Matrix-based Dynamic Updating Rough Fuzzy Approximations for Data Mining☆

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

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

Abstract

In a dynamic environment, the data collected from real applications varies not only with the amount of objects but also with the number of features, which will result in continuous change of knowledge over time. The static methods of updating knowledge need to recompute from scratch when new data are added every time. This makes it potentially very time-consuming to update knowledge, especially as the dataset grows dramatically. Calculation of approximations is one of main mining tasks in rough set theory, like frequent pattern mining in association rules. Considering the fuzzy descriptions of decision states in the universe under fuzzy environment, this paper aims to provide an efficient approach for computing rough approximations of fuzzy concepts in dynamic fuzzy decision systems (FDS) with simultaneous variation of objects and features. We firstly present a matrix-based representation of rough fuzzy approximations by a Boolean matrix associated with a matrix operator in FDS. While adding the objects and features concurrently, incremental mechanisms for updating rough fuzzy approximations are introduced, and the corresponding matrix-based dynamic algorithm is developed. Unlike the static method of computing approximations by updating the whole relation matrix, our new approach partitions it into sub-matrices and updates each sub-matrix locally by utilizing the previous matrix information and the interactive information of each sub-matrix to avoid unnecessary calculations. Experimental results on six UCI datasets shown that the proposed dynamic algorithm achieves significantly higher efficiency than the static algorithm and the combination of two reference incremental algorithms.

Keywords: Rough fuzzy set, Incremental learning, Matrix, Rough approximations.

1. Introduction

Rough Set Theory (RST) proposed by Pawlak in 1982 [1] is an efficient tool for mining knowledge from the data with uncertainty and imprecision information. Since RST based data analysis does not need any extra information about data, knowledge discovered from the data will be more objective. Nowadays, RST has been successfully applied in many fields, such as artificial intelligence [2, 3], data mining [4, 5], intelligent information processing [6, 7] and so forth.

Although the Pawlak's RST is an effective tool for dealing with the data in which the condition attributes are symbolic and decision attributes are crisp, it is difficult to process the data with real attribute values or the fuzzy decision values, which exist in many real applications, such as the disease diagnosis data [8], spacial data [9], microarray data [10]. Rough fuzzy set and fuzzy rough set were presented by Dubois et al. [11] to deal with the coarseness and fuzziness in a fuzzy environment [12, 13]. Due to the advantage of integrating two uncertainties (roughness and vagueness), these two models have been widely applied for various applications (*e.g.*, attribute reduction [14], rule induction [15],

☆This is an extended version of the paper presented at the 2015 International Joint Conference on Rough Sets (IJCRS 2015), Tianjin 2015, China.

*Corresponding author.

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

formal concept analysis [16], clustering [17], robust classifies [18], etc). When the condition attributes are nominal and decision attributes are fuzzy, rough fuzzy set depicts the fuzzy concept by lower and upper approximations in a crisp approximation space. Yang et al. extended rough fuzzy set to deal with interval-valued data based on the α-dominance relation and investigated the corresponding algorithms of attribute reduction and rule induction [19]. Sun et al. constructed the decision-theoretic rough fuzzy set by combining the probability and fuzziness in a fuzzy decision system (FDS) and proposed an approach for selecting probability parameters based on decision-making risk [20]. Li et al. integrated rough fuzzy set with two universes of discourse based on covering, tolerance, dominance and equivalence relations, respectively [21]. Huang et al. combined rough set and fuzzy set for discovering the inherent relationships among documents with different languages [22]. Petrosino et al. developed an image compression algorithm by coding and decoding the image in terms of rough fuzzy approximations [23].

In real-life applications, the data are often not static, but evolve over time. The characteristics of the evolving data can be simply summarized as three scenarios, *i.e.*, the objects are inserted or removed, the attributes are added or deleted and the attribute values are revised. For example, in an electronic health records system, new patients' records are added or outdated records are deleted, new disease features (attributes) become available due to the appearance of new medical devices or irrelevant disease features are removed, and the feature values may be revised because of the incorrect inputs. Correspondingly, dynamically updating the data will result in the changes of knowledge discovered from data. Traditional static methods retrain the whole model on the entire updated data, which make it too time-consuming to immediate decision making or predicting, etc. Incremental learning is an efficient method to improve the effectiveness of data mining models and algorithms by means of the previous accumulated knowledge and the newly updated data [24]. It has been widely employed in RST under the dynamic environment with three different data updating scenarios [25, 26, 27]. With the variation of objects, based on information entropy, Liang et al. presented an incremental attribute reduction approach with the insertion of a group objects [28]. Huang et al. proposed an incremental rule induction algorithm which can guarantee that the extracted rules were complete and no duplicate [29]. Zeng et al. investigated the incremental mechanisms of computing rough fuzzy approximations [30]. With the variation of attributes, Wang et al. presented an incremental feature selection method based on three different entropy measures [31]. Chen et al. presented two incremental methods for computing rough fuzzy approximations based on the boundary set and the cut set, respectively [32]. Yang et al. investigated an incremental approach for computing multigranulation rough approximations [33]. With the change of attribute values, Luo et al. developed a dynamic approach based on matrix for updating rough approximations in the set-valued decision systems [34]. Cai et al. designed a fast attribute reduction algorithm in the covering decision information systems [35]. However, the data may vary in the form of multi-dimensions in real-life situations, *i.e.*, objects, attributes and attribute values will vary simultaneously. Chen et al. investigated the incremental updating approximations based on decision-theoretic rough set when both the objects and attributes increase over time [36]. But the approach suffers the limitation of handling the fuzzy set. As the fuzzy information universally exist in the real applications, we investigate the incremental mechanisms of rough approximations with respect to the fuzzy concept set under the simultaneous change of objects and attributes in this paper.

Matrix is advantageous in that it is intuitional and simple for knowledge representation and reasoning in RST [37, 38, 39]. Wang et al. presented characteristic and Boolean matrices for illustrating covering approximations [40]. Zhang et al. developed a parallel method of computing composite rough approximation based on Boolean matrices [41]. Ma presented the matrix presentations of approximations of two fuzzy covering rough set models [42]. However, these matrix approaches could not be directly utilized for the computation of approximations in rough fuzzy set model. To address this limitation, we present a novel matrix operation for the construction of rough fuzzy approximations, and further develop incremental mechanisms based on matrix for maintenance of approximations when objects and attributes are added simultaneously in FDS. Specifically, the whole relation matrix is divided into four parts for updating each sub-matrix conveniently. Each main diagonal block matrix is partly updated according to the previous matrix information. The counter-diagonal matrices are updated by the interactive information of two main diagonal matrices and the related properties of relation matrix. Finally, experimental results on six UCI data sets show that the proposed dynamic algorithm can achieve better performance than the static algorithm and the combined algorithm by integrating two reference incremental algorithms with the single-dimensional variation of FDS.

The remainder of this paper is organized as follows. Section 2 introduces some basic concepts of FDS and rough fuzzy set model. Section 3 presents a matrix-based method for constructing rough fuzzy approximations. Section 4 presents incremental mechanisms for updating rough fuzzy approximations when the objects and attributes vary simultaneously, and an illustrative example is employed to show the effectiveness of the proposed method. Section 5 develops and analyzes the static and dynamic algorithms when the objects and attributes are added simultaneously. In Section 6, comparative experiments are designed for validating the efficiency of the proposed dynamic algorithm. Finally, the paper ends with conclusions and further research topics in Section 7.

2. Preliminaries

In this section, we will introduce the basic concepts of FDS and rough fuzzy set [1, 11].

Definition 1. *An FDS is 4-tuple* $S = \langle U, C \cup D, V, f \rangle$*, where* $U = \{x_i | i \in \{1, 2, \ldots, n\}\}$ *is a non-empty finite set of objects, called the universe; C is a non-empty finite set of condition attributes and D is a non-empty finite set of fuzzy decision attributes,* $C \cap D = \emptyset$*;* $V = V_C \cup V_D$*, where V is the domain of all attributes,* V_C *is the domain of condition attributes and* V_D *is the domain of decision attributes; f is an information function from* $U \times (C \cup D)$ *to V such that* $f : U \times C \to V_C$*,* $f : U \times D \to [0, 1]$.

The rough fuzzy set model was presented by Dubois and Prade to process the fuzzy concepts in a crisp approximation space [11].

