18.75	26	6	5	4	1000	10000
93.75	26	6	5	4	1000	10000
468.75	26	6	5	4	1000	10000
2343.75	26	6	5	4	1000	10000
11718.75	26	6	5	4	1000	10000
58593.75	26	6	5	4	1000	10000

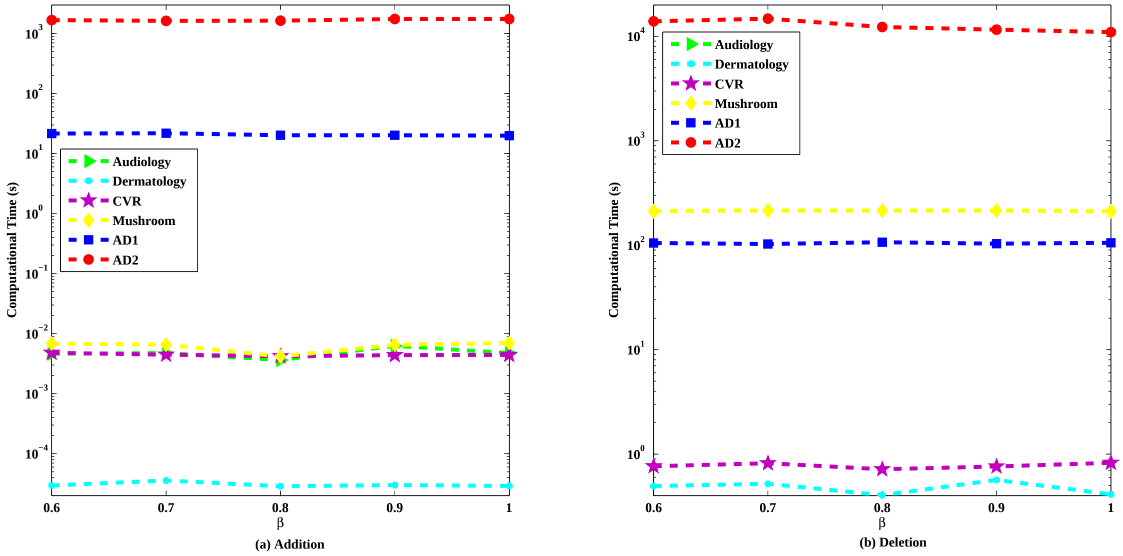


Figure 9: A comparison of Algorithms MDAUAA and MDAUAD with different β.

7. Conclusions

Due to the set values with probability distributions in some real applications, in this study, we introduced the concept of PSvIS and presented λ-tolerance relation based on Bhattacharyya distance for more reasonably characterizing the relation of objects in PSvIS. In addition, an extended VPRS in terms of λ-tolerance relation is presented for knowledge discovery in PSvIS.

To address the computation of rough approximations in dynamic PSvIS with the variation of attributes, we firstly presented a matrix-based method for computing approximations by utilizing the λ-tolerance relation matrix and matrix operators. Then we presented the incremental mechanisms by using the previous rough approximations results and partly updating the region relation matrices, which can effectively improve the computational efficiency compared with the static and Zhang's approaches. Furthermore, we have designed two incremental algorithms MDAUAA and MDAUAD for computing approximations with the addition and deletion of attributes in PSvIS, respectively. Experimental results on four UCI data sets and two artificial data sets have shown that the proposed incremental algorithms can improve computational performance. In this paper, we fixed the tolerance parameter λ for all attributes, which may be unreasonable in some practical problems. Therefore, we will investigate how to determine the parameter λ more realistically according to specific requirements in the future. Due to the difference of probability distributions of each attribute in PSvIS, it will increase the uncertainty during decision making and classification, etc. Hence, these uncertain problems will be further studied based on the the extended VPRS model.

Acknowledgements

This work is supported by the National Science Foundation of China (Nos. 61573292, 61572406, 61602327, 61603313).

References

[1] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences. 11 (5) (1982) 341–356.
[2] D. Dubois, H. Prade, Putting rough sets and fuzzy sets together, Intelligent Decision Support. 1992 203–232.

