International Journal of Approximate Reasoning 53 (2012) 620–635

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

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

Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems

Junbo Zhang [a], Tianrui Li [a,*], Da Ruan [b,c], Dun Liu [d],

[a] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[b] Belgian Nuclear Research Centre (SCK•CEN), Boeretang 200, 2400 Mol, Belgium
[c] Department of Applied Mathematics & Computer Science, Ghent University, 9000 Gent, Belgium
[d] School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China

ARTICLE INFO

Article history:
Received 22 June 2011
Received in revised form 9 October 2011
Accepted 3 January 2012
Available online 9 January 2012

Keywords:
Rough sets
Knowledge discovery
Matrix
Set-valued information systems

ABSTRACT

Set-valued information systems are generalized models of single-valued information systems. The attribute set in the set-valued information system may evolve over time when new information arrives. Approximations of a concept by rough set theory need updating for knowledge discovery or other related tasks. Based on a matrix representation of rough set approximations, a basic vector $H(X)$ is induced from the relation matrix. Four cut matrices of $H(X)$, denoted by $H^{[\mu,\nu]}(X)$, $H^{(\mu,\nu]}(X)$, $H^{[\mu,\nu)}(X)$ and $H^{(\mu,\nu)}(X)$, are derived for the approximations, positive, boundary and negative regions intuitively. The variation of the relation matrix is discussed while the system varies over time. The incremental approaches for updating the relation matrix are proposed to update rough set approximations. The algorithms corresponding to the incremental approaches are presented. Extensive experiments on different data sets from UCI and user-defined data sets show that the proposed incremental approaches effectively reduce the computational time in comparison with the non-incremental approach.

1. Introduction

Rough set theory (RST) was proposed by Pawlak [1–4] in the early 1980s. It is a powerful mathematical tool to describe the dependencies among attributes, evaluate the significance of attributes, and derive decision rules. Nowadays, many rough sets based-approaches have been successfully applied in machine learning and data mining. They have been found to be particularly useful for rule induction [5–7] and feature subset selection [8–12].

An information system is a quadruple $IS = (U, A, V, f)$, where U is a non-empty finite set of objects; A is a non-empty finite set of attributes; $V = \bigcup_{a\in A} V_a$ and V_a is a domain of attribute a; $f : U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. In many practical issues, it happens that some of the attribute values for an object are set-valued, which are always used to characterize uncertain information and missing information in the information system [13]. For an information system $IS = (U, A, V, f)$, if each attribute has a unique attribute value, then IS with $V = \bigcup_{a\in A} V_a$ is called a single-valued information system; if an information system is not a single-valued information system, it is called a set-valued (multi-valued) information system [14,15]. Due to uncertainty and incompleteness of information, there exist a lot of missing data in the information system, called incomplete information system [16,17]. Such information systems can also be regarded as a special case of set-valued information systems, in which all missing values can be represented by the set of all possible values of each attribute [15].

* Corresponding author.
E-mail addresses: JunboZhang86@163.com, jbzhang86@gmail.com (J. Zhang), trli@swjtu.edu.cn (T. Li), druan@sckcen.be, da.ruan@ugent.be (D. Ruan), newton83@163.com (D. Liu).

0888-613X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2012.01.001

Due to the dynamic characteristics of data collection, an information system varies with time. It retrains the system from scratch when the information system varies, which is known as a non-incremental approach [18]. However, the non-incremental approach is often very costly or even intractable. Alternatively, it 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. It has successfully been used to data analysis in the real-time applications and the applications with limited memory or computation ability. Since an information system is consisted of the attributes (features), the objects (instances), and the domain of attributes' values, some incremental updating approaches with respect to rough sets have been proposed in such three aspects.

- *Incremental updating approaches under the variation of the attribute set*. Chan proposed an incremental method for updating the approximations of a concept based on the lower and upper boundary sets [19]. Li et al. presented an incremental method for updating approximations of a concept in an incomplete information system through the characteristic relation [20]. Qian et al. introduced a theoretic framework based on RST, called positive approximation, which could be used to accelerate a heuristic process of attribute reduction [9]. Cheng presented two incremental methods for fast computing the rough fuzzy approximations based on boundary sets and cut sets, respectively [21].
- *Incremental updating approaches under the variation of the object set*. Shan and Ziarko presented a discernibility-matrix based incremental methodology to find all maximally generalized rules [5]. Liu et al. proposed an incremental model and approach as well as its algorithm for inducing interesting knowledge when the object set varies over time [6]. Furthermore, in business intelligent information systems, Liu et al. presented an optimization incremental approach as well as its algorithm for inducing interesting knowledge [7].
- *Incremental updating approaches under the variation of the attribute values*. Chen et al. proposed an incremental algorithm for updating the approximations of a concept under coarsening or refining of attributes' values [22].

Generally, the computation of approximations is a necessary step in knowledge representation and reduction based on rough sets. Approximations may further be applied to data mining and machine learning related work. To our best knowledge, most incremental algorithms mainly focus on the single-valued information systems, but not consider the situation of the set-valued information system. In order to compute and update approximations effectively in the set-valued information systems, it is interesting and desirable to investigate incremental approaches in this kind of information systems. To incrementally update the approximations of a set, the boundary set of a set was taken as a springboard [19,20]. Besides, Cheng took the boundary set and cut set as springboards [21]. By using the inner product method and matrix method, Liu proposed a unified axiomatic system to characterize the upper approximation operations [23]. In this paper, we define a basic vector generated by the relation matrix to drive the lower and upper approximations directly in the set-valued information system. Then we present approaches for updating the lower and upper approximations incrementally by use of the variation of the relation matrix.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts of RST in the set-valued information system. Section 3 proposes the matrix characterizations of the lower and upper approximations. Section 4 presents two approaches for updating the relation matrix and gives some illustrative examples. Section 5 proposes the non-incremental and incremental algorithms. In the Section 6, the performances of non-incremental and incremental methods are evaluated on UCI and user-defined data. The paper ends with conclusions in Section 7.

2. Preliminaries

In this section, we outline basic concepts, notations and results of set-valued information systems [1,14,15,24].

Let $IS = (U, A, V, f)$ be a set-valued information system, where U is a non-empty finite set of objects; A is a non-empty finite set of attributes; V is the set of attributes values; f is a mapping from $U \times A$ to V, where $f : U \times A \rightarrow 2^V$ is a set-valued mapping. Table 1 illustrates a set-valued information system, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $A = \{a_1, a_2, a_3, a_4\}$ and $V = \{0, 1, 2\}$.

A set-valued decision information system $(U, C \cup \{d\}, V, f)$ is a special case of a set-valued information system, where U is a non-empty finite set of objects; C is a non-empty finite set of condition attributes and d is a decision attribute with $C \cap \{d\} = \emptyset$; $V = V_C \cup V_d$, where V_C is the set of condition attribute values and V_d is the set of decision attribute values;

Table 1
A set-valued information system.

U	a_1	a_2	a_3	a_4
x_1	{0}	{0}	{1, 2}	{1, 2}
x_2	{0, 1, 2}	{0, 1, 2}	{1, 2}	{0, 1, 2}
x_3	{1, 2}	{1}	{1, 2}	{1, 2}
x_4	{0, 1}	{0, 2}	{1, 2}	{1}
x_5	{1, 2}	{1, 2}	{1, 2}	{1}
x_6	{1}	{1}	{0, 1}	{0, 1}

f is a mapping from $U \times (C \cup \{d\})$ to V such that $f : U \times C \to 2^{V_C}$ is a set-valued mapping and $f : U \times \{d\} \to V_d$ is a single-valued mapping.

Following are two common ways to present a semantic interpretation of the set-valued information systems [14,15,25–28].

Type A: For $x \in U, b \in A, f(x, b)$ is interpreted conjunctively. For instance, suppose b is the attribute "speaking a language", then $f(x, b) = \{\text{Chinese, Korean, Japanese}\}$ can be interpreted as: x speaks Chinese, Korean, and Japanese. Furthermore, when considering the attribute "feeding habits" of animals, if we denote the attribute value of herbivore as "0" and carnivore as "1", then animals possessing attribute value $\{0, 1\}$ are considered as possessing both herbivorous and carnivorous nature.

Type B: For $x \in U, b \in A, f(x, b)$ is interpreted disjunctively. For instance, suppose b is the attribute "speaking a language", then $f(x, b) = \{\text{Chinese, Korean, Japanese}\}$ can be interpreted as: x speaks Chinese, Korean or Japanese, and x can speak only one of them. Incomplete information systems with some unknown attribute values or partial known attribute values [16,17] are such a type of set-valued information systems.

Definition 1 [14,29]**.** In the set-valued information system (U, A, V, f), for $b \in A$, the tolerance relation T_b is defined as:

$$T_b = \{(x, y) | f(x, b) \cap f(y, b) \neq \emptyset\}. \tag{1}$$

and for $B \subseteq A$, the tolerance relation T_B is defined as follows:

$$T_B = \{(x, y) | \forall b \in B, f(x, b) \cap f(y, b) \neq \emptyset\} = \bigcap_{b \in B} T_b. \tag{2}$$

When $(x, y) \in T_B$, we call x and y are indiscernible or x tolerant with y w.r.t. B. Let $T_B(x) = \{y | y \in U, yT_Bx\}$, we call $T_B(x)$ the tolerance class for x w.r.t. T_B.