Definition 2. *Let* $S = \langle U, C \cup D, V, f \rangle$ *be an FDS and* $A \subseteq C$*.* $\widetilde{d}$ *is a fuzzy subset on D, where* $\widetilde{d}(x)$ $(x \in U)$ *denotes the degree of membership with respect to x in* $\widetilde{d}$*. The lower and upper approximations of* $\widetilde{d}$ *are a pair of fuzzy sets on D in terms of the equivalence relation* R_A*, and their membership functions are defined as follows:*

$$\begin{aligned} \underline{R_A}\widetilde{d}(x) &= inf\{\widetilde{d}(y) | y \in [x]_{R_A}\} \\ \overline{R_A}\widetilde{d}(x) &= sup\{\widetilde{d}(y) | y \in [x]_{R_A}\} \end{aligned} \tag{1}$$

where $R_A = \{(x, y) \in U \times U | f(x, a) = f(y, a), \forall a \in A\}$*,* $[x]_{R_A} = \{y \in U | x R_A y\}$ *denotes the equivalence class of x.*

Example 1. *Table 1 illustrates a medical diagnosis FDS,* $S = \langle U, C \cup D, V, f \rangle$*, where* $U = \{x_i | i \in \{1, 2, \ldots, 10\}\}$ *denotes the patients, the condition attributes set* $C = \{headache, muscle\ pain, sore\ throat, temperature\} = \{c_1, c_2, c_3, c_4\}$*, the fuzzy decision attributes set* $D = \{Flu, No\ Flu\}$*. The domain* $V_{c_1} = \{no, moderate, heavy\} \triangleq \{0, 1, 2\}$*,* $V_{c_2} = V_{c_3} = V_{c_4} = \{no, yes\} \triangleq \{0, 1\}$*.*

Set $A = \{c_1, c_2\} \subset C$*. The universe U can be partitioned according to the equivalence relation* R_A*:* $U/R_A = \{\{x_1, x_3, x_4\}, \{x_2, x_5, x_7, x_9, x_{10}\}, \{x_6, x_8\}\}$*. Let* $\tilde{d}$ *denote the Flu. Then the degrees of membership can be computed according to Definition 2:*

$$\begin{aligned} &\underline{R_A}\widetilde{d}(x_1) = \underline{R_A}\widetilde{d}(x_3) = \underline{R_A}\widetilde{d}(x_4) = 0.8 \wedge 1 \wedge 0.7 = 0.7; \\ &\underline{R_A}\widetilde{d}(x_2) = \underline{R_A}\widetilde{d}(x_5) = \underline{R_A}\widetilde{d}(x_7) = \underline{R_A}\widetilde{d}(x_9) = \underline{R_A}\widetilde{d}(x_{10}) \\ &\qquad = 0.3 \wedge 0.1 \wedge 0.2 \wedge 0.4 \wedge 0.2 = 0.1; \\ &\underline{R_A}\widetilde{d}(x_6) = \underline{R_A}\widetilde{d}(x_8) = 0.3 \wedge 0 = 0; \\ &\overline{R_A}\widetilde{d}(x_1) = \overline{R_A}\widetilde{d}(x_3) = \overline{R_A}\widetilde{d}(x_4) = 0.8 \vee 1 \vee 0.7 = 1; \\ &\overline{R_A}\widetilde{d}(x_2) = \overline{R_A}\widetilde{d}(x_5) = \overline{R_A}\widetilde{d}(x_7) = \overline{R_A}\widetilde{d}(x_9) = \overline{R_A}\widetilde{d}(x_{10}) \\ &\qquad = 0.3 \vee 0.1 \vee 0.2 \vee 0.4 \vee 0.2 = 0.4; \\ &\underline{R_A}\widetilde{d}(x_6) = \underline{R_A}\widetilde{d}(x_8) = 0.3 \vee 0 = 0.3. \end{aligned}$$

Additionally, the lower and upper approximations of $\widetilde{d}$ *are as follows:*

$$\begin{aligned} \underline{R_A}\widetilde{d} &= \{\frac{0.7}{x_1}, \frac{0.1}{x_2}, \frac{0.7}{x_3}, \frac{0.7}{x_4}, \frac{0.1}{x_5}, \frac{0}{x_6}, \frac{0.1}{x_7}, \frac{0}{x_8}, \frac{0.1}{x_9}, \frac{0.1}{x_{10}}\}; \\ \overline{R_A}\widetilde{d} &= \{\frac{1}{x_1}, \frac{0.4}{x_2}, \frac{1}{x_3}, \frac{1}{x_4}, \frac{0.4}{x_5}, \frac{0.3}{x_6}, \frac{0.4}{x_7}, \frac{0.3}{x_8}, \frac{0.4}{x_9}, \frac{0.4}{x_{10}}\}. \end{aligned}$$

Table 1: A fuzzy decision table.

U	c_1	c_2	c_3	c_4	Fuzzy Decision Attribute	
					Flu	No Flu
x_1	2	1	0	1	0.8	0.3
x_2	1	0	0	1	0.3	0.5
x_3	2	1	0	1	1	0.1
x_4	2	1	0	1	0.7	0.2
x_5	1	0	0	1	0.1	0.8
x_6	0	1	1	0	0.3	0.7
x_7	1	0	0	1	0.2	1
x_8	0	1	1	0	0	0.9
x_9	1	0	1	0	0.4	0.6
x_{10}	1	0	1	0	0.2	0.7

3. A matrix-based representation of approximations in the FDS

Matrix is a powerful tool, which has been widely applied to attribute reduction, rule induction and approximation computing in RST [38, 43, 44, 45]. Liu et al. showed a matrix-based representation of classical rough approximations [46]. Zhang et al. designed some matrices for describing the composite rough approximations based on different relations in composite rough set [47]. Luo et al. presented the dominant and dominated matrices for characterizing the approximations in dominance-based rough set approach [48]. Tan et al. presented boolean, characteristic and neighbor matrices for computing the set approximations in covering-based rough set [49]. In this section, a novel matrix operation based on the relation matrix are firstly presented for constructing the lower and upper approximations in rough fuzzy set. Then we discuss several matrix-based properties, which will be employed to incrementally compute approximations in next section.

Definition 3. *[46] Let $S = \langle U, C \cup D, V, f\rangle$ be an FDS, where $U = \{x_i | i \in \{1, 2, \ldots, n\}\}$, $A \subseteq C$. The corresponding relation matrix of A is denoted as $M^A = (m^A_{ij})_{n\times n}$, where*

$$m^A_{ij} = \begin{cases} 1, & x_i \in [x_j]_{R_A} \\ 0, & otherwise \end{cases} \tag{2}$$

Proposition 1. *[46] $M^A = (m^A_{ij})_{n\times n}$ is a symmetric matrix, and $m^A_{ii} = 1 (i = 1, \cdots, n)$.*

Definition 4. *Let $S = \langle U, C \cup D, V, f\rangle$ be an FDS. The corresponding relation matrices of $A, B \subseteq C$ are denoted as $M^A = (m^A_{ij})_{n\times n}$ and $M^B = (m^B_{ij})_{n\times n}$, respectively. Then the dot operation between M^A and M^B is defined as follows.*

$$M^A \bullet M^B = (m^A_{ij} \cdot m^B_{ij})_{n\times n}, \tag{3}$$

where $\bullet$ is the dot product of two matrices.

Proposition 2. *Let $S = \langle U, C \cup D, V, f\rangle$ be an FDS. The corresponding relation matrices of $A, B \subseteq C$ are $M^A = (m^A_{ij})_{n\times n}$ and $M^B = (m^B_{ij})_{n\times n}$, respectively. Then the relation matrix $M^{A\cup B}$ of $A \cup B$ equals to $M^A \bullet M^B$.*

Proof. If $m^{A\cup B}_{ij} = 1$, according to Definition 3, it follows $x_i \in [x_j]_{R_{A\cup B}}$. Then $x_i \in [x_j]_{R_A}$ and $x_i \in [x_j]_{R_B}$, i.e., $m^A_{ij} = 1$ and $m^B_{ij} = 1$. Therefore, we have $m^{A\cup B}_{ij} = 1 = m^A_{ij} \cdot m^B_{ij}$, and vice versa. If $m^{A\cup B}_{ij} = 0$, then $x_i \notin [x_j]_{R_{A\cup B}}$, that is, $x_i \notin [x_j]_{R_A}$ or $x_i \notin [x_j]_{R_B}$. Hence $m^A_{ij} = 0$ or $m^B_{ij} = 0$. Thus $m^{A\cup B}_{ij} = 0 = m^A_{ij} \cdot m^B_{ij}$, and vice versa.

