



# Composite rough sets for dynamic data mining<sup>☆</sup>



Junbo Zhang<sup>a,b</sup>, Tianrui Li<sup>a,\*</sup>, Hongmei Chen<sup>a</sup>

<sup>a</sup> School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

<sup>b</sup> Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

## ARTICLE INFO

### Article history:

Received 4 October 2012

Received in revised form 5 May 2013

Accepted 2 August 2013

Available online 21 August 2013

### Keywords:

Rough sets

Data mining

Incremental learning

Matrix

Composite information systems

## ABSTRACT

As a soft computing tool, rough set theory has become a popular mathematical framework for pattern recognition, data mining and knowledge discovery. It can only deal with attributes of a specific type in the information system by using a specific binary relation. However, there may be attributes of multiple different types in information systems in real-life applications. Such information systems are called as composite information systems in this paper. A composite relation is proposed to process attributes of multiple different types simultaneously in composite information systems. Then, an extended rough set model, called as composite rough sets, is presented. We also redefine lower and upper approximations, positive, boundary and negative regions in composite rough sets. Through introducing the concepts of the relation matrix, the decision matrix and the basic matrix, we propose matrix-based methods for computing the approximations, positive, boundary and negative regions in composite information systems, which is crucial for feature selection and knowledge discovery. Moreover, combined with the incremental learning technique, a novel matrix-based method for fast updating approximations is proposed in dynamic composite information systems. Extensive experiments on different data sets from UCI and user-defined data sets show that the proposed incremental method can process large data sets efficiently.

© 2013 Elsevier Inc. All rights reserved.

## 1. Introduction

Rough set theory, proposed by Pawlak [23–26], is a powerful mathematical tool for analyzing various types of data. It can be used in an attribute value representation model to describe the dependencies among attributes, evaluate the significance of attributes and derive decision rules [11,18,27,31].

Since the classical rough set model can only be used to deal with categorical attributes, many extended rough set models have been developed for attributes of multiple different types, such as numerical ones, set-valued ones, interval-valued ones and missing ones [6,8,9,11,13,15,32,33]. For example, Hu et al. generalized classical rough set model with a neighborhood relation to deal with numerical attributes [9,11]. Guan et al. defined a tolerance relation and used the maximal tolerance classes to derive optimal decision rules from set-valued information systems [8]. Qian et al. used a binary dominance relation to process set-valued data in set-valued ordered information systems [28]. Leung et al. defined  $\alpha$ -tolerance relations and employed the  $\alpha$ -misclassification rate for rule acquisition from interval-valued information systems [15]. In incomplete information systems, the toleration and similarity relations as well as the limited tolerance relation were proposed respectively to

<sup>☆</sup> This is an extended version of the paper presented at the 2012 International Conference on Artificial Intelligence and Computational Intelligence (AICI'12), Chengdu, 2012, China.

\* Corresponding author. Tel.: +86 28 86466426.

E-mail addresses: [JunboZhang86@163.com](mailto:JunboZhang86@163.com), [jbzhang@cs.gsu.edu](mailto:jbzhang@cs.gsu.edu) (J. Zhang), [trli@swjtu.edu.cn](mailto:trli@swjtu.edu.cn) (T. Li), [hmchen@swjtu.edu.cn](mailto:hmchen@swjtu.edu.cn) (H. Chen).

deal with missing data in [13,32,34]. Grzymała-Busse combined the toleration and similarity relations and presented characteristic relations for missing data in incomplete information systems [6].

In real-life applications, there are attributes of multiple different types in information systems, e.g., categorical ones, numerical ones, set-valued ones, interval-valued ones and missing ones. Such information systems are called as composite information systems. Most of the rough set based methods fail to deal with more than attributes of two different types. To solve this problem, Abu-Donia proposed multi knowledge based rough approximations using a family of finite number of relations [1]. In our previous work, we introduced the composite rough set model and proposed the basic idea to deal with attributes of multiple different types [38]. Here, we continue with this work and improve the composite rough set model. In addition, a matrix-based method is introduced into our work. It helps compute the approximations, positive, boundary and negative regions intuitively from the composite information system and composite decision table. To adapt to the dynamic changes of the composite information system, we employ an incremental technique and propose a matrix-based incremental method for fast updating the approximations from dynamic composite information systems. Extensive experiments on different data sets from UCI and user-defined data sets show that the proposed matrix-based incremental method can process large data sets efficiently.

The remainder of this paper is organized as follows: Section 2 introduces basic concepts of rough sets and its extended models. Section 3 proposes the composite rough set model to deal with attributes of multiple different types. Section 4 gives matrix-based methods for computing the approximations, positive, boundary and negative regions in composite information systems. Section 5 presents matrix-based incremental methods for updating the approximations in dynamic composite information systems. Section 6 designs and develops the static and incremental algorithm based on matrix for computing and updating the approximations in composite information systems. In the Section 7, the performances of static and incremental methods are evaluated on UCI and user-defined data. Section 8 discusses about some related work on rough sets using matrix-based and incremental techniques. The paper ends with conclusions and further research work in Section 9.

## 2. Rough set models

In this section, we first briefly review the concepts of rough set model as well as its extensions [7–9,11,13,23,32].

### 2.1. Classical rough set model

Given a pair  $K = (U, R)$ , where  $U$  is a finite and non-empty set called the universe, and  $R \subseteq U \times U$  is an indiscernibility relation on  $U$ . The pair  $K = (U, R)$  is called as an approximation space.  $K = (U, R)$  is characterized by an information system  $IS = (U, A, V, f)$ , 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  $a$ ;  $f: U \times A \rightarrow V$  is an information function such that  $f(x, a) \in V_a$  for every  $x \in U, a \in A$ . Let  $B \subseteq A$ , in the classical rough set model, a binary indiscernibility relation  $R_B$  is defined as follows:

$$R_B = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \quad \forall a \in B\} \quad (1)$$

$R_B$  is an equivalence relation, and  $[x]_{R_B}$  denotes an equivalence class of an element  $x \in U$  under  $R_B$ , where  $[x]_{R_B} = \{y \in U \mid x R_B y\}$ .

Classical rough set model is based on the equivalence relation. The elements in an equivalence class satisfy reflexive, symmetric and transitive. It does not allow the non-categorical data (e.g., numerical data, set-valued data, and interval-valued data) and requires the information table should be complete. However, non-categorical data appears frequently in real-life applications [6,10,13,28]. Therefore, it is necessary to investigate the situation of non-categorical data in information systems. In what follows, we introduce several representative rough set models [8,11], which will be used in our examples. More rough set models for dealing with non-categorical data are available in the literatures [7,10,13,15,21,28,37].

### 2.2. Neighborhood rough set model

To deal with numerical data in neighborhood information systems, Hu et al. first employed a neighborhood relation and proposed neighborhood rough sets [9,11].

**Definition 1** ([9,11]). Let  $B \subseteq C$  be a subset of attributes,  $x \in U$ . The neighborhood  $\delta_B(x)$  of  $x$  in  $B$  is defined as

$$\delta_B(x) = \{y \in U \mid \Delta_B(x, y) \leq \delta\} \quad (2)$$

where  $\Delta$  is a distance function.  $\forall x, y, z \in U$ , it satisfies:

- (I)  $\Delta(x, y) \geq 0, \Delta(x, y) = 0$  if and only if  $x = y$ ;
- (II)  $\Delta(x, y) = \Delta(y, x)$ ;
- (III)  $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$ .

There are three metric functions widely used in pattern recognition. Considered that  $x$  and  $y$  are two objects in an  $m$ -dimensional space  $A = \{a_1, a_2, \dots, a_m\}$ .  $f(x, a_j)$  denotes the value of sample  $x$  in the  $j$ th attribute  $a_j$ . Then a general metric, named Minkowsky distance, is defined as

$$\Delta_P(x, y) = \left( \sum_{j=1}^m |f(x, a_j) - f(y, a_j)|^P \right)^{1/P} \quad (3)$$

where (3) is called: (a) Manhattan distance  $\Delta_1$  if  $P = 1$ ; (b) Euclidean distance  $\Delta_2$  if  $P = 2$ ; (c) Chebychev distance if  $P = \infty$  [35].

The neighborhood relation is reflexive and symmetric.

### 2.3. Set-valued rough set model

To deal with set-valued data in set-valued information systems, Guan et al. employed a tolerance relation [8].

**Definition 2 8.** In the set-valued information system  $(U, A, V, f)$ , for  $b \in A$ , the tolerance relation  $T_b$  is defined as:

$$T_b = \{(x, y) | f(x, b) \cap f(y, b) \neq \emptyset\} \quad (4)$$

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

$$T_B = \{(x, y) | \forall b \in B, f(x, b) \cap f(y, b) \neq \emptyset\} = \bigcap_{b \in B} T_b \quad (5)$$

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

### 2.4. Characteristic relation-based rough set model

To deal with missing data in incomplete information systems, Grzymala integrated the tolerance relation [13,14] and the similarity relation [32] and presented a characteristic relation [7].

**Definition 3 [7].** In the incomplete information system  $(U, A, V, f)$ , for  $B \subseteq A$ , the characteristic relation  $K_B$  is defined as follows:

$$K_B = \{(x, y) | \forall b \in B, f(x, b) \neq ? \wedge (f(x, b) = f(y, b) \vee f(x, b) = * \vee f(y, b) = *)\} \quad (6)$$

where “?” and “\*” are missing values which mean the lost value and “do not care”, respectively.

The characteristic relation is reflexive, but not symmetric and transitive. It is a generalization of tolerance relation and similarity relation.

## 3. Composite rough set model

In many practical issues, there are attributes of multiple different types in the information system, we call it a composite information system in this paper. A composite information system can be written as  $CIS = (U, A, V, f)$ , where

$$\left\{ \begin{array}{ll} U, & \text{a non-empty finite set of objects} \\ A = \bigcup A_k, & \text{a union of attribute sets} \\ & \text{where } A_k \text{ is an attribute set with the same data type} \\ V = \bigcup_{A_k \subseteq A} V_{A_k}, & V_{A_k} = \bigcup_{a \in A_k} V_a, V_a \text{ is a domain of attribute } a \\ f : U \times A \rightarrow V, & \text{namely, } U \times \bigcup A_k \rightarrow \bigcup V_{A_k} \\ & \text{where } U \times A_k \rightarrow V_{A_k} \text{ is an information function} \\ & f(x, a) \text{ denotes the value of object } x \text{ on attribute } a \end{array} \right.$$

More specifically, a composite information system is also called a composite decision table if there are condition and decision attributes in the information system, which is denoted by  $CDT = (U, A \cup D, V, f)$ .

**Definition 4.** Given  $x, y \in U$  and  $B = \bigcup B_k \subseteq A$ ,  $B_k \subseteq A_k$ , the composite relation  $CR_B$  is defined as

$$CR_B = \left\{ (x, y) | (x, y) \in \bigcap_{B_k \subseteq B} R_{B_k} \right\} \quad (7)$$

where  $R_{B_k} \subseteq U \times U$  is an indiscernibility relation defined by an attribute set  $B_k$  on  $U$  [25].

When  $(x, y) \in CR_B$ , we call  $x$  and  $y$  are indiscernible w.r.t.  $B$ . Let  $CR_B(x) = \{y | y \in U, \forall B_k \subseteq B, yR_{B_k}x\}$ , we call  $CR_B(x)$  the composite class for  $x$  w.r.t.  $CR_B$ .

