Fundamenta Informaticae 132 (2014) 401–422
DOI 10.3233/FI-2014-1051
IOS Press

Incremental Maintenance of Rough Fuzzy Set Approximations under the Variation of Object Set

Anping Zeng*, Tianrui Li†, Junbo Zhang, Hongmei Chen
School of Information Science and Technology
Southwest Jiaotong University
Chengdu 610031, China
zengap@126.com; trli@swjtu.edu.cn; JunboZhang86@163.com; hm_chen_swjtu@163.com

Abstract. The lower and upper approximations in rough set theory will change dynamically over time due to the variation of the information system. Incremental methods for updating approximations in rough set theory and its extensions have received much attention recently. Most existing incremental methods have difficulties in dealing with fuzzy decision systems which decision attributes are fuzzy. This paper introduces an incremental algorithm for updating approximations of rough fuzzy sets under the variation of the object set in fuzzy decision systems. In experiments on 6 data sets from UCI, comparisons of the incremental and non-incremental methods for updating approximations are conducted. The experimental results show that the incremental method effectively reduces the computational time.

Keywords: Rough Fuzzy Sets, Approximations, Incremental Learning, Fuzzy Decision System

1. Introduction

The central question in inductive learning is whether one can reproduce the expert classification of a given set of objects only on the basis of selected attributes and attribute values of objects. Rough Set Theory (RST), introduced by Pawlak [1], has proved to be a useful tool for classification and rule extraction in decision situations. However, there are several general cases that Pawlak's RST can not deal with, e.g.,

*Also works: School of Computer and Information Engineering, Yibin University, Yibin 644007, China
†Address for correspondence: School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

the decision attribute values are fuzzy and each object belongs to multiple decision classes by different probabilities. Therefore, an extended model of RST, rough fuzzy sets, was proposed to deal with these cases [2, 3].

In real-life applications, data in information systems is huge and often generated and collected dynamically, and the knowledge needs to be updated accordingly [4, 5]. Incremental learning, namely learning from new data without forgetting prior knowledge, is a feasible solution. Since an information system consists of attributes (features), objects (instances), and attributes' values, incremental updating algorithms with respect to rough sets have been proposed in following three aspects: (1) The variation of attributes; (2) The variation of attributes' values; (3) The variation of objects.

To the variation of attributes, Chan proposed an incremental method for updating the approximations of a concept based on the lower and upper boundary sets [6]. Li et al. presented an incremental method for updating approximations through the characteristic relation when the attribute set varies over time, which can deal with the case of adding and removing some attributes simultaneously in an incomplete information system [7]. In addition, Zhang et al. discussed the strategies and propositions under Variable Precision Rough Sets (VPRS) when attributes are changed [8]. Li et al. introduced a dominance matrix to calculate P-dominating sets and P-dominated sets in Dominance-based Rough Sets Approach (DRSA), and proposed the incremental algorithms for updating approximations of DRSA when the attribute set varies [9]. Luo et al. focused on maintaining approximations dynamically in set-valued ordered decision systems under the attribute generalization [10]. In Rough Fuzzy Sets (RFS), Cheng proposed approaches for incremental updating approximations when the attribute set evolves over time [11]. There may be attributes of multiple different types in information systems in real-life applications. Zhang et al. presented a composite relation and proposed an extended rough set model, called as Composite Rough Sets (CRS). Moreover, combined with the incremental learning technique, a novel matrix-based method for fast updating approximations of CRS was proposed in [12].

Since the domain of attributes may change in real-life applications, attributes' values may be added to or deleted from the domain. Chen et al. proposed an incremental algorithm for updating the approximations of a concept on coarsening or refining of attributes' values [13]. Furthermore, Chen et al. proposed the incremental updating approaches of approximations concerning attributes' values coarsening or refining in the complete and incomplete information systems [14]. In [15], the variation of an attribute's domain is also considered to perform incremental updating for approximations under VPRS. Liu et al. proposed a probabilistic rough set approach for incremental learning knowledge under coarsening and refining of attribute values [16]. Furthermore, Liu et al. proposed several incremental learning strategies and algorithms for inducing the interesting knowledge when attributes' values are changed [17].

The variation of objects has been considered widely in incremental learning based on RST. Shan et al. presented a discernibility-matrix based incremental methodology to find all maximally generalized rules [18]. Bang et al. proposed an incremental inductive learning algorithm to find a minimal set of rules for a decision table without recalculating all the set of instances when a new instance is added into the universe [19]. Liu et al. proposed a model and an incremental approach for inducing interesting knowledge when the object set varies over time [20]. Zheng et al. presented an effective incremental approach for knowledge acquisition based on the rule tree [21]. Hu et al. proposed a novel incremental attribute reduction algorithm when new objects are added into a decision information system [22]. Wang et al. constructed an incremental rule acquisition algorithm based on VPRS while inserting new objects into the information system [23]. Zhang et al. proposed an incremental rule acquisition algorithm based on neighborhood rough sets when the object set evolves over time [24]. Li et al. proposed a dynamic maintenance of approximations in DRSA when adding or deleting objects in the universe [25]. In addition, Luo et al. proposed an incremental approaches for updating approximations in set-valued ordered information systems [26]. However, there is no report about the incremental approach for updating approximations of RFS under the variation of objects so far. Firstly, we discuss the principles of incremental updating approximations when the objects in the universe change (immigrate or emigrate) dynamically in the rough fuzzy sets. Then, two incremental updating algorithms are proposed based on the principles. Finally, the performances of two incremental algorithms are evaluated on several UCI data sets.

