Int. J. Mach. Learn. & Cyber. (2017) 8:1131–1144
DOI 10.1007/s13042-015-0484-9

ORIGINAL ARTICLE

Knowledge reduction of dynamic covering decision information systems caused by variations of attribute values

Mingjie Cai[1,2] · Qingguo Li[1] · Jianmin Ma[3]

Received: 22 May 2015 / Accepted: 16 December 2015 / Published online: 31 December 2015

Abstract In practical situations, it is time-consuming to conduct knowledge reduction of dynamic covering decision information systems caused by variations of attribute values with the non-incremental approaches. In this paper, motivated by the need for knowledge reduction of dynamic covering decision information systems, we introduce incremental approaches to computing the type-1 and type-2 characteristic matrices for constructing the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces caused by revising attribute attributes. We also employ several examples to explain how to compute the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. Then we propose the incremental algorithms for computing the second and sixth lower and upper approximations of sets and employ experimental results to illustrate the incremental algorithms are effective to calculate the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. Finally, we give two examples to show how to conduct knowledge reduction of dynamic covering decision information systems caused by altering attribute values.

Keywords Boolean matrix · Characteristic matrix · Dynamic covering approximation space · Dynamic covering decision information system · Rough set

✉ Qingguo Li
liqingguoli@aliyun.com

Mingjie Cai
cmjlong@163.com

Jianmin Ma
cjm-zm@126.com

1 College of Mathematics and Econometrics, Hunan University, Changsha 410082, Hunan, People's Republic of China

2 Department of Computer Science, University of Regina, Regina, SK S4S 0A2, Canada

3 Department of Mathematics and Information Science, Faculty of Science, Chang'an University, Xi'an 710064, Shaan'xi, People's Republic of China

1 Introduction

Nowadays covering approximation spaces as generalizations of classical approximation spaces have attracted increasing attentions, and a great deal of approximation operators [4, 16–19, 25, 26, 32, 35, 43, 55, 56, 60–62, 64, 68, 70–75] have been proposed for computing the lower and upper approximations of sets in covering approximation spaces. For example, Zakowski [64] proposed the classical lower and upper approximation operators for covering approximation spaces by extending Pawlak's model. Pomykala [32] presented two pair of dual approximation operators by modifying Zakowski's definition. Tsang et al. [43] gave the concept of the lower and upper approximation operators using the minimal descriptions. Zhu et al. [72, 74, 75] provided three types of lower and upper approximation operators. They also systematically investigated these six types of approximation operators and presented relationships among them. Subsequently, Chen et al. [4] presented a new covering to construct the lower and upper approximations of an arbitrary set with respect to the covering application background of rough sets. Lin et al. [16, 17] investigated neighborhood-based multi-granulation rough sets to deal with data sets with hybrid attributes. Liu et al. [25, 26] proposed covering fuzzy rough set based on multigranulation rough sets. Wang et al. [55] studied the second and sixth lower and upper approximation operators of covering approximation spaces using characteristic matrices. Yao [60] presented a more systematic formulation of covering based rough sets from three aspects: the element, the granule and the subsystem. So far covering-based rough set theory has been applied to many fields such as data mining and knowledge discovery, and the application fields are being increasing with the development of computer sciences and covering-based rough set theory.

Knowledge reduction of information systems as the major work of rough set theory has attracted more attentions in recent years, and researchers have presented different reducts with respect to different criterions and proposed effective algorithms for conducting knowledge reduction of information systems [3, 5, 6, 8–10, 27, 31, 33, 34, 41, 42, 44–50, 53, 54, 57, 63, 69]. For example, Chen et al. [3] presented the concept of reducts for consistent and inconsistent covering decision information systems with covering rough sets. Qian et al. [33] gave the concepts of the lower and upper approximation reducts of decision information systems. Slezak et al. [42] investigated decision reduct of decision information systems. Zhang et al. [69] proposed the concept of assignment reducts and maximum assignment reducts for decision information systems. Consequently, Kryszkiewicz et al. [6] provided the notion of discernibility matrix and discernibility function for computing decision reducts. Leung et al. [10] discussed dependence space-based attribute reduction in inconsistent decision information systems. Miao et al. [31] presented the generalized discernibility matrix and discernibility function of three types of relative reducts. Skowron et al. [41] proposed the concept of the classical discernibility matrix and discernibility function for constructing relative reducts of decision information systems. In practice, information systems vary with the time due to the dynamic characteristic of data collections, and it is time-consuming to conduct knowledge reduction of dynamic information systems with the non-incremental approaches. To solve this issue, researchers [1–3, 11–15, 20–24, 28–30, 36–40, 51, 52, 58, 59, 65–67] focus on investigating knowledge reduction of dynamic information systems using incremental approaches. For example, when coarsening and refining attribute values and varying sets of attribute, Chen et al. [1–3] constructed approximations of sets and provided an effective approach to knowledge reduction of dynamic information systems. Lang et al. [7] presented incremental approaches for computing approximations of sets in dynamic covering approximation spaces caused by variations of object sets and conducted knowledge reduction of dynamic covering decision information systems with the immigration and emigration of objects. Li et al. [14] extended rough sets for incrementally updating decision rules which handles dynamic maintenance of decision rules in data mining based on characteristic relations. Liu et al. [20, 21, 24] presented incremental approaches for knowledge reduction of dynamic information systems and dynamic incomplete information systems. Yang et al. [58] studied the neighborhood system for knowledge reduction of incomplete information systems from the perspective of knowledge engineering and neighborhood systems-based rough sets. Zhang et al. [65] presented matrix-based approaches for computing the approximations, positive, boundary and negative regions in composite information systems. In practice, covering approximation spaces vary with the time because of variations of attribute values. For example, two specialists A and B decided the quality of five cars $U=\{A,B,C,D,E\}$ as follows: $good=\{A,C\}, middle=\{C,E\}, bad=\{B,D,E\}$, and obtained the covering approximation space $(U,\mathscr{C})$, where $\mathscr{C}=\{good, middle, bad\}$. By considering time variations, the specialists found that the quality of C was very bad, and $(U,\mathscr{C})$ should be revised into dynamic covering approximation space $(U,\mathscr{C}^*)$, where $\mathscr{C}^*=\{good^*, middle^*, bad^*\}$, $good^*=\{A\}, middle^*=\{E\}$, and $bad^*=\{B,C,D,E\}$. But it is time-consuming to construct the type-1 and type-2 characteristic matrices of $\mathscr{C}^*$ with the non-incremental approach. The experimental results have demonstrated the incremental approaches are effective to conduct knowledge reduction of dynamic information systems, because it reduces the computational times greatly. Such an observation motivates us to compute approximations of sets in dynamic covering approximation spaces and knowledge reduction of dynamic covering decision information systems using the incremental approaches.

The purpose of this paper is to study knowledge reduction of dynamic covering decision information systems caused by altering attribute values. First, we investigate structures of the type-1 and type-2 characteristic matrices of dynamic covering approximation spaces because of variations of attribute values and present incremental approaches to computing the type-1 and type-2 characteristic matrices of dynamic coverings. We also employ several examples to illustrate the process of calculating the type-1 and type-2 characteristic matrices can be simplified greatly by utilizing the incremental approaches. Second, we provide incremental algorithms for constructing the type-1 and type-2 characteristic matrices-based approximations of sets in dynamic covering approximation spaces caused by variations of attribute values. We also compare the time complexities of the incremental algorithms with those of non-incremental algorithms. Third, we perform experiments on ten dynamic covering approximation spaces generated randomly and employ the experimental results to illustrate the incremental approaches are effective to calculate the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces with the variation of attribute values. Finally, we employ two examples to show how to conduct knowledge reduction of dynamic covering decision information systems with the incremental approaches.

The rest of this paper is organized as follows: Sect. 2 briefly reviews the basic concepts of covering-based rough set theory. Section 3 introduces incremental approaches for computing the type-1 and type-2 characteristic matrices of dynamic coverings because of varying attribute values. Section 4 presents non-incremental and incremental algorithms for calculating the second and sixth lower and upper approximations of sets using the type-1 and type-2 characteristic matrices. Section 5 performs experiments to show the incremental approaches are effective to compute the second and sixth approximations of sets in dynamic covering approximation spaces. Section 6 devotes to knowledge reduction of dynamic covering decision information systems. We give the conclusions in Sect. 7.

2 Preliminaries

A brief summary of concepts related to covering-based rough sets is given in this section.

Definition 2.1 [64] Let U be a finite universe of discourse, and $\mathscr{C}$ a family of subsets of U. If none of elements of $\mathscr{C}$ is empty and $\bigcup\{C|C \in \mathscr{C}\} = U$, then $\mathscr{C}$ is referred to as a covering of U. In addition, $(U, \mathscr{C})$ is called a covering approximation space if $\mathscr{C}$ is a covering of U.

Definition 2.2 [55] Let $(U, \mathscr{C})$ be a covering approximation space, and $N(x) = \bigcap\{C_i|x \in C_i \in \mathscr{C}\}$. For any $X \subseteq U$, the second and sixth upper and lower approximations of X with respect to $\mathscr{C}$ are defined as follows:

(1) $SH_{\mathscr{C}}(X) = \bigcup\{C \in \mathscr{C} \mid C \cap X \neq \emptyset\}, SL_{\mathscr{C}}(X) = [SH_{\mathscr{C}}(X^c)]^c$;
(2) $XH_{\mathscr{C}}(X) = \{x \in U \mid N(x) \cap X \neq \emptyset\}, XL_{\mathscr{C}}(X) = \{x \in U \mid N(x) \subseteq X\}$.