**Definition 5.** Given a composite information system  $CIS = (U, A, V, f)$ ,  $\forall X \subseteq U, B \subseteq A$ , the lower and upper approximations of  $X$  in terms of the composite relation  $CR_B$  are defined as

$$\underline{CR}_B(X) = \{x \in U | CR_B(x) \subseteq X\} \quad (8)$$

$$\overline{CR}_B(X) = \{x \in U | CR_B(x) \cap X \neq \emptyset\} \quad (9)$$

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

$$\begin{cases} POS_{CR_B}(X) = \underline{CR}_B(X) \\ BND_{CR_B}(X) = \overline{CR}_B(X) - \underline{CR}_B(X) \\ NEG_{CR_B}(X) = U - \overline{CR}_B(X) \end{cases} \quad (10)$$

**Definition 6.** Given a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ . Let  $U/D = \{D_1, D_2, \dots, D_r\}$  be a partition over the decision  $D$ . Then the lower and upper approximations of the decision  $D$  with respect to attributes  $B$  are defined as

$$\underline{CR}_B(D) = \bigcup_{j=1}^r \underline{CR}_B(D_j) \quad (11)$$

$$\overline{CR}_B(D) = \bigcup_{j=1}^r \overline{CR}_B(D_j) \quad (12)$$

where

$$\underline{CR}_B(D_j) = \{x \in U | CR_B(x) \subseteq D_j\} \quad (13)$$

$$\overline{CR}_B(D_j) = \{x \in U | CR_B(x) \cap D_j \neq \emptyset\} \quad (14)$$

The positive region  $POS_{CR_B}(D)$ , the boundary region  $BND_{CR_B}(D)$  and the negative region  $NEG_{CR_B}(D)$  of  $D$  with respect to the attribute subset  $B$  are defined as

$$\begin{cases} POS_{CR_B}(D) = \underline{CR}_B(D) \\ BND_{CR_B}(D) = \overline{CR}_B(D) - \underline{CR}_B(D) \\ NEG_{CR_B}(D) = U - \overline{CR}_B(D) \end{cases} \quad (15)$$

**Example 1.** A composite decision table  $CDT = (U, A \cup D, V, f)$  is presented in Table 1. Let  $B = \bigcup_{k=1,2,3,4} B_k$ , where  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2\}$ ,  $B_3 = \{a_3, a_4\}$ ,  $B_4 = \{a_5\}$ . We set the neighborhood parameter  $\delta = 0.15$ .

**Table 1**

A composite decision table.

| $U$   | $a_1$ | $a_2$         | $a_3$ | $a_4$ | $a_5$ | $D$ |
|-------|-------|---------------|-------|-------|-------|-----|
| $x_1$ | $y$   | $\{1, 2\}$    | 0.2   | 0.1   | *     | Yes |
| $x_2$ | $y$   | $\{1\}$       | 0.2   | 0.3   | ?     | No  |
| $x_3$ | $y$   | $\{0\}$       | 0.1   | 0.1   | Small | Yes |
| $x_4$ | $y$   | $\{0, 1, 2\}$ | 0.1   | 0.2   | Small | Yes |
| $x_5$ | $n$   | $\{1\}$       | 0.1   | 0.3   | Large | Yes |
| $x_6$ | $n$   | $\{0, 2\}$    | 0.2   | 0.2   | Large | No  |

**Table 2**

Results of composite classes.

| $x_i$ | $R_{B_1}(x_i) = [x_i]_{R_{B_1}}$ | $R_{B_2}(x_i) = T_{B_2}(x_i)$      | $R_{B_3}(x_i) = \delta_{B_3}(x_i)$ | $R_{B_4}(x_i) = K_{B_4}(x_i)$ | $CR_B(x_i)$         |
|-------|----------------------------------|------------------------------------|------------------------------------|-------------------------------|---------------------|
| $x_1$ | $\{x_1, x_2, x_3, x_4\}$         | $\{x_1, x_2, x_4, x_5, x_6\}$      | $\{x_1, x_3, x_4, x_6\}$           | $U$                           | $\{x_1, x_4\}$      |
| $x_2$ | $\{x_1, x_2, x_3, x_4\}$         | $\{x_1, x_2, x_4, x_5\}$           | $\{x_2, x_4, x_5, x_6\}$           | $U$                           | $\{x_2, x_4\}$      |
| $x_3$ | $\{x_1, x_2, x_3, x_4\}$         | $\{x_3, x_4, x_6\}$                | $\{x_1, x_3, x_4, x_6\}$           | $\{x_1, x_3, x_4\}$           | $\{x_3, x_4\}$      |
| $x_4$ | $\{x_1, x_2, x_3, x_4\}$         | $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ | $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ | $\{x_1, x_3, x_4\}$           | $\{x_1, x_3, x_4\}$ |
| $x_5$ | $\{x_5, x_6\}$                   | $\{x_1, x_2, x_4, x_5\}$           | $\{x_2, x_4, x_5, x_6\}$           | $\{x_1, x_5, x_6\}$           | $\{x_5\}$           |
| $x_6$ | $\{x_5, x_6\}$                   | $\{x_1, x_3, x_4, x_6\}$           | $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ | $\{x_1, x_5, x_6\}$           | $\{x_6\}$           |

According to the introduction in Section 2, it is easy to know that  $R_{B_1}, R_{B_2}, R_{B_3}, R_{B_4}$  are equivalence relation, neighborhood relation, tolerance relation and characteristic relation by the attribute sets  $B_1, B_2, B_3$  and  $B_4$ , respectively. By Definition 4,  $\forall x_i \in U, CR_B(x_i) = \bigcap_{k=1}^4 R_{B_k}(x_i)$ . The results are listed in Table 2.

It is easy to obtain that  $U/D = \{D_1, D_2\}$ , where  $D_1 = \{x_1, x_3, x_4, x_5\}$  and  $D_2 = \{x_2, x_6\}$ .

Since  $CR_B(x_1) \subseteq D_1, CR_B(x_2) \not\subseteq D_1, CR_B(x_3) \subseteq D_1, CR_B(x_4) \subseteq D_1, CR_B(x_5) \subseteq D_1$  and  $CR_B(x_6) \not\subseteq D_1$ , then  $\underline{CR}_B(D_1) = \{x_1, x_3, x_4, x_5\}$ . Since  $\forall i \in \{1, 2, 3, 4, 5\}, CR_B(x_i) \cap D_1 \neq \emptyset$  and  $CR_B(x_6) \cap D_1 = \emptyset$ , then  $\overline{CR}_B(D_1) = \{x_1, x_2, x_3, x_4, x_5\}$ .

Similarly, we can compute the approximations of  $D_2, \underline{CR}_B(D_2) = \{x_6\}$  and  $\overline{CR}_B(D_2) = \{x_2, x_6\}$ .

Therefore,  $POS_{CR_B}(D) = \underline{CR}_B(D_1) \cup \underline{CR}_B(D_2) = \{x_1, x_3, x_4, x_5, x_6\}, BND_{CR_B}(D) = \{x_2\}$  and  $NEG_{CR_B}(D) = \emptyset$ .

#### 4. Matrix-based method in composite information systems

In this section, we present the matrix representation of the lower and upper approximations in the composite information system.

Firstly, we review the matrix-based methods in rough sets. In 2006, a set of axioms were constructed to characterize classical rough set upper approximation from the matrix point of view by Liu [19]. Then, in our previous work, we defined a basic vector  $H(X)$ , which was induced from the relation matrix. And four cut matrices of  $H(X)$ , denoted by  $H^{[l, \mu, \nu]}(X), H^{[u, \mu, \nu]}(X), H^{[l, \mu, \nu]}(X)$  and  $H^{[u, \mu, \nu]}(X)$ , were derived for the computation of the approximations, positive, boundary and negative regions intuitively in set-valued information systems [40]. In this paper, we use matrix-based methods to deal with the composite data. The difference is to construct the relation matrix with a composite relation.

**Definition 7** 40. Given two  $\mu \times \nu$  matrices  $Y = (y_{ij})_{\mu \times \nu}$  and  $Z = (z_{ij})_{\mu \times \nu}$ , Minimum of two matrices is defined as

$$\min(Y, Z) = (\min(y_{ij}, z_{ij}))_{\mu \times \nu} \quad (16)$$

**Definition 8** 40. Let  $U = \{x_1, x_2, \dots, x_n\}$ , and  $X$  be a subset of  $U$ . The characteristic function  $G(X) = (g_1, g_2, \dots, g_n)^T$  ( $T$  denotes the transpose operation) is defined as

$$g_i = \begin{cases} 1, & x_i \in X \\ 0, & x_i \notin X \end{cases} \quad (17)$$

where  $G(X)$  assigns 1 to an element that belongs to  $X$  and 0 to an element that does not belong to  $X$ . For example, if  $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$  and  $X = \{x_1, x_3, x_4, x_5\}$ , then  $G(X) = (1, 0, 1, 1, 1, 0)^T$ .

**Definition 9** 20. Given an information system  $IS = (U, A, V, f)$ . Let  $B \subseteq A$  and  $R_B$  be an indiscernibility relation on  $U, M_{n \times n}^{R_B} = (\zeta_{ij})_{n \times n}$  be an  $n \times n$  matrix representing  $R_B$ , called the relation matrix w.r.t.  $B$ . Then

$$\zeta_{ij} = \begin{cases} 1, & (x_i, x_j) \in R_B \\ 0, & (x_i, x_j) \notin R_B \end{cases} \quad (18)$$

**Lemma 1.** Given a composite information system  $CIS = (U, A, V, f)$ , where  $A = \bigcup A_k$ . Let  $B = \bigcup B_k \subseteq A, B_k \subseteq A_k$  and  $CR_B$  be a composite relation on  $U$  by the attribute set  $B, M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$  be an  $n \times n$  matrix representing  $CR_B$ , called the relation matrix w.r.t.  $B$ . Then

$$M_{n \times n}^{CR_B} = \min_{B_k \subseteq B} M_{n \times n}^{R_{B_k}} \quad (19)$$

**Corollary 1.** Let  $M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$  and  $CR_B$  be a composite relation on  $U$ . Then  $m_{ii} = 1, 1 \leq i \leq n$ .

**Definition 10** 40. Let  $B \subseteq A$  and  $CR_B$  be a composite relation on  $U, A_{n \times n}^{CR_B}$  be an induced diagonal matrix of  $M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$ . Then

$$A_{n \times n}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_n}\right) \quad (20)$$

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

**Corollary 2.**  $A_{n \times n}^{CR_B} = \text{diag}\left(\frac{1}{|CR_B(x_1)|}, \frac{1}{|CR_B(x_2)|}, \dots, \frac{1}{|CR_B(x_n)|}\right)$  and  $1 \leq |CR_B(x_i)| = |\bigcap_{B_k \subseteq B} R_{B_k}(x_i)| \leq n, 1 \leq i \leq n$ .

**Definition 11** [40]. The  $n$ -column vector called a basic vector, denoted by  $H(X)$ , is defined as:

$$H(X) = A_{n \times n}^{CR_B} \bullet \left( M_{n \times n}^{CR_B} \bullet G(X) \right) = A_{n \times n}^{CR_B} \bullet \Omega_{n \times 1}^{CR_B} \quad (21)$$

where  $\bullet$  is the dot product of matrices and  $\Omega_{n \times 1}^{CR_B} = M_{n \times n}^{CR_B} \bullet G(X)$ , called an intermediate vector.