The rest of this paper is organized as follows. In Section 2, some basic concepts of RST and RFS are introduced. In Section 3, the updating principles of upper and lower approximations in RFS are analyzed under the variation of the object set in fuzzy decision systems. In Section 4, experimental evaluations with six data sets from UCI are given. In Section 5, we conclude the paper.

2. Preliminaries

In this section, we briefly introduce basic concepts of rough sets and RFS [1, 3].

Definition 1. Let (U, R) be a Pawlak approximation space, where the universe $U \neq \emptyset$ and $R \subseteq U \times U$ is an equivalence relation on U. U/R denotes the family of all equivalence classes, and $[x]_R$ denotes an equivalence class of R containing an element $x \in U$. For any $X \subseteq U$, the lower and upper approximations of X are defined, respectively, as follows:

$$\begin{aligned} &\underline{R}X = \{x \in U | [x]_R \subseteq X\}; \\ &\overline{R}X = \{x \in U | [x]_R \cap X \neq \emptyset\}. \end{aligned} \tag{1}$$

In order to describe a fuzzy concept in a crisp approximation space, Dubois and Prade introduced an extended model of RST, called as RFS [2, 3].

Definition 2. Let (U, R) be a Pawlak approximation space, and R be an equivalence relation on U. If $\tilde{X}$ is a fuzzy subset on U, $\tilde{X}(x)$ denotes the degree of membership of x in $\tilde{X}$. The lower and upper approximations of $\tilde{X}$ are a pair of fuzzy sets on U, and their membership functions are defined as follows:

$$\begin{aligned} &\underline{R}\tilde{X}(x) = \inf\{\tilde{X}(y) | y \in [x]_R\}; \\ &\overline{R}\tilde{X}(x) = \sup\{\tilde{X}(y) | y \in [x]_R\}. \end{aligned} \tag{2}$$

Example 2.1. Table 1 is a decision table with condition attributes (Headache, Muscle Pain, Temperature, denoted as $a,\ b,\ c$, respectively) and a fuzzy decision attribute d including two classes (Flu and No Flu). Let R be the equivalence relation under the attribute a. The partitions (or equivalence classes) can be obtained as follows:

$$U/R = \{E_1, E_2, E_3\} = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6, x_7, x_8\}\}.$$

Let $\tilde{X}$ be a fuzzy subset about Flu on U and $\tilde{X} = \{0.7/x_1, 0/x_2, 0.7/x_3, 0.8/x_4, 0/x_5, 0.5/x_6, 0.2/x_7, 0.6/x_8\}$.

Table 1. A decision table with fuzzy decision attributes.

U	**Headache(a)**	**Muscle Pain(b)**	**Temperature(c)**	**Fuzzy attribute d**	
				Flu	**No Flu**
x_1	Heavy	Yes	High	0.7	0.3
x_2	Heavy	Yes	High	0	1
x_3	Moderate	Yes	High	0.7	0.4
x_4	Moderate	Yes	Normal	0.8	0.2
x_5	Moderate	No	Normal	0	1
x_6	No	No	Normal	0.5	0.5
x_7	No	No	Normal	0.2	0.8
x_8	No	Yes	High	0.6	0.4

According to Definition 2, the degrees of membership are calculated as follows.

$$\begin{aligned}
&\underline{R}\tilde{X}(x_1) = \underline{R}\tilde{X}(x_2) = \inf\{\tilde{X}(y)|y \in [x_1]_R\} = 0.7 \wedge 0 = 0;\\
&\underline{R}\tilde{X}(x_3) = \underline{R}\tilde{X}(x_4) = \underline{R}\tilde{X}(x_5) = \inf\{\tilde{X}(y)|y \in [x_3]_R\} = 0.7 \wedge 0.8 \wedge 0 = 0;\\
&\underline{R}\tilde{X}(x_6) = \underline{R}\tilde{X}(x_7) = \underline{R}\tilde{X}(x_8) = \inf\{\tilde{X}(y)|y \in [x_6]_R\} = 0.5 \wedge 0.2 \wedge 0.6 = 0.2;\\
&\overline{R}\tilde{X}(x_1) = \overline{R}\tilde{X}(x_2) = \sup\{\tilde{X}(y)|y \in [x_1]_R\} = 0 \vee 0.7 = 0.7;\\
&\overline{R}\tilde{X}(x_3) = \overline{R}\tilde{X}(x_4) = \overline{R}\tilde{X}(x_5) = \sup\{\tilde{X}(y)|y \in [x_3]_R\} = 0.7 \vee 0.8 \vee 0 = 0.8;\\
&\overline{R}\tilde{X}(x_6) = \overline{R}\tilde{X}(x_7) = \overline{R}\tilde{X}(x_8) = \sup\{\tilde{X}(y)|y \in [x_7]_R\} = 0.5 \vee 0.2 \vee 0.6 = 0.6.
\end{aligned}$$