Example 1. A set-valued information system is presented in Table 1. Let $B = A$. Here, tolerance classes of objects in U can be computed by Definition 1. Therefore, $T_B(x_1) = \{x_1, x_2, x_4\}$, $T_B(x_2) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $T_B(x_3) = T_B(x_6) = \{x_2, x_3, x_5, x_6\}$, $T_B(x_4) = \{x_1, x_2, x_5, x_6\}$, $T_B(x_5) = \{x_2, x_3, x_4, x_5, x_6\}$.

Definition 2 [29]**.** Given a set-valued information system $IS = (U, A, V, f)$, $\forall X \subseteq U, B \subseteq A$, the lower and upper approximations of X in terms of tolerance relation T_B are defined as

$$\underline{T_B}(X) = \{x \in U | T_B(x) \subseteq X\} \tag{3}$$

$$\overline{T_B}(X) = \{x \in U | T_B(x) \cap X \neq \emptyset\} \tag{4}$$

Intuitively, these two approximations divide the universe U into three disjoint regions: the positive region $POS_{T_B}(X)$, the boundary region $BND_{T_B}(X)$ and the negative region $NEG_{T_B}(X)$, respectively.

$$\begin{cases} POS_{T_B}(X) = \underline{T_B}(X) \\ BND_{T_B}(X) = \overline{T_B}(X) - \underline{T_B}(X) \\ NEG_{T_B}(X) = U - \overline{T_B}(X) \end{cases} \tag{5}$$

The positive region $POS_{T_B}(X)$ is defined by the lower approximation, the boundary region $BND_{T_B}(X)$ is defined by the difference between the upper and lower approximations, and the negative region $NEG_{T_B}(X)$ is defined by the complement of the upper approximation.

Example 2 (Continuation of Example 1)**.** Let $X = \{x_1, x_2, x_3, x_4, x_5\}$. Since $T_B(x_1) \subseteq X$, $T_B(x_2) \not\subseteq X$, $T_B(x_3) \not\subseteq X$, $T_B(x_4) \subseteq X$, $T_B(x_5) \not\subseteq X$ and $T_B(x_6) \not\subseteq X$, then $\underline{T_B}(X) = \{x_1, x_4\}$; since $\forall i \in \{1, 2, 3, 4, 5, 6\}$, $T_B(x_i) \cap X \neq \emptyset$, then $\overline{T_B}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. Hence, $POS_{T_B}(X) = \underline{T_B}(X) = \{x_1, x_4\}$, $BND_{T_B}(X) = \overline{T_B}(X) - \underline{T_B}(X) = \{x_2, x_3, x_5, x_6\}$ and $NEG_{T_B}(X) = U - \overline{T_B}(X) = \emptyset$.

However, the above definitions are not robust enough for tolerating the noisy samples in real applications. Followed by the idea of probabilistic rough sets [30], the rough sets can also be generalized by introducing a measure of inclusion degree. Given two sets X and Y in the universe U, X's inclusion degree in Y is defined as

$$I(X, Y) = \frac{|X \cap Y|}{|X|},$$

where $|\cdot|$ stands for the cardinal number of a set and $X \neq \emptyset$.

Definition 3. Given a set-valued information system $IS = (U, A, V, f)$, $\forall X \subseteq U$, $B \subseteq A$, the (α, β)-probabilistic lower and upper approximation of X in terms of tolerance relation T_B are defined as

$$\underline{T}_B^{(\alpha,\beta)}(X) = \{x \in U | I(T_B(x), X) \geq \alpha\} \tag{6}$$

$$\overline{T}_B^{(\alpha,\beta)}(X) = \{x \in U | I(T_B(x), X) > \beta\} \tag{7}$$

Similarly, the (α, β)-probabilistic positive, boundary and negative regions can be defined by the (α, β)-probabilistic lower and upper approximations, respectively.

$$\begin{cases} POS_{T_B}^{(\alpha,\beta)}(X) = \{x \in U | I(T_B(x), X) \geq \alpha\} \\ BND_{T_B}^{(\alpha,\beta)}(X) = \{x \in U | \beta < I(T_B(x), X) < \alpha\} \\ NEG_{T_B}^{(\alpha,\beta)}(X) = \{x \in U | I(T_B(x), X) \leq \beta\} \end{cases} \tag{8}$$

Clearly, the parameters α and β allow certain acceptable level of errors, and the probabilistic neighborhood rough set model makes the process of decision making more reasonable. Specially, it degrades to the classical rough model if $\alpha = 1$ and $\beta = 0$. In addition, the values of α and β can be automatically computed by using the Bayesian minimum conditional risk criterion in the decision-theoretic rough set model [31].

3. Matrix representation of the approximations

In this section, we give the matrix representation of the lower and upper approximations in the set-valued information system. We propose a matrix approach for computing the approximations in the set-valued information system.

Definition 4. Given two $\mu \times \nu$ matrices $Y = (y_{ij})_{\mu \times \nu}$ and $Z = (z_{ij})_{\mu \times \nu}$, preference relations "$\preceq$" and "$\succeq$" are defined as follows respectively.
(1) Preference relations "$\preceq$": $Y \preceq Z$ if and only if $y_{ij} \leq z_{ij}$, $i = 1, 2, \ldots, \mu$, $j = 1, 2, \ldots, \nu$;
(2) Preference relations "$\succeq$": $Y \succeq Z$ if and only if $y_{ij} \geq z_{ij}$, $i = 1, 2, \ldots, \mu$, $j = 1, 2, \ldots, \nu$;
(3) Minimum min: $\min(Y, Z) = (\min(y_{ij}, z_{ij}))_{\mu \times \nu}$.

Definition 5 [23]. Let $U = \{x_1, x_2, \ldots, x_n\}$, and X be a subset of U. The characteristic function $G(X) = (g_1, g_2, \ldots, g_n)^T$ (T denotes the transpose operation) is defined as

$$g_i = \begin{cases} 1, & x_i \in X \\ 0, & x_i \notin X \end{cases} \tag{9}$$

where $G(X)$ assigns 1 to an element that belongs to X and 0 to an element that does not belong to X.

Example 3. Let $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $X = \{x_1, x_2, x_3, x_4, x_5\}$. Then $G(X) = (1, 1, 1, 1, 1, 0)^T$.

Definition 6 [32]. Given a set-valued information system $IS = (U, A, V, f)$. Let $b \in A$ and T_b be a tolerance relation on U, $M_{n\times n}^{T_b} = (\varphi_{ij})_{n\times n}$ be an $n \times n$ matrix representing T_b, called the relation matrix w.r.t. b. Then

$$\varphi_{ij} = \begin{cases} 1, & (x_i, x_j) \in T_b \\ 0, & (x_i, x_j) \notin T_b \end{cases} \tag{10}$$

Definition 7 [32]. Let $B \subseteq A$ and T_B be a tolerance relation on U, $M_{n\times n}^{T_B} = (m_{ij})_{n\times n}$ be an $n \times n$ matrix representing T_B, called the relation matrix w.r.t. B. Then

$$m_{ij} = \begin{cases} 1, & (x_i, x_j) \in T_B \\ 0, & (x_i, x_j) \notin T_B \end{cases} \tag{11}$$

Corollary 1 [33]. *Let $M_{n\times n}^{T_B} = (m_{ij})_{n\times n}$ and T_B be a tolerance relation on U. Then $m_{ii} = 1$ and $m_{ij} = m_{ji}$, $1 \leq i, j \leq n$.*

Lemma 1. *Let $b_1, b_2 \in A$ and $T_{\{b_1,b_2\}}, T_{b_1}, T_{b_2}$ be the tolerance relation on U. Then $M_{n\times n}^{T_{\{b_1,b_2\}}} = \min\left\{M_{n\times n}^{T_{b_1}}, M_{n\times n}^{T_{b_2}}\right\}$.*

Corollary 2. *Let $B \subseteq A$. Then $M_{n\times n}^{T_B} = \min_{b\in B} M_{n\times n}^{T_b}$.*

Example 4. We continue Example 1. Let $B = A = \{a_1, a_2, a_3, a_4\}$. Then

$$M_{6\times 6}^{T_{a_1}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad M_{6\times 6}^{T_{a_2}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

$$M_{6\times 6}^{T_{a_3}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad M_{6\times 6}^{T_{a_4}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad M_{6\times 6}^{T_B} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Here, it is easy to verify that $M_{6\times 6}^{T_B} = \min\left\{M_{6\times 6}^{T_{a_1}}, M_{6\times 6}^{T_{a_2}}, M_{6\times 6}^{T_{a_3}}, M_{6\times 6}^{T_{a_4}}\right\}$.

Definition 8. Let $B \subseteq A$ and T_B be a tolerance relation on U, $\Lambda_{n\times n}^{T_B}$ be an induced diagonal matrix of $M_{n\times n}^{T_B} = (m_{ij})_{n\times n}$. Then

$$\Lambda_{n\times n}^{T_B} = \mathrm{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}\right) = \mathrm{diag}\left(\frac{1}{\sum_{j=1}^{n} m_{1j}}, \frac{1}{\sum_{j=1}^{n} m_{2j}}, \ldots, \frac{1}{\sum_{j=1}^{n} m_{nj}}\right) \tag{12}$$

where $\lambda_i = \sum_{j=1}^{n} m_{ij}$, $1 \le i \le n$.