**Definition 12** [40]. 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.

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

$$h'_i = \begin{cases} 1, & \mu \leq h_i \leq \nu \\ 0, & \text{else} \end{cases} \quad (22)$$

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

$$h'_i = \begin{cases} 1, & \mu < h_i \leq \nu \\ 0, & \text{else} \end{cases} \quad (23)$$

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

$$h'_i = \begin{cases} 1, & \mu \leq h_i < \nu \\ 0, & \text{else} \end{cases} \quad (24)$$

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

$$h'_i = \begin{cases} 1, & \mu < h_i < \nu \\ 0, & \text{else} \end{cases} \quad (25)$$

**Remark 1.** These four cut matrices are Boolean matrices.

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

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

$$G(\underline{CR}_B(X)) = H^{[1, 1]}(X) \quad (26)$$

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

$$G(\overline{CR}_B(X)) = H^{(0, 1]}(X) \quad (27)$$

**Corollary 3.** The positive region  $POS_{CR_B}(X)$ , the boundary region  $BND_{CR_B}(X)$ , and the negative region  $NEG_{CR_B}(X)$  can also be generated from the cut matrix of  $H(X)$ , respectively as follows.

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

$$G(POS_{CR_B}(X)) = H^{[1, 1]}(X) \quad (28)$$

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

$$G(BND_{CR_B}(X)) = H^{(0, 1)}(X) \quad (29)$$

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

$$G(NEG_{CR_B}(X)) = H^{[0, 0]}(X) \quad (30)$$

#### 4.1. Matrix-based method in composite decision table

**Definition 13.** Given a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ , let  $U/D = \{D_1, D_2, \dots, D_r\}$  be a partition over the decision  $D$ .  $\forall D_j \in U/D$ ,  $G(D_j)$  is an  $n$ -column boolean vector of  $D_j$ . Let  $GD_{n \times r} = (G(D_1), G(D_2), \dots, G(D_r))$  be an  $n \times r$  boolean matrix, called a decision matrix.

**Corollary 4.** Given a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ , let  $U/D = \{D_1, D_2, \dots, D_r\}$  be a partition over the decision  $D$ .  $\forall D_j \in U/D$ ,  $H(D_j)$  is the basic vector. Let  $HD = (H(D_1), H(D_2), \dots, H(D_r))$  be an  $n \times r$  matrix, called a basic matrix. Then,  $HD$  can be computed as follows.

$$HD = A_{n \times n}^{CR_B} \bullet (M_{n \times n}^{CR_B} \bullet GD_{n \times r}) = A_{n \times n}^{CR_B} \bullet \Omega_{n \times r}^{CR_B} \quad (31)$$

where  $\Omega_{n \times r}^{CR_B} = M_{n \times n}^{CR_B} \bullet GD_{n \times r}$ , called an intermediate matrix.

**Corollary 5.** Given a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ . Let  $U/D = \{D_1, D_2, \dots, D_r\}$  be a partition over the decision  $D$ , and  $HD$  be a basic matrix.  $\forall j = 1, 2, \dots, r$ , the upper and lower approximations of  $D_j$  in the composite information system can be generated from the cut matrix of  $HD$  as follows.

(1) The  $n$ -column boolean vector  $G(\underline{CR}_B(D_j))$  of the lower approximation  $\underline{CR}_B(D_j)$ :

$$G(\underline{CR}_B(D_j)) = H^{[1,1]}(D_j) \quad (32)$$

(2) The  $n$ -column boolean vector  $G(\overline{CR}_B(D_j))$  of the upper approximation  $\overline{CR}_B(D_j)$ :

$$G(\overline{CR}_B(D_j)) = H^{(0,1)}(D_j) \quad (33)$$

The positive region  $POS_{CR_B}(D)$ , the boundary region  $BND_{CR_B}(D)$ , and the negative region  $NEG_{CR_B}(D)$  can also be generated from the cut matrix of  $HD$ , respectively as follows.

(3) The  $n$ -column boolean vector  $G(POS_{CR_B}(D))$  of the positive region:

$$G(POS_{CR_B}(D)) = \sum_{j=1}^r H^{[1,1]}(D_j) \quad (34)$$

(4) The  $n$ -column boolean vector  $G(BND_{CR_B}(D))$  of the boundary region:

$$G(BND_{CR_B}(D)) = \sum_{j=1}^r H^{(0,1)}(D_j) - \sum_{j=1}^m H^{[1,1]}(D_j) = \sum_{j=1}^m H^{(0,1)}(D_j) \quad (35)$$

(5) The  $n$ -column boolean vector  $G(NEG_{CR_B}(D))$  of the negative region:

$$G(NEG_{CR_B}(D)) = J - \sum_{j=1}^r H^{(0,1)}(D_j) = \left( \sum_{j=1}^r H^{(0,1)}(D_j) \right)^{[0,0]} \quad (36)$$

where  $J = (1, 1, \dots, 1)^T$ .

**Example 2.** A composite decision table  $CDT = (U, A \cup D, V, f)$  is presented in Table 1. Let  $B = \bigcup_{k=1,2,3,4} B_k$ , where  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2\}$ ,  $B_3 = \{a_3, a_4\}$ ,  $B_4 = \{a_5\}$ . We set the neighborhood parameter  $\delta = 0.15$ .

#### • The construction of the relation matrix

According to Definition 9, we have

$$M_{6 \times 6}^{R_{B_1}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad M_{6 \times 6}^{R_{B_2}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$M_{6 \times 6}^{R_{B_3}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \text{ and } M_{6 \times 6}^{R_{B_4}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

By Lemma 1,

$$M_{6 \times 6}^{CR_B} = \min_{k=1,2,3,4} M_{6 \times 6}^{R_{B_k}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

• **The calculation of the induced diagonal matrix**

According to Definition 10, we have  $\Lambda_{6 \times 6}^{CR_B} = \text{diag}(1/2, 1/2, 1/2, 1/3, 1, 1)$ .

• **The construction of the decision matrix**

Since  $U/D = \{D_1, D_2\}$ , where  $D_1 = \{x_1, x_3, x_4, x_5\}$  and  $D_2 = \{x_2, x_6\}$ , then we have  $G(D_1) = (1, 0, 1, 1, 1, 0)^T$  and  $G(D_2) = (0, 1, 0, 0, 0, 1)^T$ .

• **The calculation of the intermediate matrix**

According to Corollary 4, we have

$$\Omega_{6 \times 2}^{CR_B} = M_{6 \times 6}^{CR_B} \bullet GD_{6 \times 2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

• **The calculation of the basic matrix**

Since  $\Lambda_{6 \times 6}^{CR_B} = \text{diag}(1/2, 1/2, 1/2, 1/3, 1, 1)$ , then

$$HD = \Lambda_{6 \times 6}^{CR_B} \bullet \Omega_{6 \times 2}^{CR_B} = \text{diag}(1/2, 1/2, 1/2, 1/3, 1, 1) \bullet \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

• **The calculation of approximations, positive region, boundary region and negative region from HD**

According to Corollary 5, we have

| HD   | $G(\underline{CR}_B(D_1))$                                 | $G(\underline{CR}_B(D_2))$                                 | $G(\overline{CR}_B(D_1))$                                  | $G(\overline{CR}_B(D_2))$                                  | $G(POS_{CR_B}(D))$   | $G(BND_{CR_B}(D))$   | $G(NEG_{CR_B}(D))$   |
|--|--|--|--|--|--|--|--|
| $\begin{bmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ |

In other words,  $\underline{CR}_B(D_1) = \{x_1, x_3, x_4, x_5\}$ ,  $\underline{CR}_B(D_2) = \{x_6\}$ ,  $\overline{CR}_B(D_1) = \{x_1, x_2, x_3, x_4, x_5\}$ ,  $\overline{CR}_B(D_2) = \{x_2, x_6\}$ ,  $POS_{CR_B}(D) = \{x_1, x_3, x_4, x_5, x_6\}$ ,  $BND_{CR_B}(D) = \{x_2\}$  and  $NEG_{CR_B}(D) = \emptyset$ .

## 5. A matrix-based incremental method for dynamic composite information systems

Usually, objects in the dynamic information systems evolve over time. To adapt to this situation, we give a matrix-based incremental method in dynamic composite information systems. The variation of objects, contains two situations, adding objects and deleting objects, respectively. Since the lower approximation, upper approximation, positive, boundary and



negative regions can be induced from the basic vector  $HD$  directly in the composite decision table. In addition, the key step for computing basic vector  $HD$  is to construct the relation matrix, the induced diagonal matrix, the decision matrix, and the intermediate matrix. Thus, it reduces the running time if we can renew these matrices with an incremental updating strategy rather than reconstructing them from scratch. We discuss how to update these matrices incrementally under the variation of objects as follows.

### 5.1. Updating the relation matrix and decision matrix when adding objects

For a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ , suppose there are  $n^+$  objects and  $r^+$  decisions after adding objects.

**Corollary 6.** Let  $M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$  and  $M_{n^+ \times n^+}^{CR_B} = (m_{ij})_{n^+ \times n^+}$  be relation matrices before and after adding objects, respectively. Then

$$M_{n^+ \times n^+}^{CR_B} = \left[ \begin{array}{c|c} M_{n \times n}^{CR_B} & P \\ \hline Q & R \end{array} \right] \quad (37)$$

where

$$P = \begin{bmatrix} m_{1,n+1} & m_{1,n+2} & \cdots & m_{1,n^+} \\ m_{2,n+1} & m_{2,n+2} & \cdots & m_{2,n^+} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,n+1} & m_{n,n+2} & \cdots & m_{n,n^+} \end{bmatrix}, Q = \begin{bmatrix} m_{n+1,1} & m_{n+1,2} & \cdots & m_{n+1,n} \\ m_{n+2,1} & m_{n+2,2} & \cdots & m_{n+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n^+,1} & m_{n^+,2} & \cdots & m_{n^+,n} \end{bmatrix}$$

and

$$R = \begin{bmatrix} m_{n+1,n+1} & m_{n+1,n+2} & \cdots & m_{n+1,n^+} \\ m_{n+2,n+1} & m_{n+2,n+2} & \cdots & m_{n+2,n^+} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n^+,n+1} & m_{n^+,n+2} & \cdots & m_{n^+,n^+} \end{bmatrix}, m_{ij} = \begin{cases} 1, & (x_i, x_j) \in CR_B \\ 0, & (x_i, x_j) \notin CR_B \end{cases}, \quad 1 \leq i, j \leq n^+$$

**Corollary 7.** Let  $\Lambda_{n \times n}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_n}\right)$  and  $\Lambda_{n^+ \times n^+}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1^+}, \frac{1}{\lambda_2^+}, \dots, \frac{1}{\lambda_{n^+}^+}\right)$  be induced diagonal matrices of relation matrices before and after adding objects, respectively. Then

$$\lambda_i^+ = \begin{cases} \lambda_i + \sum_{j=n+1}^{n^+} m_{ij}, & 1 \leq i \leq n \\ \sum_{j=1}^{n^+} m_{ij}, & n+1 \leq i \leq n^+ \end{cases} \quad (38)$$

After adding new objects, it may form new decision classes. We suppose that there are  $r^+$  decisions after adding objects. It is easy to know that  $r^+ \geq r$  and  $r^+ = r$  if it does not form a new decision classes; Otherwise  $r^+ > r$ .