Therefore, the lower and upper approximations of $\tilde{X}$ are as follows:

$$\begin{aligned}
&\underline{R}\tilde{X} = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8\};\\
&\overline{R}\tilde{X} = \{0.7/x_1, 0.7/x_2, 0.8/x_3, 0.8/x_4, 0.8/x_5, 0.6/x_6, 0.6/x_7, 0.6/x_8\}.
\end{aligned}$$

3. Incremental maintenance of rough fuzzy set approximations under the variation of object set

To illustrate the dynamic approach for updating rough fuzzy set approximations, we divide the work into three parts: We present several assumptions and the basic structure of the approach in Section 3.1; Then, we design an algorithm for Non-dynamic Maintenance of Approximations based on RFS (NDMARFS), which can deal with the variation of the object set in Section 3.2; Finally, we discuss the principles of dynamic maintenance of approximations based on RFS under the variation of objects in Section 3.3.

3.1. Assumptions and the basic structure for the dynamic maintenance approach

To reduce the time of updating approximations and facilitate the design of incremental algorithms, we divide the updating approximations into two steps: Approximations of Pawlak's rough set are updated in Step 1; The memberships of different objects are updated in Step 2.

Proposition 1. Let $\tilde{X}$ be a fuzzy subset on U, $X = \{x \in U|\tilde{X}(x) \neq 0\}$. The lower and upper approximations of $\tilde{X}$ are calculated as follows.

$$\underline{R}\tilde{X}(x) = \begin{cases} \inf\{\tilde{X}(y)|y \in [x]_R\}, & x \in \underline{R}X, \\ 0, & x \notin \underline{R}X; \end{cases} \tag{3}$$

$$\overline{R}\tilde{X}(x) = \begin{cases} \sup\{\tilde{X}(y)|y \in [x]_R\}, & x \in \overline{R}X, \\ 0, & x \notin \overline{R}X. \end{cases} \tag{4}$$

Proof:

(1) Because $X = \{x \in U|\tilde{X}(x) \neq 0\}$, $\forall x \in U$, if $x \notin \underline{R}X$, then $[x]_R \nsubseteq X$. Therefore, $\exists z \in [x]_R, \tilde{X}(z) = 0$. According to Definition 2, $\underline{R}\tilde{X}(x) = 0$. Therefore, when $x \notin \underline{R}X$, $\underline{R}\tilde{X}(x) = 0$.

(2) $\forall x \in U$, if $x \in \underline{R}X$, then $[x]_R \subseteq X$. Therefore, $\forall z \in [x]_R, \tilde{X}(z) \neq 0$. According to Definition 2, $\underline{R}\tilde{X}(x) = \inf\{\tilde{X}(y)|y \in [x]_R\}$.

(3) $\forall x \in U$, if $x \in \overline{R}X$, then $[x]_R \cap X \neq \emptyset$. Therefore, $\exists z \in [x]_R, \tilde{X}(z) \neq 0$. According to Definition 2, $\overline{R}\tilde{X}(x) = \sup\{\tilde{X}(y)|y \in [x]_R\}$.

(4) $\forall x \in U$, if $x \notin \overline{R}X$, then $[x]_R \cap X = \emptyset$. Therefore, $\forall z \in [x]_R, \tilde{X}(z) = 0$. According to Definition 2, $\overline{R}\tilde{X}(x) = 0$. □

Example 3.1. Based on Example 2.1, $X = \{x \in U|\tilde{X}(x) \neq 0\} = \{x_1, x_3, x_4, x_6, x_7, x_8\}$. According to Definiton 1, the lower and upper approximations of X can be obtained.

$$\underline{R}X = \{x_6, x_7, x_8\};$$
$$\overline{R}X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}.$$