Corollary 3. $\Lambda_{n\times n}^{T_B} = \begin{bmatrix} \frac{1}{|T_B(x_1)|} & 0 & \cdots & 0 \\ 0 & \frac{1}{|T_B(x_2)|} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{|T_B(x_n)|} \end{bmatrix}$ *and $1 \le |T_B(x_i)| \le n$, $1 \le i \le n$.*

Example 5 Continuation of Example 4**.** Here we compute the induced diagonal matrix of $M_{6\times 6}^{T_B}$ by Definition 8. $\Lambda_{6\times 6}^{T_B} = \mathrm{diag}(1/3, 1/6, 1/4, 1/4, 1/5, 1/4)$.

Definition 9. The n-column vector called basic vector, denoted by $H(X)$, is defined as:

$$H(X) = \Lambda_{n\times n}^{T_B} \bullet (M_{n\times n}^{T_B} \bullet G(X)) = \Lambda_{n\times n}^{T_B} \bullet \Omega_{n\times 1}^{T_B} \tag{13}$$

where $\bullet$ is dot product of matrices and $\Omega_{n\times 1}^{T_B} = M_{n\times n}^{T_B} \bullet G(X)$.

Corollary 4. *Suppose $H(X) = (h_1, h_2, \cdots, h_n)^T$. Let $\Lambda_{n\times n}^{T_B} = \mathrm{diag}(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n})$, $M_{n\times n}^{T_B} = (m_{ij})_{n\times n}$, $G(X) = (g_1, g_2, \ldots, g_n)^T$, $\Omega_{n\times 1}^{T_B} = (\omega_1, \omega_2, \ldots, \omega_n)^T$, where $\lambda_i = \sum_{j=1}^{n} m_{ij}$, $1 \le i \le n$. Then, $\forall i \in \{1, 2, \ldots, n\}$, $\omega_i = \sum_{j=1}^{n} m_{ij} g_j$, $h_i = \frac{\omega_i}{\lambda_i} = \frac{\sum_{j=1}^{n} m_{ij} g_j}{\sum_{j=1}^{n} m_{ij}}$ and $0 \le h_i \le 1$.*

Definition 10. Let $0 \leq \mu \leq \nu \leq 1$. Four cut matrices of $H(X)$, denoted by $H^{[\mu,\nu]}(X)$, $H^{(\mu,\nu]}(X)$, $H^{[\mu,\nu)}(X)$ and $H^{(\mu,\nu)}(X)$, are defined as follows.

(1) $H^{[\mu,\nu]}(X) = (h'_i)_{n\times 1}$

$$h'_i = \begin{cases} 1, & \mu \leq h_i \leq \nu \\ 0, & \text{else} \end{cases} \tag{14}$$

(2) $H^{(\mu,\nu]}(X) = (h'_i)_{n\times 1}$

$$h'_i = \begin{cases} 1, & \mu < h_i \leq \nu \\ 0, & \text{else} \end{cases} \tag{15}$$

(3) $H^{[\mu,\nu)}(X) = (h'_i)_{n\times 1}$

$$h'_i = \begin{cases} 1, & \mu \leq h_i < \nu \\ 0, & else \end{cases} \tag{16}$$

(4) $H^{(\mu,\nu)}(X) = (h'_i)_{n\times 1}$

$$h'_i = \begin{cases} 1, & \mu < h_i < \nu \\ 0, & else \end{cases} \tag{17}$$

Remark 1. These four cut matrices are Boolean matrices.

Lemma 2. *Given any subset $X \subseteq U$ in a set-valued information system $IS = (U, A, V, f)$, where $U = \{x_1, x_2, \ldots, x_n\}$. $B \subseteq A$ and T_B is a tolerance relation on U. $H(X) = (h_1, h_2, \ldots, h_n)^T$ is the basic vector. Then the lower and upper approximations of X in the set-valued information system can be computed from the cut matrix of $H(X)$ as follows.*

(1) *The n-column boolean vector $G(\underline{T_B}(X))$ of the lower approximation $\underline{T_B}(X)$:*

$$G(\underline{T_B}(X)) = H^{[1,1]}(X) \tag{18}$$

(2) *The n-column boolean vector $G(\overline{T_B}(X))$ of the upper approximation $\overline{T_B}(X)$:*

$$G(\overline{T_B}(X)) = H^{(0,1]}(X) \tag{19}$$

Proof. Suppose $G(X) = (g_1, g_2, \ldots, g_n)^T$, $G(\underline{T_B}(X)) = (g'_1, g'_2, \ldots, g'_n)^T$, $H^{[1,1]}(X) = (h'_1, h'_2, \ldots, h'_n)^T$.

(1) "$\Rightarrow$": $\forall i \in \{1, 2, \ldots, n\}$, if $g'_i = 1$, namely, $x_i \in \underline{T_B}(X)$, then $T_B(x_i) \subseteq X$. Therefore, $\forall x_j \in T_B(x_i)$, we have $(x_j, x_i) \in T_B$ and $x_j \in X$, which means $m_{ij} = 1$, $g_j = 1$. That is, $m_{ij} = m_{ij}g_j$. According to Corollary 4 and Definition 10, $h_i = \frac{\sum_{j=1}^n m_{ij}g_j}{\sum_{j=1}^n m_{ij}} = 1$ and $h'_i = 1$. Hence, $\forall i \in \{1, 2, \ldots, n\}$, $g'_i \leq h'_i$, that is, $G(\underline{T_B}(X)) \preceq H^{[1,1]}(X)$.

"$\Leftarrow$": $\forall i \in \{1, 2, \cdots, n\}$, if $h'_i = 1$, then $h_i = 1$, namely, $\frac{\sum_{j=1}^n m_{ij}g_j}{\sum_{j=1}^n m_{ij}} = h_i = 1$. Obviously, $\forall j \in \{1, 2, \ldots, n\}$, if $m_{ij} = 1$, then $g_j = 1$, in other words, $\forall x_j \in T_B(x_i)$, we have $x_j \in X$. Then, $T_B(x_i) \subseteq X$, namely, $x_i \in \underline{T_B}(X)$ and $g'_i = 1$. Thus $\forall i \in \{1, 2, \ldots, n\}$, $g'_i \geq h'_i$. Therefore, $G(\underline{T_B}(X)) \succeq H^{[1,1]}(X)$.

Thus, $G(\underline{T_B}(X)) = H^{[1,1]}(X)$.

(2) The proof is similar to that of (1). □

Corollary 5. *The positive region $POS_{T_B}(X)$, the boundary region $BND_{T_B}(X)$, and the negative region $NEG_{T_B}(X)$ can also be generated from the cut matrix of $H(X)$, respectively as follows.*

(1) *The n-column boolean vector $G(POS_{T_B}(X))$ of the positive region:*

$$G(POS_{T_B}(X)) = H^{[1,1]}(X) \tag{20}$$

(2) *The n-column boolean vector $G(BND_{T_B}(X))$ of the boundary region:*

$$G(BND_{T_B}(X)) = H^{(0,1)}(X) \tag{21}$$

(3) *The n-column boolean vector* $G(NEG_{T_B}(X))$ *of the negative region:*

$$G(NEG_{T_B}(X)) = H^{[0,0]}(X) \tag{22}$$

Proof. The proof is similar to that of Lemma 2. □

Example 6 Continuation of Examples 4 and 5**.** Since X and $M^{T_B}_{6\times 6}$ have been constructed in Examples 3 and 4, respectively. Then $H(X) = \Lambda^{T_B}_{6\times 6} \bullet (M^{T_B}_{6\times 6} \bullet X) = \text{diag}(1/3, 1/6, 1/4, 1/4, 1/5, 1/4) \bullet (3, 5, 3, 4, 4, 3)^T = (1, 5/6, 3/4, 1, 4/5, 3/4)^T$. We obtain n-column boolean vectors of approximations from $H(X)$ as follows.

$H(X)$	$G(\underline{T_B}(X)) = H^{[1,1]}(X)$	$G(\overline{T_B}(X)) = H^{(0,1]}(X)$	$G(POS_{T_B}(X)) = H^{[1,1]}(X)$	$G(BND_{T_B}(X)) = H^{(0,1)}(X)$	$G(NEG_{T_B}(X)) = H^{[0,0]}(X)$
$\begin{bmatrix} 1 \\ 5/6 \\ 3/4 \\ 1 \\ 4/5 \\ 3/4 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

In other words, $\underline{T_B}(X) = \{x_1, x_4\}$, $\overline{T_B}(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $POS_{T_B}(X) = \{x_1, x_4\}$, $BND_{T_B}(X) = \{x_2, x_3, x_5, x_6\}$ and $NEG_{T_B}(X) = \emptyset$, respectively.