**Corollary 8.** Let  $GD_{n \times r} = (G(D_1), G(D_2), \dots, G(D_r)) = (d_{ij})_{n \times r}$  and  $GD_{n^+ \times r^+} = (d_{ij})_{n^+ \times r^+}$  be decision matrices before and after adding objects, respectively. Then

$$GD_{n^+ \times r^+} = \left[ \begin{array}{c|c} GD_{n \times r} & P' \\ \hline Q' & R' \end{array} \right] \quad (39)$$

where  $P' = [0]_{n \times (r^+ - r)}$  is a zero matrix,  $Q' = \begin{bmatrix} d_{n+1,1} & d_{n+1,2} & \cdots & d_{n+1,r} \\ d_{n+2,1} & d_{n+2,2} & \cdots & d_{n+2,r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n^+,1} & d_{n^+,2} & \cdots & d_{n^+,r} \end{bmatrix}$ , and  $R' = \begin{bmatrix} d_{n+1,r+1} & d_{n+1,r+2} & \cdots & d_{n+1,r^+} \\ d_{n+2,r+1} & d_{n+2,r+2} & \cdots & d_{n+2,r^+} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n^+,r+1} & d_{n^+,r+2} & \cdots & d_{n^+,r^+} \end{bmatrix}$ ,

$$d_{ij} = \begin{cases} 1, & f(x_i, d) = j, \quad n+1 \leq i \leq n^+, 1 \leq j \leq r^+. \\ 0, & \text{else} \end{cases}$$

**Corollary 9.** Let  $\Omega_{n \times r}^{CR_B} = M_{n \times n}^{CR_B} \bullet GD_{n \times r} = (\omega_{ij})_{n \times r}$  and  $\Omega_{n^+ \times r^+}^{CR_B} = M_{n^+ \times n^+}^{CR_B} \bullet GD_{n^+ \times r^+} = (\omega_{ij}^+)_{n^+ \times r^+}$  be intermediate matrices before and after adding objects, respectively. Then

$$\Omega_{n^+ \times r^+}^{CR_B} = \left[ \frac{\Omega_{n \times r}^{CR_B} + P \bullet Q'}{Q \bullet GD_{n \times r} + R \bullet Q'} \middle| \frac{M_{n \times n}^{CR_B} \bullet P' + P \bullet R'}{Q \bullet P' + R \bullet R'} \right] \quad (40)$$

where  $M_{n^+ \times n^+}^{CR_B} = \left[ \frac{M_{n \times n}^{CR_B}}{Q} \middle| \frac{P}{R} \right]$  and  $GD_{n^+ \times r^+} = \left[ \frac{GD_{n \times r}}{Q'} \middle| \frac{P'}{R'} \right]$ .

or, equivalently,

$$\omega_{ij}^+ = \begin{cases} \omega_{ij} + \sum_{k=n+1}^{n^+} m_{ik} d_{kj}, & 1 \leq i \leq n, 1 \leq j \leq r \\ \sum_{k=1}^{n^+} m_{ik} d_{kj}, & \text{else} \end{cases} \quad (41)$$

**Proof.** Let  $M_{n \times n}^{CR_B}$  and  $M_{n^+ \times n^+}^{CR_B}$  be relation matrices before and after adding objects, respectively, and  $GD_{n \times r}$  and  $GD_{n^+ \times r^+}$  be decision matrices before and after adding objects, respectively.

Because of  $M_{n^+ \times n^+}^{CR_B} = \left[ \frac{M_{n \times n}^{CR_B}}{Q} \middle| \frac{P}{R} \right]$  and  $GD_{n^+ \times r^+} = \left[ \frac{GD_{n \times r}}{Q'} \middle| \frac{P'}{R'} \right]$ , then we have

$$\Omega_{n^+ \times r^+}^{CR_B} = M_{n^+ \times n^+}^{CR_B} \bullet GD_{n^+ \times r^+} = \left[ \frac{M_{n \times n}^{CR_B}}{Q} \middle| \frac{P}{R} \right] \bullet \left[ \frac{GD_{n \times r}}{Q'} \middle| \frac{P'}{R'} \right] = \left[ \frac{M_{n \times n}^{CR_B} \bullet GD_{n \times r} + P \bullet Q'}{Q \bullet GD_{n \times r} + R \bullet Q'} \middle| \frac{M_{n \times n}^{CR_B} \bullet P' + P \bullet R'}{Q \bullet P' + R \bullet R'} \right] = \left[ \frac{\Omega_{n \times r}^{CR_B} + P \bullet Q'}{Q \bullet GD_{n \times r} + R \bullet Q'} \middle| \frac{M_{n \times n}^{CR_B} \bullet P' + P \bullet R'}{Q \bullet P' + R \bullet R'} \right]$$

or, equivalently,

$$\omega_{ij}^+ = \sum_{k=1}^{n^+} m_{ik} d_{kj} = \begin{cases} \sum_{k=1}^n m_{ik} d_{kj} + \sum_{k=n+1}^{n^+} m_{ik} d_{kj}, & 1 \leq i \leq n, 1 \leq j \leq r \\ \sum_{k=1}^{n^+} m_{ik} d_{kj}, & \text{else} \end{cases} = \begin{cases} \omega_{ij} + \sum_{k=n+1}^{n^+} m_{ik} d_{kj}, & 1 \leq i \leq n, 1 \leq j \leq r \\ \sum_{k=1}^{n^+} m_{ik} d_{kj}, & \text{else} \end{cases} \quad \square$$

**Example 3** (Continuation of Examples 1 and 2). The composite decision table is shown Table 1. Let  $U_1 = \{x_1, x_2, x_3, x_4\}$  and  $U_2 = \{x_5, x_6\}$  be the original object set and the added object set, respectively.

#### • The update of the relation matrix

According to Example 2, we know that the relation matrix of  $U_1$  is  $M_{4 \times 4}^{CR_B} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$ .

After adding  $U_2$ , we have  $P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ ,  $Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ , and  $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

According to Corollary 6, the updated relation matrix is  $M_{6 \times 6}^{CR_B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ .

#### • The update of the induced diagonal matrix

The induced diagonal matrix of  $U_1$  is  $\lambda_{4 \times 4}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_4}\right) = \text{diag}(1/2, 1/2, 1/3, 1/2)$ .

After adding  $U_2$ , according to Corollary 7, we have  $\begin{cases} \lambda_1^+ = \lambda_1 + \sum_{j=5}^6 m_{1j} = 2 + 0 = 2 \\ \lambda_2^+ = \lambda_2 + \sum_{j=5}^6 m_{2j} = 2 + 0 = 2 \\ \lambda_3^+ = \lambda_3 + \sum_{j=5}^6 m_{3j} = 3 + 0 = 2 \\ \lambda_4^+ = \lambda_4 + \sum_{j=5}^6 m_{4j} = 2 + 0 = 2 \\ \lambda_5^+ = \sum_{j=1}^6 m_{5j} = 1 \\ \lambda_6^+ = \sum_{j=1}^6 m_{6j} = 1 \end{cases}$ .

Thus,  $\lambda_{6 \times 6}^{CR_B} = \text{diag}(1/2, 1/2, 1/2, 1/3, 1, 1)$ .

• **The update of the decision matrix**

The decision matrix of  $U_1$  is  $GD_{4 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$ .

After adding  $U_2$ ,  $P'$  and  $R'$  are empty because it does not form a new decision class and  $Q' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

By Corollary 8, we have  $GD_{6 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

• **The update of the intermediate matrix**

The intermediate matrix of  $U_1$  is  $\Omega_{4 \times 2}^{CR_B} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \end{bmatrix}$ .

After adding  $U_2$ ,  $\Omega_{n^+ \times r^+}^{CR_B} = \left[ \frac{\Omega_{n \times r}^{CR_B} + P \bullet Q'}{Q \bullet GD_{n \times r} + R \bullet Q'} \mid \frac{M_{n \times n}^{CR_B} \bullet P' + P \bullet R'}{Q \bullet P' + R \bullet R'} \right] = \left[ \frac{\Omega_{n \times r}^{CR_B} + P \bullet Q'}{Q \bullet GD_{n \times r} + R \bullet Q'} \right]$  because  $P'$  and  $R'$  are empty.

Since

$$P \bullet Q' = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q \bullet GD_{4 \times 2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{and -}$$

$$R \bullet Q' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{we have } \Omega_{6 \times 2}^{CR_B} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

## 5.2. Updating the relation matrix and decision matrix when deleting objects

For a composite decision table  $CDT = (U, A \cup D, V, f)$ ,  $B \subseteq A$ , we suppose the object set  $U' = \{x_{n^+ + 1}, x_{n^+ + 2}, \dots, x_n\}$  is deleted, the number of rest objects is  $n^-$  and the number of rest decisions is  $r^-$ .

**Corollary 10.** Let  $M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$  and  $M_{n^- \times n^-}^{CR_B} = (m_{ij})_{n^- \times n^-}$  be relation matrices before and after deleting objects, respectively.  $\forall x_{i^*} \in U'$ , when it is deleted from the composite information system, we only need to delete Line  $i^*$  and Column  $i^*$  of the relation matrix, e.g.,

$$M_{(n-1) \times (n-1)}^{CR_B} = \begin{bmatrix} m_{11} & m_{12} & \cdots & \cancel{m_{1i^*}} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & \cancel{m_{2i^*}} & \cdots & m_{2n} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \cancel{m_{i^*1}} & \cancel{m_{i^*2}} & \cdots & \cancel{m_{i^*i^*}} & \cdots & \cancel{m_{i^*n}} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & \cancel{m_{ni^*}} & \cdots & m_{nn} \end{bmatrix}.$$

Therefore, it executes  $|U'|$  delete operations repeatedly to obtain  $M_{n^- \times n^-}^{CR_B} = (m_{ij})_{n^- \times n^-}$ .

**Corollary 11.** Let  $\Lambda_{n \times n}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_n}\right)$  and  $\Lambda_{n^- \times n^-}^{CR_B} = \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_{n^-}}\right)$  be induced diagonal matrices before and after deleting, respectively. Suppose  $U'$  is deleted. Then

$$\lambda_i^- = \begin{cases} \lambda_i - \sum_{j=n^+ + 1}^n m_{ij}, & 1 \leq i \leq n^- \\ \text{deleted}, & n^- + 1 \leq i \leq n \end{cases} \quad (42)$$

After deleting objects, it may remove one or more decision classes. We suppose that there are  $r^-$  decisions after deleting objects. It is easy to know that  $r^- \leq r$  and  $r^- = r$  if it does not remove decision class; Otherwise  $r^- < r$ .