The second and sixth lower and upper approximation operators are typical approximation operators for covering approximation spaces, and they are also dual operators. Furthermore, researchers have established the foundation for further studying the second and sixth lower and upper approximation operators in dynamic environment.

Definition 2.3 Let $(U, \mathscr{C})$ be a covering approximation space, where $U = \{x_1, x_2, \ldots, x_n\}$ and $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$. Then the representation matrix of $\mathscr{C}$ is defined as: $M_{\mathscr{C}} = (a_{ij})_{n \times m}$, where $a_{ij} = \begin{cases} 1, & x_i \in C_j, \\ 0, & x_i \notin C_j. \end{cases}$

According to Definition 2.3, a covering may induce different representation matrices due to different positions of blocks in $\mathscr{C}$. Furthermore, the characteristic function of $X \subseteq U$ is defined as: $\mathcal{X}_X = [a_1 a_2 \cdots a_n]^T$, where $a_i = \begin{cases} 1, & x_i \in X, \\ 0, & x_i \notin X, \end{cases} i = 1, 2, \ldots, n.$

Definition 2.4 Let $(U, \mathscr{C})$ be a covering approximation space, and $M_{\mathscr{C}} = (a_{ij})_{n \times m}$ a matrix representation of $\mathscr{C}$, where $U = \{x_1, x_2, \ldots, x_n\}$, $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$, and $a_{ij} = \begin{cases} 1, & x_i \in C_j, \\ 0, & x_i \notin C_j, \end{cases}$. Then

(1) $\Gamma(\mathscr{C}) = M_{\mathscr{C}} \bullet M_{\mathscr{C}}^T = (b_{ij})_{n \times n}$ is called the type-1 characteristic matrix of $\mathscr{C}$, where $b_{ij} = \bigvee_{k=1}^{m} (a_{ik} \cdot a_{jk})$.
(2) $\prod(\mathscr{C}) = M_{\mathscr{C}} \odot M_{\mathscr{C}}^T = (c_{ij})_{n \times n}$ is called the type-2 characteristic matrix of $\mathscr{C}$, where $c_{ij} = \bigwedge_{k=1}^{m} (a_{jk} - a_{ik} + 1)$.

By Definition 2.4, we see the type-1 and type-2 characteristic matrices are symmetric and asymmetric, respectively. Furthermore, we show the second and sixth lower and upper approximations of sets using the type-1 and type-2 characteristic matrices, respectively, as follows:

Definition 2.5 [55] Let $(U, \mathscr{C})$ be a covering approximation space, and $\mathcal{X}_X$ the characteristic function of X in U. Then

(1) $\mathcal{X}_{SH(X)} = \Gamma(\mathscr{C}) \bullet \mathcal{X}_X, \mathcal{X}_{SL(X)} = \Gamma(\mathscr{C}) \odot \mathcal{X}_X$;
(2) $\mathcal{X}_{XH(X)} = \prod(\mathscr{C}) \bullet \mathcal{X}_X, \mathcal{X}_{XL(X)} = \prod(\mathscr{C}) \odot \mathcal{X}_X$.

By Definition 2.5, Lang et al. presented the concepts of type-1 and type-2 reducts of covering decision information systems as follows:

Definition 2.6 [7] Let $(U, \mathscr{D} \cup U/d)$ be a covering decision information system, where $\mathscr{D} = \{\mathscr{C}_i|i \in I\}$, $U/d = \{D_i|i \in J\}$, $I = \{1, 2, \ldots, n_1\}$, $J = \{1, 2, \ldots, n_2\}$ two integer sets, and $\mathscr{P} \subseteq \mathscr{D}$. $\mathscr{P}$ is called a type-1 reduct of $(U, \mathscr{D} \cup U/d)$ if it satisfies (1) and (2) simultaneously as follows:

(1) $\Gamma(\mathscr{D}) \bullet \mathcal{X}_{D_i} = \Gamma(\mathscr{P}) \bullet \mathcal{X}_{D_i}$ and $\Gamma(\mathscr{D}) \odot \mathcal{X}_{D_i} = \Gamma(\mathscr{P}) \odot \mathcal{X}_{D_i}, \forall i \in J$,
(2) $\Gamma(\mathscr{D}) \bullet \mathcal{X}_{D_i} \neq \Gamma(\mathscr{P}') \bullet \mathcal{X}_{D_i}$ and $\Gamma(\mathscr{D}) \odot \mathcal{X}_{D_i} \neq \Gamma(\mathscr{P}') \odot \mathcal{X}_{D_i}, \forall \mathscr{P}' \subset \mathscr{P}$.

Definition 2.7 [7] Let $(U, \mathscr{D} \cup U/d)$ be a covering decision information system, where $\mathscr{D} = \{\mathscr{C}_i|i \in I\}$, $U/d = \{D_i|i \in J\}$, $I = \{1, 2, \ldots, n_1\}$, $J = \{1, 2, \ldots, n_2\}$ two integer sets, and $\mathscr{P} \subseteq \mathscr{D}$. $\mathscr{P}$ is called a type-2 reduct of $(U, \mathscr{D} \cup U/d)$ if it satisfies (1) and (2) simultaneously as follows:

(1) $\prod(\mathscr{D}) \bullet \mathcal{X}_{D_i} = \prod(\mathscr{P}) \bullet \mathcal{X}_{D_i} and \prod(\mathscr{D}) \odot \mathcal{X}_{D_i} = \prod(\mathscr{P}) \odot \mathcal{X}_{D_i}, \forall i \in J$,

(2) $\prod(\mathscr{D}) \bullet \mathcal{X}_{D_i} \neq \prod(\mathscr{P}') \bullet \mathcal{X}_{D_i} and \prod(\mathscr{D}) \odot \mathcal{X}_{D_i} \neq \prod(\mathscr{P}') \odot \mathcal{X}_{D_i}, \forall \mathscr{P}' \subset \mathscr{P}$.

For simplicity, we define the operators $+$ and $-$ between $A = (a_{ij})_{n\times m}$ and $B = (b_{ij})_{n\times m}$ as follows: $A + B = (a_{ij} + b_{ij})_{n\times m}$ and $A - B = (a_{ij} - b_{ij})_{n\times m}$. Furthermore, we define the operators $\bullet$ and $\odot$ between $C = (c_{ij})_{n\times m}$ and $D = (d_{jk})_{m\times p}$ as follows: $C \bullet D = (e_{ik})_{n\times p}$ and $C \odot D = (f_{ik})_{n\times p}$, where $e_{ik} = \bigvee_{j=1}^{m}(c_{ij} \cdot d_{jk})$ and $f_{ik} = \bigwedge_{j=1}^{m}(d_{jk} - c_{ij} + 1)$.

3 Incremental approaches for computing the second and sixth approximations of sets

In this section, we present incremental approaches for computing the second and sixth lower and upper approximations of sets.

Definition 3.1 Let $(U, \mathscr{C})$ and $(U, \mathscr{C}^*)$ be covering approximation spaces, where $U = \{x_1, x_2, \ldots, x_n\}$, $\mathscr{C} = \{C_1, C_2, ..., C_m\}$, $\mathscr{C}^* = \{C_1^*, C_2^*, \ldots, C_m^*\}$, and $C_i^* - \{x_k\} = C_i - \{x_k\}(1 \leq i \leq m)$, where $x_k \in U$. Then $(U, \mathscr{C}^*)$ is called a dynamic covering approximation space of $(U, \mathscr{C})$.

According to Definition 3.1, the dynamic covering approximation space $(U, \mathscr{C}^*)$ is generated by revising attribute values of x_k. In practice, variations of attribute values maybe result in $|\mathscr{C}^*| < |\mathscr{C}|$, $|\mathscr{C}^*| = |\mathscr{C}|$ or $|\mathscr{C}^*| > |\mathscr{C}|$. For example, it will result in new blocks or combing different blocks into a block. In this work, we only discuss the situation $|\mathscr{C}^*| = |\mathscr{C}|$ caused by revising attribute values of an object.

Below, we discuss the relationship between $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^*)$. For convenience, we denote $M_{\mathscr{C}} = (a_{ij})_{n\times m}$, $M_{\mathscr{C}^*} = (b_{ij})_{n\times m}$, $\Gamma(\mathscr{C}) = (c_{ij})_{n\times n}$, and $\Gamma(\mathscr{C}^*) = (d_{ij})_{n\times n}$.