Example 2. *(Continuation of Example 1) Let* $A = \{c_1, c_2\}$*,* $B = \{c_3, c_4\}$*. Then according to Definition 3, the relation matrices* M^A *and* M^B *can be calculated as follows.*

$$M^A = \begin{pmatrix} 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 1&0&1&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 0&0&0&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 0&0&0&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 0&1&0&0&1&0&1&0&1&1 \end{pmatrix} \quad M^B = \begin{pmatrix} 1&1&1&1&1&0&1&0&0&0\\ 1&1&1&1&1&0&1&0&0&0\\ 1&1&1&1&1&0&1&0&0&0\\ 1&1&1&1&1&0&1&0&0&0\\ 1&1&1&1&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&1&1\\ 1&1&1&1&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&1&1\\ 0&0&0&0&0&1&0&1&1&1\\ 0&0&0&0&0&1&0&1&1&1 \end{pmatrix}$$

According to Definition 4

$$M^A \bullet M^B = \begin{pmatrix} 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&1&1 \end{pmatrix}$$

It is easy to demonstrate that the relation matrix $M^{A\cup B} = M^A \bullet M^B$*.*

Definition 5. *Let* $S = \langle U, C \cup D, V, f\rangle$ *be an FDS, where* $U = \{x_i | i \in \{1, 2, \ldots, n\}\}$*.* $\widetilde{d}$ *is a fuzzy subset on D,* $\widetilde{d}(x)$ $(x \in U)$ *is the degree of membership of x in* $\widetilde{d}$ *and* $M^A = (m^A_{ij})_{n\times n}$ *is the relation matrix of the attribute set* $A \subseteq C$*.* $M^A \otimes_{max} \widetilde{d}$ *is a column vector, and the i-th element in this vector is defined as follows.*

$$(M^A \otimes_{max} \widetilde{d})(i) = max\{m^A_{i1} \cdot \widetilde{d}(x_1), m^A_{i2} \cdot \widetilde{d}(x_2), \ldots, m^A_{in} \cdot \widetilde{d}(x_n)\} \quad (i = 1, 2, \ldots, n) \tag{4}$$

where max operation takes the maximum value among n numbers and T denotes the transpose operation.

Theorem 1. *Let* $S = \langle U, C \cup D, V, f\rangle$ *be an FDS, where* $U = \{x_i | i \in \{1, 2, \ldots, n\}\}$*.* $\widetilde{d}$ *is a fuzzy subset on D.* M^A *is the relation matrix of the attribute set* $A \subseteq C$*. The upper and lower approximations of* $\widetilde{d}$ *are calculated as follows.*

$$\overline{R_A}\widetilde{d} = M^A \otimes_{max} \widetilde{d}; \tag{5}$$

$$\underline{R_A}\widetilde{d} = L - M^A \otimes_{max} \widetilde{d}^c \tag{6}$$

where L is the column vector that all elements are one and $\widetilde{d}^c$ *is the complement of* $\widetilde{d}$*.*

Proof. According to Definition 5, $\forall x_i \in U$, $(M^A \otimes_{max} \widetilde{d})(i) = max\{m^A_{i1} \cdot \widetilde{d}(x_1), m^A_{i2} \cdot \widetilde{d}(x_2), \ldots, m^A_{in} \cdot \widetilde{d}(x_n)\}$. If $m^A_{ij} = 1$, then $x_j \in [x_i]_{R_A}$, *i.e.*, $m^A_{ij} \cdot \widetilde{d}(x_j) = \widetilde{d}(x_j)$; otherwise $m^A_{ij} \cdot \widetilde{d}(x_j) = 0$. Therefore, we have $(M^A \otimes_{max} \widetilde{d})(i) = \overline{R_A}\widetilde{d}(x_i)$ based on Definition 2. In addition, according to $\underline{R_A}\widetilde{d} = \sim \overline{R_A}\widetilde{d}^c$ and Equation (5), it is clearly that $\underline{R_A}\widetilde{d}(x_i) = L - (M^A \otimes_{max} \widetilde{d}^c)(i)$.

Example 3. *Given an FDS* S=$\langle U, C \cup D, V, f\rangle$ *as shown in Table 1. Let* $A = \{c_1, c_2\}$ *and* $\widetilde{d}$ *be the Flu. Then* $\widetilde{d} = \{\frac{0.8}{x_1}, \frac{0.3}{x_2}, \frac{1}{x_3}, \frac{0.7}{x_4}, \frac{0.1}{x_5}, \frac{0.3}{x_6}, \frac{0.2}{x_7}, \frac{0}{x_8}, \frac{0.4}{x_9}, \frac{0.2}{x_{10}}\}$*. From the results of Example 2, we have*

$$\overline{R_A}\widetilde{d} = M^A \otimes_{max} \widetilde{d} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \otimes_{max} \begin{pmatrix} 0.8 \\ 0.3 \\ 1 \\ 0.7 \\ 0.1 \\ 0.3 \\ 0.2 \\ 0 \\ 0.4 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.4 \\ 1 \\ 1 \\ 0.4 \\ 0.3 \\ 0.4 \\ 0.3 \\ 0.4 \\ 0.4 \end{pmatrix}$$

$$\underline{R_A}\widetilde{d} = L - M^A \otimes_{max} \widetilde{d}^c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \otimes_{max} \begin{pmatrix} 0.2 \\ 0.7 \\ 0 \\ 0.3 \\ 0.9 \\ 0.7 \\ 0.8 \\ 1 \\ 0.6 \\ 0.8 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.1 \\ 0.7 \\ 0.7 \\ 0.1 \\ 0 \\ 0.1 \\ 0 \\ 0.1 \\ 0.1 \end{pmatrix}$$

4. Dynamically updating approximations under the variation of objects and attributes

In this section, an incremental method for computing approximations based on matrix is proposed in the FDS associated with the addition of attributes and objects simultaneously. As mentioned above, the key step for the computation of rough fuzzy approximations is how to update the relation matrix. If we can dynamically compute the updated relation matrix based an incremental updating strategy rather than reconstructing it from scratch, then the runtime will be reduced. In what follows, we discuss how to update the relation matrix incrementally while the attribute and object sets simultaneously vary over time.

Let $S^t = \langle U^t, C^t \cup D^t, V^t, f^t\rangle$ be an FDS at time t. $S^{t+1} = \langle U^{t+1}, C^{t+1} \cup D^{t+1}, V^{t+1}, f^{t+1}\rangle$ denotes the FDS at time $t+1$, where $U^{t+1} = U^t \cup \Delta U$, $C^{t+1} = C^t \cup \Delta C$, $D^{t+1} = D^t \cup \Delta D$. To incrementally compute the relation matrix, we partition the system S^{t+1} into two subsystems. One is $S^{\Delta U} = \langle \Delta U, C^{t+1} \cup \Delta D, V^{\Delta U}, f^{\Delta U}\rangle$ and the other is $S^{U^t} = \langle U^t, C^{t+1} \cup D^t, V^{U^t}, f^{U^t}\rangle$. Then the $S^{U^t} = \langle U^t, C^{t+1} \cup D^t, V^{U^t}, f^{U^t}\rangle$ is again partitioned into two subsystems: $S^t = \langle U^t, C^t \cup D^t, V^t, f^t\rangle$ and $S^{\Delta C} = \langle U^t, \Delta C \cup D^t, V^{\Delta C}, f^{\Delta C}\rangle$. Suppose $|\ U^{t+1} |= n'$, $|\ U^t |= n$, $|\ \Delta U |= n^+$, $|\ C^{t+1} |= m'$, $|\ C^t |= m$, $|\ \Delta C |= m^+$, then we have $n' = n + n^+$, $m' = m + m^+$.

Theorem 2. *Let* $M^{C^{t+1}}$ *denote the relation matrix of FDS* S^{t+1}*. Then* $M^{C^{t+1}}$ *can be partitioned four parts, i.e.,* $M^{C^{t+1}} = \left(\begin{array}{c:c} (M^{C^{t+1}}_{U_t})_{n\times n} & (M^{C^{t+1}}_{U_t,\Delta U})_{n\times n^+} \\ \hdashline (M^{C^{t+1}}_{U_t,\Delta U})^T_{n^+\times n} & (M^{C^{t+1}}_{\Delta U})_{n^+\times n^+} \end{array}\right)$*, where* $M^{C^{t+1}}_{U_t}$ *denotes the relation matrix of* U^t *under* C^{t+1}*,* $M^{C^{t+1}}_{U_t,\Delta U}$ *denotes the relation between* U^t *and* ΔU *under* C^{t+1} *and* $M^{C^{t+1}}_{\Delta U}$ *denotes the relation matrix of* ΔU *under* C^{t+1}*.*

Proof. According to Definition 3, it is easy to see that the relation matrix $M^{C^{t+1}}$ can be divided into four parts. Each part can be obtained according to Definition 3 directly.

In order to dynamically compute the relation matrix, we partition the relation matrix into four parts according to Theorem 2. Then the first part $(M_{U_t}^{C^{t+1}})_{n\times n}$ can be incrementally updated as follows.