**Corollary 12.** Let  $GD_{n \times r} = (G(D_1), G(D_2), \dots, G(D_r)) = (d_{ij})_{n \times r}$  and  $GD_{n^- \times r^-} = (d_{ij}^-)_{n^- \times r^-}$  be decision matrices before and after deleting objects, respectively.  $\forall x_{i^*} \in U'$ , when it is deleted from the composite information system, we only need to delete Line  $i^*$  of the decision matrix, e.g.,

$$GD_{(n-1) \times r} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1i^*} & \cdots & d_{1r} \\ d_{21} & d_{22} & \cdots & d_{2i^*} & \cdots & d_{2r} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ d_{i^*-1,1} & d_{i^*-1,2} & \cdots & d_{i^*-1,i^*} & \cdots & d_{i^*-1,r} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{ni^*} & \cdots & d_{nr} \end{bmatrix}.$$

Here, it executes  $|U'|$  delete operations repeatedly to obtain  $GD_{n^- \times r^-}$ . Then, we can obtain  $GD_{n^- \times r^-}$  after deleting the column whose entries are all zero.

**Corollary 13.** Let  $\Omega_{n \times r}^{CR_B} = M_{n \times n}^{CR_B} \bullet GD_{n \times r} = (\omega_{ij})_{n \times r}$  and  $\Omega_{n^- \times r^-}^{CR_B} = M_{n^- \times n^-}^{CR_B} \bullet GD_{n^- \times r^-} = (\omega_{ij}^-)_{n^- \times r^-}$  be intermediate matrices before and after deleting objects, respectively. Then

$$\Omega_{n \times r}^{CR_B} = \left[ \begin{array}{c|c} \Omega_{n^- \times r^-}^{CR_B} + P \bullet Q' & M_{n^- \times n^-}^{CR_B} \bullet P' + P \bullet R' \\ \hline Q \bullet GD_{n^- \times r^-} + R \bullet Q' & Q \bullet P' + R \bullet R' \end{array} \right] \quad (43)$$

where  $M_{n \times n}^{CR_B} = \left[ \begin{array}{c|c} M_{n^- \times n^-}^{CR_B} & P \\ \hline Q & R \end{array} \right]$ ,  $GD_{n \times r} = \left[ \begin{array}{c|c} GD_{n^- \times r^-} & P' \\ \hline Q' & R' \end{array} \right]$ , and

$$\begin{cases} P = \begin{bmatrix} m_{1,n^-+1} & m_{1,n^-+2} & \cdots & m_{1,n} \\ m_{2,n^-+1} & m_{2,n^-+2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n^-,n^-+1} & m_{n^-,n^-+2} & \cdots & m_{n^-,n} \end{bmatrix} & P' = [0]_{n^- \times (r-r^-)} \text{ is a zero matrix} \\ Q = \begin{bmatrix} m_{n^-+1,1} & m_{n^-+1,2} & \cdots & m_{n^-+1,n^-} \\ m_{n^-+2,1} & m_{n^-+2,2} & \cdots & m_{n^-+2,n^-} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n^-} \end{bmatrix} & Q' = \begin{bmatrix} d_{n^-+1,1} & d_{n^-+1,2} & \cdots & d_{n^-+1,r} \\ d_{n^-+2,1} & d_{n^-+2,2} & \cdots & d_{n^-+2,r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,r} \end{bmatrix} \\ R = \begin{bmatrix} m_{n^-+1,n^-+1} & m_{n^-+1,n^-+2} & \cdots & m_{n^-+1,n} \\ m_{n^-+2,n^-+1} & m_{n^-+2,n^-+2} & \cdots & m_{n^-+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,n^-+1} & m_{n,n^-+2} & \cdots & m_{n,n} \end{bmatrix} & R' = \begin{bmatrix} d_{n^-+1,r^-+1} & d_{n^-+1,r^-+2} & \cdots & d_{n^-+1,r} \\ d_{n^-+2,r^-+1} & d_{n^-+2,r^-+2} & \cdots & d_{n^-+2,r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,r^-+1} & d_{n,r^-+2} & \cdots & d_{n,r} \end{bmatrix} \end{cases}$$

or, equivalently,

$$\omega_{ij}^- = \begin{cases} \omega_{ij} - \sum_{k=n^-+1}^n m_{ik} d_{kj}, & 1 \leq i \leq n^-, 1 \leq j \leq r^- \\ \text{deleted,} & \text{else} \end{cases} \quad (44)$$

Then, we can obtain  $\omega_{n^- \times r^-}$  after deleting the column whose entries are all zero.

**Proof.** The proof is similar to that of Corollary 9.  $\square$

**Example 4** (Continuation of Example 1). The composite decision table is shown in Table 1. Let  $U_1 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$  and  $U_2 = \{x_5, x_6\}$  be the original object set and the deleted object set, respectively.

#### • The update of the relation matrix

According Example 2, the relation matrix of  $U_1$  is  $M_{6 \times 6}^{CR_B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$

After deleting  $U_2$ , by [Corollary 10](#), we have  $M_{4 \times 4}^{CR_B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$

• **The update of the induced diagonal matrix**

The induced diagonal matrix of  $U_1$  is  $\Lambda_{6 \times 6}^{CR_B} = \text{diag}(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_6}) = \text{diag}(1/2, 1/2, 1/2, 1/3, 1/1, 1).$

After deleting  $U_2$ , by [Corollary 11](#), we have  $\begin{cases} \lambda_1^- = \lambda_1 - \sum_{j=5}^6 m_{1j} = 2 - 0 = 2 \\ \lambda_2^- = \lambda_2 - \sum_{j=5}^6 m_{2j} = 2 - 0 = 2 \\ \lambda_3^- = \lambda_3 - \sum_{j=5}^6 m_{3j} = 3 - 0 = 2 \\ \lambda_4^- = \lambda_4 - \sum_{j=5}^6 m_{4j} = 2 - 0 = 2 \end{cases}$ .  $\lambda_5^-$  and  $\lambda_6^-$  are deleted.

Therefore,  $\Lambda_{4 \times 4}^{CR_B} = \text{diag}(1/2, 1/2, 1/3, 1/2).$

• **The update of the decision matrix**

The decision matrix of  $U_1$  is  $GD_{6 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$

After deleting  $U_2$ , by [Corollary 12](#), we have  $GD_{4 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}.$

• **The update of the intermediate matrix**

The intermediate matrix of  $U_1$  is  $\Omega_{6 \times 2}^{CR_B} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$

After deleting  $U_2$ , according to [Corollary 13](#), since  $P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$  and  $Q' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$

$$\Omega_{4 \times 2}^{CR_B} = \Omega_{4 \times 2}^{CR_B} - P \bullet Q' = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 0 \end{bmatrix}$$

## 6. Development of static and incremental algorithms for computing approximations based on the matrix

In this section, we design static and incremental algorithms for computing approximations based on the matrix in the composite decision table corresponding to [Sections 4 and 5](#), respectively.

### 6.1. Static algorithm MSACA (Matrix-based Static Algorithm for Computing Approximations)

Algorithm MSACA (see [Algorithm 1](#)) is a static (non-incremental) algorithm for computing approximations of the decision classes based on the matrix while the composite decision table is constant.

Step 2 is to construct the relation matrix and its key steps are to compute the equivalence relation, the neighborhood relation, the tolerance relation, the characteristic relation. Suppose  $B = \bigcup_{k=1,2,3,4} B_k$  and  $B_1, B_2, B_3, B_4$  generate the equivalence relation, the neighborhood relation, the tolerance relation, the characteristic relation, respectively, on the universe. The time complexity of computing the equivalence relation is  $O(|U||B_1|)$  [36]; The time complexity of computing the neighborhood relation is  $O(|U|\log|U||B_2|)$  [11]; The time complexity of computing the tolerance relation is

$O(|U||B_3| + \sum_{j=1}^{|B_3|} \sum_{k=1}^{j-1} |sv_{b_k}| |V_{b_k}|) \approx O(|U||B_3|^2)$  where  $|sv_{b_k}|$  is the number of objects with set-value under the attribute  $b_k$ , and  $|V_{b_k}|$  represents the number of values under the attribute  $b_k$  [30]; The time complexity of computing the characteristic relation is  $O(|U||B_4| + \sum_{j=1}^{|B_4|} \sum_{k=1}^{j-1} |*_{b_k}| |V_{b_k}|) \approx O(|U||B_4|^2)$  where  $|*_{b_k}|$  is the number of objects with missing value  $*$  under the attribute  $b_k$ , and  $|V_{b_k}|$  represents the number of values (is not equal to  $*$  or  $?$ ) under the attribute  $b_k$  [29]. After that, all four relations generate the relation matrix with the time complexity  $O(|U|^2)$ . Therefore, the time complexity of Step 2 is  $O(|U||B_1| + |U|\log|U||B_2| + |U||B_3|^2 + |U||B_4|^2) = O(|U|(|B_1| + |B_2|\log|U| + |B_3|^2 + |B_4|^2 + |U|))$ .

Step 3 is to compute the induced diagonal matrix and its time complexity is  $O(|U|^2)$ .

Step 4 is to construct the decision matrix and its time complexity is  $O(|U|r)$ .

Step 5 is to compute the intermediate matrix and its time complexity is  $O(|U|^2r)$ .

Step 6 is to compute the basic matrix and its time complexity is  $O(|U|r)$ .

Step 7 is to generate and output the approximations and its time complexity is  $O(|U|r)$ .

Hence, the total time complexity is

$$O(|U|(|B_1| + |B_2|\log|U| + |B_3|^2 + |B_4|^2) + |U|^2 + |U|r + |U|^2r + |U|r + nr) = O(|U|(|B_1| + |B_2|\log|U| + |B_3|^2 + |B_4|^2 + |U|r))$$

**Algorithm 1.** Matrix-based Static Algorithm for Computing Approximations of decision classes in the composite decision table (MSACA)

---

**Input:** A composite decision table  $CDT = (U, A \cup D, V, f)$ , and attribute subset  $B \subseteq A$ .

**Output:** The lower and upper approximations of the decision classes.

1 **begin**

2     **Construct** the relation matrix:  $M_{n \times n}^{CR_b} = (m_{ik})_{n \times n}$  // According to Definition 9 and Lemma 1

3     **Compute** the induced diagonal matrix of  $M_{n \times n}^{CR_b}$ :  $A_{n \times n}^{CR_b} = \text{diag}(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_n})$  // According to Definition 10 and

Corollary 2

4     **Construct** the decision matrix:  $GD_{n \times r} = (d_{kj})_{n \times r}$  // According to Definition 13

5     **Compute** the intermediate matrix:  $\Omega_{n \times r}^{CR_b} = M_{n \times n}^{CR_b} \bullet GD_{n \times r}$  // According to Corollary 4

6     **Compute** the basic matrix:  $HD = A_{n \times n}^{CR_b} \bullet \Omega_{n \times r}^{CR_b}$  // According to Corollary 4

7     **Generate and Output** the approximations by using the cut matrices of  $HD$  // According to Corollary 5

8 **end**

---

## 6.2. Incremental algorithm MIAUA-A (Matrix-based Incremental Algorithm for Updating Approximations when Adding objects)

Algorithm MIAUA-A (see Algorithm 2) is a dynamic (incremental) algorithm for updating approximations while the objects enter in the composite decision table. Steps 2–5 are to update the relation matrix, the induced diagonal matrix, the decision matrix and intermediate matrix according to Corollaries 6–9, respectively. Suppose the added object set is  $U'$  and  $U^* = U \cup U'$ . The time complexity of Step 2 is

$$\begin{aligned} &O(|U'|(|B_1| + |B_2|\log|U'| + |B_3|^2 + |B_4|^2 + |U'|) + |U||U'| |B| + |U'| |U| |B|) \\ &= O(|U'|(|B_1| + |B_2|\log|U| + |B_3|^2 + |B_4|^2 + |U'| + |U| |B|)) \end{aligned}$$

The time complexity of Step 3 is  $O(|U'| |U| + |U'| |U^*|) = O(|U'| |U^*|)$ .