Theorem 3.2 *Let $(U, \mathscr{C}^*)$ be a dynamic covering approximation space of $(U, \mathscr{C})$, $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^*)$ the type-1 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^*$, respectively. Then*

$$\Gamma(\mathscr{C}^*) = \Gamma(\mathscr{C}) + \Delta\Gamma(\mathscr{C}),$$

where

$$\Delta\Gamma(\mathscr{C}) = \begin{bmatrix} 0 & 0 & \cdots & d_{1k}^* & \ldots & 0 \\ 0 & 0 & \cdots & d_{2k}^* & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ d_{k1}^* & d_{k2}^* & \cdots & d_{kk}^* & \cdots & d_{kn}^* \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \cdots & d_{nk}^* & \cdots & 0 \end{bmatrix},$$

$$d_{kj}^* = d_{jk}^* = [\, b_{k1} \quad b_{k2} \quad \cdots \quad b_{km} \,] \bullet [\, b_{1j} \quad b_{2j} \quad \cdots \quad b_{mj} \,]^T - c_{kj}.$$

Proof By Definition 2.4, $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^*)$ are presented as follows:

$$\begin{aligned} \Gamma(\mathscr{C}) &= M_{\mathscr{C}} \bullet M_{\mathscr{C}}^T \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \bullet \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}^T \\ &= \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \Gamma(\mathscr{C}^*) &= M_{\mathscr{C}^*} \bullet M_{\mathscr{C}^*}^T \\ &= \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix} \bullet \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}^T \\ &= \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}. \end{aligned}$$

By Definition 2.4, since $a_{ij} = b_{ij}$ for $i \neq k$, we have $c_{ij} = d_{ij}$ for $i \neq k, j \neq k$. To compute $\Gamma(\mathscr{C}^*)$ on the basis of $\Gamma(\mathscr{C})$, we only need to compute $(d_{ij})_{(i=k, 1\leq j\leq n)}$ and $(d_{ij})_{(1\leq i\leq n, j=k)}$. Since $\Gamma(\mathscr{C}^*)$ is symmetric, we only need to compute $(d_{ij})_{(i=k,1\leq j\leq n)}$. In other words, we need to compute $\Delta\Gamma(\mathscr{C})$, where

$$\Delta\Gamma(\mathscr{C}) = \begin{bmatrix} 0 & 0 & \cdots & d_{1k}^* & \ldots & 0 \\ 0 & 0 & \cdots & d_{2k}^* & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ d_{k1}^* & d_{k2}^* & \cdots & d_{kk}^* & \cdots & d_{kn}^* \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \cdots & d_{nk}^* & \cdots & 0 \end{bmatrix},$$

$$d_{kj}^* = d_{jk}^* = [\, b_{k1} \quad b_{k2} \quad \cdots \quad b_{km} \,] \bullet [\, b_{1j} \quad b_{2j} \quad \cdots \quad b_{mj} \,]^T - c_{kj}.$$

Therefore, we have

$$\Gamma(\mathscr{C}^*) = \Gamma(\mathscr{C}) + \Delta\Gamma(\mathscr{C}).$$

□

The following example shows the process of constructing the second lower and upper approximations of sets.

Example 3.3 Let $U = \{x_1, x_2, x_3, x_4\}$, $\mathscr{C} = \{C_1, C_2, C_3\}$, and $\mathscr{C}^* = \{C_1^*, C_2^*, C_3^*\}$, where $C_1 = \{x_1, x_4\}$, $C_2 = \{x_1, x_2, x_4\}$, $C_3 = \{x_3, x_4\}$, $C_1^* = \{x_1, x_3, x_4\}$, $C_2^* = \{x_1, x_2, x_3, x_4\}$, $C_3^* = \{x_4\}$, and $X = \{x_3, x_4\}$. By Definition 2.4, we first obtain

$$\Gamma(\mathscr{C}) = M_{\mathscr{C}} \bullet M_{\mathscr{C}}^T = (c_{ij})_{4\times 4} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

According to Definition 2.3, we get

$$M_{\mathscr{C}^*} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Second, we obtain

$$\begin{aligned} [d^*_{31} \quad d^*_{32} \quad d^*_{33} \quad d^*_{34}] &= [1 \quad 1 \quad 0] \bullet M^T_{\mathscr{C}^*} - [c_{31} \quad c_{32} \quad c_{33} \quad c_{34}] \\ &= [1 \quad 1 \quad 0] \bullet \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} - [0 \quad 0 \quad 1 \quad 1] \\ &= [1 \quad 1 \quad 1 \quad 1] - [0 \quad 0 \quad 1 \quad 1] \\ &= [1 \quad 1 \quad 0 \quad 0], \\ [d^*_{13} \quad d^*_{23} \quad d^*_{33} \quad d^*_{43}] &= [d^*_{31} \quad d^*_{32} \quad d^*_{33} \quad d^*_{34}]. \end{aligned}$$

By Theorem 3.2, we have

$$\Delta\Gamma(\mathscr{C}) = \begin{bmatrix} 0 & 0 & d^*_{13} & 0 \\ 0 & 0 & d^*_{23} & 0 \\ d^*_{31} & d^*_{32} & d^*_{33} & d^*_{34} \\ 0 & 0 & d^*_{43} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, we obtain

$$\Gamma(\mathscr{C}^*) = \Gamma(\mathscr{C}) + \Delta\Gamma(\mathscr{C}) = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

By Definition 2.5, we have

$$\mathcal{X}_{SH(X)} = \Gamma(\mathscr{C}^*) \bullet \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = [1 \quad 1 \quad 1 \quad 1]^T,$$

$$\mathcal{X}_{SL(X)} = \Gamma(\mathscr{C}^*) \odot \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = [0 \quad 0 \quad 0 \quad 0]^T.$$

Therefore, $SH(X) = \{x_1, x_2, x_3, x_4\}$ and $SL(X) = \emptyset$.

In Example 3.3, we only need to compute $\Delta\Gamma(\mathscr{C})$ by Theorem 3.2. But there is a need to compute all elements in $\Gamma(\mathscr{C}^*)$ by Definition 2.4. Therefore, the computational time of the incremental algorithm is less than the non-incremental algorithm. Subsequently, we discuss the construction of $\prod(\mathscr{C}^*)$ based on $\prod(\mathscr{C})$. For convenience, we denote $\prod(\mathscr{C}) = (e_{ij})_{n\times n}$ and $\prod(\mathscr{C}^*) = (f_{ij})_{n\times n}$.

Theorem 3.4 *Let $(U, \mathscr{C}^*)$ be a dynamic covering approximation space of $(U, \mathscr{C})$, $\prod(\mathscr{C})$ and $\prod(\mathscr{C}^*)$ the type-2 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^*$, respectively. Then*

$$\prod(\mathscr{C}^*) = \prod(\mathscr{C}) + \Delta\prod(\mathscr{C}),$$

where

$$\Delta\prod(\mathscr{C}) = \begin{bmatrix} 0 & 0 & \cdots & f^*_{1k} & \cdots & 0 \\ 0 & 0 & \cdots & f^*_{2k} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ f^*_{k1} & f^*_{k2} & \cdots & f^*_{kk} & \cdots & f^*_{kn} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & f^*_{nk} & \cdots & 0 \end{bmatrix},$$

$$[f^*_{k1} \quad f^*_{k2} \quad \cdots \quad f^*_{kn}] = [b_{k1} \quad b_{k2} \quad \cdots \quad b_{km}] \odot M^T_{\mathscr{C}^*} - [e_{k1} \quad e_{k2} \quad \cdots \quad e_{kn}],$$
$$[f^*_{1k} \quad f^*_{2k} \quad \cdots \quad f^*_{nk}]^T = M_{\mathscr{C}^*} \odot [b_{1k} \quad b_{2k} \quad \cdots \quad b_{mk}]^T - [e_{1k} \quad e_{2k} \quad \cdots \quad e_{nk}]^T.$$

Proof According to Definition 2.4, $\prod(\mathscr{C})$ and $\prod(\mathscr{C}^*)$ are presented as follows:

$$\prod(\mathscr{C}) = M_{\mathscr{C}} \odot M^T_{\mathscr{C}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \odot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}^T = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ e_{n1} & e_{n2} & \cdots & e_{nn} \end{bmatrix},$$

$$\prod(\mathscr{C}^*) = M_{\mathscr{C}^*} \odot M^T_{\mathscr{C}^*} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}^T = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix}.$$

By Definition 2.4, we have $e_{ij} = f_{ij}$ for $i \neq k, j \neq k$ since $a_{ij} = b_{ij}$ for $i \neq k$. To compute $\prod(\mathscr{C}^*)$ on the basis of $\prod(\mathscr{C})$, we only need to compute $(f_{ij})_{(i=k, 1\leq j\leq n)}$ and $(f_{ij})_{(1\leq i\leq n, j=k)}$. In other words, we need to compute $\Delta\prod(\mathscr{C})$, where

$$\Delta\prod(\mathscr{C})=\begin{bmatrix}0 & 0 & \cdots & f^*_{1k} & \cdots & 0\\ 0 & 0 & \cdots & f^*_{2k} & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ f^*_{k1} & f_{k2} & \cdots & f^*_{kk} & \cdots & f^*_{kn}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & f^*_{nk} & \cdots & 0\end{bmatrix},$$

$$[f^*_{k1}\ \ f^*_{k2}\ \ \cdots\ \ f^*_{kn}]=[b_{k1}\ \ b_{k2}\ \ \cdots\ \ b_{km}]\odot M^T_{\mathscr{C}^*}-[e_{k1}\ \ e_{k2}\ \ \cdots\ \ e_{kn}],$$
$$[f^*_{1k}\ \ f^*_{2k}\ \ \cdots\ \ f^*_{nk}]^T=M_{\mathscr{C}^*}\odot[b_{1k}\ \ b_{2k}\ \ \cdots\ \ b_{mk}]^T-[e_{1k}\ \ e_{2k}\ \ \cdots\ \ e_{nk}]^T.$$

Therefore, we have

$$\prod(\mathscr{C}^*)=\prod(\mathscr{C})+\Delta\prod(\mathscr{C}).$$

□

The following example is employed to show the process of constructing the sixth lower and upper approximations of sets.