Theorem 3. *Suppose* $M_{U_t}^{C^{t+1}} = (m_{ij}^{C^{t+1}})_{n\times n}$, $M_{U_t}^{C^t} = (m_{ij}^{C^t})_{n\times n}$, $M_{U_t}^{\Delta C} = (m_{ij}^{\Delta C})_{n\times n}$ *are the relation matrices with respect to* S^{U^t}, S^t, $S^{\Delta C}$, *respectively. The relation matrix* $M_{U_t}^{C^{t+1}}$ *can be updated as follows.*

(1) *if* $m_{ij}^{C^t} = 0$, *then* $m_{ij}^{C^{t+1}} = m_{ij}^{C^t}$;

(2) *if* $m_{ij}^{C^t} = 1$, *and* $m_{ij}^{\Delta C} = 1$, *then* $m_{ij}^{C^{t+1}} = m_{ij}^{C^t}$;

(3) *if* $m_{ij}^{C^t} = 1$, *and* $m_{ij}^{\Delta C} = 0$, *then* $m_{ij}^{C^{t+1}} = 0$.

Proof. It follows directly from Proposition 2.

By utilizing the accumulated matrix information $M_{U_t}^{C^t}$ and the newly added matrix information $M_{U_t}^{\Delta C}$, we compute the matrix $M_{U_t}^{C^{t+1}}$ only by updating the third scenario of Theorem 3 instead of recomputing the whole matrix, which can improve the efficiency of computing.

Theorem 4. *Given the relation matrices* $M_{\Delta U}^{C^{t+1}} = (m_{ij}^{\Delta U})_{n^+\times n^+}$ *and* $M_{U_t}^{C^{t+1}} = (m_{ij}^{C^{t+1}})_{n\times n}$. *Then* $M_{U_t,\Delta U}^{C^{t+1}} = (m_{ij}^{U_t,\Delta U})_{n\times n^+}$ *can be updated as follows.*

(1) *if* x_i *and* x_j *are equivalent under* C^{t+1}, *where* $x_i \in U^t$, $i \in \{1, 2, \ldots, n\}$, $x_j \in \Delta U$, $j \in \{n+1, n+2, \ldots, n+n^+\}$, *then* $m_{[i:]}^{U_t,\Delta U} = m_{[(j-n):]}^{\Delta U}$, $m_{[:(j-n)]}^{U_t,\Delta U} = m_{[:i]}^{C^{t+1}}$. *In addition, if* x_i *and* $x_{i'}$ *are equivalent under* C^{t+1}, x_j *and* $x_{j'}$ *are equivalent under* ΔC, *where* $i' \in \{1, 2, \ldots, n\}$, $j' \in \{n+1, n+2, \ldots, n+n^+\}$, *then* $m_{[i':]}^{U_t,\Delta U} = m_{[(j-n):]}^{\Delta U}$, $m_{[:(j'-n)]}^{U_t,\Delta U} = m_{[:i]}^{C^{t+1}}$.

(2) *if* x_i *and* x_j *do not satisfy the aforementioned conditions, then* $m_{i(j-n)}^{U_t,\Delta U} = 0$. *Furthermore, if* x_i *and* $x_{i'}$ *are equivalent under* C^{t+1}, x_j *and* $x_{j'}$ *are equivalent under* ΔC, *then* $m_{i'(j-n)}^{U_t,\Delta U} = m_{i(j'-n)}^{U_t,\Delta U} = m_{i'(j'-n)}^{U_t,\Delta U} = 0$.

where $m_{[i:]}^{U_t,\Delta U}$ *denotes the* i*th row in the matrix* $M_{U_t,\Delta U}^{C^{t+1}}$, $m_{[:j]}^{U_t,\Delta U}$ *denotes the* j*th column in the matrix* $M_{U_t,\Delta U}^{C^{t+1}}$.

Proof. If x_i and x_j are equivalent, then according to Theorem 2 and Propoisition 1, we have $m_{[i:]}^{U_t,\Delta U} = m_{[(j-n):]}^{\Delta U}$. In addition, according to Proposition 1, we have $m_{[:(j-n)]}^{U_t,\Delta U} = m_{[:i]}^{C^{t+1}}$. Furthermore, if x_i and $x_{i'}$, x_j and $x_{j'}$ are in the same equivalence class, respectively. Then we have $m_{[i':]}^{U_t,\Delta U} = m_{[i:]}^{U_t,\Delta U} = m_{[(j-n):]}^{\Delta U}$, $m_{[:(j'-n)]}^{U_t,\Delta U} = m_{[:(j-n)]}^{U_t,\Delta U} = m_{[:i]}^{C^{t+1}}$ according to Proposition 1. The proof of case 2 is analogous.

By the utilization of the symmetry property of the relation matrix and the previous updated matrix results, we can update the whole row or the whole column of the matrix $M_{U_t,\Delta U}^{C^{t+1}}$, not rather one by one, which can reduce the computing overhead.

In order to describe the mechanisms of incremental computing approximations more clearly, here is an example to introduce the process of computing rough fuzzy approximations.

Example 4. *Let* $S^t = \langle U^t, C^t \cup D^t, V^t, f^t\rangle$ *be an FDS at time t, where* $U = \{x_i | i \in \{1, 2, \ldots, 10\}\}$, $C^t = \{c_i, 1 \le i \le 4\}$ *(see Table 1). At the time* $t+1$, *the attribute set* $\Delta C = \{runny\ noses, cough\} = \{c_5, c_6\}$ *and the objects* $\Delta U = \{x_{11}, x_{12}, x_{13}\}$ *are added to* S^t, *where* $V_{c_5} = V_{c_6} = \{No, Yes\} = \{0, 1\}$ *(see Table 2).*

Firstly, the result $M_{U_t}^{C^t}$ *of Example 2 and the relation matrix* $M_{U_t}^{\Delta C}$ *can be obtained according to Definition 3.*

$$M_{U_t}^{C^t} = \begin{pmatrix} 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&0&1&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&1&1 \end{pmatrix} \quad M_{U_t}^{\Delta C} = \begin{pmatrix} 1&0&1&0&0&1&0&0&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 1&0&1&0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0&1&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 1&0&1&0&0&1&0&0&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 0&0&0&1&0&0&0&1&0&0\\ 0&1&0&0&1&0&1&0&1&1\\ 0&1&0&0&1&0&1&0&1&1 \end{pmatrix}$$

Table 2: A decision table with fuzzy decision attributes at time $t+1$.

U	c_1	c_2	c_3	c_4	c_5	c_6	Fuzzy Decision Attribute	
							Flu	No Flu
x_1	2	1	0	1	1	0	0.8	0.3
x_2	1	0	0	1	0	1	0.3	0.5
x_3	2	1	0	1	1	0	1	0.1
x_4	2	1	0	1	0	0	0.7	0.2
x_5	1	0	0	1	0	1	0.1	0.8
x_6	0	1	1	0	1	0	0.3	0.7
x_7	1	0	0	1	0	1	0.2	1
x_8	0	1	1	0	0	0	0	0.9
x_9	1	0	1	0	0	1	0.4	0.6
x_{10}	1	0	1	0	0	1	0.2	0.7
x_{11}	1	0	0	1	0	1	0.4	0.8
x_{12}	1	0	0	1	0	1	0.1	0.7
x_{13}	2	1	0	1	1	1	0.9	0.3

*Secondly, according to Proposition 1, we only compute the elements under the principal diagonal of the matrix $M_{U_t}^{C^{t+1}}$. We judge the elements which values are **"1"** under the principal diagonal of the matrix $M_{U_t}^{C^t}$ whether change or not according to Theorem 3. Then the matrix $M_{U_t}^{C^{t+1}}$ can be obtained.*

$$M_{U_t}^{C^{t+1}} = \begin{array}{c} \begin{matrix} 1 & \boldsymbol{2} & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \underline{\boldsymbol{0}} & 0 & \underline{\boldsymbol{0}} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \underline{\boldsymbol{0}} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

where three elements are changed under the principal diagonal of the matrix $M_{U_t}^{C^t}$.