The time complexity of Step 4 is  $O(|U'| r^+)$ .

The time complexity of Step 5 is  $O(|U^+|(|U^+| r^+ - |U| r) + |U| |U'| r) = O(|U^+|^2 r^+ - |U|^2 r)$ .

Step 6 is to compute the basic matrix and its time complexity is  $O(|U^+| r^+)$ .

Step 7 is to generate and output the approximations and its time complexity is  $O(|U^+| r^+)$ .

Therefore, the total time complexity is

$$O(|U'|(|B_1| + |B_2|\log|U| + |B_3|^2 + |B_4|^2 + |U'| + |U| |B|) + |U^+|^2 r^+ - |U|^2 r)$$

The incremental algorithm always needs extra space to store the intermediate result, which include the relation matrix, the induced diagonal matrix, the decision matrix, the intermediate matrix. The space complexities of the relation matrix, the induced diagonal matrix, the decision matrix, and the intermediate matrix are  $O(|U|^2)$ ,  $O(|U|)$ ,  $O(|U|r)$ , and  $O(|U|r)$ , respectively. Therefore, the total space complexity is

$$O(|U|^2 + |U| + |U|r + |U|r) = O(|U|(|U| + r))$$

Since the relation matrix and the decision matrix are boolean matrices, the space would be reduced if we use smaller storage unit to store these values in the real applications. For example, we can use “byte/bit” to storage rather than “integer”.

After adding objects, the total space complexity becomes  $O(|U^+|(|U^+| + r^+))$  for next update.

**Algorithm 2.** Matrix-based Incremental Algorithm for Updating Approximations when Adding objects (MIAUA-A)

---

**Input:** (1) A composite decision table  $CDT = (U, A \cup D, V, f)$ , and attribute subset  $B \subseteq A$ .  
 (2) Intermediate results:  $M_{n \times n}^{CR_B} = (m_{ik})_{n \times n}$  (relation matrix),  $A = \text{diag}(1/\lambda_i)$  (induced diagonal matrix),  $GD = (d_{kj})_{n \times r}$  (decision matrix) and  $\Omega_{n \times r}^{CR_B}$  (intermediate matrix), which have been computed before;  
 (3) The added object set:  $U'$ , where  $|U \cup U'| = n^+$ .  
**Output:** The updated lower and upper approximations of the decision classes.  
**1 begin**  
**2 Update** the relation matrix:  $M_{n^+ \times n^+}^{CR_B} = \left[ \begin{array}{c|c} M_{n \times n}^{CR_B} & P \\ \hline Q & R \end{array} \right]$ . // According to Corollary 6  
**3 Update** the induced diagonal matrix:  $A_{n^+ \times n^+}^{CR_B} = \text{diag}(\frac{1}{\lambda_1^+}, \frac{1}{\lambda_2^+}, \dots, \frac{1}{\lambda_{n^+}^+})$ , where  

$$\lambda_i^+ = \begin{cases} \lambda_i + \sum_{k=n+1}^{n^+} m_{ik}, & 1 \leq i \leq n \\ \sum_{k=1}^{n^+} m_{ik}, & n+1 \leq i \leq n^+ \end{cases}$$
 // According to Corollary 7  
**4 Update** the decision matrix:  $GD_{n^+ \times r^+} = \left[ \begin{array}{c|c} GD_{n \times r} & P' \\ \hline Q' & R' \end{array} \right]$ . // According to Corollary 8  
**5 Update** the intermediate matrix:  $\Omega_{n^+ \times r^+}^{CR_B} = \left[ \begin{array}{c|c} \Omega_{n \times r}^{CR_B} + P \bullet Q' & M_{n \times n}^{CR_B} \bullet P' + P \bullet R' \\ \hline Q \bullet GD_{n \times r} + R \bullet Q' & Q \bullet P' + R \bullet R' \end{array} \right]$ . // According to Corollary 9.  
**6 Compute** the basic matrix:  $HD = A_{n^+ \times n^+}^{CR_B} \bullet \Omega_{n^+ \times r^+}^{CR_B}$ . // According to Corollary 4  
**7 Compute and Output** the approximations by using the cut matrices of  $HD$ . // According to Corollary 5  
**8 end**

---

**6.3. Incremental algorithm MIAUA-D (Matrix-based Incremental Algorithm for Updating Approximations when Deleting objects)**

Algorithm MIAUA-D (see Algorithm 3) is a dynamic (incremental) algorithm for updating approximations while the objects get out of the composite decision table. The update process of deleting objects can be viewed as inverse process of adding objects. Hence, the intermediate matrix is updated first, and then followed by the update of the decision matrix, the induced diagonal matrix and the relation matrix in turn, according to Corollaries 13, 12, 11, 10 respectively.

Suppose the deleted object set is  $U'$  and  $U^- = U - U'$ . The delete operation is very fast, e.g., the time complexity of deleting  $|U^-|$  is  $O(|U^-|)$ .

Step 2 is to update the intermediate matrix and its time complexity is  $O(|U'| |U^-| r^- + |U'| + |U^-| r^-) = O(|U'| |U^-| r^-)$ .

Step 3 is to update the decision matrix and its time complexity is  $O(|U^-| + |U^-| r^-) = O(|U^-| r^-)$ .

Step 4 is to update the induced diagonal matrix its time complexity is  $O(|U^-| |U'| + |U^-|)$ .

Step 5 is to update the relation matrix and its time complexity is  $O(|U^-| + |U^-|) = O(|U^-|)$ .

Step 6 is to compute the basic matrix and its time complexity is  $O(|U^-| r^-)$ .

Step 7 is to generate and output the approximations and its time complexity is  $O(|U^-| r^-)$ .

Hence, the total time complexity is

$$O(|U'| |U^-| r^- + |U^-| r^- + |U^-| |U'| + |U^-| + |U^-| + |U^-| r^- + |U^-| r^-) = O(|U'| |U^-| r^-)$$

The total space complexity is  $O(|U|(|U| + r))$ , which is used to store the intermediate results, i.e., the intermediate matrix, the decision matrix, the induced diagonal matrix and the relation matrix.

After deleting objects, the total space complexity becomes  $O(|U^-|(|U^-| + r^-))$  for next update.

**Algorithm 3.** Matrix-based Incremental Algorithm for Updating Approximations when Deleting objects (MIAUA-D)

---

**Input:** (1) A composite decision table  $CDT = (U, A \cup D, V, f)$ , and attribute subset  $B \subseteq A$ .  
 (2) Intermediate results:  $M_{n \times n}^{CR_B} = (m_{ik})_{n \times n}$  (relation matrix),  $A = \text{diag}(1/\lambda_i)$  (induced diagonal matrix),  $GD = (d_{kj})_{n \times r}$  (decision matrix), and  $\Omega_{n \times r}^{CR_B}$  (intermediate matrix), which have been computed before;  
 (3) The deleted object set:  $U'$ , where  $|U - U'| = n^-$ .

---

(continued on next page)

**Appendix 3** (continued)

**Output** The updated lower and upper approximations of the decision classes.

**1 begin**

**2 Update** the intermediate matrix:  $\Omega_{n^- \times r^-}^{CR_B} = (\omega_{ij}^-)_{n^- \times r^-}$ , where  $\omega_{ij}^- = \begin{cases} \omega_{ij} - \sum_{k=n^-+1}^n m_{ik} d_{kj}, & 1 \leq i \leq n^-, 1 \leq j \leq r^- \\ \text{deleted}, & \text{else} \end{cases}$ ,

then delete the column whose entries are all zero.// According to [Corollary 13](#)

**3 Update** the decision matrix:  $GD_{n^- \times r^-}$  by executing  $|U|$  delete operations repeatedly and delete the column whose entries are all zero.// According to [Corollary 12](#)

**4 Update** the induced diagonal matrix:  $A_{n^- \times n^-}^{CR_B} = \text{diag}(\frac{1}{\lambda_1^-}, \frac{1}{\lambda_2^-}, \dots, \frac{1}{\lambda_{n^-}^-})$ , where

$$\lambda_i^- = \begin{cases} \lambda_i - \sum_{j=n^-+1}^n m_{ij}, & 1 \leq i \leq n^- \\ \text{deleted}, & n^- + 1 \leq i \leq n \end{cases} // \text{According to Corollary 11}$$

**5 Update** the relation matrix:  $M_{n^- \times n^-}^{CR_B}$  by executing  $|U|$  delete operations repeatedly.// According to [Corollary 10](#)

**6 Compute** the basic matrix:  $HD = A_{n^- \times n^-}^{CR_B} \bullet \Omega_{n^- \times r^-}^{CR_B}$ .// According to [Corollary 4](#)

**7 Compute and Output** the approximations by using the cut matrices of  $HD$ .// According to [Corollary 5](#)

**8 end**

**7. Experimental analysis**

The experiments are carried out on a PC with the operation system Ubuntu 12.04 (64-bit), which has 8 GB main memory and uses Inter (R) Core (TM) i7-2670QM (4 cores and 8 threads in all) with a clock frequency of 2.20 GHz. All the algorithms are coded in C++ and compiled with g++, whose version is 4.6.3.

To test and compare the performances of the incremental and static algorithms, we download four data sets from the machine learning data repository, University of California at Irvine [22]. The set-valued data generator [40] is employed and one set-valued data set SVD1 is generated. Besides, we develop a composite data generator, which generates four different composite data sets named CIS1, CIS2, CIS3 and CIS4. All these data sets are outlined in [Table 3](#).

**7.1. A comparison of incremental and static algorithms when adding data**

To compare incremental and static algorithms when adding data, we divide each of these nine data sets in [Table 3](#) into ten parts of equal size. The combination of first five parts are regarded as the original data set. And each time, we add one part, two parts, ..., five parts to the original data set in sequence. Let  $R_a$  be the ratio of the number of adding data and original data, called adding ratio. It means that  $R_a$  is equal to 20%, 40%, 60%, 80%, 100%, respectively. We run both two algorithms on these data sets, see [Fig. 1](#).

From [Fig. 1](#), the computational time of both two algorithms increase with the increasing size of adding data. As shown in each sub-figure of [Fig. 1](#), the incremental algorithm consistently is faster than the static algorithm on updating approximations when the same data is added to the original data set in the composite decision table. Obviously, if the added data is smaller, the incremental algorithm is more efficient than the static one.

To show the superiority of the incremental algorithm more clearly, an incremental speedup is given by the following formula:

$$IncS = \frac{T_s}{T_i} \quad (45)$$

where  $T_s$  is the execution time of the static algorithm,  $T_i$  is the execution time of the incremental algorithm.