Example 3.5 Let $U=\{x_1,x_2,x_3,x_4\}$, $\mathscr{C}=\{C_1,C_2,C_3\}$, and $\mathscr{C}^*=\{C^*_1,C^*_2,C^*_3\}$, where $C_1=\{x_1,x_4\}$, $C_2=\{x_1,x_2,x_4\}$, $C_3=\{x_3,x_4\}$, $C^*_1=\{x_1,x_3,x_4\}$, $C^*_2=\{x_1,x_2,x_3,x_4\}$, $C^*_3=\{x_4\}$, and $X=\{x_3,x_4\}$. By Definition 2.4, we first have

$$\prod(\mathscr{C})=M_{\mathscr{C}}\odot M^T_{\mathscr{C}}=(e_{ij})_{4\times 4}$$
$$=\begin{bmatrix}1&1&0\\0&1&0\\0&0&1\\1&1&1\end{bmatrix}\odot\begin{bmatrix}1&0&0&1\\1&1&0&1\\0&0&1&1\end{bmatrix}=\begin{bmatrix}1&0&0&1\\1&1&0&1\\0&0&1&1\\0&0&0&1\end{bmatrix}.$$

According to Definition 2.2, we have

$$M_{\mathscr{C}^*}=\begin{bmatrix}1&1&0\\0&1&0\\1&1&0\\1&1&1\end{bmatrix}.$$

Second, we get

$$[f^*_{31}\ \ f^*_{32}\ \ f^*_{33}\ \ f^*_{34}]=[1\ \ 1\ \ 0]\odot M^T_{\mathscr{C}^*}-[e_{31}\ \ e_{32}\ \ e_{33}\ \ e_{34}]$$
$$=[1\ \ 1\ \ 0]\odot\begin{bmatrix}1&0&1&1\\1&1&1&1\\0&0&0&1\end{bmatrix}-[0\ \ 0\ \ 1\ \ 1]$$
$$=[1\ \ 0\ \ 1\ \ 1]-[0\ \ 0\ \ 1\ \ 1]$$
$$=[1\ \ 0\ \ 0\ \ 0],$$

$$[f^*_{13}\ \ f^*_{23}\ \ f^*_{33}\ \ f^*_{43}]^T=\begin{bmatrix}1&1&0\\0&1&0\\1&1&0\\1&1&1\end{bmatrix}\odot\begin{bmatrix}1\\1\\0\end{bmatrix}-[e_{13}\ \ e_{23}\ \ e_{33}\ \ e_{43}]^T$$
$$=[1\ \ 1\ \ 1\ \ 0]^T-[0\ \ 0\ \ 1\ \ 0]^T$$
$$=[1\ \ 1\ \ 0\ \ 0]^T.$$

By Theorem 3.4, we have

$$\Delta\prod(\mathscr{C})=\begin{bmatrix}0&0&f^*_{13}&0\\0&0&f^*_{23}&0\\f^*_{31}&f^*_{32}&f^*_{33}&f^*_{34}\\0&0&f^*_{43}&0\end{bmatrix}=\begin{bmatrix}0&0&1&0\\0&0&1&0\\1&0&0&0\\0&0&0&0\end{bmatrix}.$$

Thus, we obtain

$$\prod(\mathscr{C}^*)=\prod(\mathscr{C})+\Delta\prod(\mathscr{C})$$
$$=\begin{bmatrix}1&0&0&1\\1&1&0&1\\0&0&1&1\\0&0&0&1\end{bmatrix}+\begin{bmatrix}0&0&1&0\\0&0&1&0\\1&0&0&0\\0&0&0&0\end{bmatrix}=\begin{bmatrix}1&0&1&1\\1&1&1&1\\1&0&1&1\\0&0&0&1\end{bmatrix}.$$

By Definition 2.5, we have

$$\mathcal{X}_{SH(X)}=\prod(\mathscr{C}^*)\bullet\mathcal{X}_X$$
$$=\begin{bmatrix}1&0&1&1\\1&1&1&1\\1&0&1&1\\0&0&0&1\end{bmatrix}\bullet\begin{bmatrix}0\\0\\1\\1\end{bmatrix}=[1\ \ 1\ \ 1\ \ 1]^T,$$

$$\mathcal{X}_{SL(X)}=\prod(\mathscr{C}^*)\odot\mathcal{X}_X$$
$$=\begin{bmatrix}1&0&1&1\\1&1&1&1\\1&0&1&1\\0&0&0&1\end{bmatrix}\odot\begin{bmatrix}0\\0\\1\\1\end{bmatrix}=[0\ \ 0\ \ 0\ \ 1]^T.$$

Therefore, we have $SH(X)=\{x_1,x_2,x_3,x_4\}$ and $SL(X)=\{x_4\}$.

In Example 3.5, we only need to compute $\Delta\prod(\mathscr{C})$ by Theorem 3.4. But there is a need to compute all elements in $\prod(\mathscr{C}^*)$ by Definition 2.4. Therefore, the computational time of the incremental algorithm is less than the non-incremental algorithm.

4 Non-incremental and incremental algorithms of computing approximations of sets

In this section, we present non-incremental and incremental algorithms of computing the second and sixth lower and upper approximations of sets.

Algorithm 4.1: Non-incremental algorithm of computing the second lower and upper approximations of sets(NIS)

Input: $(U,\mathscr{C}^*)$ and $X\subseteq U$.
Output: $\mathcal{X}_{SH(X)}$ and $\mathcal{X}_{SL(X)}$.
Construct $M_{\mathscr{C}^*}$ based on $\mathscr{C}^*$;
Compute $\Gamma(\mathscr{C}^*)=M_{\mathscr{C}^*}\bullet M^T_{\mathscr{C}^*}$;
Obtain $\mathcal{X}_{SH(X)}=\Gamma(\mathscr{C}^*)\bullet\mathcal{X}_X$ and $\mathcal{X}_{SL(X)}=\Gamma(\mathscr{C}^*)\odot\mathcal{X}_X$.

Algorithm 4.2: Incremental algorithm of computing the second lower and upper approximations of sets(IS)

Input: $(U,\mathscr{C})$, $\Gamma(\mathscr{C})$, $(U,\mathscr{C}^*)$, $X \subseteq U$.
Output: $\mathcal{X}_{SH(X)}$ and $\mathcal{X}_{SL(X)}$.
1 Construct $M^*_{\mathscr{C}}=(b_{ij})_{n\times m}$ based on $\mathscr{C}^*$;
2 Denote $row_k=[b_{k1},b_{k2},\ldots,b_{km}]$;
3 Compute $\Delta row_k=row_k \bullet M^T_{\mathscr{C}^*}$;
4 Let $\Gamma(\mathscr{C}^*)=\Gamma(\mathscr{C})$;
5 Set k-th row of $\Gamma(\mathscr{C}^*)$ as Δrow_k;
6 Set k-th col of $\Gamma(\mathscr{C}^*)$ as $(\Delta row_k)^T$;
7 Obtain $\mathcal{X}_{SH(X)}=\Gamma(\mathscr{C}^*) \bullet \mathcal{X}_X$ and $\mathcal{X}_{SL(X)}=\Gamma(\mathscr{C}^*) \odot \mathcal{X}_X$.

In Algorithm 4.1, the time complexity of step 2 is $O(mn^2)$; the time complexity of step 3 is $O(2n^2)$. The total time complexity is $O((m+2)n^2)$. In Algorithm 4.2, the time complexity of step 3 is $O(nm)$; the time complexity of step 5 is $O(n)$; the time complexity of step 6 is $O(n)$; the time complexity of step 7 is $O(2n^2)$. The total time complexity is $O(2n^2+nm+2n)$. Furthermore, $O((m+2)n^2)$ is the time complexity of the non-incremental algorithm. Therefore, the incremental algorithm is more effective than the non-incremental algorithm.

Algorithm 4.3: Non-incremental algorithm of computing the sixth lower and upper approximations of sets(NIX)

Input: $(U,\mathscr{C}^*)$ and $X \subseteq U$.
Output: $\mathcal{X}_{XH(X)}$ and $\mathcal{X}_{XL(X)}$.
1 Construct $M_{\mathscr{C}^*}$ based on $\mathscr{C}^*$;
2 Compute $\prod(\mathscr{C}^*) = M_{\mathscr{C}^*} \odot M^T_{\mathscr{C}^*}$;
3 Obtain $\mathcal{X}_{XH(X)} = \prod(\mathscr{C}^*) \bullet \mathcal{X}_X$ and $\mathcal{X}_{XL(X)} = \prod(\mathscr{C}^*) \odot \mathcal{X}_X$.

Algorithm 4.4: Incremental algorithm of computing the sixth lower and upper approximations of sets(IX)

Input: $(U,\mathscr{C})$, $\prod(\mathscr{C})$, $(U,\mathscr{C}^*)$ and $X \subseteq U$.
Output: $\mathcal{X}_{XH(X)}$ and $\mathcal{X}_{XL(X)}$.
1 Construct $M^*_{\mathscr{C}}=(b_{ij})_{n\times m}$ based on $\mathscr{C}^*$;
2 Denote $row_k=[b_{k1},b_{k2},\ldots,b_{km}]$;
3 Compute $\Delta row_k=row_k \odot M^T_{\mathscr{C}^*}$;
4 Denote $col_k=[b_{1k},b_{2k},\ldots,b_{mk}]^T$;
5 Compute $\Delta col_k=M_{\mathscr{C}^*} \odot col_k$;
6 Let $\prod(\mathscr{C}^*)=\prod(\mathscr{C})$;
7 Set k-th row of $\prod(\mathscr{C}^*)$ as Δrow_k;
8 Set k-th col of $\prod(\mathscr{C}^*)$ as Δcol_k;
9 Obtain $\mathcal{X}_{XH(X)}=\prod(\mathscr{C}^*) \bullet \mathcal{X}_X$ and $\mathcal{X}_{XL(X)}=\prod(\mathscr{C}^*) \odot \mathcal{X}_X$.