According to Proposition 1, the degrees of membership are calculated as follows.
(1) Because $x_1, x_2, x_3, x_4, x_5 \notin \underline{R}X$, $\underline{R}\tilde{X}(x_1) = \underline{R}\tilde{X}(x_2) = \underline{R}\tilde{X}(x_3) = \underline{R}\tilde{X}(x_4) = \underline{R}\tilde{X}(x_5) = 0$.
$x_6, x_7, x_8 \in \underline{R}X$. Then, $\underline{R}\tilde{X}(x_6) = \underline{R}\tilde{X}(x_7) = \underline{R}\tilde{X}(x_8) = \inf\{\tilde{X}(y)|y \in [x_6]_R\} = 0.2$.
(2) $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \in \overline{R}X$. Then,
$\overline{R}\tilde{X}(x_1) = \overline{R}\tilde{X}(x_2) = \sup\{\tilde{X}(y)|y \in [x_1]_R\} = 0 \vee 0.7 = 0.7$;
$\overline{R}\tilde{X}(x_3) = \overline{R}\tilde{X}(x_4) = \overline{R}\tilde{X}(x_5) = \sup\{\tilde{X}(y)|y \in [x_3]_R\} = 0.7 \vee 0.8 \vee 0 = 0.8$;
$\overline{R}\tilde{X}(x_6) = \overline{R}\tilde{X}(x_7) = \overline{R}\tilde{X}(x_8) = \sup\{\tilde{X}(y)|y \in [x_7]_R\} = 0.5 \vee 0.2 \vee 0.6 = 0.6$.
Therefore, the lower and upper approximations of $\tilde{X}$ are as follows:

$$\underline{R}\tilde{X} = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8\};$$
$$\overline{R}\tilde{X} = \{0.7/x_1, 0.7/x_2, 0.8/x_3, 0.8/x_4, 0.8/x_5, 0.6/x_6, 0.6/x_7, 0.6/x_8\}.$$

Note that the approximations are the same as Example 2.1.

We assume the incrementally learning process lasts from time t to time $t + 1$. The variation of objects is divided into two parts: the immigration and the emigration of objects. The former case means the objects enter into the Fuzzy Decision System (FDS) at time t + 1, and the universe is expanding; The latter case means the objects get out of the FDS, and the universe is contracting at time t + 1.

Given an FDS $= (U, C \cup D, V, f)$ at time t, $U \neq \emptyset$ and $C \cap D = \emptyset$. Suppose there are some objects immigrate into FDS or emigrate from FDS at time $t + 1$. The equivalence classes will be changed. Then, the lower and upper approximations will be changed too. For each fuzzy subset $\tilde{X}$ on U, the lower and upper approximations of $\tilde{X}$ are denoted by $\underline{R^{(t)}}\tilde{X}$ and $\overline{R^{(t)}}\tilde{X}$ at time t, respectively. Let $X = \{x \in U|\tilde{X}(x) \neq 0\}$. $\underline{R^{(t)}}X$ and $\overline{R^{(t)}}X$ denote the lower and upper approximations of X at time t, respectively. After the immigration and emigration, the FDS will be changed into $\text{FDS}' = (U', C' \cup D', V', f')$. The lower and upper approximations of $\tilde{X}$ are denoted by $\underline{R^{(t+1)}}\tilde{X}$ and $\overline{R^{(t+1)}}\tilde{X}$, respectively. $\underline{R^{(t+1)}}X$ and $\overline{R^{(t+1)}}X$ denote the lower and upper approximations of X at time $t+1$, respectively. Table 2 shows the main symbols used in the incrementally learning process.

Table 2. The main symbols used in the incrementally learning process.

$\tilde{X}$	Fuzzy subset of U	X	Object subset of U and $X = \{x \in U\|\tilde{X}(x) \neq 0\}$
$\underline{R^{(t)}}\tilde{X}$	Lower approximation of $\tilde{X}$ at time t	$\underline{R^{(t)}}X$	Lower approximation of X at time t
$\overline{R^{(t)}}\tilde{X}$	Upper approximation of $\tilde{X}$ at time t	$\overline{R^{(t)}}X$	Upper approximation of X at time t
$\underline{R^{(t+1)}}\tilde{X}$	Lower approximation of $\tilde{X}$ at time $t+1$	$\underline{R^{(t+1)}}X$	Lower approximation of X at time $t+1$
$\overline{R^{(t+1)}}\tilde{X}$	Upper approximation of $\tilde{X}$ at time $t+1$	$\overline{R^{(t+1)}}X$	Upper approximation of X at time $t+1$

In the following, we focus on the algorithms for updating approximations of the fuzzy concept $\tilde{X}$ when (1) objects enter into the FDS at time $t+1$; (2) objects get out of the FDS at time $t+1$.

3.2. A non-dynamic algorithm for maintenance of approximations

First, we design the NDMARFS algorithm, which can process the above two mentioned variations of the object set. When objects immigrate or emigrate, the NDMARFS algorithm first generates the equivalence classes. Then, according to Definition 1, objects which should be included in the upper or lower approximations of X can be obtained. And then, according to Definition 2, the memberships can be computed. Finally, the upper or lower approximations of $\tilde{X}$ can be obtained. The detailed process is outlined in Algorithm 1.