Thirdly, the relation matrix $M_{\Delta U}^{C^{t+1}}$ can be obtained according to Definition 3.

$$M_{\Delta U}^{C^{t+1}} = \begin{matrix} \boldsymbol{1} \\ 2 \\ 3 \end{matrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then according to Theorem 4,

1. *Because x_2 and x_{11} are equivalent, then $m_{[2:]}^{U_t,\Delta U} = m_{[1:]}^{\Delta U}$, $m_{[:1]}^{U_t,\Delta U} = m_{[:2]}^{C^{t+1}}$.*

2. *According to* $m_{25}^{C^{t+1}} = 1$, $m_{27}^{C^{t+1}} = 1$, *we have* $m_{[5:]}^{U_t,\Delta U} = m_{[1:]}^{\Delta U}$, $m_{[7:]}^{U_t,\Delta U} = m_{[1:]}^{\Delta U}$, *and according to* $m_{11,12}^{\Delta U} = 1$, *we have* $m_{[:2]}^{U_t,\Delta U} = m_{[:2]}^{C^{t+1}}$.
3. *Because* x_2 *and* x_{13} *are not equivalent, then* $m_{[13]}^{U_t,\Delta U} = 0$. *The others are the same.*

Therefore, we obtain the relation matrix

$$M_{U_t,\Delta U}^{C^{t+1}} = \begin{array}{c} \\ 1 \\ \boldsymbol{2} \\ 3 \\ 4 \\ \boldsymbol{5} \\ 6 \\ \boldsymbol{7} \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{1} & \boldsymbol{2} & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

where the first and second columns are the same to the second column of $M_{U_t}^{C^{t+1}}$, *and the second, fifth and seventh rows are the same to the first row of* $M_{\Delta U}^{C^{t+1}}$. *It apparently reduces the computing time than reconstructing the entire matrix.*

Lastly, according to Theorems 1 and 2, we have the upper and lower approximations of $\widetilde{d}^{t+1}$.

$$\underline{R_{C_{t+1}}}\widetilde{d}^{t+1} = \{\frac{0.8}{x_1}, \frac{0.1}{x_2}, \frac{0.8}{x_3}, \frac{0.7}{x_4}, \frac{0.1}{x_5}, \frac{0.3}{x_6}, \frac{0.1}{x_7}, \frac{0}{x_8}, \frac{0.2}{x_9}, \frac{0.2}{x_{10}}, \frac{0.1}{x_{11}}, \frac{0.1}{x_{12}}, \frac{0.9}{x_{13}}\}$$
$$\overline{R_{C_{t+1}}}\widetilde{d}^{t+1} = \{\frac{1}{x_1}, \frac{0.4}{x_2}, \frac{1}{x_3}, \frac{0.7}{x_4}, \frac{0.4}{x_5}, \frac{0.3}{x_6}, \frac{0.4}{x_7}, \frac{0}{x_8}, \frac{0.4}{x_9}, \frac{0.4}{x_{10}}, \frac{0.4}{x_{11}}, \frac{0.4}{x_{12}}, \frac{0.9}{x_{13}}\}$$

5. Dynamic and static algorithms based on the matrix for computing approximations under the addition of objects and attributes

According to the incremental mechanisms for computing approximations while objects and attributes increase over time in FDS, the static and dynamic algorithms are developed and analyzed in this section, respectively.

Let $S^t = \langle U^t, C^t \cup D^t, V^t, f^t \rangle$ be an FDS at time t. The attribute sets ΔC and ΔD and the object set ΔU are added into S^t simultaneously at time $t+1$. Suppose $|U^t| = n$, $|C^t| = m$, $|\Delta U| = n^+$, $|\Delta C| = m^+$, then $|U^{t+1}| = n' = n + n^+$ and $|C^{t+1}| = m' = m + m^+$, where $U^{t+1} = U^t \cup \Delta U$, $C^{t+1} = C^t \cup \Delta C$.

Algorithm 1 is a matrix-based static algorithm for computing approximations while objects and attributes are increased simultaneously. Steps 2-14 are to compute the relation matrix $M^{C^{t+1}}$ according to Definition 3, whose time complexity is $O(\frac{n'(n'+1)m'}{2})$. Steps 15-19 are to calculate the lower and upper approximations by Theorem 1. Hence the total time complexity is $O(\frac{n'(n'+1)m'}{2} + (n')^2)$.

Algorithm 2 is a matrix-based incremental algorithm for updating approximations when objects and attributes are added simultaneously. Steps 2-15 are to update the relation matrix $M_{U_t}^{C^{t+1}}$ in terms of Theorem 3, whose time complexity is $O(\frac{\eta m^+}{2})$, where η denotes the numbers of "1" in the matrix $M_{U_t}^{C^t}$ and $\eta \leqslant n^2$. Steps 17-22 are to compute the relation matrix $M_{\Delta U}^{C^{t+1}}$ by Definition 3, whose time complexity is $O(\frac{n^+(n^++1)(m+m^+)}{2})$. Steps 23-27 are to update the interactive matrix $M_{U_t,\Delta U}^{C^{t+1}}$ according to Theorem 4, which are the crucial steps for updating approximations in FDS, and the time complexity of updating operation is $O(|U^t/R_{C^{t+1}}||\Delta U/R_{C^{t+1}}|)$. Step 28 is to calculate the lower and upper approximations according to Theorem 1, whose time complexity is $O((n')^2)$. Therefore, the total time complexity of Algorithm 2 is $O(\frac{\eta m^+}{2} + \frac{n^+(n^++1)(m+m^+)}{2} + (n')^2 + |U^t/R_{C^{t+1}}||\Delta U/R_{C^{t+1}}|(m+m^+))$. To compare with the static algorithm more clearly, the complexity of Algorithm 1 is divided into five parts, namely, $O(\frac{n'(n'+1)m'}{2} + (n')^2) = O(\frac{n(n+1)m^+}{2} + \frac{n^+(n^++1)(m+m^+)}{2} + (n')^2 + \frac{n(n+1)m}{2} + nn^+(m+m^+))$. It can be seen that the time complexity of Algorithm 2 is better than that of Algorithm 1. When the size of added object and attribute set is small, i.e., n^+ and m^+ are far less than n and m, respectively, it is obvious that the time complexity of Algorithm 2 is almost identical to that of Algorithm 1. However, if n^+ and m^+ are increased to n and m, respectively, the last two terms of time complexity in

Algorithm 1: The static algorithm for computing approximations in FDS while attributes and objects are added simultaneously

Input:
1. An FDS $S = (U, C \cup D, V, f)$;
2. The added object set ΔU, the condition attribute set ΔC and the decision attribute set ΔD.

Output: The lower and upper approximations in FDS.

```
begin
    for 1 ≤ i ≤ n′ do                      // Compute the relation matrix M^{C^{t+1}} by Definition 3;
        for i ≤ j ≤ n′ do
            if j = i then                  // According to Proposition 1;
                m_{ii}^{C^{t+1}} = 1;
            else
                if x_i ∈ [x_j]_{R_{C^{t+1}}} then
                    m_{ij}^{C^{t+1}} = m_{ji}^{C^{t+1}} = 1;
                else
                    m_{ij}^{C^{t+1}} = m_{ji}^{C^{t+1}} = 0
                end
            end
        end
    end
    for 1 ≤ i ≤ n′ do
        for 1 ≤ j ≤ n′ do
            Compute \underline{R_{C^{t+1}}}\tilde{d}, \overline{R_{C^{t+1}}}\tilde{d};   // Compute the approximations by Theorem 1;
        end
    end
    return \underline{R_{C^{t+1}}}\tilde{d}, \overline{R_{C^{t+1}}}\tilde{d}.
end
```

Algorithm 1 could not be disregarded. Evidently, the last term $|U^t/R_{C^{t+1}}||\Delta U/R_{C^{t+1}}|(m + m^+)$ of time complexity in Algorithm 2 is far less than the sum of last two terms, i.e., $\frac{n(n+1)m}{2} + nn^+(m + m^+)$, in Algorithm 1. Thus, the dynamic algorithm (Algorithm 2) is more efficient than the static algorithm (Algorithm 1) when a large number of new objects and attributes are added concurrently.

6. Experimental evaluations

To demonstrate the effectiveness of our proposed incremental algorithm, the comparative experiments are designed and the results are discussed in this section. Six data sets are obtained from the UCI Repository of Machine Learning Databases (www.ics.uci.edu/∼mlearn/MLRepository.html). The description of data sets is listed in Table 3, which is ordered by the number of samples in an ascending order. Since there is no fuzzy membership information on decision attributes of the experimental data, the membership of each object is calculated according to $A_i(x) = 1 - \frac{d(x,c_i)}{\max(d(x,c_1),d(x,c_2),\cdots,d(x,c_m))}$, where $A_i(x)$ is the membership of the object x in the ith decision class, c_i is the center of ith decision class which is computed by the mean of the ith decision class, and $d(x, c_i)$ is the Euclid distance between the object x and the ith class center c_i. All experiments are performed on personal computer with Intel Core i5-4200U CPU 1.60GHZ, 4.0 GB of memory, running Win 7. Algorithms are coded in Matlab 2012.