In Algorithm 4.3, the time complexity of step 2 is $O(mn^2)$; the time complexity of step 3 is $O(n^2)$. The total time complexity is $O((m+2)n^2)$. In Algorithm 4.4, the time complexity of step 3 is $O(nm)$; the time complexity of step 5 is $O(nm)$; the time complexity of step 7 is $O(n)$; the time complexity of step 8 is $O(n)$; the time complexity of step 9 is $O(2n^2)$. The total time complexity is $O(2n^2+2nm+2n)$. Furthermore, $O((m+2)n^2)$ is the time complexity of the non-incremental algorithm. Therefore, the incremental algorithm is more effective than the non-incremental algorithm.

5 Experimental analysis

In this section, we perform experiments to validate the effectiveness of Algorithms 4.2 and 4.4 for computing the second and sixth approximations of sets in dynamic covering approximation spaces caused by varying attribute values.

In practical situations, a large amount of computational time is used to transform incomplete information systems into covering approximation spaces, and the main objective of this study is to illustrate the efficiency of the Algorithms 4.2 and 4.4 in computing approximations of sets, ten covering approximation spaces are generated randomly with information shown in Table 1 to evaluate the performance of Algorithms 4.2 and 4.4, where $|U_i|$ denotes the number of objects in U_i and $|\mathscr{C}_i|$ means the cardinality of $\mathscr{C}_i$. All experiments are run on a PC with 64-bit Windows 7, Inter(R) Core(TM) i5-4200M CPU@2.50 GHZ and 4 GB memory; the computational software is Matlab R2013b 64-bit.

We apply Algorithms 4.1–4.4 to the covering approximation space $(U_i,\mathscr{C}_i)$, where $i=1,2,\ldots,10$, and compare the computational times by Algorithms 4.1 and 4.3 with those of Algorithms 4.2 and 4.4, respectively. First, we calculate $\Gamma(\mathscr{C}_i)$ and $\prod(\mathscr{C}_i)$ by Definition 2.4, where $i=1,2,\ldots,10$. Then we obtain the dynamic covering approximation space $(U_i,\mathscr{C}^*_i)$ because of revising attribute values of x_k, where $C^*_j = C_j \cup \{x_k\}$ or C_j, $C^*_j \in \mathscr{C}^*_i$ and $C_j \in \mathscr{C}_i$. Subsequently, we get $\Gamma(\mathscr{C}^*_i)$ and $\prod(\mathscr{C}^*_i)$ by Algorithms 4.1 and 4.3, respectively, where $i=1,2,\ldots,10$. Second, we calculate $SH(X)$, $SL(X)$, $XH(X)$, and $XL(X)$ based on $\Gamma(\mathscr{C}^*_i)$ and $\prod(\mathscr{C}^*_i)$ for $X \subseteq U_i$, respectively, where $i=1,2,\ldots,10$. Third, we obtain $\Gamma(\mathscr{C}^*_i)$ and $\prod(\mathscr{C}^*_i)$ by Algorithms 4.2 and 4.4, respectively. Fourth, we calculate $SH(X)$, $SL(X)$, $XH(X)$, and $XL(X)$ based on $\Gamma(\mathscr{C}^*_i)$ and $\prod(\mathscr{C}^*_i)$ for $X \subseteq U_i$, respectively, where $i=1,2,\ldots,10$. We conduct all experiments ten times and show the results in Table 2 and Fig. 1. In Table 2, the measure of time is in seconds; $\bar{t}$ indicates the average time of ten experiments; SD indicates the standard deviations of ten experimental results; in Fig. 1, i stands for the experimental number in x axis; in Fig. 1, i refers to the

Table 1 Covering approximation spaces

No.	Name	$\lvert U_i\rvert$	$\lvert\mathscr{C}_i\rvert$
1	$(U_1,\mathscr{C}_1)$	2000	100
2	$(U_2,\mathscr{C}_2)$	4000	200
3	$(U_3,\mathscr{C}_3)$	6000	300
4	$(U_4,\mathscr{C}_4)$	8000	400
5	$(U_5,\mathscr{C}_5)$	10,000	500
6	$(U_6,\mathscr{C}_6)$	12,000	600
7	$(U_7,\mathscr{C}_7)$	14,000	700
8	$(U_8,\mathscr{C}_8)$	16,000	800
9	$(U_9,\mathscr{C}_9)$	18,000	900
10	$(U_{10},\mathscr{C}_{10})$	20,000	1000

Table 2 Computational times using Algorithms 4.1–4.4 in $(U_i, \mathscr{C}_i)$

No	Algo.	1	2	3	4	5	6	7	8	9	10	$\bar{t}$	SD
1	NIS	0.4578	0.4213	0.4279	0.4223	0.4271	0.4236	0.4235	0.4263	0.4236	0.4273	0.4281	0.0107
	NIX	0.4681	0.4671	0.4636	0.4646	0.4668	0.4651	0.4651	0.4681	0.4668	0.4720	0.4667	0.0024
	IS	0.0044	0.0026	0.0033	0.0040	0.0029	0.0028	0.0031	0.0030	0.0028	0.0028	0.0032	0.0006
	IX	0.0351	0.0339	0.0333	0.0339	0.0340	0.0334	0.0340	0.0335	0.0338	0.0333	0.0338	0.0005
2	NIS	1.8902	1.8452	1.8610	1.8203	1.8179	1.8257	1.8223	1.8224	1.8294	1.8189	1.8353	0.0237
	NIX	2.0389	2.0437	2.0314	2.0237	2.0378	2.0331	2.0531	2.0565	2.0583	2.0641	2.0440	0.0134
	IS	0.0091	0.0118	0.0102	0.0100	0.0098	0.0127	0.0110	0.0099	0.0099	0.0096	0.0104	0.0011
	IX	0.2035	0.2018	0.2013	0.2018	0.2034	0.1992	0.2018	0.2006	0.1987	0.2035	0.2016	0.0017
3	NIS	4.2030	4.1889	4.1905	4.1457	4.1446	4.1681	4.1518	4.1765	4.2310	4.1604	4.1760	0.0277
	NIX	4.6993	4.7126	4.6838	4.6895	4.6941	4.7000	4.7025	4.6711	4.7039	4.6939	4.6951	0.0116
	IS	0.0177	0.0210	0.0211	0.0199	0.0199	0.0199	0.0199	0.0205	0.0200	0.0197	0.0200	0.0009
	IX	0.5259	0.5059	0.5076	0.5056	0.5089	0.5055	0.5106	0.5080	0.5059	0.5078	0.5092	0.0061
4	NIS	7.5968	7.5550	7.7428	7.6536	7.6756	7.7031	7.6304	7.6051	7.6118	7.7013	7.6475	0.0581
	NIX	8.6892	8.7967	8.8918	9.0384	8.7810	8.7764	8.6300	9.2821	8.6324	8.6121	8.8130	0.2112
	IS	0.0428	0.0338	0.0350	0.0394	0.0378	0.0386	0.0345	0.0345	0.0346	0.0348	0.0366	0.0029
	IX	0.9813	0.9681	0.9694	0.9677	0.9669	0.9731	0.9654	0.9683	0.9648	0.9685	0.9694	0.0048
5	NIS	12.0856	11.9662	11.9944	11.9200	11.9992	11.9683	11.9321	11.9008	11.8811	11.8839	11.9532	0.0633
	NIX	13.8290	13.6560	13.7430	13.7308	13.6831	13.6816	13.7970	13.6794	13.8141	13.7338	13.7348	0.0614
	IS	0.0675	0.0530	0.0549	0.0537	0.0551	0.0537	0.0536	0.0523	0.0535	0.0540	0.0551	0.0044
	IX	1.6266	1.6193	1.6163	1.6138	1.6189	1.6057	1.6230	1.6213	1.6172	1.6211	1.6183	0.0057
6	NIS	17.8842	17.8858	18.0800	17.6753	17.5945	17.5710	17.7019	18.2036	17.5415	17.9582	17.8096	0.2277
	NIX	20.1684	20.1404	20.0242	20.0022	20.0277	20.0598	20.0897	20.2560	21.6223	22.1194	20.4510	0.7613
	IS	0.0977	0.0748	0.0746	0.0744	0.0803	0.0727	0.0753	0.0735	0.0738	0.0723	0.0770	0.0076
	IX	2.4011	2.3671	2.4204	2.3771	2.3679	2.3662	2.3737	2.3644	2.3614	2.3692	2.3769	0.0189
7	NIS	24.2936	24.3201	24.4603	25.2946	24.4922	24.5153	24.3296	25.0792	24.6210	24.2059	24.5612	0.3554
	NIX	27.9154	28.2049	28.2523	28.2664	28.7698	28.2559	28.1121	28.4234	28.6467	29.2779	28.4125	0.3921
	IS	0.1071	0.1014	0.1017	0.0996	0.1015	0.1018	0.1025	0.1007	0.1020	0.1009	0.1019	0.0020
	IX	3.4572	3.3194	3.3070	3.3030	3.2899	3.3109	3.2777	3.2753	3.2790	3.2758	3.3095	0.0544
8	NIS	33.2714	33.3024	33.2390	33.2370	33.3127	33.3602	33.3527	33.2599	33.4496	33.3485	33.3133	0.0664
	NIX	39.0763	39.0729	39.1256	39.1677	39.1382	39.5114	39.2732	38.9632	39.1487	38.8493	39.1327	0.1765
	IS	0.1267	0.1243	0.1293	0.1242	0.1248	0.1239	0.1259	0.1234	0.1226	0.1284	0.1254	0.0022
	IX	6.1013	5.3888	5.3412	5.3710	5.2641	5.3158	5.3229	5.3422	5.2858	5.4398	5.4173	0.2456
9	NIS	44.2060	43.5990	43.2590	44.3375	43.9165	43.6185	44.3864	44.4667	44.2301	45.2159	44.1236	0.5542
	NIX	50.1711	50.8559	50.4446	49.7286	50.6871	50.3282	50.5291	49.5770	50.0544	50.3550	50.2731	0.4021
	IS	0.2048	0.1611	0.1628	0.1620	0.1607	0.1607	0.1605	0.1612	0.1615	0.1615	0.1657	0.0138
	IX	6.1794	5.8323	5.8586	5.7428	5.8902	5.8318	5.8949	5.7688	5.7606	5.8051	5.8564	0.1249
10	NIS	55.6793	55.8107	55.6728	55.9174	55.5917	58.1981	59.1824	56.0537	55.7757	55.5664	56.3448	1.2663
	NIX	64.8043	65.7104	65.2075	64.5169	64.7856	64.7118	65.0349	64.4148	64.7802	64.3155	64.8282	0.4101
	IS	0.2716	0.1941	0.1944	0.1924	0.1938	0.1956	0.1936	0.1917	0.1947	0.1948	0.2017	0.0246
	IX	8.3148	7.6287	7.3082	7.9581	7.2058	7.4084	7.1585	7.2874	7.1620	7.2413	7.4673	0.3879