[3] S. Greco, B. Matarazzo, R. Slowinski, Rough sets theory for multicriteria decision analysis, European Journal of Operational Research. 129 (1) (2001) 1–47.

[4] T.Y. Lin, Q. Liu, Rough approximate operators: Axiomatic rough set theory, Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer London, 1994, 256–260.

[5] A. Skowron, J. Stepaniuk, Tolerance approximation spaces, Fundamenta Informaticae. 27 (2, 3) (1996) 245–253.

[6] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Information Sciences. 111 (1) (1998) 239–259.

[7] X.H. Zhang, D.R. Miao, C.H. Liu, M.L. Le, Constructive methods of Rough approximation operators and multigranulation rough sets, Knowledge- Based Systems. 91 (2016) 114–125.

[8] W. Kotlowski, R. Slowinski, On nonparametric ordinal classification with monotonicity constraints, IEEE Transactions on Knowledge and Data Engineering. 25 (11) (2013) 2576–2589.

[9] X.B. Yang, Y. Qi, D.J. Yu, H.L. Yu, J.Y. Yang, α-Dominance relation and rough sets in interval-valued information systems, Information Sciences. 294 (2015) 334–347.

[10] T. Beaubouef, R. Ladner, F. Petry, Rough set spatial data modeling for data mining, International Journal of Intelligent Systems. 19 (7) (2004) 567–584.

[11] P. Maji, A rough hypercuboid approach for feature selection in approximation spaces, IEEE Transactions on Knowledge and Data Engineering. 26 (1) (2014) 16–29.

[12] Y. Y. Yao, Rough sets and three-way decisions, in: Rough Sets and Knowledge Technology, Springer, 2015, pp. 62–73.

[13] E. Orłowska, Z. Pawlak, Representation of nondeterministic information, Theoretical Computer Science. 29 (1) (1984) 27–39.

[14] Y.Y. Yao, N. Noroozi, A unified model for set-based computations, in: Soft Computing: 3rd International Workshop on Rough Sets and Soft Computing, 1994, pp. 252–255.

[15] Y.Y. Guan, H.K. Wang, Set-valued information systems, Information Sciences. 176 (17) (2006) 2507–2525.

[16] J.H. Dai, H.W. Tian, Fuzzy rough set model for set-valued data, Fuzzy Sets and Systems. 229 (2013) 54–68.

[17] W. Wei, J.B. Cui, J.Y. Liang, J.H. Wang, Fuzzy rough approximations for set-valued data, Information Sciences. 360 (2016) 181–201.

[18] Y.H. Qian, C.Y. Dang, J.Y. Liang, D.W. Tang, Set-valued ordered information systems, Information Sciences. 179 (16) (2009) 2809–2832.

[19] H.Y. Zhang, S.Y. Yang, Feature selection and approximate reasoning of large-scale set-valued decision tables based on α-dominance-based quantitative rough sets, Information Sciences. 378 (2017) 328–347.

[20] H.W. Wang, M.L. Chen, X.J. Shi, N. Li, Principal component analysis for normal-distribution-valued symbolic data, IEEE Transactions on Cybernetics. 46 (2016) 356–365.

[21] W. Shao, F.D. Salim, A. Song, A. Bouguettaya, Clustering Big Spatiotemporal-Interval Data, IEEE Transactions on Big Data. 1 (2016) 190–203.

[22] J. Perolat, I. Couso, K. Loquin, O. Strauss, Generalizing the Wilcoxon rank-sum test for interval data, International Journal of Approximate Reasoning. 56 (2015) 108–121.

[23] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences. 46 (1) (1993) 39–59.

[24] F.E. Tay, L. Shen, Economic and financial prediction using rough sets model, European Journal of Operational Research. 141 (3) (2002) 641–659.

[25] J.H. Cheng, H.P. Chen, Y.M. Lin, A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4.5, Expert Systems with Applications. 37 (3) (2010) 1814–1820.

[26] C.T. Su, J.H. Hsu, Precision parameter in the variable precision rough sets model: An application, Omega. 34 (2) (2006) 149–157.

[27] J. Zhou, D. Miao, β-interval attribute reduction in variable precision rough set model, Soft Computing. 15 (8) (2011) 1643–1656.