Input: $\text{FDS}' = (U', C' \cup D', V', f')$ at time $t+1$.
Output: $\underline{R^{(t+1)}}\tilde{X}, \overline{R^{(t+1)}}\tilde{X}$ at time $t+1$.
Generating equivalence classes in U';
for *each* $E_i \in U'/R$ **do**
  if $E_i \cap X \neq \emptyset$ **then**
    $\overline{R^{(t+1)}}X = \overline{R^{(t+1)}}X \cup E_i$;
    for *each* $x_j \in E_i$ **do**
      $\overline{R^{(t+1)}}\tilde{X}(x_j) = \sup\{\tilde{X}(y)|y \in E_i\}$;
    end
    if $E_i \subseteq X$ **then**
      $\underline{R^{(t+1)}}X = \underline{R^{(t+1)}}X \cup E_i$;
      $\underline{R^{(t+1)}}\tilde{X}(x_j) = \inf\{\tilde{X}(y)|y \in E_i\}$;
    end
  end
end

Algorithm 1: The NDMARFS algorithm.

Let $n = |U|$ in Algorithm 1. There are one sup function and one inf function. It is easy to proof that the time complexity of sup and inf functions is $O(n\log n)$. Therefore, the time complexity of Algorithm 1 is $O(n\log n)$.

3.3. Dynamic maintenance of approximations based on RFS

Because the variation of objects includes the immigration and the emigration, we will discuss the dynamic maintenance principles of approximations when objects immigrate or emigrate, respectively.

3.3.1. The immigration of objects

Suppose there are some objects U_{im} that enter into the FDS at time $t+1$. So $U^{'} = U \cup U_{im}$, $\forall B \subseteq C$, $U^{'}/R = \{E_{l_1}, ..., E_{l_p}, E^{+}_{k_1}, ..., E^{+}_{k_q}, E^{+}_{g_1}, ..., E^{+}_{g_r}\}$. $E_{l_1}, ..., E_{l_p}$ denote equivalence classes whose objects are not changed. $E^{+}_{k_1}, ..., E^{+}_{k_q}$ denote the equivalence classes that objects immigrate into. $E^{+}_{g_1}, ..., E^{+}_{g_r}$ denote the new generated equivalence classes. $E_{k_1}, ..., E_{k_q}$ denote the equivalence classes before the immigration. $\forall E^{+}_{k_j} \in \{E^{+}_{k_1}, ..., E^{+}_{k_q}\}$, $E^{+}_{k_j} = E_{k_j} \cup \{x^{+}_{k_{j1}}, ..., x^{+}_{k_{js}}\}, E_{k_j} \cap \{x^{+}_{k_{j1}}, ..., x^{+}_{k_{js}}\} = \emptyset$, where $x^{+}_{k_{j1}}, ..., x^{+}_{k_{js}}$ immigrate into E_{k_j} at time $t+1$.

Proposition 2. $\forall E_{l_j} \in \{E_{l_1}, ..., E_{l_p}\}$, $\forall x_i \in E_{l_j}$, the following results hold:

$$\underline{R^{(t+1)}}\tilde{X}(x_i) = \underline{R^{(t)}}\tilde{X}(x_i);$$
$$\overline{R^{(t+1)}}\tilde{X}(x_i) = \overline{R^{(t)}}\tilde{X}(x_i). \tag{5}$$

Proposition 3. When some objects $x^{+}_{1}, ..., x^{+}_{m}$ enter into FDS = $(U, C \cup D, V, f)$, $\forall B \subseteq C$. The following results hold:

(1) If $\exists x^{+} \in \{x^{+}_{1}, ..., x^{+}_{m}\}$, $\forall x \in U, f(x^{+}, B) \neq f(x, B)$. It means that the immigration of $\{x^{+}_{1}, ..., x^{+}_{m}\}$ can form new equivalence classes.

(2) If $\forall x^{+} \in \{x^{+}_{1}, ..., x^{+}_{m}\}$, $\exists x \in U, f(x^{+}, B) = f(x, B)$. It means that the immigration of $\{x^{+}_{1}, ..., x^{+}_{m}\}$ does not form new equivalence classes.

Proposition 4. $\forall E^{+}_{g_j} \in \{E^{+}_{g_1}, ..., E^{+}_{g_r}\}$, the following results hold:

$$(1)\ \underline{R^{(t+1)}}X = \begin{cases} \underline{R^{(t)}}X, & E^{+}_{g_j} \nsubseteq X \\ \underline{R^{(t)}}X \cup E^{+}_{g_j}, & E^{+}_{g_j} \subseteq X, \end{cases} \tag{6}$$

$$\underline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \inf\{\tilde{X}(y) | y \in E^{+}_{g_j}\}, & x_i \in E^{+}_{g_j}\ and\ x_i \in \underline{R^{(t+1)}}X \\ 0, & x_i \in E^{+}_{g_j}\ and\ x_i \notin \underline{R^{(t+1)}}X \end{cases} \tag{7}$$

$$(2)\ \overline{R^{(t+1)}}X = \begin{cases} \overline{R^{(t)}}X, & E^{+}_{g_j} \cap X = \emptyset \\ \overline{R^{(t)}}X \cup E^{+}_{g_j}, & E^{+}_{g_j} \cap X \neq \emptyset, \end{cases} \tag{8}$$

$$\overline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \sup\{\tilde{X}(y) | y \in E^{+}_{g_j}\}, & x_i \in E^{+}_{g_j}\ and\ x_i \in \overline{R^{(t+1)}}X \\ 0, & x_i \in E^{+}_{g_j}\ and\ x_i \notin \overline{R^{(t+1)}}X \end{cases} \tag{9}$$