covering approximation space $(U_i, \mathscr{C}_i)$ in $x-$axis; while $y-$axes corresponds to the computational time to compute approximations of sets in dynamic covering approximation spaces; *NIS*, *IS*, *NIX*, and *IX* stand for the time of constructing the second and sixth lower and upper approximations of sets by Algorithms 4.1, 4.2, 4.3 and 4.3, respectively.

In Table 2, we see that Algorithms 4.1–4.4 are stable to compute approximations of sets in all experiments. That is, the computational times of each algorithm are almost the same. Moreover, the computational times of approximations of sets using incremental algorithms are much smaller than those of the non-incremental algorithms. In other words, the computational times of Algorithms 4.2 and 4.4

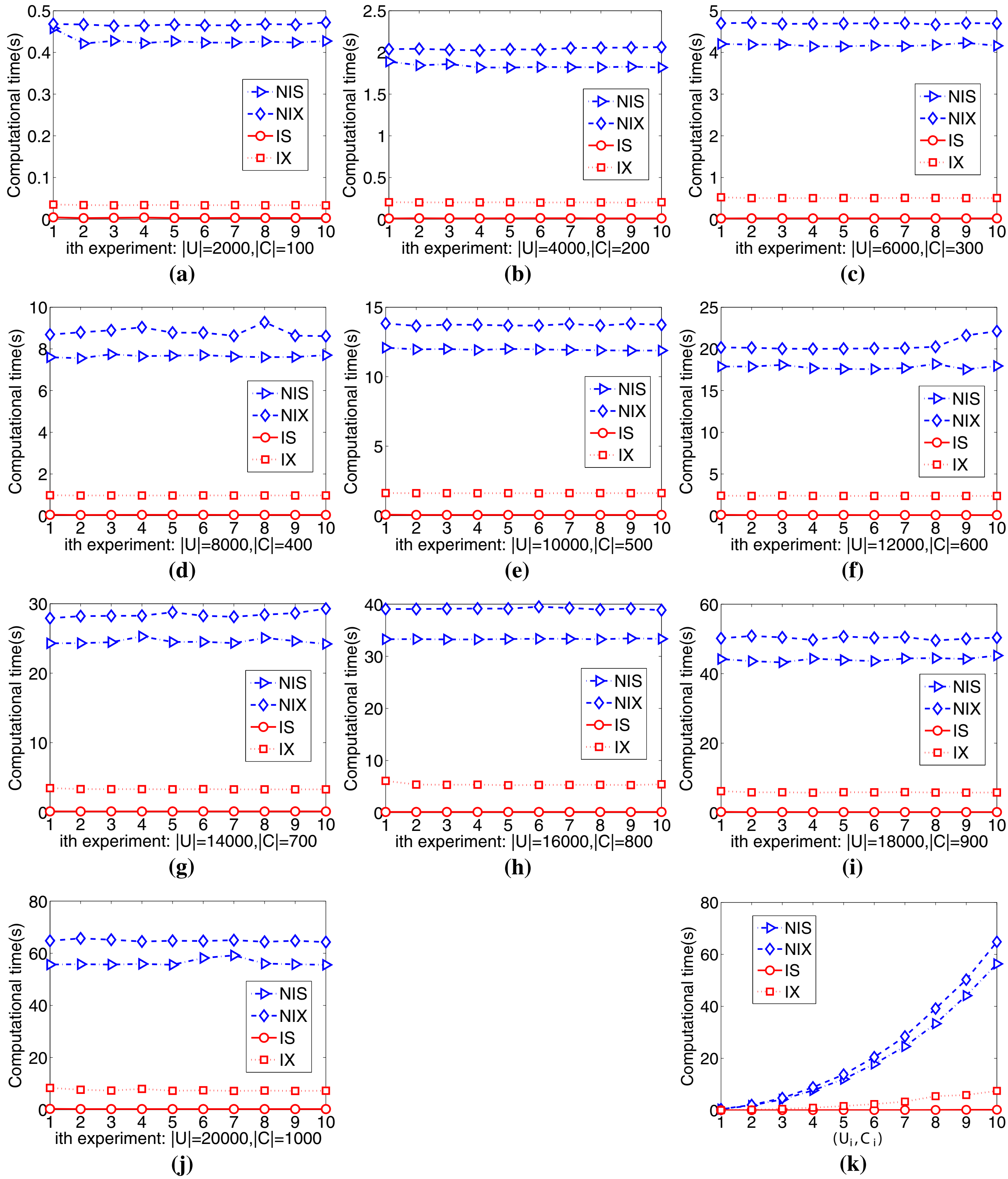

Fig. 1 Computational times using Algorithms 4.1–4.4 in $(U_i, \mathscr{C}_i)$

are far less than those of Algorithms 4.1 and 4.3, respectively. Therefore, the incremental algorithms are effective to construct approximations of sets in the dynamic covering approximation space $(U_i, \mathscr{C}_i^*)$, where $i = 1, 2, \ldots, 10$.

In Fig. 1, we observe that the average times of the incremental and non-incremental algorithms rise monotonically with the increasing cardinalities of object sets and coverings. The incremental algorithms perform always faster than the non-incremental algorithms in all experiments, and the average times of the incremental algorithms are much smaller than those of the non-incremental algorithms. Moreover, the speed-up ratios of times using the non-incremental algorithms are higher than the incremental algorithms with the increasing cardinalities of object sets and coverings. Especially, there exists little influence of the cardinalities of object sets and coverings on computing the second and sixth lower and upper approximations of sets using Algorithm 4.2 and 4.4. In other words, the incremental algorithms are effective to construct the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces when varying attribute values. All experimental results demonstrate Algorithms 4.2 and 4.4 are effective to compute the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces with varying attribute values.

6 Knowledge reduction of dynamic covering decision information systems caused by variations of attribute values

In this section, we employ examples to illustrate how to compute the type-1 and type-2 reducts of dynamic covering decision information systems caused by variations of attribute values.

Example 6.1 Let $(U, \mathscr{D} \cup U/d)$ be a covering decision information system, where $\mathscr{D} = \{\mathscr{C}_1, \mathscr{C}_2, \mathscr{C}_3, \mathscr{C}_4\}$, $\mathscr{C}_1 = \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}$, $\mathscr{C}_2 = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}\}$, $\mathscr{C}_3 = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\}$, $\mathscr{C}_4 = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}$, and $U/d = \{D_1, D_2\}$, where $D_1 = \{x_1, x_2\}$ and $D_2 = \{x_3, x_4, x_5\}$. According to Definition 2.3, we have

$$M_{\mathscr{D}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By Definition 2.4, we obtain

$$\Gamma(\mathscr{D}) = M_{\mathscr{D}} \bullet M_{\mathscr{D}}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

According to Definition 2.5, we get

$$\mathcal{X}_{SH(D_1)} = \Gamma(\mathscr{D}) \bullet \mathcal{X}_{D_1} = [1 \quad 1 \quad 1 \quad 1 \quad 1]^T,$$
$$\mathcal{X}_{SL(D_1)} = \Gamma(\mathscr{D}) \odot \mathcal{X}_{D_1} = [0 \quad 0 \quad 0 \quad 0 \quad 0]^T,$$
$$\mathcal{X}_{SH(D_2)} = \Gamma(\mathscr{D}) \bullet \mathcal{X}_{D_2} = [1 \quad 1 \quad 1 \quad 1 \quad 1]^T,$$
$$\mathcal{X}_{SL(D_2)} = \Gamma(\mathscr{D}) \odot \mathcal{X}_{D_2} = [0 \quad 0 \quad 0 \quad 0 \quad 0]^T.$$

By Definition 2.4, we obtain

$$\Gamma(\mathscr{D}/\mathscr{C}_4) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}/\{\mathscr{C}_2, \mathscr{C}_4\}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}/\{\mathscr{C}_1, \mathscr{C}_2, \mathscr{C}_4\}) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}/\{\mathscr{C}_2, \mathscr{C}_3, \mathscr{C}_4\}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

According to Definition 2.5, we have

$$\begin{aligned}
\Gamma(\mathscr{D}/\mathscr{C}_4)\bullet\mathcal{X}_{D_1}&=\mathcal{X}_{SH(D_1)},\Gamma(\mathscr{D}/\mathscr{C}_4)\odot\mathcal{X}_{D_1}=\mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}/\mathscr{C}_4)\bullet\mathcal{X}_{D_2}&=\mathcal{X}_{SH(D_2)},\Gamma(\mathscr{D}/\mathscr{C}_4)\odot\mathcal{X}_{D_2}=\mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1}&=\mathcal{X}_{SH(D_1)},\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_1}=\mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2}&=\mathcal{X}_{SH(D_2)},\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_2}=\mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1}&\neq\mathcal{X}_{SH(D_1)},\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_1}=\mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2}&=\mathcal{X}_{SH(D_2)},\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_2}\neq\mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1}&\neq\mathcal{X}_{SH(D_1)},\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\odot\mathcal{X}_{D_1}=\mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2}&=\mathcal{X}_{SH(D_2)},\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\odot\mathcal{X}_{D_2}\neq\mathcal{X}_{SL(D_2)}.
\end{aligned}$$

By Definition 2.6, we get $\{\mathscr{C}_1,\mathscr{C}_3\}$ is a type-1 reduct of $(U,\mathscr{D}\cup U/d)$. Similarly, we can obtain a type-2 reduct of $(U,\mathscr{D}\cup U/d)$.