[Table 4](#) shows the incremental speedup versus the adding ratio of the data. It is easy to see that the proposed incremental algorithm achieves 2.6–4.7× speedup over the static algorithm when  $R_a$  is equal to 20%. The speedup becomes smaller and

**Table 3**

A description of data sets.

|   | Data sets  | Samples | Attributes  |           |            |         |       | Classes |
|---|------------|---------|-------------|-----------|------------|---------|-------|---------|
|   |            |         | Categorical | Numerical | Set-valued | Missing | Total |         |
| 1 | Nursery    | 12,960  | 8           | 0         | 0          | 0       | 8     | 5       |
| 2 | Mushroom   | 8124    | 21          | 0         | 0          | 1       | 22    | 2       |
| 3 | Magic      | 19,020  | 0           | 10        | 0          | 0       | 10    | 2       |
| 4 | Ionosphere | 351     | 0           | 34        | 0          | 0       | 34    | 2       |
| 5 | SVD1       | 10,000  | 0           | 0         | 80         | 0       | 80    | 5       |
| 6 | CIS1       | 1000    | 10          | 10        | 10         | 10      | 40    | 10      |
| 7 | CIS2       | 2000    | 20          | 20        | 20         | 20      | 80    | 10      |
| 8 | CIS3       | 4000    | 40          | 40        | 40         | 40      | 160   | 10      |
| 9 | CIS4       | 8000    | 80          | 80        | 80         | 80      | 320   | 10      |



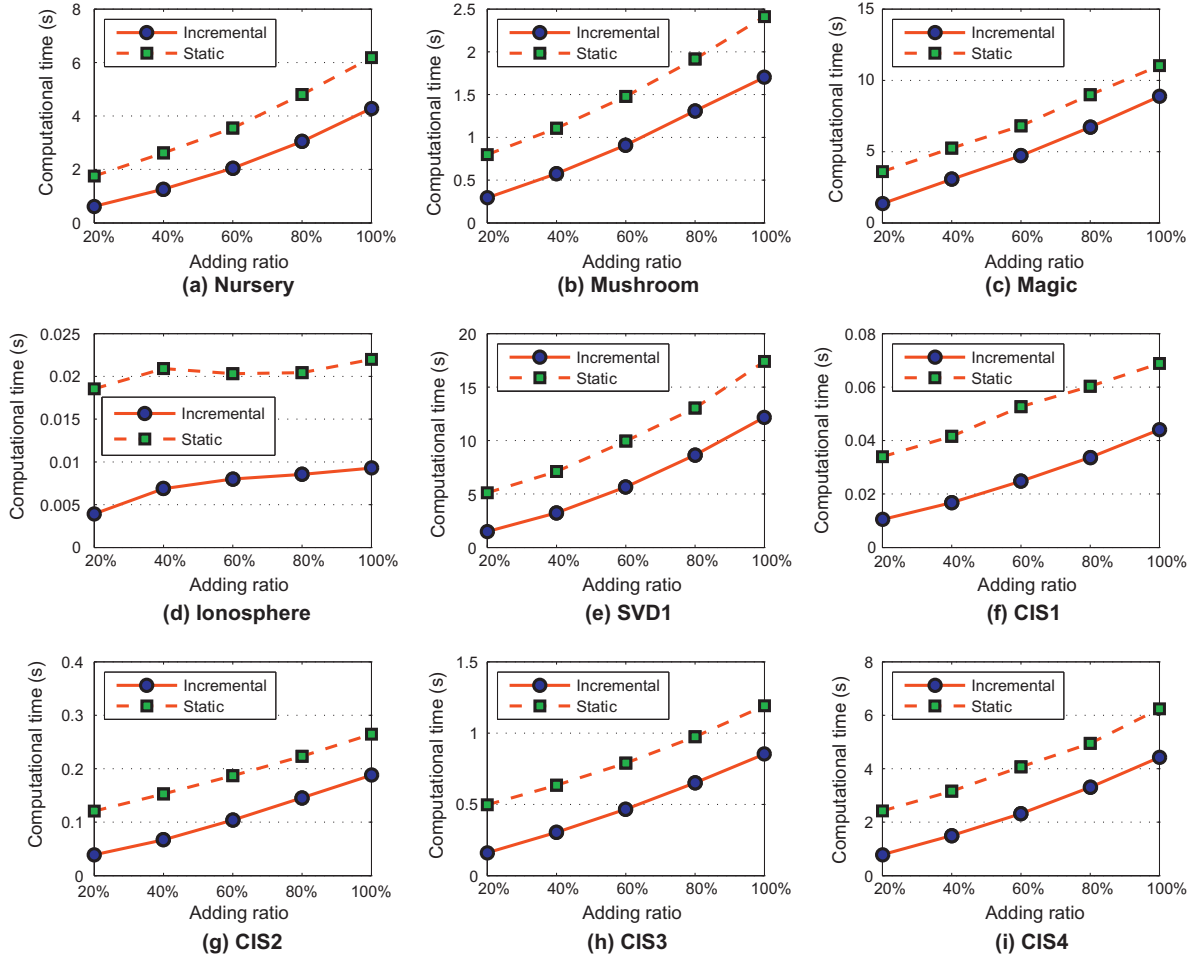


Fig. 1. A comparison of incremental and static algorithms versus the adding ratio of the data.

Table 4

The incremental speedup ( $IncS$ ) versus the adding ratio of the data.

| Data sets  | $R_a$ (Adding ratio) |       |       |       |       |
|------------|----------------------|-------|-------|-------|-------|
|            | 20%                  | 40%   | 60%   | 80%   | 100%  |
| Nursery    | 2.821                | 2.075 | 1.732 | 1.578 | 1.443 |
| Mushroom   | 2.731                | 1.921 | 1.625 | 1.463 | 1.417 |
| Magic      | 2.663                | 1.714 | 1.443 | 1.341 | 1.244 |
| Ionosphere | 4.715                | 3.046 | 2.536 | 2.392 | 2.371 |
| SVD1       | 3.451                | 2.198 | 1.758 | 1.508 | 1.428 |
| CIS1       | 3.238                | 2.482 | 2.123 | 1.790 | 1.561 |
| CIS2       | 3.095                | 2.284 | 1.799 | 1.534 | 1.408 |
| CIS3       | 3.090                | 2.092 | 1.694 | 1.498 | 1.397 |
| CIS4       | 3.087                | 2.106 | 1.753 | 1.498 | 1.411 |
| Average    | 3.210                | 2.213 | 1.829 | 1.622 | 1.520 |

smaller with the increase of the adding ratio. And yet for all that, the average speedup is still  $1.5\times$  even when the data volume is doubling ( $R_a = 100\%$ ).

## 7.2. A comparison of incremental and static algorithms when deleting data

To compare incremental and static algorithms when deleting data, we divide each of these nine data sets in Table 3 into ten parts of equal size. The combination of all ten parts are regarded as the original data set and each time, we delete one

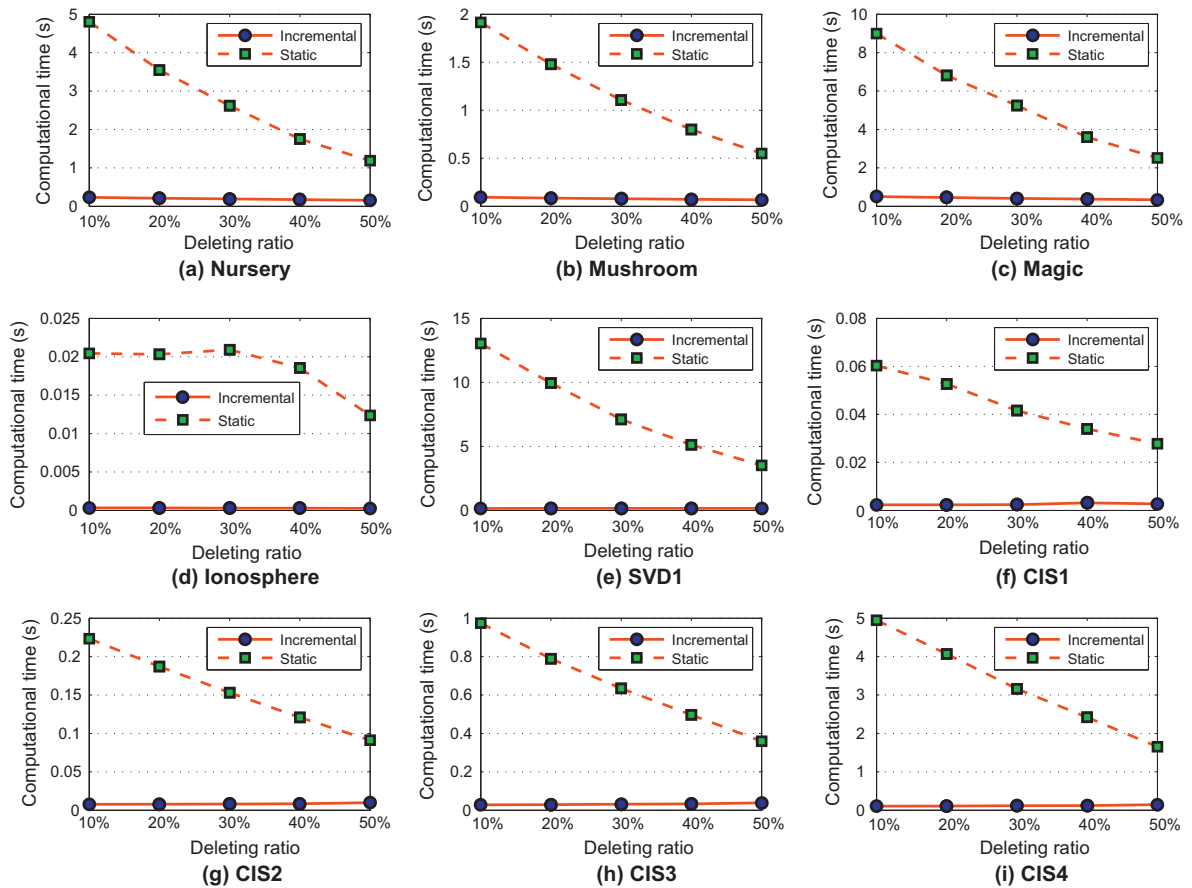


Fig. 2. A comparison of incremental and static algorithms versus the deleting ratio of the data.

Table 5

The incremental speedup (*IncS*) versus the deleting ratio of the data.

| Data sets  | $R_d$ (Deleting ratio) |        |        |        |        |
|------------|------------------------|--------|--------|--------|--------|
|            | 10%                    | 20%    | 30%    | 40%    | 50%    |
| Nursery    | 20.634                 | 16.867 | 13.674 | 10.109 | 7.472  |
| Mushroom   | 20.379                 | 17.243 | 14.015 | 10.963 | 8.132  |
| Magic      | 17.899                 | 14.994 | 12.697 | 9.629  | 7.434  |
| Ionosphere | 60.897                 | 63.538 | 70.849 | 64.964 | 45.180 |
| SVD1       | 85.502                 | 59.881 | 44.575 | 32.565 | 22.634 |
| CIS1       | 25.651                 | 22.941 | 17.421 | 10.774 | 10.182 |
| CIS2       | 29.246                 | 24.057 | 19.219 | 14.344 | 9.290  |
| CIS3       | 34.175                 | 26.953 | 20.375 | 15.126 | 9.294  |
| CIS4       | 46.875                 | 37.114 | 27.323 | 19.780 | 11.371 |
| Average    | 37.918                 | 31.510 | 26.683 | 20.917 | 14.554 |

part, two parts, etc., five parts from the original data set in sequence. Let  $R_d$  be the ratio of the number of deleting data and original data, called deleting ratio. It means that  $R_d$  is equal to 10%, 20%, 30%, 40%, 50%, respectively. We run both two algorithms on these data sets, see Fig. 2.