Example 3.2. Shown as Table 3, two new objects $x^{+}_{1} = (x_9, Slight, Yes, Moderate, 0, 0.5)$, $x^{+}_{2} = (x_{10}, Very\ Slight, Yes, Moderate, 0.6, 0.5)$ immigrate into the FDS based on Example 2.1. $\tilde{X}(x_9) = 0$, and $\tilde{X}(x_{10}) = 0.6$. Therefore, $x_9 \notin X$, and $x_{10} \in X$.

Table 3. The immigration of objects forms new classes.

U	**Headache(a)**	**Muscle Pain(b)**	**Temperature(c)**	**Fuzzy attribute d**		
				Flu	**No Flu**	
x_1	Heavy	Yes	High	0.7	0.3	
x_2	Heavy	Yes	High	0	1	
x_3	Moderate	Yes	High	0.7	0.4	
x_4	Moderate	Yes	Normal	0.8	0.2	
x_5	Moderate	No	Normal	0	1	
x_6	No	No	Normal	0.5	0.7	
x_7	No	No	Normal	0.2	0.8	
x_8	No	Yes	High	0.6	0.4	
x_9	Slight	Yes	High	0	0.5	new object
x_{10}	Very Slight	Yes	Moderate	0.6	0.5	new object

$U'/R = \{E_1, E_2, E_3, E_4^+, E_5^+\} = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6, x_7, x_8\}, \{x_9\}, \{x_{10}\}\}$. Therefore, the immigration of $\{x_1^+, x_2^+\}$ forms new classes E_4^+ and E_5^+.

Because $x_9 \notin X$, new class $E_4^+ \nsubseteq X$, according to Proposition 4, $x_9 \notin \underline{R^{(t+1)}}X$, and $x_9 \notin \overline{R^{(t+1)}}X$. Therefore, $\underline{R^{(t+1)}}\tilde{X}(x_9) = \overline{R^{(t+1)}}\tilde{X}(x_9) = 0$.

Because $x_{10} \in X$, new class $E_5^+ \subset X$, $\underline{R^{(t+1)}}\tilde{X}(x_{10}) = \tilde{X}(x_{10}) = 0.6, \overline{R^{(t+1)}}\tilde{X}(x_{10}) = \tilde{X}(x_{10}) = 0.6$. Therefore,

$\underline{R^{(t+1)}}\tilde{X} = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8, 0/x_9, 0.6/x_{10}\}$;

$\overline{R^{(t+1)}}\tilde{X} = \{0.7/x_1, 0.7/x_2, 0.8/x_3, 0.8/x_4, 0.8/x_5, 0.6/x_6, 0.6/x_7, 0.6/x_8, 0/x_9, 0.6/x_{10}\}$.

Proposition 5. $\forall E_{k_j}^+ \in \{E_{k_1}^+, ..., E_{k_q}^+\}$, let E_{k_j} be a partition at time t, $x_{k_{j1}}^+, ..., x_{k_{js}}^+$ be the objects immigrating into E_{k_j} and $E_{k_j}^+ = E_{k_j} \cup \{x_{k_{j1}}^+, ..., x_{k_{js}}^+\}$. The following results hold:

$$\underline{R^{(t+1)}}X = \begin{cases} \underline{R^{(t)}}X \cup \{x_{k_{j1}}^+, ..., x_{k_{js}}^+\}, & E_{k_j}^+ \subseteq X \\ \underline{R^{(t)}}X - E_{k_j}, & E_{k_j} \subseteq X \textit{ and } E_{k_j}^+ \nsubseteq X \\ \underline{R^{(t)}}X, & \textit{otherwise}. \end{cases} \tag{10}$$

$$\underline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \{\underline{R^{(t)}}\tilde{X}(y) | y \in E_{k_j}\} \wedge \inf\{\tilde{X}(x_{k_{j1}}^+), ..., \tilde{X}(x_{k_{js}}^+)\}, & x_i \in E_{k_j}^+ \cap \underline{R^{(t+1)}}X \\ 0, & x_i \in E_{k_j}^+ \textit{ and } x_i \notin \underline{R^{(t+1)}}X. \end{cases} \tag{11}$$

Proof:

(1) If $\forall E_{k_j}^+ \in \{E_{k_1}^+, ..., E_{k_q}^+\}$, $E_{k_j}^+ \subseteq X$, because $E_{k_j}^+ = E_{k_j} \cup \{x_{k_{j1}}^+, ..., x_{k_{js}}^+\}$, $E_{k_j} \subseteq X$ and $\{x_{k_{j1}}^+, ..., x_{k_{js}}^+\} \subseteq X$. It is clear that $E_{k_j} \subseteq \underline{R^{(t)}}X$, $E_{k_j} \subseteq \underline{R^{(t+1)}}X$, $\{x_{k_{j1}}^+, ..., x_{k_{js}}^+\} \subseteq \underline{R^{(t+1)}}X$. Therefore, $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup \{x_{k_{j1}}^+, ..., x_{k_{js}}^+\}$.