6.1. *A comparison of computational efficiency between the static and dynamic algorithms under the addition of different sizes of objects and attributes*

In this subsection, to compare the performance between static and dynamic algorithms when adding different sizes of data set, we select 50% attributes and objects from the whole data set as the basic data set and 10%, 20%, . . . , 100% attributes and objects are taken out from the remaining 50% data set as the incremental data set. The experimental results are illustrated in Table 4 and Figure 1. Table 4 indicates the comparison of computational time between static and dynamic algorithms on six data sets, which are listed in Table 3. The tendency of running time with the addition of different sizes of data sets are shown in Figure 1. In each sub-figure of Figure 1, the x-coordinate pertains to the

Algorithm 2: The dynamic algorithm for computing approximations in FDS while attributes and objects are added simultaneously

Input:
1. An FDS $S = (U, C \cup D, V, f)$;
2. The relation matrix at time t: $M_{U_t}^{C^t} = (m_{ij}^{C^t})_{n \times n}$;
3. Original unions of equivalent classes at time t: $U^t/R_{C^t} = \{E_1, E_2, \ldots, E_r\}$;
4. The added object set ΔU, the condition attribute set ΔC and the decision attribute set ΔD.

Output: The lower and upper approximations in FDS.

begin
  for $1 \le i \le n$ **do** // Compute the relation matrix $M_{U_t}^{\Delta C} = (m_{ij}^{\Delta C})_{n \times n}, M_{U_t}^{C^{t+1}} = (m_{ij}^{C^{t+1}})_{n \times n}$;
    for $i \le j \le n$ **do**
      Compute $m_{ij}^{\Delta C}$; // Compute the relation matrix by Definition 3;
      if $m_{ij}^{C^t} == 0$ **then** // Update the relation matrix $M_{U_t}^{C^{t+1}}$ by Theorem 3;
        $m_{ij}^{C^{t+1}} = m_{ji}^{C^{t+1}} = m_{ij}^{C^t}$ are constant;
      else
        if $m_{ij}^{\Delta C}{==}1$ **then**
          $m_{ij}^{C^{t+1}} = m_{ji}^{C^{t+1}} = m_{ij}^{C^t}$ are constant;
        else
          $m_{ij}^{C^{t+1}} = m_{ji}^{C^{t+1}} = 0$
        end
      end
    end
    Compute the equivalence classes $U^t/R_{C^{t+1}}$.
  end
  for $1 \le i \le n^+$ **do** // Compute the relation matrix $M_{\Delta U}^{C^{t+1}} = (m_{ij}^{\Delta U})_{n^+ \times n^+}$;
    for $i \le j \le n^+$ **do**
      Compute $m_{ij}^{\Delta U}$; // Compute the relation matrix by Definition 3;
    end
    Compute the equivalence classes $\Delta U/R_{C^{t+1}}$.
  end
  for $1 \le i \le |U^t/R_{C^{t+1}}|$ **do** // Compute the relation matrix $M_{U_t,\Delta U}^{C^{t+1}} = (m_{ij}^{U_t,\Delta U})_{n \times n^+}$;
    for $1 \le j \le |\Delta U/R_{C^{t+1}}|$ **do**
      Compute $m_{ij}^{U_t,\Delta U}$; // Update the relation matrix by Theorem 4;
    end
  end
  Compute the lower and upper approximations $\underline{R_{C^{t+1}}}\widetilde{d}, \overline{R_{C^{t+1}}}\widetilde{d}$. // Compute the approximations by Theorem 1;
  return $\underline{R_{C^{t+1}}}\widetilde{d}$ *and* $\overline{R_{C^{t+1}}}\widetilde{d}$.
end

Table 3: A description of data sets

	Data sets	Abbreviation	Samples	Attributes	Classes	Source
1	Phishing Websites	Phishing	2456	30	2	UCI
2	Chess	Chess	3196	36	2	UCI
3	Optical Recognition of Handwritten Digits	Optical	5620	64	10	UCI
4	Turkiye Student Evaluation	Turkiye	5820	32	13	UCI
5	Statlog (Landsat Satellite)	Statlog	6435	36	7	UCI
6	Musk (Version 2)	Musk	6598	168	2	UCI

Table 4: A comparison of static and incremental algorithms versus different updating ratios when adding the objects and attributes simultaneously.

	Phishing		Chess		Optical		Turkiye		Statlog		Musk	
Insert rate	Static	Dynamic	Static	Dynamic	Static	Dynamic	Static	Dynamic	Static	Dynamic	Static	Dynamic
10%	5.155	2.426	11.574	4.911	31.996	25.188	30.366	24.629	31.649	27.196	34.277	3.351
20%	6.225	3.081	13.967	7.552	38.655	29.254	39.081	29.711	31.580	28.503	41.510	5.914
30%	7.467	3.997	16.394	8.693	51.230	31.661	50.271	34.743	48.920	31.523	52.451	8.746
40%	9.202	4.695	19.516	10.776	60.266	37.379	55.351	37.464	61.282	34.924	60.930	11.494
50%	10.664	5.703	22.219	12.338	69.054	40.177	72.027	41.859	70.126	40.511	78.201	14.507
60%	12.415	6.566	25.696	13.537	82.285	43.763	80.684	45.459	82.432	43.712	95.161	18.573
70%	14.808	7.763	30.126	15.401	95.829	51.416	102.225	50.992	97.113	49.184	113.914	20.901
80%	17.217	8.743	32.957	16.856	112.820	59.495	114.431	56.244	114.266	55.491	137.987	24.639
90%	18.519	9.949	38.234	19.783	130.904	63.281	136.791	63.314	132.167	62.729	165.719	31.620
100%	22.671	11.092	42.449	23.652	144.559	73.251	154.597	70.234	158.107	75.069	188.678	33.390

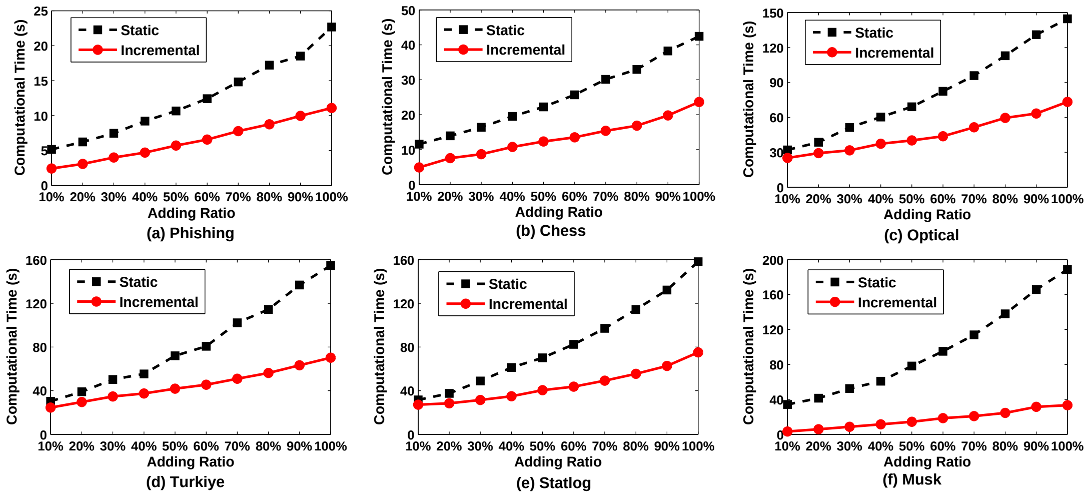

Figure 1: A comparison of computational time between the static algorithm (Algorithm 1) and incremental algorithm (Algorithm 2) with increasing size of attributes and objects simultaneously.

rate of the objects and attributes added to the basic data set from the rest of data set and the *y*-coordinate pertains to the computational time of static and dynamic algorithms.

It is easy to observe from Table 4 and Figure 1 that the running times of static and incremental algorithms rise with increasing ratio of data set. And the gap becomes greater when the size of data set increases. Furthermore, the computational time of the incremental algorithm keeps persistently lower than the static algorithm when the same rate of data are added from the rest of data set. Hence, the experimental results demonstrate that the dynamic algorithm is more efficient than the static algorithm when keeping the basic data set unchanged and adding different ratios of objects and attributes simultaneously.