Corollary 6. *The* (α, β)*-probabilistic lower and upper approximations and* (α, β)*-probabilistic positive, boundary and negative regions are generated from the cut matrix of* $H(X)$ *as follows.*

(1) *The n-column boolean vector* $G(\underline{T_B}^{(\alpha,\beta)}(X))$ *of the lower approximation* $\underline{T_B}^{(\alpha,\beta)}(X)$*:*

$$G\left(\underline{T_B}^{(\alpha,\beta)}(X)\right) = H^{[\alpha,1]}(X) \tag{23}$$

(2) *The n-column boolean vector* $G(\overline{T_B}^{(\alpha,\beta)}(X))$ *of the upper approximation* $\overline{T_B}^{(\alpha,\beta)}(X)$*:*

$$G\left(\overline{T_B}^{(\alpha,\beta)}(X)\right) = H^{(\beta,1]}(X) \tag{24}$$

(3) *The n-column boolean vector* $G(POS^{(\alpha,\beta)}_{T_B}(X))$ *of the positive region:*

$$G\left(POS^{(\alpha,\beta)}_{T_B}(X)\right) = H^{[\alpha,1]}(X) \tag{25}$$

(4) *The n-column boolean vector* $G(BND^{(\alpha,\beta)}_{T_B}(X))$ *of the boundary region:*

$$G\left(BND^{(\alpha,\beta)}_{T_B}(X)\right) = H^{(\alpha,\beta)}(X) \tag{26}$$

(5) *The n-column boolean vector* $G(NEG^{(\alpha,\beta)}_{T_B}(X))$ *of the negative region:*

$$G\left(NEG^{(\alpha,\beta)}_{T_B}(X)\right) = H^{[0,\beta]}(X) \tag{27}$$

Proof. The proof is similar to that of Lemma 2. □

4. Updating approximations in set-valued information systems incrementally based on the matrix

In this section, we consider the problem of updating approximations of a subset X of U in terms of variation of the attribute set in set-valued information systems. The variation of the attribute set, also called the attribute generalization [19], contains two situations, adding attributes and removing attributes, respectively. Since the lower approximation, upper approximation, positive, boundary and negative regions can be induced from the basic vector $H(X)$ directly. The key step for computing basic vector $H(X)$ is to construct the relation matrix. Thus, it reduces running time if we can renew the relation matrix with an incremental updating strategy rather than reconstructing it from scratch. In the following, we discuss how to update the relation matrix incrementally while the attribute set varies.

Lemma 3. *Let P and Q be two attribute sets. Then $M_{n\times n}^{T_P} \succeq M_{n\times n}^{T_{P\cup Q}}$, $M_{n\times n}^{T_Q} \succeq M_{n\times n}^{T_{P\cup Q}}$.*

4.1. Updating approximations incrementally when adding an attribute set

Corollary 7. *Let $P, Q \subseteq A$ and $Q \cap P = \emptyset$. Suppose $M_{n\times n}^{T_{P\cup Q}} = (m_{ij}^{\uparrow})_{n\times n}$ is a relation matrix representing $T_{P\cup Q}$, and $M_{n\times n}^{T_P} = (m_{ij})_{n\times n}$ a relation matrix representing T_P. The relation matrix by adding Q to P can be updated as:*

$$m_{ij}^{\uparrow} = \begin{cases} 0, & m_{ij} = 0 \vee (x_i, x_j) \notin T_Q \\ 1, & m_{ij} = 1 \wedge (x_i, x_j) \in T_Q \end{cases} \tag{28}$$

Remark 2. When adding Q to P, m_{ij} is constant while $m_{ij} = 0$.

Example 7 (Continuation of Example 4). Let $P = \{a_1, a_4\}$, $Q = \{a_2, a_3\}$. Here, we update the matrix by adding Q to P. According to Definition 7, we have $M_{6\times 6}^{T_P} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$. According to Corollary 7, if $m_{ij} = 0$, then $m_{ij}^{\uparrow} = 0$. Moreover, $m_{ii} = 1$ and $m_{ij} = m_{ji}$ by Corollary 1. Hence, we just update the m_{ij} which value is equal to 1 in $M_{6\times 6}^{T_P}$ and $1 \le i \le n,\ i < j \le n$. Hence, $M_{6\times 6}^{T_{P\cup Q}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \underline{0} & 1 & 1 \\ 1 & 1 & \underline{0} & 1 & 1 & \underline{0} \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \underline{0} & 1 & 1 \end{bmatrix}$, where the four $\underline{0}$s are changed.

Corollary 8. *Let $P, Q \subseteq A$, $Q \cap P = \emptyset$ and $\Lambda_{n\times n}^{T_P} = \mathrm{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}\right)$, where $\lambda_i = \sum_{j=1}^{n} m_{ij}$. Suppose $\Lambda_{n\times n}^{T_{P\cup Q}} = \mathrm{diag}\left(\frac{1}{\lambda_1^{\uparrow}}, \frac{1}{\lambda_2^{\uparrow}}, \ldots, \frac{1}{\lambda_n^{\uparrow}}\right)$. Then $\lambda_i^{\uparrow} \le \lambda_i, 1 \le i \le n$. Moreover, $\lambda_i^{\uparrow} = \lambda_i - \sum_{j=1}^{n} m_{ij} \oplus m_{ij}^{\uparrow}$, where $\oplus$ is the logical operation exclusive disjunction (exclusive or).*

Proof. According to Remark 2, m_{ij} is constant while $m_{ij} = 0$ when adding Q to P. Hence, m_{ij} is constant or m_{ij} changes from 1 to 0. (a) When m_{ij} is constant, λ_i is constant; (b) When m_{ij} changes from 1 to 0, λ_i decreases because $\lambda_i = \sum_{j=1}^{n} m_{ij}$. Thus, $\lambda_i^{\uparrow} = \lambda_i - \sum_{j=1}^{n} m_{ij} \oplus m_{ij}^{\uparrow}$. □

Example 8 (Continuation of Example 7). $\Lambda_{6\times 6}^{T_P} = \mathrm{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \cdots, \frac{1}{\lambda_6}\right) = \mathrm{diag}(1/3, 1/6, 1/5, 1/6, 1/5, 1/5)$. After adding Q to P, by Corollary 8, we have $\lambda_1^{\uparrow} = \lambda_1 = 3$, $\lambda_2^{\uparrow} = \lambda_2 = 6$, $\lambda_3^{\uparrow} = \lambda_3 - 1 = 4$, $\lambda_4^{\uparrow} = \lambda_4 - 2 = 4$, $\lambda_5^{\uparrow} = \lambda_5 = 5$ and $\lambda_6^{\uparrow} = \lambda_6 - 1 = 4$. Hence, $\Lambda_{6\times 6}^{T_{P\cup Q}} = \mathrm{diag}(1/3, 1/6, 1/4, 1/4, 1/5, 1/4)$.

Corollary 9. *Let $P, Q \subseteq A$, $Q \cap P = \emptyset$ and $\Omega_{n\times 1}^{T_P} = M_{n\times n}^{T_P} \bullet G(X) = (\omega_1, \omega_2, \ldots, \omega_n)^T$, where $\omega_i = \sum_{j=1}^{n} m_{ij} g_j$ and $G(X) = \{g_1, g_2, \ldots, g_n\}$. Suppose $\Omega_{n\times n}^{T_{P\cup Q}} = (\omega_1^{\uparrow}, \omega_2^{\uparrow}, \ldots, \omega_n^{\uparrow})^T$. Then $\omega_i^{\uparrow} \le \omega_i$, $1 \le i \le n$, and $\omega_i^{\uparrow} = \omega_i - \sum_{j=1}^{n}((m_{ij} \oplus m_{ij}^{\uparrow}) \wedge g_j)$.*

Proof. The proof is similar to that of Corollary 8. □

Example 9 (Continuation of Example 7). Since $G(X) = (1, 1, 1, 1, 1, 0)^T$, then $\Omega^{T_P}_{6\times 1} = M^{T_P}_{6\times 6} \bullet G(X) = (\omega_1, \omega_2, \cdots, \omega_6)^T = (3, 5, 4, 5, 4, 4)^T$. When adding Q to P, according to Corollary 9, $\omega^{\uparrow}_1 = \omega_1 = 3$, $\omega^{\uparrow}_2 = \omega_2 = 5$, $\omega^{\uparrow}_3 = \omega_3 - 1 = 3$, $\omega^{\uparrow}_4 = \omega_4 - 2 = 4$, $\omega^{\uparrow}_5 = \omega_5 = 4$ and $\omega^{\uparrow}_6 = \omega_6 - 1 = 3$. Thus, $\Omega^{T_{P\cup Q}}_{6\times 1} = (3, 5, 3, 4, 4, 3)^T$.

In the subsection, we discuss about how to update $M^{T_P}_{n\times n}$, $\Lambda^{T_P}_{n\times n}$ and $\Omega^{T_P}_{n\times 1}$ when adding an attribute set Q to P. Then, $H(X)$ can be obtained by Corollary 4 and approximations can be computed from the cut matrix of $H(X)$.

4.2. Updating approximations incrementally when deleting an attribute set

Corollary 10. *Let $Q \subset P \subseteq A$. Suppose $M^{T_{P-Q}}_{n\times n} = (m^{\downarrow}_{ij})_{n\times n}$ is a relation matrix representing T_{P-Q}, and $M^{T_P}_{n\times n} = (m_{ij})_{n\times n}$ is a relation matrix representing T_P. The relation matrix by deleting Q from P can be updated as:*

$$m^{\downarrow}_{ij} = \begin{cases} 0, & m_{ij} = 0 \wedge (x_i, x_j) \notin T_{P-Q} \\ 1, & m_{ij} = 1 \vee (x_i, x_j) \in T_{P-Q} \end{cases} \tag{29}$$

Remark 3. When deleting Q from P, m_{ij} is constant while $m_{ij} = 1$.