We employ the following example to illustrate how to compute a type-1 reduct of dynamic covering decision information system.

Example 6.2 (Continued from Example 6.1) Let $(U,\mathscr{D}^*\cup U/d)$ be a dynamic covering decision information system, where $\mathscr{D}^*=\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\}$, $\mathscr{C}_1^*=\{\{x_1,x_2,x_3,x_4\},\{x_5\}\}$, $\mathscr{C}_2^*=\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$, $\mathscr{C}_3^*=\{\{x_1,x_2,x_3,x_5\},\{x_4\}\}$, and $\mathscr{C}_4^*=\{\{x_1,x_2\},\{x_3,x_4\},\{x_5\}\}$. According to Definition 2.3, we have

$$M_{\mathscr{D}^*}=\begin{bmatrix}1&0&1&0&1&0&1&0&0\\1&0&1&0&1&0&1&0&0\\1&0&0&1&1&0&0&1&0\\1&0&0&1&0&1&0&1&0\\0&1&0&1&1&0&0&0&1\end{bmatrix}.$$

Furthermore, we obtain

$$\Delta\Gamma(\mathscr{D})=\begin{bmatrix}0&0&d_{13}^*&0&0\\0&0&d_{23}^*&0&0\\d_{31}^*&d_{32}^*&d_{33}^*&d_{34}^*&d_{35}^*\\0&0&d_{43}^*&0&0\\0&0&d_{53}^*&0&0\end{bmatrix}=\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}.$$

By Theorem 3.2, we have

$$\Gamma(\mathscr{D}^*)=\Gamma(\mathscr{D})+\Delta\Gamma(\mathscr{D})=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}.$$

By Definition 2.5, we get

$$\begin{aligned}
\mathcal{X}_{SH(D_1)}&=\Gamma(\mathscr{D}^*)\bullet\mathcal{X}_{D_1}=[1\quad 1\quad 1\quad 1\quad 1]^T,\\
\mathcal{X}_{SL(D_1)}&=\Gamma(\mathscr{D}^*)\odot\mathcal{X}_{D_1}=[0\quad 0\quad 0\quad 0\quad 0]^T,\\
\mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D}^*)\bullet\mathcal{X}_{D_2}=[1\quad 1\quad 1\quad 1\quad 1]^T,\\
\mathcal{X}_{SL(D_2)}&=\Gamma(\mathscr{D}^*)\odot\mathcal{X}_{D_2}=[0\quad 0\quad 0\quad 0\quad 0]^T.
\end{aligned}$$

Moreover, we obtain

$$\Gamma(\mathscr{D}^*/\mathscr{C}_4^*)=\Gamma(\mathscr{D}/\mathscr{C}_4)+\Delta\Gamma(\mathscr{D}/\mathscr{C}_4)=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix},$$

$$\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_4^*\})=\Gamma(\mathscr{D}/\mathscr{C}_4)+\Delta\Gamma(\mathscr{D}/\mathscr{C}_4)=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&0\\1&1&1&0&1\end{bmatrix},$$

$$\Gamma(\mathscr{D}^*/\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_4^*\})=\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})+\Delta\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})$$
$$=\begin{bmatrix}1&1&1&0&1\\1&1&1&0&1\\1&1&1&0&1\\0&0&0&1&0\\1&1&1&0&1\end{bmatrix},$$

$$\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\})=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})+\Delta\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})$$
$$=\begin{bmatrix}1&1&1&1&0\\1&1&1&1&0\\1&1&1&1&0\\1&1&1&1&0\\0&0&0&0&1\end{bmatrix}.$$

According to Definition 2.5, we have

$$\begin{aligned}
\Gamma(\mathscr{D}^*/\mathscr{C}_4^*) \bullet \mathcal{X}_{D_1} &= \mathcal{X}_{SH(D_1)}, & \Gamma(\mathscr{D}^*/\mathscr{C}_4^*) \odot \mathcal{X}_{D_1} &= \mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}^*/\mathscr{C}_4^*) \bullet \mathcal{X}_{D_2} &= \mathcal{X}_{SH(D_2)}, & \Gamma(\mathscr{D}^*/\mathscr{C}_4^*) \odot \mathcal{X}_{D_2} &= \mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_1} &= \mathcal{X}_{SH(D_1)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_1} &= \mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_2} &= \mathcal{X}_{SH(D_2)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_2} &= \mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_1} &\neq \mathcal{X}_{SH(D_1)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_1} &= \mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_2} &= \mathcal{X}_{SH(D_2)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_1^*,\mathscr{C}_2^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_2} &\neq \mathcal{X}_{SL(D_2)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_1} &\neq \mathcal{X}_{SH(D_1)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_1} &= \mathcal{X}_{SL(D_1)},\\
\Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\}) \bullet \mathcal{X}_{D_2} &= \mathcal{X}_{SH(D_2)}, & \Gamma(\mathscr{D}^*/\{\mathscr{C}_2^*,\mathscr{C}_3^*,\mathscr{C}_4^*\}) \odot \mathcal{X}_{D_2} &\neq \mathcal{X}_{SL(D_2)}.
\end{aligned}$$

By Definition 2.6, we have $\{\mathscr{C}_1^*, \mathscr{C}_3^*\}$ is the type-1 reduct of $(U, \mathscr{D}^* \cup U/d)$.

In Example 6.2, there is no need to compute all elements in the type-1 characteristic matrices using the incremental approach, and it illustrates the incremental approach is effective to compute the type-1 reducts of dynamic covering decision information systems caused by variations of attribute values. Similarly, we can obtain a type-2 reduct of dynamic covering decision information systems caused by variations of attribute values.

7 Conclusions

In this paper, we have introduced the incremental approaches for computing the type-1 and type-2 characteristic matrices in dynamic covering approximation spaces caused by revising attribute values. We have presented the non-incremental and incremental algorithms for computing the second and sixth lower and upper approximations of sets and compared the computational complexities of the non-incremental algorithms with those of incremental algorithms. We have applied the incremental algorithms to compute the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. Experimental results have illustrated the incremental approaches are effective to compute the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. We have demonstrated how to conduct knowledge reduction of dynamic covering decision information systems with the incremental approaches.

In the future, we will improve the effectiveness of Algorithms 4.2 and 4.4 and test them in large-scale dynamic covering approximation spaces. Furthermore, we will present incremental approaches to computing the second and sixth lower and upper approximations of sets in complex dynamic covering approximation spaces and reducts of dynamic covering decision information systems caused by variation attribute values.

Acknowledgments We would like to thank the anonymous reviewers very much for their professional comments and valuable suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 11201490, 11371130, 11401052, 11401195, 11201137, 11526039), the Postdoctoral Science Foundation of China (NO. 2015M580353), the Scientific Research Fund of Hunan Provincial Education Department (No. 14C0049), the Planned Science and Technology Project of Hunan Province (No. 2015JC3055) and the China Scholarship Council (CSC).

References

1. Chen HM, Li TR, Qiao SJ, Ruan D (2010) A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values. Int J Intell Syst 25(10):1005–1026
2. Chen HM, Li TR, Ruan D (2012) Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining. Knowl Based Syst 31:140–161
3. Chen HM, Li TR, Ruan D, Lin JH, Hu CX (2013) A rough-set based incremental approach for updating approximations under dynamic maintenance environments. IEEE Trans Knowl Data Eng 25(2):174–184
4. Chen DG, Wang CZ, Hu QH (2007) A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. Inf Sci 177:3500–3518
5. Jia XY, Shang L, Zhou B, Yao YY (2016) Generalized attribute reduction in rough set theory. Knowl Based Syst 91:204–218
6. Kryszkiewicz M (2001) Comparative study of alternative types of knowledge reduction in inconsistent systems. Int J Intell Syst 16:105–120
7. Lang GM, Li QG, Cai MJ, Yang T (2015) Characteristic matrices-based knowledge reduction in dynamic covering decision information systems. Knowl Based Syst 85:1–26
8. Lang GM, Li QG, Guo LK (2015) Homomorphisms between covering approximation spaces. Fundamenta Informaticae 137:351–371
9. Lang GM, Li QG, Guo LK (2015) Homomorphisms-based attribute reduction of dynamic fuzzy covering information systems. Int J Gen Syst 44(7–8):791–811
10. Leung Y, Ma JM, Zhang WX, Li TJ (2008) Dependence-space-based attribute reductions in inconsistent decision information systems. Int J Approx Reason 49:623–630