Table 5: The incremental speedup ratio versus each test set

	Data Set					
Test Set	Phishing	Chess	Optical	Turkiye	Statlog	Musk
1	1.922	2.502	1.825	1.434	1.001	1.025
2	2.797	3.265	1.780	1.592	1.038	1.085
3	2.794	2.479	1.569	1.285	1.081	1.094
4	2.102	2.571	1.388	1.198	1.081	1.023
5	1.992	2.144	1.167	1.157	1.088	1.109
6	1.888	1.940	1.291	1.135	1.099	1.110
7	1.873	1.960	1.356	1.131	1.102	1.105
8	1.841	1.673	1.347	1.147	1.109	1.117
9	1.676	1.596	1.214	1.154	1.106	1.113
average	2.098	2.237	1.438	1.248	1.078	1.087

6.2. *Performance comparison with the growing sizes of data sets*

In this subsection, to compare running times between the static and incremental algorithms when the size of original data set grows, we extract 10%, 20%, . . . , 90% data as test set 1, test set 2, . . . , test set 9 from each data set which is listed in Table 3, respectively, *i.e.*, the original data set is increased by 10% gradually. Then the data of which size is the 5% of test set is appended to the test set from the rest of each data set. In order to show the advantage of incremental algorithm, the speedup ratio on each test set is depicted in Table 5. The computational time of static and incremental algorithms when objects and attributes are added simultaneously are shown in Figure 2, where *x*-coordinate pertains to the number of test set and the *y*-coordinate pertains to the computational time of static and dynamic algorithms.

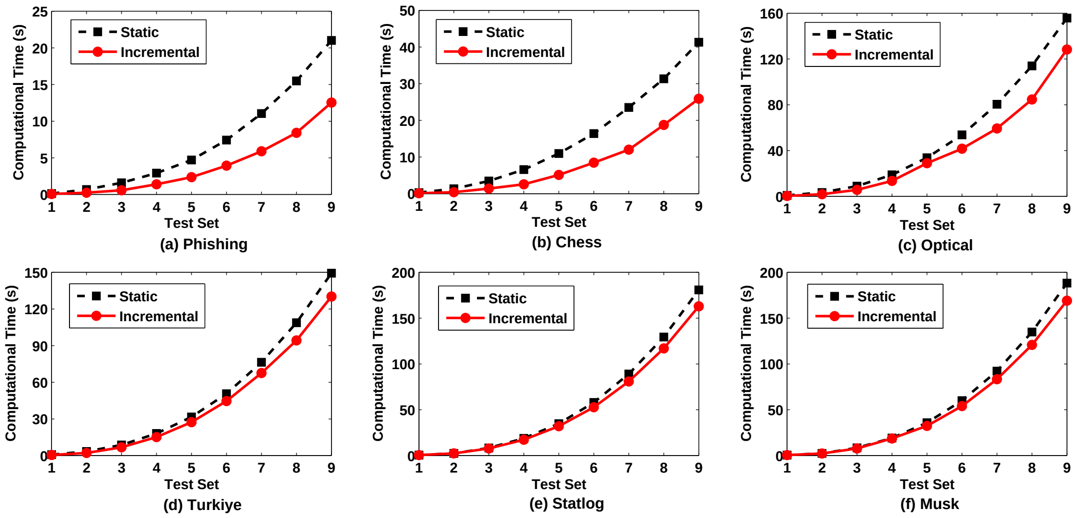

Figure 2: A comparison of computational time between static algorithm (Algorithm 1) and incremental algorithm (Algorithm 2) when the original data set grows.

In Table 5, the incremental algorithm achieves 2.797-1.676 speedup over the static algorithm on the Phishing data set. And the average speedup ratio on each different data set is from 1.078- 2.237. Obviously, the incremental algorithm performs better than the static algorithm when the original data set grows. Figure 2 shows that the computational time of incremental algorithm is much faster than the static algorithm with the growth of original data set. In addition, the differences are getting larger with the increasing size of test sets, which demonstrates the much better performance of our presented algorithm when the original data set becomes larger.

6.3. Comparisons with the reference algorithm

Although there were not related algorithms for maintenance of rough approximations in FDS where both objects and attributes increase over time, Chen et al. introduced an incremental method for computing approximations with the variation of attributes (CIA for short) [32] and Zeng et al. presented a dynamic algorithm for updating approximations under the variation of objects (ZIO for short) [30]. In order to compare our proposed algorithm with these two incremental algorithms with single-dimensional variation of FDS, we combine them to deal with the simultaneous variation of objects and attributes in FDS. The combined algorithm is abbreviated as CIA+ZIO for convenience in this paper.

Table 6 shows the speedup ratio between the computational times of our proposed algorithm and CIA+ZIO when inserting different proportion of data. It is evidently that our method is much faster than the combined method. And the average speedup ratio increases with the number of samples of each data set. Figure 3 shows the running time of our incremental algorithm and CIA+ZIO with the different adding ratio. Clearly, the performance of our method is better than the combined method. Since that the combined method only considers the incremental mechanism with the single-dimensional variation of objects or attributes, it omits the interactive information which can reduce the runtime when adding objects and attributes simultaneously. However, our proposed approach not only handles the individual variation' impacts on the structure of approximations, but also takes into account the interaction between adding attributes and inserting objects. Moreover, according to Theorem 4, the updating of interactive matrix is that multiple matrix elements or even several rows and columns of matrix are renewed by utilizing the previous matrix information. Hence, our method is more efficient than the combined method through updating the structure of approximations under the independent variation of objects and attributes.

Table 6: The incremental speedup ratio between the computational times of our incremental algorithm and CIA+ZIO

	Data Set					
Insert rate	Phishing	Chess	Optical	Turkiye	Statlog	Musk
10%	18.106	19.252	3295.830	5072.693	6634.820	27612.330
20%	14.964	16.909	2843.597	4233.077	6458.600	17402.735
30%	12.895	21.727	2665.665	4092.389	6368.522	12386.122
40%	12.163	28.369	2308.833	4829.983	5816.108	9914.626
50%	10.710	28.132	2187.322	4364.921	5079.422	7937.799
60%	11.066	26.389	2020.844	4145.912	5156.450	6455.063
70%	12.240	24.396	1767.120	3852.931	4919.663	5959.542
80%	13.148	22.481	1545.254	3502.400	7111.295	5113.529
90%	15.243	20.512	1478.272	3119.385	7155.473	3993.029
100%	18.321	17.893	1301.878	2847.486	6267.194	3978.997
average	13.886	22.606	2141.461	4006.118	6096.755	10075.377

7. Conclusions

In this paper, we defined a novel matrix operator for the construction of rough fuzzy approximations. To improve the efficiency of computing approximations in FDS when objects and attributes are added simultaneously, we proposed the dynamic mechanisms for maintenance of rough fuzzy approximations based on matrix. Then we developed

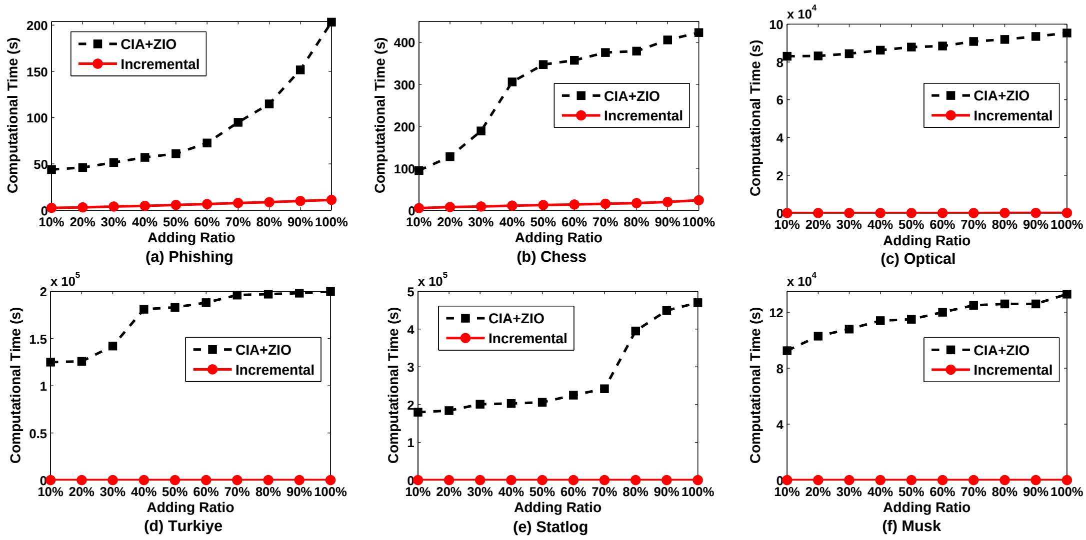

Figure 3: A comparison of computational time between our incremental algorithm and CIA+ZIO.

a matrix-based incremental algorithm for updating approximations in FDS. Finally, we designed comparative experiments for validating the effectiveness of the proposed incremental algorithm. Experimental results demonstrated that the performance of dynamic algorithm is better than the static and combined algorithms. Furthermore, the more objects and attributes are added to the data set, the more efficiency of dynamic algorithm will be achieved. Considering attributes with preference-ordered domains in FDS, we will integrate the proposed method and dominance-based rough set model to update approximations under dynamic fuzzy environments in the future.