Example 10. We continue Example 4. Let $P = \{a_1, a_2, a_3, a_4\}$, $Q = \{a_2, a_3\}$. Here, we update the matrix by deleting Q from P. By Example 4, $M^{T_P}_{6\times 6} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$. Then, $m_{ii} = 1$ and $m_{ij} = m_{ji}$ by Corollary 1. Furthermore, according to Corollary 10, $m^{\downarrow}_{ij} = 1$ when $m_{ij} = 1$. Therefore, we just update the m_{ij} whose value is equal to 0 in $M^{T_P}_{6\times 6}$ and $1 \le i \le n, i < j \le n$. We have $M^{T_{P-Q}}_{6\times 6} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \underline{1} & 1 & 1 \\ 1 & 1 & \underline{1} & 1 & 1 & \underline{1} \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \underline{1} & 1 & 1 \end{bmatrix}$, where the four $\underline{1}$s are changed.

Corollary 11. *Let $Q \subset P \subseteq A$ and $\Lambda^{T_P}_{n\times n} = \mathrm{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}\right)$, where $\lambda_i = \sum_{j=1}^{n} m_{ij}$. Suppose $\Lambda^{T_{P-Q}}_{n\times n} = \mathrm{diag}\left(\frac{1}{\lambda^{\downarrow}_1}, \frac{1}{\lambda^{\downarrow}_2}, \ldots, \frac{1}{\lambda^{\downarrow}_n}\right)$. Then $\lambda^{\downarrow}_i \ge \lambda_i$, $1 \le i \le n$. Moreover, $\lambda^{\downarrow}_i = \lambda_i + \sum_{j=1}^{n} m_{ij} \oplus m^{\downarrow}_{ij}$.*

Proof. The proof is similar to that of Corollary 8. □

Example 11 (Continuation of Example 10). $\Lambda^{T_P}_{6\times 6} = \mathrm{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_6}\right) = \mathrm{diag}(1/3, 1/6, 1/4, 1/4, 1/5, 1/4)$. After deleting Q from P, by Corollary 11, we have $\lambda^{\downarrow}_1 = \lambda_1 = 3$, $\lambda^{\downarrow}_2 = \lambda_2 = 6$, $\lambda^{\downarrow}_3 = \lambda_3 + 1 = 5$, $\lambda^{\downarrow}_4 = \lambda_4 + 2 = 6$, $\lambda^{\downarrow}_5 = \lambda_5 = 5$ and $\lambda^{\downarrow}_6 = \lambda_6 + 1 = 5$. Hence, $\Lambda^{T_{P-Q}}_{6\times 6} = \mathrm{diag}(1/3, 1/6, 1/5, 1/6, 1/5, 1/5)$.

Corollary 12. *Let $Q \subset P \subseteq A$ and $\Omega^{T_P}_{n\times 1} = M^{T_P}_{n\times n} \bullet G(X) = (\omega_1, \omega_2, \ldots, \omega_n)^T$, where $\omega_i = \sum_{j=1}^{n} m_{ij}g_j$ and $G(X) = \{g_1, g_2, \ldots, g_n\}$. Suppose $\Omega^{T_{P-Q}}_{n\times n} = \left(\omega_1^{\downarrow}, \omega_2^{\downarrow}, \ldots, \omega_n^{\downarrow}\right)^T$. Then $\omega_i^{\downarrow} \geq \omega_i$, $1 \leq i \leq n$, and $\omega_i^{\downarrow} = \omega_i + \sum_{j=1}^{n}((m_{ij} \oplus m_{ij}^{\downarrow}) \wedge g_j)$.*

Proof. The proof is similar to that of Corollary 8. □

Example 12 (Continuation of Example 10)**.** Since $G(X) = (1, 1, 1, 1, 1, 0)^T$, then $\Omega^{T_P}_{6\times 1} = M^{T_P}_{6\times 6} \bullet G(X) = (\omega_1, \omega_2, \cdots, \omega_6)^T = (3, 5, 3, 4, 4, 3)^T$. When deleting Q from P, according to Corollary 12, $\omega_1^{\downarrow} = \omega_1 = 3$, $\omega_2^{\downarrow} = \omega_2 = 5$, $\omega_3^{\downarrow} = \omega_3 + 1 = 3$, $\omega_4^{\downarrow} = \omega_4 + 2 = 4$, $\omega_5^{\downarrow} = \omega_5 = 4$ and $\omega_6^{\downarrow} = \omega_6 + 1 = 3$. Thus, $\Omega^{T_{P-Q}}_{6\times 1} = (3, 5, 4, 5, 4, 4)^T$.

In this subsection, we proposed the methods of updating $M^{T_P}_{n\times n}$, $\Lambda^{T_P}_{n\times n}$ and $\Omega^{T_P}_{n\times 1}$ when deleting an attribute set Q from P. Then, $H(X)$ can be obtained by Corollary 4 and approximations can be computed from the cut matrix of $H(X)$.

5. Development of static and dynamic algorithms for computing approximations based on the matrix

In this section, we design static and dynamic algorithms for computing approximations based on the matrix in the set-valued decision information system corresponding to Sections 3 and 4, respectively.

5.1. Algorithm SACAM (The static algorithm for computing approximations based on the matrix)

Algorithm 1: The static algorithm for computing approximations of decision classes in the set-valued decision information system based on the matrix (SACAM)

Input: (1) A set-valued decision information system with a condition attribute set P and a decision attribute d;
(2) The thresholds α and β.
Output: The lower and upper approximations of the decision classes in the set-valued decision information system.

1 **begin**
2   **for** $i = 1$ **to** n **do** // Construct the relation matrix with respect to T_P, $M^{T_P}_{n\times n} = (m_{ij})_{n\times n}$
3     $m_{ii} = 1$; // According Corollary 1, $m_{ii} = 1$
4   **for** $j = i + 1$ **to** n **do** // According to Definitions 1 and 7
5   **if** $(x_i, x_j) \in T_P$ **then** $m_{ij} = m_{ji} = 1$;
6   **else** $m_{ij} = m_{ji} = 0$;
7   **end**
8   **end**
9   **for** $i = 1$ **to** n **do** // Compute the induced diagonal matrix of $M^{T_P}_{n\times n}$: $\Lambda = \mathrm{diag}(1/\lambda_1, 1/\lambda_2, \cdots, 1/\lambda_n)$
10     $\lambda_i = \sum_{j=1}^{n} m_{ij}$;
11   **end**
12   From the set-valued decision information system,
13   **for** *each class X_k in the decision attribute* **do** // Compute the approximations of the decision classes
14   Construct the n-column Boolean vector of X_k: $X_k = (g_{k1}, g_{k2}, \ldots, g_{kn})^T$;
15   **for** $i = 1$ **to** n **do** // Compute intermediate values $\Omega = (\omega_{ki})$
16       $\omega_{ki} = \sum_{j=1}^{n} m_{ij}g_{kj}$;
17   **end**
18   **Let** $H(X_k) = (h_{k1}, h_{k2}, \ldots, h_{kn})^T$;
19   **for** $i = 1$ **to** n **do** // Compute $H(X_k)$
20       $h_{ki} = \frac{\omega_{ki}}{\lambda_i}$;
21   **end**
22   **Compute** the n-column Boolean vector of lower approximation: $G(\underline{T_P}^{(\alpha,\beta)}(X_k)) = H^{[\alpha,1]}(X_k)$;
23   **Compute** the n-column Boolean vector of upper approximation: $G(\overline{T_P}^{(\alpha,\beta)}(X_k)) = H^{(\beta,1]}(X_k)$;
24   **Output** the approximations $\underline{T_P}^{(\alpha,\beta)}(X_k)$ and $\overline{T_P}^{(\alpha,\beta)}(X_k)$ by $G(\underline{T_P}^{(\alpha,\beta)}(X_k))$ and $G(\overline{T_P}^{(\alpha,\beta)}(X_k))$, respectively.
25   **end**
26 **end**

Algorithm SACAM (see Algorithm 1) is a static (non-incremental) algorithm for computing approximations of the decision classes based on the matrix while the set-valued decision information system is constant. Steps 2-8 are to construct the relation matrix $M^{T_P}_{n\times n}$ by Definition 7 and Corollary 1. According to Definition 8, steps 9-11 are to compute the induced diagonal matrix. Step 14 is to construct the n-column Boolean vector of a concept X. Steps 15-17 are to compute the intermediate values Ω. Steps 18-21 are to compute the basic vector $H(X)$ by Definition 9. Steps 22-25 are to compute approximations by Corollary 6.

5.2. *Algorithm DAUAM-A (The dynamic algorithm for updating approximations based on the matrix when adding an attribute set)*

Algorithm 2: The dynamic algorithm for updating approximations based on the matrix when adding an attribute set (DAUAM-A)

Input: (1) A set-valued decision information system with a condition attribute set P and a decision attribute d;
(2) The thresholds α and β;
(3) Intermediate results: $M = (m_{ij})_{n\times n}$ (relation matrix), $\Lambda = \mathrm{diag}(1/\lambda_i)$ (induced matrix), $\Omega = (\omega_{ki})$, $X_k = (g_{ki})^T$ (n-column Boolean vector of X_k), which have been computed by Algorithm SACAM;
(4) Adding an attribute set Q and $Q \cap P = \emptyset$.
Output: Updated lower and upper approximations of the decision classes in the set-valued decision information system.