[28] I.T. R. Yanto, P. Vitasari, T. Herawan, M.M. Deris, Applying variable precision rough set model for clustering student suffering studys anxiety, Expert Systems with Applications. 39 (1) (2012) 452–459.

[29] J.L. Wang, P.L. Zhao, S.C. Hoi, R. Jin, Online feature selection and its applications, IEEE Transactions on Knowledge and Data Engineering. 26 (3) (2014) 698–710.

[30] X.D. Wu, K. Yu, W. Ding, H. Wang, X.Q. Zhu, Online feature selection with streaming features, IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (5) (2013) 1178–1192.

[31] S.F. Fang, L.Da Xu, Y.Q. Zhu, J. Ahati, H. Pei, J.W. Yan, Z.H. Liu, An integrated system for regional environmental monitoring and management based on internet of things, IEEE Transactions on Industrial Informatics. 10 (2) (2014) 1596–1605.

[32] V. Ganti, R. Ramakrishnan, Mining and monitoring evolving data, in: Handbook of massive data sets, Springer, 2002, pp. 593–642.

[33] M. Ristin, M. Guillaumin, J. Gall, L.Van Gool, Incremental learning of random forests for large-scale image classification, IEEE Transactions on Pattern Analysis and Machine Intelligence. 38 (3) (2016) 490–503.

[34] C. Luo, T.R. Li, H.M. Chen, H. Fujita, Y. Zhang, Efficient updating of probabilistic approximations with incremental objects, Knowledge-Based Systems. 109 (2016) 71–83.

[35] 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, Knowledge-Based Systems. 20 (5) (2007) 485–494.

[36] C. Luo, T.R. Li, H.M. Chen, Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization, Information Sciences. 257 (2014) 210–228.

[37] X.B. Yang, Y. Qi, H.L. Yu, X.N. Song, J.Y. Yang, Updating multigranulation rough approximations with increasing of granular structures, Knowledge-Based Systems. 64 (2014) 59–69.

[38] F. Wang, J.Y. Liang, Y.H. Qian, Attribute reduction: a dimension incremental strategy, Knowledge-Based Systems. 39 (2013) 95–108.

[39] W.H. Shu, H. Shen, Updating attribute reduction in incomplete decision systems with the variation of attribute set, International Journal of Approximate Reasoning. 55 (3) (2014) 867–884.

[40] T.R. Li, C. Luo, H.M. Chen, J.B. Zhang, PICKT: A solution for big data analysis, Lecture Notes in Computer Science. 9436 (2016) 15-25.

[41] J.M. Ma, Y.Y. Yao, Rough set approximations in multi-granulation fuzzy approximation spaces, Fundamenta Informaticae. 142 (1-4) (2015) 145–160.

[42] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters. 24 (6) (2003) 833–849.

[43] G.L. Liu, Matrix approaches for variable precision rough approximations, in: International Conference on Rough Sets and Knowledge Technology, Springer, 2015, pp. 214–221.

[44] L.W. Ma, Two fuzzy covering rough set models and their generalizations over fuzzy lattices, Fuzzy Sets and Systems. 294 (2016) 1–17.

[45] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, International Journal of Approximate Reasoning. 53 (4) (2012) 620–635.

[46] S.P. Wang, W. Zhu, Q.X. Zhu, F. Min, Characteristic matrix of covering and its application to boolean matrix decomposition, Information Sciences. 263 (2014) 186–197.

[47] A.H. Tan, J.J. Li, Y.J. Lin, G.P. Lin, Matrix-based set approximations and reductions in covering decision information systems, International Journal of Approximate Reasoning. 59 (2015) 68–80.

[48] C. Luo, T.R. Li, Y. Zhang, H. Fujita, Matrix approach to decision-theoretic rough sets for evolving data, Knowledge-Based Systems. 99 (2016) 123–134.

[49] Rudin, Walter, Real and complex analysis, Tata McGraw-Hill Education, 1987.

[50] T. Kailath, The Divergence and Bhattacharyya Distance Measures in Signal Selection, IEEE Transactions on Communication Technology. 15 (1967) 52–60.