In Fig. 2, it is easy to know that the computational time of both two algorithms decrease with the increasing size of deleting data. As shown in each sub-figure of Fig. 2, the incremental algorithm consistently is faster than the static algorithm on updating approximations when the same data is deleted from the original data set in the composite decision table. It is obvious that the incremental algorithm is more efficient than the static one when the deleted data becomes smaller.

We here also use the incremental speedup (*IncS*) to test the performance of the incremental algorithm. The incremental speedup versus the deleting ratio of the data is shown in Table 5. The proposed incremental algorithm achieves 17.8–85.5 $\times$  speedup over the static algorithm when  $R_d$  is equal to 10%. The *IncS* is pretty well when deleting data because the delete

operation is always very fast. The speedup becomes smaller with the increase of the deleting ratio. But the average speedup is still  $14.5\times$  even when  $R_d$  is equal to 50%.

## 8. Related work

Matrix-based methods in rough sets were introduced by Liu [19]. By using the inner product method and matrix method, Liu proposed a unified axiomatic system to characterize the upper approximation operations [19]. Then, the matrix-based methods were extended to deal with numerical and set-valued data by using neighborhood and set-valued rough set models, respectively, in our previous work [40,41]. Specially, we discussed about the incremental methods under the variation of the attribute set [40]. To be different from that, in this paper, we mainly discussed about the incremental methods under the variation of the object set on the composite rough set model, which is very useful in dealing with attributes of multiple different types.

In previous research work, incremental methods for dynamic information systems have been widely studied. (1) To adapt to the variation of the object set, Shan and Ziarko put forward a discernibility-matrix based incremental methodology to find all maximally generalized rules [31]. Zheng et al. developed a rough set and rule tree based incremental knowledge acquisition algorithm RRIA [42]. Liu et al. presented an optimization incremental method for inducing interesting knowledge [18]. Zhang et al. proposed an incremental method for dynamic data mining based on neighborhood rough sets [39]. Based on multi-dominance discernibility matrices, Jia et al. proposed a non-incremental algorithm and an incremental algorithm by means of dominance-based rough set approach [12]. (2) To adapt to the variation of the attribute set, Chan presented an incremental mining algorithm for learning classification rules efficiently [2]. Li et al. put forward a method for updating approximations of a concept in an incomplete information system under the characteristic relation [17]. Cheng took the boundary set and cut set as springboards and proposed an incremental model for fast computing the rough fuzzy approximations [5]. Zhang et al. gave a matrix-based incremental method for fast updating the approximations in set-valued information systems. Li et al. introduced a kind of dominance matrix to calculate P-dominating sets and P-dominated sets in dominance-based rough sets approach (DRSA), then proposed incremental approaches and algorithms for updating approximations in DRSA [16]. (3) To adapt to the variation of the attribute values, Chen et al. proposed an incremental algorithm for updating the approximations under coarsening or refining of attributes' values in information systems [3]. Chen et al. also presented the principles of dynamically updating approximations w.r.t. attribute values' coarsening and refining in incomplete ordered decision systems under the extended dominance characteristic relation based rough sets, and proposed algorithms for incremental updating approximations of an upward union and downward union of classes [4].

## 9. Conclusions

In this paper, we defined composite information systems that contained attributes of multiple different types. A composite relation was given to deal with attributes of multiple different types simultaneously. The composite rough set model under the composite relation was presented and its approximations, positive, boundary and negative regions were defined. The relation matrix, the decision matrix and the basic matrix were introduced to derive the approximations, positive, boundary and negative regions from the composite information system intuitively. Finally, to adapt to the dynamic variation of the composite information system, we employed an incremental technique and presented a matrix-based incremental method for fast updating the approximations when many objects enter into or get out of the composite information system. Experimental studies pertaining to different UCI data sets and user-defined data sets showed that the proposed algorithms significantly reduce computational time of updating approximations. In future, we will study on rule acquisition and attribute reduction in composite information systems. Our another future work is to develop a parallel method to mine knowledge from massive composite data.

## Acknowledgments

This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117) and NSAF (No. U1230117), the Fundamental Research Funds for the Central Universities (No. SWJTU11ZT08), the Research Fund of Traction Power State Key Laboratory, Southwest Jiaotong University (No. 2012TPL\_T15), the Science and Technology Planning Project of Sichuan Province (No. 2012RZ0009) and the Fostering Foundation for the Excellent Ph.D. Dissertation of Southwest Jiaotong University (No. 2012ZJB), China.

## References

- [1] H.M. Abu-Donia, Multi knowledge based rough approximations and applications, *Knowledge-Based Systems* 26 (2012) 20–29.
- [2] C.C. Chan, A rough set approach to attribute generalization in data mining, *Information Sciences* 107 (1998) 169–176.
- [3] H.M. Chen, T.R. Li, S.J. Qiao, D. Ruan, A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values, *International Journal of Intelligent Systems* 25 (2010) 1005–1026.
- [4] H.M. Chen, T.R. Li, D. Ruan, Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining, *Knowledge-Based Systems* 31 (2012) 140–161.
- [5] Y. Cheng, The incremental method for fast computing the rough fuzzy approximations, *Data & Knowledge Engineering* 70 (2011) 84–100.

- [6] J. Grzymala-Busse, Characteristic relations for incomplete data: a generalization of the indiscernibility relation, in: S. Tsumoto, R. Slowinski, J. Komorowski, J. Grzymala-Busse (Eds.), *Rough Sets and Current Trends in Computing*, Lecture Notes in Computer Science, vol. 3066, Springer, Berlin/Heidelberg, 2004, pp. 244–253.
- [7] J. Grzymala-Busse, Characteristic relations for incomplete data: a generalization of the indiscernibility relation, in: J. Peters, A. Skowron (Eds.), *Transactions on Rough Sets IV*, Lecture Notes in Computer Science, vol. 3700, Springer, Berlin/Heidelberg, 2005, pp. 58–68.
- [8] Y. Guan, H. Wang, Set-valued information systems, *Information Sciences* 176 (2006) 2507–2525.
- [9] Q.H. Hu, J.F. Liu, D.R. Yu, Mixed feature selection based on granulation and approximation, *Knowledge-Based Systems* 21 (2008) 294–304.
- [10] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, *Pattern Recognition* 40 (2007) 3509–3521.
- [11] Q.H. Hu, D.R. Yu, J.F. Liu, C.X. Wu, Neighborhood rough set based heterogeneous feature subset selection, *Information Sciences* 178 (2008) 3577–3594.
- [12] X.Y. Jia, L. Shang, J.J. Chen, X.Y. Dai, Incremental versus non-incremental: data and algorithms based on ordering relations, *International Journal of Computational Intelligence Systems* 4 (2011) 112–122.
- [13] M. Kryszkiewicz, Rough set approach to incomplete information systems, *Information Sciences* 112 (1998) 39–49.
- [14] M. Kryszkiewicz, Rules in incomplete information systems, *Information Sciences* 113 (1999) 271–292.
- [15] Y. Leung, M.M. Fischer, W.Z. Wu, J.S. Mi, A rough set approach for the discovery of classification rules in interval-valued information systems, *International Journal of Approximate Reasoning* 47 (2008) 233–246.
- [16] S.Y. Li, T.R. Li, D. Liu, Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set, *Knowledge-Based Systems* 40 (2013) 17–26.
- [17] 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 (2007) 485–494.
- [18] D. Liu, T.R. Li, D. Ruan, J.B. Zhang, Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, *Journal of Global Optimization* 51 (2011) 325–344, <http://dx.doi.org/10.1007/s10898-010-9607-8>.
- [19] G.L. Liu, The axiomatization of the rough set upper approximation operations, *Fundamenta Informaticae* 69 (2006) 331–342.
- [20] G.L. Liu, Axiomatic systems for rough sets and fuzzy rough sets, *International Journal of Approximate Reasoning* 48 (2008) 857–867.
- [21] J.S. Mi, W.X. Zhang, An axiomatic characterization of a fuzzy generalization of rough sets, *Information Sciences* 160 (2004) 235–249.
- [22] 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://archive.ics.uci.edu/ml/>>.
- [23] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, System Theory, Knowledge Engineering and Problem Solving, vol. 9, Kluwer Academic Publishers, Dordrecht, 1991.
- [24] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, *Information Sciences* 177 (2007) 41–73.
- [25] Z. Pawlak, A. Skowron, Rough sets: some extensions, *Information Sciences* 177 (2007) 28–40.
- [26] Z. Pawlak, A. Skowron, Rudiments of rough sets, *Information Sciences* 177 (2007) 3–27.
- [27] J. Qian, D.Q. Miao, Z.H. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, *International Journal of Approximate Reasoning* 52 (2011) 212–230.
- [28] Y.H. Qian, C.Y. Dang, J.Y. Liang, D.W. Tang, Set-valued ordered information systems, *Information Sciences* 179 (2009) 2809–2832.
- [29] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, *Artificial Intelligence* 174 (2010) 597–618.
- [30] Y.H. Qian, J.Y. Liang, F. Wang, A positive-approximation based accelerated algorithm to feature selection from incomplete decision tables, *Chinese Journal of Computers* 34 (2011) 435–442.
- [31] N. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, *Computational Intelligence* 11 (1995) 357–370.
- [32] J. Stefanowski, A. Tsoukiàs, On the extension of rough sets under incomplete information, in: *Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, RSFDGrC '99*, Springer-Verlag, London, UK, UK, 1999, pp. 73–81.
- [33] J. Stefanowski, A. Tsoukiàs, Incomplete information tables and rough classification, *Computational Intelligence* 17 (2001) 545–566.
- [34] G.Y. Wang, Extension of rough set under incomplete information systems, in: *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1098–1103.
- [35] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, *Journal of Artificial Intelligence Research* 6 (1997) 1–34.
- [36] Z.Y. Xu, Z.P. Liu, B.R. Yang, W. Song, A quick attribute reduction algorithm with complexity of  $\max(O(|C||U|), O(|C|^2|U/C|))$ , *Chinese Journal of Computers* 29 (2006) 391–398.
- [37] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, *Information Sciences* 111 (1998) 239–259.
- [38] J.B. Zhang, T.R. Li, H.M. Chen, Composite rough sets, in: J. Lei, F. Wang, H. Deng, D. Miao (Eds.), *Artificial Intelligence and Computational Intelligence*, Lecture Notes in Computer Science, vol. 7530, Springer, Berlin/Heidelberg, 2012, pp. 150–159.
- [39] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Neighborhood rough sets for dynamic data mining, *International Journal of Intelligent Systems* 27 (2012) 317–342.
- [40] 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 (2012) 620–635.
- [41] J.B. Zhang, T.R. Li, Y. Yang, L. Wang, Neighborhood rough sets based matrix approach for calculation of the approximations, in: *Proceedings of the 6th International Conference on Rough Sets and Knowledge Technology, RSKT'11*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 166–171.
- [42] Z. Zheng, G.Y. Wang, RRIA: a rough set and rule tree based incremental knowledge acquisition algorithm, *Fundamenta Informaticae* 59 (2004) 299–313.