1 **begin**
2 $P \longleftarrow P \cup Q$; `// An attribute set` Q `is added to the attribute set` P
3 **for** $i = 1$ **to** n **do**
4 $m_{ii} = 1$; `// According Corollary 1,` $m_{ii} = 1$ `is constant`
5 **for** $j = i + 1$ **to** n **do**
6 **if** $m_{ij} == 0$ **then**
7 $m_{ij}=m_{ji}=0$; `// According to Corollary 7, when adding` Q `to` P`,` m_{ij} `is constant while` $m_{ij}=0$
8 **else**
9 **if** $(x_i, x_j) \notin T_Q$ **then**
10 $m_{ij} = m_{ji} = 0$; `// Update relation matrix:` $M = (m_{ij})_{n\times n}$
11 $\lambda_i = \lambda_i - 1$ and $\lambda_j = \lambda_j - 1$; `// Update induced matrix:` $\Lambda = \mathrm{diag}(1/\lambda_i)$
12 **for** *each* k **do** `// Update intermediate result:` $\Omega = (\omega_{ki})$
13 **if** $g_{ki} == 1$ **then** $\omega_{ki} = \omega_{ki} - 1$;
14 **if** $g_{kj} == 1$ **then** $\omega_{kj} = \omega_{kj} - 1$;
15 **end**
16 **else**
17 $m_{ij} = m_{ji} = 1$; `// According to Corollary 7`
18 **end**
19 **end**
20 **end**
21 **end**
22 From the set-valued decision information system,
23 **for** *each class* X_k *in the decision attribute* **do** `// Compute the approximations of the decision classes`
24 **Let** $H(X_k) = (h_{k1}, h_{k2}, \ldots, h_{kn})^T$;
25 **for** $i = 1$ **to** n **do** `// Compute` $H(X_k)$
26 $h_{ki} = \frac{\omega_{ki}}{\lambda_i}$;
27 **end**
28 **Compute** the n-column Boolean vector of lower approximation: $G(\underline{T_P}^{(\alpha,\beta)}(X_k)) = H^{[\alpha,1]}(X_k)$;
29 **Compute** the n-column Boolean vector of upper approximation: $G(\overline{T_P}^{(\alpha,\beta)}(X_k)) = H^{(\beta,1]}(X_k)$;
30 **Output** the approximations $\underline{T_P}^{(\alpha,\beta)}(X_k)$ and $\overline{T_P}^{(\alpha,\beta)}(X_k)$ by $G(\underline{T_P}^{(\alpha,\beta)}(X_k))$ and $G(\overline{T_P}^{(\alpha,\beta)}(X_k))$, respectively.
31 **end**
32 **end**

Algorithm DAUAM-A (see Algorithm 2) is a dynamic (incremental) algorithm for updating approximations while adding an attribute set. Step 2 is to update the condition attribute set P. Steps 3-21 are to update intermediate results: $M = (m_{ij})_{n\times n}$ (relation matrix), $\Lambda = \mathrm{diag}(1/\lambda_i)$ (induced matrix), $\Omega = (\omega_{ki})$ when adding some attributes, which are the key steps in the Algorithm DAUAM-A. Specially, Step 10 is to update the relation matrix, Step 11 is to update the induced matrix and Steps 12-15 are to update the intermediate result Ω according to Corollaries 7, 8 and 9, respectively. Steps 22-31 are to compute basic vectors of the decision classes and approximations by Corollary 6.

5.3. *Algorithm DAUAM-D (The dynamic algorithm for updating approximations based on the matrix when deleting an attribute set)*

Algorithm DAUAM-D (see Algorithm 3) is a dynamic (incremental) algorithm for updating approximations while deleting an attribute set. Step 2 is to update the condition attribute set P. Steps 3-21 are to update intermediate results: $M = (m_{ij})_{n\times n}$ (relation matrix), $\Lambda = \mathrm{diag}(1/\lambda_i)$ (induced matrix), $\Omega = (\omega_{ki})$ when deleting some attributes, which are the key steps in the Algorithm DAUAM-A. Specially, Step 10 is to update the relation matrix, Step 11 is to update the induced matrix and

Algorithm 3: The dynamic algorithm for updating approximations based on the matrix when deleting an attribute set (DAUAM-D)

Input: (1) A set-valued decision information system with a condition attribute set P and a decision attribute d;
(2) The thresholds α and β;
(3) Intermediate result: $M = (m_{ij})_{n\times n}$ (relation matrix), $\Lambda = \text{diag}(1/\lambda_i)$ (induced matrix), $\Omega = (\omega_{ki})$, $X_k = (g_{ki})^T$ (n-column Boolean vector of X_k), which have been computed by Algorithm SACAM;
(4) Deleting an attribute set Q and $Q \subseteq P$.
Output: Updated lower and upper approximations of the decision classes in the set-valued decision information system.

1 **begin**
2 $P \longleftarrow P - Q$; // An attribute set Q is deleted from the attribute set P
3 **for** $i = 1$ **to** n **do**
4 $m_{ii} = 1$; // According Corollary 1, $m_{ii} = 1$ is constant
5 **for** $j = i + 1$ **to** n **do**
6 **if** $m_{ij} = 1$ **then**
7 m_{ij}=m_{ji}=1;// According to Corollary 10, when deleting Q, m_{ij} is constant while m_{ij}=1
8 **else**
9 **if** $(x_i, x_j) \in T_{P-Q}$ **then**
10 $m_{ij} = m_{ji} = 1$; // Update relation matrix: $M = (m_{ij})_{n\times n}$
11 $\lambda_i = \lambda_i + 1$ and $\lambda_j = \lambda_j + 1$; // Update induced matrix: $\Lambda = \text{diag}(1/\lambda_i)$
12 **for** *each* k **do** // Update intermediate result: $\Omega = (\omega_{ki})$
13 **if** $g_{ki} == 1$ **then** $\omega_{ki} = \omega_{ki} + 1$;
14 **if** $g_{kj} == 1$ **then** $\omega_{kj} = \omega_{kj} + 1$;
15 **end**
16 **else**
17 $m_{ij} = m_{ji} = 0$; // According to Corollary 10
18 **end**
19 **end**
20 **end**
21 **end**
22 From the set-valued decision information system,
23 **for** *each class* X_k *in the decision attribute* **do** // Compute the approximations of the decision classes
24 **Let** $H(X_k) = (h_{k1}, h_{k2}, \ldots, h_{kn})^T$;
25 **for** $i = 1$ **to** n **do** // Compute $H(X_k)$
26 $h_{ki} = \frac{\omega_{ki}}{\lambda_i}$;
27 **end**
28 **Compute** the n-column Boolean vector of lower approximation: $G(\underline{T_P}^{(\alpha,\beta)}(X_k)) = H^{[\alpha,1]}(X_k)$;
29 **Compute** the n-column Boolean vector of upper approximation: $G(\overline{T_P}^{(\alpha,\beta)}(X_k)) = H^{(\beta,1]}(X_k)$;
30 **Output** the approximations $\underline{T_P}^{(\alpha,\beta)}(X_k)$ and $\overline{T_P}^{(\alpha,\beta)}(X_k)$ by $G(\underline{T_P}^{(\alpha,\beta)}(X_k))$ and $G(\overline{T_P}^{(\alpha,\beta)}(X_k))$, respectively.
31 **end**
32 **end**

Steps 12-15 are to update the intermediate result Ω according to Corollaries 10, 11 and 12, respectively. Steps 22-31 are to compute basic vectors of the decision classes and approximations by Corollary 6.

6. Experimental evaluations

To test performance of Algorithms SACAM, DAUAM-A and DAUAM-D, we download five data sets from the machine learning data repository, University of California at Irvine [34], which are all symbolic data with missing values. Here, all missing values are represented by the set of all possible values of each attribute, then the incomplete information system is regarded as a special case of the set-valued information system [15]. Besides, the set-valued data generator is developed and four different set-valued data sets are generated. All these data sets are outlined in Table 2.

In what follows, we compare the computational times of non-incremental and incremental algorithms. To distinguish the computational times, we divide each of these nine data sets into five parts of equal size. The first part is regarded as the 1st data set, the combination of the 1st data set and the second part is viewed as the 2nd data set, the combination of the 2nd data set and the third part is regarded as the 3rd data set, $\cdots$, the combination of all five parts is viewed as the 5th data set. These data sets can be used to calculate the computational time used by non-incremental and incremental algorithms and show it vis-a-vis the sizes of the universe. In addition, by Algorithms SACAM, DAUAM-A and DAUAM-D, since the (α, β)-probabilistic approximations can be computed from $H(X)$ with two thresholds α, β directly, then different thresholds cost the same operations. Thus, we test performance of these three algorithms with the constant thresholds $\alpha = 1$ and $\beta = 0$. The computations are conducted on a PC with Windows 7 and Inter(R) Core(TM)2 CPU P8400 and 2 GB memory. Algorithms are coded in C++ and the software being used is Microsoft Visual 2008.