Acknowledgements

This work is supported by the National Science Foundation of China (Nos. 61573292, 61572406, 61602327, 61603313), the China Postdoctoral Science Foundation (No. 2016M60268).

References

[1] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (5) (1982) 341–356.
[2] R. Wang, D. G. Chen, S. Kwong, Fuzzy-rough-set-based active learning, IEEE Transactions on Fuzzy Systems 22 (6) (2014) 1699–1704.
[3] W. Pedrycz, S. M. Chen, Information Granularity, Big Data, and Computational Intelligence, Springer, 2015.
[4] J. T. Yao, N. Azam, Web-based medical decision support systems for three-way medical decision making with game-theoretic rough sets, IEEE Transactions on Fuzzy Systems 23 (1) (2015) 3–15.
[5] Y. Y. Yao, N. Zhong, Potential applications of granular computing in knowledge discovery and data mining, Proceedings of World Multiconference on Systemics, Cybernetics and Informatics 5 (1999) 573–580.
[6] S. Salehi, A. Selamat, H. Fujita, Systematic mapping study on granular computing, Knowledge-Based Systems 80 (2015) 78–97.
[7] L. A. Zadeh, Computing with words in Information/Intelligent systems 1: Foundations, Physica, 2013.
[8] P. Maji, S. K. Pal, Rough-fuzzy pattern recognition: applications in bioinformatics and medical imaging, John Wiley and Sons, 2011.
[9] H. X. Bai, Y. Ge, J. F. Wang, D. Y. Li, Y. L. Liao, X. Y. Zheng, A method for extracting rules from spatial data based on rough fuzzy sets, Knowledge-Based Systems 57 (2014) 28-40.
[10] G. Peters, M. Lampart, R. Weber, Evolutionary rough k-medoid clustering, Transactions on rough sets VIII, Springer Berlin Heidelberg, 2008, 289–306.
[11] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets*, International Journal of General System 17 (2-3) (1990) 191–209.
[12] Y. Y. Yao, A comparative study of fuzzy sets and rough sets, Information Sciences 109 (1998) 227–242.
[13] D. S. Yeung, D. G. Chen, E. C. Tsang, J. W. Lee, X. Z. Wang, On the generalization of fuzzy rough sets, IEEE Transactions on Fuzzy Systems 13 (3) (2005) 343–361.
[14] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 824–838.
[15] Y. Cheng, Forward approximation and backward approximation in fuzzy rough sets, Neurocomputing 148 (2015) 340–353.

[16] W. H. Xu, W. T. Li, Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets, IEEE Transactions on Cybernetics 46(2) (2016) 366-379.
[17] T. Glenn, A. Zare, P. Gader, Bayesian fuzzy clustering, IEEE Transactions on Fuzzy Systems 23 (2015) 1545–1561.
[18] Q. H. Hu, L. Zhang, S. An, D. Zhang, D. R. Yu, On robust fuzzy rough set models, IEEE Transactions on Fuzzy Systems 20 (4) (2012) 636–651.
[19] X. B. Yang, Y. Qi, D. J. Yu, H. L. Yu, J. Y. Yang, α-dominance relation and rough sets in interval-valued information systems, Information Sciences 294 (2015) 334–347.
[20] B. Z. Sun, W. M. Ma, H. Y. Zhao, Decision-theoretic rough fuzzy set model and application, Information Sciences 283 (2014) 180–196.
[21] T. J. Li, W. X. Zhang, Rough fuzzy approximations on two universes of discourse, Information Sciences 178 (3) (2008) 892–906.
[22] H. H. Huang, Y. H. Kuo, Cross-lingual document representation and semantic similarity measure: a fuzzy set and rough set based approach, IEEE Transactions on Fuzzy Systems 18 (6) (2010) 1098–1111.
[23] A. Petrosino, A. Ferone, Rough fuzzy set-based image compression, Fuzzy Sets and Systems 160 (10) (2009) 1485–1506.
[24] G. Krempl, I. Žliobaite, D. Brzeziński, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou *et al.*, Open challenges for data stream mining research, ACM SIGKDD Explorations Newsletter 16 (1) (2014) 1–10.
[25] C. C. Chan, A rough set approach to attribute generalization in data mining, Information Sciences 107 (1998) 169–176.
[26] I. Saha, U. Maulik, Incremental learning based multiobjective fuzzy clustering for categorical data, Information Sciences 267 (2014) 35–57.
[27] S. Tsumoto, Incremental rule induction based on rough set theory, Foundations of Intelligent Systems, Springer Berlin Heidelberg, 2011, 70–79.
[28] J. Y. Liang, F. Wang, C. Y. Dang, Y. H. Qian, A group incremental approach to feature selection applying rough set technique, IEEE Transactions on Knowledge and Data Engineering 26 (2) (2014) 294–308.
[29] C. C. Huang, T. L. B. Tseng, Y. N. Fan, C. H. Hsu, Alternative rule induction methods based on incremental object using rough set theory, Applied Soft Computing 13 (2013) 372–389.
[30] A. P. Zeng, T. R. Li, J. B. Zhang, H. M. Chen, Incremental maintenance of rough fuzzy set approximations under the variation of object set, Fundamenta Informaticae 132 (3) (2014) 401–422.
[31] F. Wang, J. Y. Liang, Y. H. Qian, Attribute reduction: a dimension incremental strategy, Knowledge-Based Systems 39 (2013) 95–108.
[32] Y. Cheng, The incremental method for fast computing the rough fuzzy approximations, Data and Knowledge Engineering 70 (2011) 84–100.
[33] X. B. Yang, Y. Qi, H. L. Yu, X. N. Song, J. Y. Yang, Updating multigranulation rough approximations with increasing of granular structures, Knowledge-Based Systems 64 (2014) 59–69.
[34] C. Luo, T. R. Li, H. M. Chen, X. L. Lu, Fast algorithms for computing rough approximations in set-valued decision systems while updating criteria values, Information Sciences 299 (2015) 221–242.
[35] M. J. Cai, Knowledge reduction of dynamic covering decision information systems with varying attribute values, arXiv preprint arXiv:1504.02930, 2015.
[36] H. M. Chen, T. R. Li, C. Luo, S. J. Horng, G. Y. Wang, A decision-theoretic rough set approach for dynamic data mining, IEEE Transactions on Fuzzy Systems 23 (6) (2015) 1958–1970.
[37] J. B. Zhang, T. R. Li, D. Ruan, D. Liu, Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, International Journal of Approximate Reasoning 53 (4) (2012) 620–635.
[38] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, Intelligent Decision Support, Springer Netherlands, 1992, 331–362.
[39] A. H. Tan, J. J. Li, G. P. Lin, Y. J. Lin, Fast approach to knowledge acquisition in covering information systems using matrix operations, Knowledge-Based Systems 79 (2015) 90–98.
[40] S. P. Wang, W. Zhu, Q. X. Zhu, F. Min, Characteristic matrix of covering and its application to boolean matrix decomposition, Information Sciences 263 (2014) 186–197.
[41] J. B. Zhang, Y. Zhu, Y. Pan, T. R. Li, Efficient parallel boolean matrix based algorithms for computing composite rough set approximations, Information Sciences 329 (2016) 287–302.
[42] L. Ma, Two fuzzy covering rough set models and their generalizations over fuzzy lattices, Fuzzy Sets and Systems 294 (2016) 1-17.
[43] R. W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (6) (2003) 833–849.
[44] X. Z. Wang, E. C. Tsang, S. Y. Zhao, D. G. Chen, D. S. Yeung, Learning fuzzy rules from fuzzy samples based on rough set technique, Information sciences 177 (20) (2007) 4493–4514.
[45] C. H. Chen, Z. M. Rao, Mrm: A matrix representation and mapping approach for knowledge acquisition, Knowledge-Based Systems 21 (2008) 284–293.
[46] G. L. Liu, The axiomatization of the rough set upper approximation operations, Fundamenta Informaticae 69 (3) (2006) 331–342.
[47] J. B. Zhang, T. R. Li, H. M. Chen, Composite rough sets for dynamic data mining, Information Sciences 257 (2014) 81–100.
[48] C. Luo, T. R. Li, H. M. Chen, Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization, Information Sciences 257 (2014) 210–228.
[49] A. H. Tan, J. J. Li, Y. J. Lin, G. P. Lin, Matrix-based set approximations and reductions in covering decision information systems, International Journal of Approximate Reasoning 59 (2015) 68–80.