If $E_{k_j} \subseteq X$ *and* $E_{k_j}^+ \nsubseteq X$, because $E_{k_j} \subseteq X$, $E_{k_j} \subseteq \underline{R^{(t)}}X$, $E_{k_j}^+ \nsubseteq \underline{R^{(t+1)}}X$. It is clear that $E_{k_j} \nsubseteq \underline{R^{(t+1)}}X$. Therefore, $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X - E_{k_j}$.

If $E_{k_j} \nsubseteq X$, then $E^+_{k_j} \nsubseteq X$. It is clear that $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X$.

(2) According to Definition 2, $\forall x_i \in E^+_{k_j} \cap \underline{R^{(t+1)}}X$, $\underline{R^{(t+1)}}\tilde{X}(x_i) = \inf\{\tilde{X}(y)|y \in E^+_{k_j}\} = \inf\{\tilde{X}(y)|y \in E_{k_j} \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}\} = \inf\{\tilde{X}(y)|y \in E_{k_j}\} \wedge \inf\{\tilde{X}(x^+_{k_{j1}}), ..., \tilde{X}(x^+_{k_{js}})\}$.

If $\forall x_i \in E^+_{k_j}$ and $x_i \notin \underline{R^{(t+1)}}X$, it is clear that $\underline{R^{(t+1)}}\tilde{X}(x_i) = 0$. □

Proposition 6. $\forall E^+_{k_j} \in \{E^+_{k_1}, ..., E^+_{k_q}\}$, let E_{k_j} be a partition at time t, $x^+_{k_{j1}}, ..., x^+_{k_{js}}$ be the objects immigrating into E_{k_j} and $E^+_{k_j} = E_{k_j} \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}$. The following results hold:

$$\overline{R^{(t+1)}}X = \begin{cases} \overline{R^{(t)}}X \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}, & E_{k_j} \cap X \neq \emptyset \\ \overline{R^{(t)}}X \cup E^+_{k_j}, & E_{k_j} \cap X = \emptyset \; and \; E^+_{k_j} \cap X \neq \emptyset \\ \overline{R^{(t)}}X, & otherwise, \end{cases} \tag{12}$$

$$\overline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \overline{R^{(t)}}\tilde{X}(x_i) \vee \sup\{\tilde{X}(x^+_{k_{j1}}), ..., \tilde{X}(x^+_{k_{js}})\}, & \\ \qquad x_i \in E^+_{k_j}, E_{k_j} \cap X \neq \emptyset, x_i \in \overline{R^{(t+1)}}X & \\ \sup\{\tilde{X}(y)|y \in E^+_{k_j}\}, \quad x_i \in E^+_{k_j}, E_{k_j} \cap X = \emptyset \; and \; x_i \in \overline{R^{(t+1)}}X & \\ 0, \qquad x_i \in E^+_{k_j} \; and \; x_i \notin \overline{R^{(t+1)}}X. & \end{cases} \tag{13}$$

Proof:

(1) If $\forall E^+_{k_j} \in \{E^+_{k_1}, ..., E^+_{k_q}\}$, $E_{k_j} \cap X \neq \emptyset$, because $E^+_{k_j} = E_{k_j} \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}$, $E^+_{k_j} \cap X \neq \emptyset$. It is clear that $E_{k_j} \subseteq \overline{R^{(t)}}X$, $E_{k_j} \subseteq \overline{R^{(t+1)}}X$, $\{x^+_{k_{j1}}, ..., x^+_{k_{js}}\} \subseteq \overline{R^{(t+1)}}X$. Therefore, $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}$.

If $E_{k_j} \cap X = \emptyset$ and $E^+_{k_j} \cap X \neq \emptyset$, $E^+_{k_j} \subseteq \overline{R^{(t+1)}}X$, it is clear that $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X \cup E^+_{k_j}$.

If $E_{k_j} \cap X = \emptyset$ and $E^+_{k_j} \cap X = \emptyset$, it is clear that $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X$.

(2) According to Definition 2, $\forall x_i \in E^+_{k_j}, E_{k_j} \cap X \neq \emptyset$ and $x_i \in \overline{R^{(t+1)}}X, \overline{R^{(t+1)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E^+_{k_j}\} = \sup\{\tilde{X}(y)|y \in E_{k_j} \cup \{x^+_{k_{j1}}, ..., x^+_{k_{js}}\}\} = \sup\{\tilde{X}(y)|y \in E_{k_j}\} \vee \sup\{\tilde{X}(x^+_{k_{j1}}), ..., \tilde{X}(x^+_{k_{js}})\}$.