11. Li SY, Li TR, Liu D (2013) Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set. Knowl Based Syst 40:17–26
12. Li SY, Li TR, Liu D (2013) Dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set. Int J Intell Syst 28(8):729–751
13. Li TR, Ruan D, Geert W, Song J, Xu Y (2007) A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowl Based Syst 20(5):485–494
14. Li TR, Ruan D, Song J (2007) Dynamic maintenance of decision rules with rough set under characteristic relation. In: International conference on wireless communications, networking and mobile computing, 2007, pp 3713–3716
15. Liang JY, Wang F, Dang CY, Qian YH (2014) A group incremental approach to feature selection applying rough set technique. IEEE Trans Knowl Data Eng 26(2):294–308
16. Lin GP, Qian YH, Li JJ (2012) NMGRS: neighborhood-based multigranulation rough sets. Int J Approx Reason 53(7):1080–1093
17. Lin GP, Liang JY, Qian YH (2013) Multigranulation rough sets: from partition to covering. Inf Sci 241:101–118
18. Liu GL (2015) Special types of coverings and axiomatization of rough sets based on partial orders. Knowl Based Syst 85:316–321
19. Liu GL, Zhu K (2014) The relationship among three types of rough approximation pairs. Knowl Based Syst 60:28–34
20. Liu D, Li TR, Ruan D, Zhang JB (2011) Incremental learning optimization on knowledge discovery in dynamic business intelligent systems. J Glob Optim 51(2):325–344
21. Liu D, Li TR, Ruan D, Zou WL (2009) An incremental approach for inducing knowledge from dynamic information systems. Fundamenta Informaticae 94(2):245–260
22. Liu D, Li TR, Zhang JB (2014) A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems. Int J Approx Reason 55(8):1764–1786
23. Liu D, Li TR, Zhang JB (2015) Incremental updating approximations in probabilistic rough sets under the variation of attributes. Knowl Based Syst 73:81–96
24. Liu D, Liang DC, Wang CC (2016) A novel three-way decision model based on incomplete information system. Knowl Based Syst 91:32–45
25. Liu CH, Wang MZ (2011) Covering fuzzy rough set based on multi-granulations. In: International conference on uncertainty reasoning and knowledge engineering, pp 146–149
26. Liu CH, Miao DQ (2011) Covering rough set model based on multigranulations. In: Proceedings of the 13th international conference on rough sets, fuzzy sets, data mining and granular computing, LNCS (LNAI) 6743, pp 87–90
27. Lu SX, Wang XZ, Zhang GQ, Zhou X (2015) Effective algorithms of the moore-penrose inverse matrices for extreme learning machine. Intell Data Anal 19(4):743–760
28. Luo C, Li TR, Chen HM (2014) Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization. Inf Sci 257:210–228
29. Luo C, Li TR, Chen HM, Liu D (2013) Incremental approaches for updating approximations in set-valued ordered information systems. Knowl Based Syst 50:218–233
30. Luo C, Li TR, Chen HM, Lu LX (2015) Fast algorithms for computing rough approximations in set-valued decision systems while updating criteria values. Inf Sci 299:221–242
31. Miao DQ, Zhao Y, Yao YY, Li HX, Xu FF (2009) Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Inf Sci 179:4140–4150
32. Pomykala JA (1987) Approximation operations in approximation space. Bull Pol Acad Sci 35:653–662
33. Qian YH, Liang JY, Li DY, Wang F, Ma NN (2010) Approximation reduction in inconsistent incomplete decision tables. Knowl Based Syst 23:427–433
34. Qian WB, Shu WH, Xie YH (2015) Feature selection using compact discernibility matrix-based approach in dynamic incomplete decision systems. J Inf Sci Eng 31(2):509–527
35. Samanta P, Chakraborty MK (2009) Covering based approaches to rough sets and implication lattices. In: Proceedings of the 12th international conference on tough sets, fuzzy sets, data mining and granular computing, LNCS(LNAI) 5908, pp 127–134
36. Shan N, Ziarko W (1995) Data-based acquisition and incremental modification of classification rules. Comput Intell 11(2):357–370
37. Sang YL, Liang JY, Qian YH (2016) Decision-theoretic rough sets under dynamic granulation. Knowl Based Syst 91:84–92
38. Shu WH, Qian WB (2015) An incremental approach to attribute reduction from dynamic incomplete decision systems in rough set theory. Data Knowl Eng 100:116–132
39. Shu WH, Shen H (2013) Updating attribute reduction in incomplete decision systems with the variation of attribute set. Int J Approx Reason 55(3):867–884
40. Shu WH, Shen H (2014) Incremental feature selection based on rough set in dynamic incomplete data. Pattern Recognit 47(12):3890–3906
41. Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. Intell Decis Support 11:331–362
42. Slezak D (2000) Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44:291–319
43. Tsang E, Cheng D, Lee J, Yeung D (2004) On the upper approximations of covering generalized rough sets. In: Proceedings of the 3rd international conference machine learning and cybernetics, pp 4200–4203
44. Tan AH, Li JJ, Lin YJ, Lin GP (2015) Matrix-based set approximations and reductions in covering decision information systems. Int J Approx Reason 59:68–80
45. Tan AH, Li JJ, Lin GP, Lin YJ (2015) Fast approach to knowledge acquisition in covering information systems using matrix operations. Knowl Based Syst 79:90–98
46. Wang XZ (2015) Uncertainty in learning from big data-editorial. J Intell Fuzzy Syst 28(5):2329–2330
47. Wang XZ, Aamir R, Fu AM (2015) Fuzziness based sample categorization for classifier performance improvement. J Intell Fuzzy Syst 29:1185–1196
48. Wang XZ, He Q, Chen DG, Yeung D (2005) A genetic algorithm for solving the inverse problem of support vector machines. Neurocomputing 68:225–238
49. Wang XZ, Hong JR (1998) On the handling of fuzziness for continuous-valued attributes in decision tree generation. Fuzzy Sets Syst 99(3):283–290
50. Wang CZ, He Q, Chen DG, Hu QH (2014) A novel method for attribute reduction of covering decision systems. Inf Sci 254:181–196
51. Wang F, Liang JY, Dang CY (2013) Attribute reduction for dynamic data sets. Appl Soft Comput 13:676–689
52. Wang F, Liang JY, Qian YH (2013) Attribute reduction: a dimension incremental strategy. Knowl Based Syst 39:95–108
53. Wang CZ, Shao MW, Sun BQ, Hu QH (2015) An improved attribute reduction scheme with covering based rough sets. Appl Soft Comput 26:235–243
54. Wang XZ, Xing HJ, Li YH, Hua Q, Dong CR, Pedrycz W (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst 23(5):1638–1654
55. Wang SP, Zhu W, Zhu QX, Min F (2014) Characteristic matrix of covering and its application to Boolean matrice decomposition. Inf Sci 263(1):186–197

56. Xu WH, Zhang WX (2007) Measuring roughness of generalized rough sets induced by a covering. Fuzzy Sets Syst 158:2443–2455
57. Yang T, Li QG (2010) Reduction about approximation spaces of covering generalized rough sets. Int J Approx Reason 51(3):335–345
58. Yang XB, Zhang M, Dou HL, Yang JY (2011) Neighborhood systems-based rough sets in incomplete information system. Knowl Based Syst 24(6):858–867
59. Yang XB, Qi Y, Yu HL, Song XN, Yang JY (2014) Updating multigranulation rough approximations with increasing of granular structures. Knowl Based Syst 64:59–69
60. Yao YY, Yao BX (2012) Covering based rough set approximations. Inf Sci 200:91–107
61. Yao YY (1998) Relational interpretations of neighborhood operators and rough set approximation operators. Inf Sci 101:239–259
62. Yao YY (2003) On generalizing rough set theory. In: Proceedings of the 9th international conference on rough sets, fuzzy sets, data mining and granular computing, LNCS(LNAI) 2639, pp 44–51
63. Yao YY, Zhao Y (2009) Discernibility matrix simplification for constructing attribute reducts. Inf Sci 179(7):867–882
64. Zakowski W (1983) Approximations in the space (u, π). Demonstr Math 16:761–769
65. Zhang JB, Li TR, Chen HM (2014) Composite rough sets for dynamic data mining. Inf Sci 257:81–100
66. Zhang JB, Li TR, Ruan D, Liu D (2012) Rough sets based matric approaches with dynamic attribute variation in set-valued information systems. Int J Approx Reason 53(4):620–635
67. Zhang JB, Li TR, Ruan D, Liu D (2012) Neighborhood rough sets for dynamic data mining. Int J Intell Syst 27(4):317–342
68. Zhang YL, Li JJ, Wu WZ (2010) On axiomatic characterizations of three pairs of covering based approximation operators. Inf Sci 180(2):274–287
69. Zhang WX, Mi JS, Wu WZ (2003) Approaches to knowledge reductions in inconsistent systems. Int J Intell Syst 18(9):989–1000
70. Zhu W, Wang FY (2007) On three types of covering rough sets. IEEE Trans Knowl Data Eng 19(8):1131–1144
71. Zhu P (2011) Covering rough sets based on neighborhoods: an approach without using neighborhoods. Int J Approx Reason 52(3):461–472
72. Zhu W (2007) Topological approaches to covering rough sets. Inf Sci 177(6):1499–1508
73. Zhu W (2009) Relationship among basic concepts in covering-based rough sets. Inf Sci 179(14):2478–2486
74. Zhu W (2009) Relationship between generalized rough sets based on binary relation and coverings. Inf Sci 179(3):210–225
75. Zhu W, Wang FY (2006) A new type of covering rough sets. In: Proceedings of the 3rd international IEEE conference on intelligent systems, pp 444–449