Table 2
A description of data sets.

	Data sets	Abbreviation	Samples	Attributes	Classes	Source
1	Congressional voting records	CVR	435	16	2	UCI
2	Mushroom	Mushroom	8124	22	2	UCI
3	Soybean-large	Soybean	307	35	19	UCI
4	Audiology (Standardized)	Audiology	226	69	24	UCI
5	Dermatology	Dermatology	336	34	6	UCI
6	Set-valued data one	SVD1	2000	5	5	Data generator
7	Set-valued data two	SVD2	200	100	4	Data generator
8	Set-valued data three	SVD3	500	10	8	Data generator
9	Set-valued data four	SVD4	800	8	20	Data generator

Table 3
A description of adding attribute set.

	Data sets	Attributes	Original attribute set	Adding attribute set
1	CVR	16	$\{a_7, a_8, \ldots, a_{16}\}$	$\{a_1, a_2, \ldots, a_6\}$
2	Mushroom	22	$\{a_1, a_2, \ldots, a_{15}\}$	$\{a_{16}, a_{17}, \ldots, a_{22}\}$
3	Soybean	35	$\{a_1, a_2, \ldots, a_{30}\}$	$\{a_{31}, a_{32}, \ldots, a_{35}\}$
4	Audiology	69	$\{a_1, a_2, \ldots, a_{20}, a_{31}, a_{32}, \ldots, a_{69}\}$	$\{a_{21}, a_{22}, \cdots, a_{30}\}$
5	Dermatology	34	$\{a_1, a_2, \ldots, a_{27}\}$	$\{a_{28}, a_{29}, \ldots, a_{34}\}$
6	SVD1	5	$\{a_1, a_2, a_3\}$	$\{a_4, a_5\}$
7	SVD2	100	$\{a_{11}, a_{12}, \ldots, a_{100}\}$	$\{a_1, a_2, \ldots, a_{10}\}$
8	SVD3	10	$\{a_1, a_2, a_3, a_7, \ldots, a_{10}\}$	$\{a_4, a_5, a_6\}$
9	SVD4	8	$\{a_1, a_3, a_4, a_6, a_7\}$	$\{a_2, a_5, a_8\}$

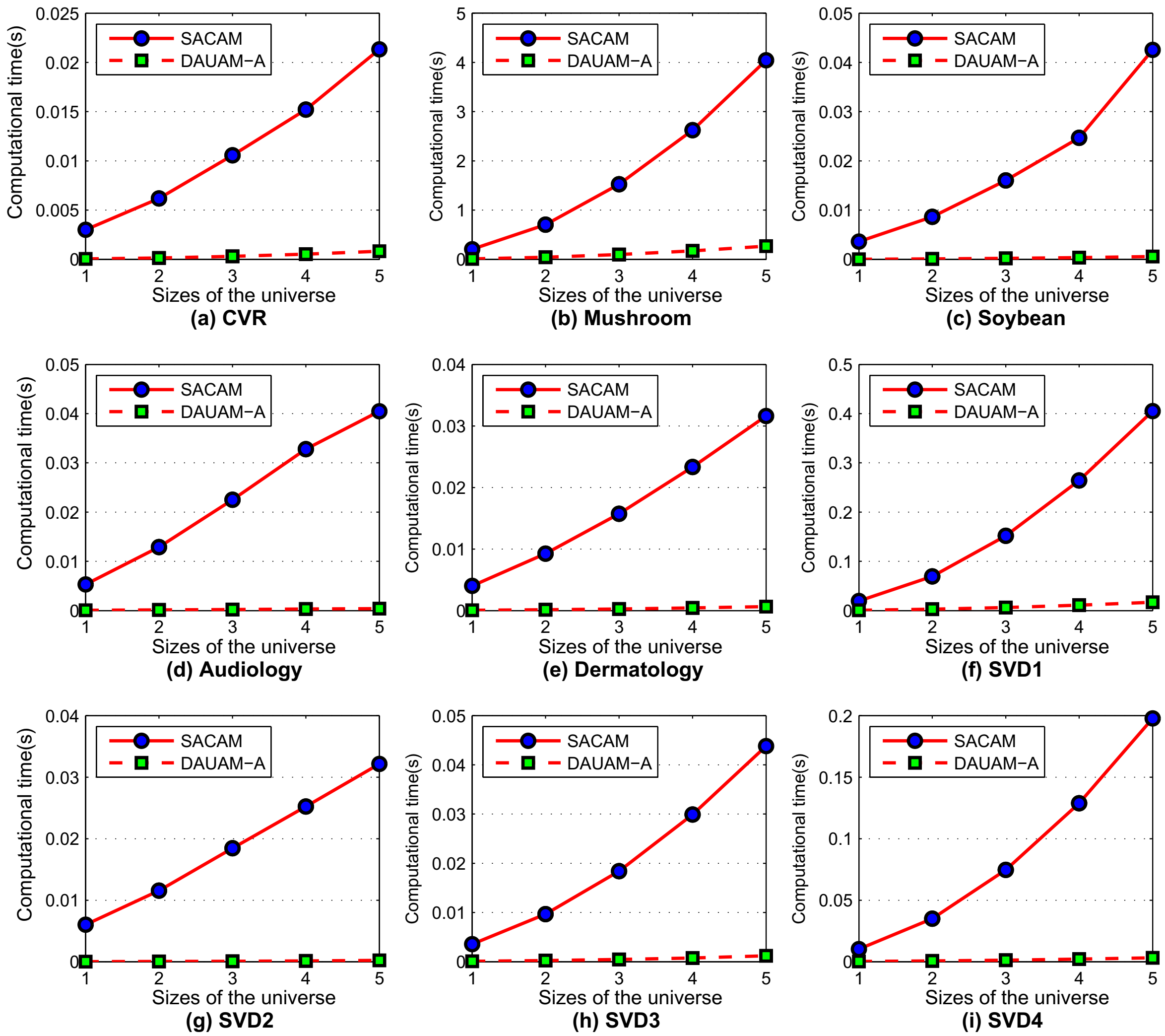

Fig. 1. A comparison of Algorithm SACAM and DAUAM-A versus the size of data when adding an attribute set.

Table 4
A description of deleting attribute set.

	Data sets	Attributes	Original attribute set	Deleting attribute set
1	CVR	16	$\{a_1, a_2, \ldots, a_{16}\}$	$\{a_1, a_2, \ldots, a_6\}$
2	Mushroom	22	$\{a_1, a_2, \ldots, a_{22}\}$	$\{a_{16}, a_{17}, \ldots, a_{22}\}$
3	Soybean	35	$\{a_1, a_2, \ldots, a_{35}\}$	$\{a_{31}, a_{32}, \ldots, a_{35}\}$
4	Audiology	69	$\{a_1, a_2, \ldots, a_{69}\}$	$\{a_{21}, a_{22}, \ldots, a_{30}\}$
5	Dermatology	34	$\{a_1, a_2, \ldots, a_{34}\}$	$\{a_{28}, a_{29}, \ldots, a_{34}\}$
6	SVD1	5	$\{a_1, a_2, a_3, a_4, a_5\}$	$\{a_4, a_5\}$
7	SVD2	100	$\{a_1, a_2, \ldots, a_{100}\}$	$\{a_1, a_2, \ldots, a_{10}\}$
8	SVD3	10	$\{a_1, a_2, \ldots, a_{10}\}$	$\{a_4, a_5, a_6\}$
9	SVD4	8	$\{a_1, a_2, \ldots, a_8\}$	$\{a_2, a_5, a_8\}$

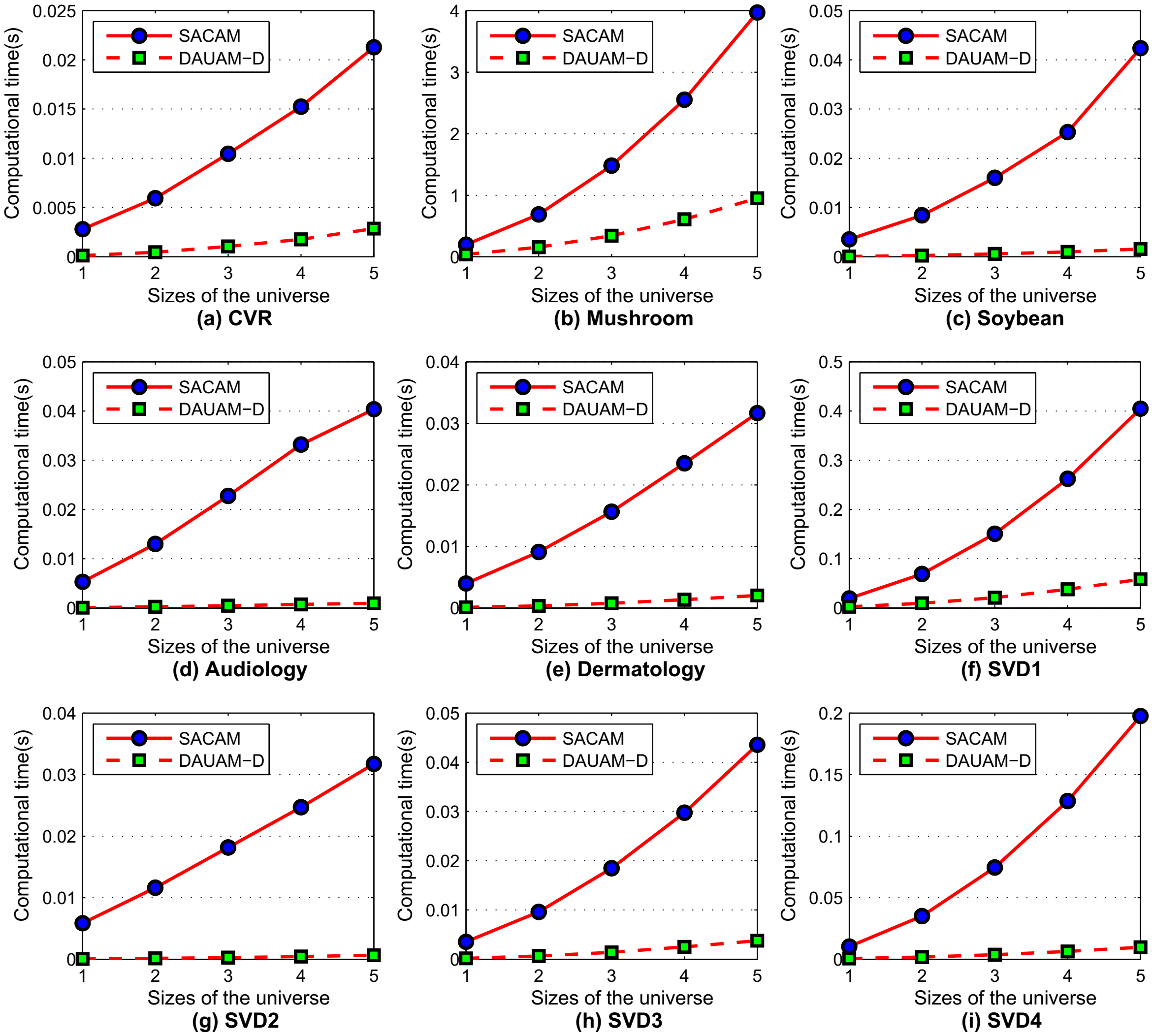

Fig. 2. A comparison of Algorithm SACAM and DAUAM-D versus the size of data when deleting an attribute set.

6.1. SACAM and DAUAM-A

We compare Algorithm SACAM with DAUAM-A on the nine data sets shown in Table 2 when adding an attribute set to the original attribute set. The original and adding attribute sets and the experimental results of these nine data sets are shown in Table 3 and Figure 1, respectively. In Figure 1, we display detailed change trend line of each of two algorithms with the increasing size of the data set. In each of these sub-figures (a)–(i), the x-coordinate pertains to the size of the data set (the five data sets starting from the smallest one), while the y-coordinate concerns the computational time.

From Figure 1, the computational time of each of these two algorithms increases with the increasing size of data. As shown in each sub-figure of Figure 1, Algorithm DAUAM-A consistently is faster than Algorithm SACAM on updating approximations when the same attribute set is added to the original attribute set in the set-valued decision information system. Moreover, the differences become larger and larger when the size of the data set increases.

6.2. SACAM and DAUAM-D

We compare Algorithm SACAM with DAUAM-D on the nine data sets shown in Table 2 when deleting an attribute set. The original and deleting attribute sets and the experimental results of these nine data sets are shown in Table 4 and Figure 2, respectively.

The computational time of each of these two algorithms increases with the increasing size of data from Figure 2. In each sub-figure of Figure 2, Algorithm DAUAM-D is faster than Algorithm SACAM on updating approximations when the same attribute set is deleted from the original attribute set in the set-valued decision information system. Furthermore, the differences become larger and larger when the size of the data set increases.

7. Conclusions

In this paper, we defined a basic vector $H(X)$ and four cut matrices of $H(X)$ which were used to derive the approximations, positive, boundary and negative regions intuitively. They can also be used to derive the (α, β)-probabilistic approximations, positive, boundary and negative regions directly. The key for generating $H(X)$ is to construct the relation matrix. We presented a basic method for constructing the relation matrix. Since the system will vary with time, the attribute set may vary simultaneously. Then, we obtained an incremental approach for updating the relation matrix by investigating the properties about the variation of the relation matrix while attribute set varies. We also proposed the corresponding algorithms for updating approximations. Experimental studies pertaining to different UCI and user-defined data sets showed that the proposed algorithms effectively reduce computational time of updating approximations. Our future work will focus on the time complexity of the proposed approach based on the matrix and try to extend our method to incomplete information systems.

Acknowledgments

This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047 and 61100117), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630127), the Fundamental Research Funds for the Central Universities (No. SWJTU11ZT08) and the Doctoral Innovation Foundation of Southwest Jiaotong University (No. 2012ZJB), the Young Software Innovation Foundation of Sichuan Province (No. 2011-017), China. The authors also thank Xiaoguang Gu for preparing the manuscript.

References

[1] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Kluwer Academic Publishers, Dordrecht, 1991.
[2] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (1) (2007) 3–27.
[3] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (1) (2007) 41–73.
[4] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (1) (2007) 28–40.
[5] N. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, Computational Intelligence 11 (2) (1995) 357–370.
[6] D. Liu, T. Li, D. Ruan, W. Zou, An incremental approach for inducing knowledge from dynamic information systems, Fundamenta Informaticae 94 (2) (2009) 245–260.
[7] D. Liu, T. Li, D. Ruan, J. Zhang, Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, Journal of Global Optimization 51 (2011) 325–344.
[8] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.
[9] Y. Qian, J. Liang, W. Pedrycz, C. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artificial Intelligence 174 (9-10) (2010) 597–618.
[10] T. Yang, Q. Li, Reduction about approximation spaces of covering generalized rough sets, International Journal of Approximate Reasoning 51 (3) (2010) 335–345.
[11] Y. Shi, L. Yao, J. Xu, A probability maximization model based on rough approximation and its application to the inventory problem, International Journal of Approximate Reasoning 52 (2) (2011) 261–280.
[12] J. Qian, D. Miao, Z. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, International Journal of Approximate Reasoning 52 (2) (2011) 212–230.
[13] W. Zhang, J. Ma, S. Fan, Variable threshold concept lattices, Information Sciences 177 (22) (2007) 4883–4892.
[14] Y. Guan, H. Wang, Set-valued information systems, Information Sciences 176 (17) (2006) 2507–2525.
[15] Y. Qian, C. Dang, J. Liang, D. Tang, Set-valued ordered information systems, Information Sciences 179 (16) (2009) 2809–2832.
[16] M. Kryszkiewicz, Rough set approach to incomplete information systems, Information Sciences 112 (1–4) (1998) 39–49.
[17] M. Kryszkiewicz, Rules in incomplete information systems, Information Sciences 113 (3–4) (1999) 271–292.
[18] R. Michalski, Knowledge repair mechanisms: evolution vs. revolution, in: Proceedings of ICML'85, 1985, pp. 116–119.
[19] C. Chan, A rough set approach to attribute generalization in data mining, Information Sciences 107 (1–4) (1998) 169–176.
[20] T. 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.
[21] Y. Cheng, The incremental method for fast computing the rough fuzzy approximations, Data & Knowledge Engineering 70 (1) (2011) 84–100.
[22] H. Chen, T. Li, S. Qiao, D. Ruan, A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values, International Journal of Intelligent Systems 25 (2010) 1005–1026.
[23] G. Liu, The axiomatization of the rough set upper approximation operations, Fundamenta Informaticae 69 (3) (2006) 331–342.

[24] Y. Leung, W. Wu, W. Zhang, Knowledge acquisition in incomplete information systems: a rough set approach, European Journal of Operational Research 168 (1) (2006) 164–180.
[25] E. Orlowska, Z. Pawlak, Representation of nondeterministic information, Theoretical Computer Science 29 (1–2) (1984) 27–39.
[26] I. Duntsch, G. Gediga, E. Orlowska, Relational attribute systems, International Journal of Human-Computer Studies 55 (3) (2001) 293–309.
[27] W. Lipski, On databases with incomplete information, Journal of the ACM 26 (1981) 41–70.
[28] E. Orlowska, Semantic analysis of inductive reasoning, Theoretical Computer Science 43 (1986) 81–89.
[29] W. Zhang, J. Mi, Incomplete information system and its optimal selections, Computers & Mathematics with Applications 48 (5–6) (2004) 691–698.
[30] Y.Y. Yao, Probabilistic rough set approximations, International Journal of Approximate Reasoning 49 (2) (2008) 255–271.
[31] Y.Y. Yao, S.K.M. Wong, A decision theoretic framework for approximating concepts, International Journal of Man-Machine Studies 37 (6) (1992) 793–809.
[32] G. Liu, Axiomatic systems for rough sets and fuzzy rough sets, International Journal of Approximate Reasoning 48 (3) (2008) 857–867.
[33] B. Kolman, R.C. Busby, S.C. Ross, Discrete Mathematical Structures, fifth ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
[34] D. Newman, S. Hettich, C. Blake, C. Merz, UCI Repository of Machine Learning Databases, University of California, Department of Information and Computer Science, Irvine, CA, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.