According to Definition 2, $\forall x_i \in E^+_{k_j}$, $E_{k_j} \cap X \neq \emptyset$ and $x_i \in \overline{R^{(t+1)}}X$, $\overline{R^{(t+1)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E^+_{k_j}\}$.

Finally, it is clear that $\forall x_i \in E^+_{k_j}$ and $x_i \notin \overline{R^{(t+1)}}X, \overline{R^{(t+1)}}\tilde{X}(x_i) = 0$. □

Example 3.3. Shown as Table 4, three new objects, $x^+_3 = (x_{11}, Slight, Yes, Moderate, 0, 0.3)$, $x^+_4 = (x_{12}, Very\ \ Slight, Yes, Moderate, 0, 0.5)$, $x^+_5 = (x_{13}, No, Yes, Moderate, 0.7, 0.5)$ immigrate into the FDS based on Example 3.2. $\tilde{X}(x_{11}) = 0, \tilde{X}(x_{12}) = 0, \tilde{X}(x_{13}) = 0.7$. Therefore, $x_{11} \notin X$, $x_{12} \notin X$, and $x_{13} \in X$.

$U'/R = \{E_1, E_2, E^+_3, E^+_4, E^+_5\} = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6, x_7, x_8, x_{13}\}, \{x_9, x_{11}\}, \{x_{10}, x_{12}\}\}$. There are three equivalence classes E^+_3, E^+_4, E^+_5 which have objects immigrating into, $x^+_3 \in E^+_4, x^+_4 \in E^+_5, x^+_5 \in E^+_3$.

Table 4. The immigration of objects does not form new classes.

U	**Headache**(a)	**Muscle Pain**(b)	**Temperature**(c)	**Fuzzy attribute** d		
				Flu	**No Flu**	
x_1	Heavy	Yes	High	0.7	0.3	
x_2	Heavy	Yes	High	0	1	
x_3	Moderate	Yes	High	0.7	0.4	
x_4	Moderate	Yes	Normal	0.8	0.2	
x_5	Moderate	No	Normal	0	1	
x_6	No	No	Normal	0.5	0.7	
x_7	No	No	Normal	0.2	0.8	
x_8	No	Yes	High	0.6	0.4	
x_9	Slight	Yes	High	0	0.5	
x_{10}	Very Slight	Yes	Moderate	0.6	0.5	
x_{11}	Slight	Yes	High	0	0.3	new object
x_{12}	Very Slight	Yes	Moderate	0	0.5	new object
x_{13}	No	Yes	Moderate	0.7	0.5	new object

Because $x_{11} \notin X$, $E_4^+ \not\subseteq X$, according to Propositions 5 and 6, $x_{11} \notin \underline{R}^{(t+1)}X$, and $x_{11} \notin \overline{R}^{(t+1)}X$. Therefore, $\underline{R}^{(t+1)}\tilde{X}(x_{11}) = \overline{R}^{(t+1)}\tilde{X}(x_{11}) = 0$.

Because $x_{12} \notin X$, $E_5 \subset X$, $E_5^+ \not\subseteq X$, according to Proposition 5, x_{10} should be deleted from $\underline{R}^{(t+1)}X$. Therefore, $\underline{R}^{(t+1)}\tilde{X}(x_{12}) = \underline{R}^{(t+1)}\tilde{X}(x_{10}) = 0$.

Because $x_{10} \in X$, $E_5^+ \cap X \neq \emptyset$, according to Proposition 6, x_{12} should be added into $\overline{R}^{(t+1)}X$. $\overline{R}^{(t+1)}\tilde{X}(x_{12}) = \overline{R}^{(t+1)}\tilde{X}(x_{10}) = \overline{R}^{(t)}\tilde{X}(x_{10}) \vee \tilde{X}(x_{12}) = 0.6 \vee 0 = 0.6$.

Because $x_{13} \in X$, $E_3 \subset X$, and $E_3^+ \subset X$, according to Propositions 5 and 6,

$\underline{R}^{(t+1)}\tilde{X}(x_{13}) = \underline{R}^{(t+1)}\tilde{X}(x_6) = \underline{R}^{(t+1)}\tilde{X}(x_7) = \underline{R}^{(t+1)}\tilde{X}(x_8) = \underline{R}^{(t)}\tilde{X}(E_3) \wedge \tilde{X}(x_{13}) = 0.2 \wedge 0.7 = 0.2$.

$\overline{R}^{(t+1)}\tilde{X}(x_{13}) = \overline{R}^{(t+1)}\tilde{X}(x_6) = \overline{R}^{(t+1)}\tilde{X}(x_7) = \overline{R}^{(t+1)}\tilde{X}(x_8) = \overline{R}^{(t)}\tilde{X}(E_3) \vee \tilde{X}(x_{13}) = 0.6 \vee 0.7 = 0.7$.

Therefore,

$\underline{R}^{(t+1)}\tilde{X} = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8, 0/x_9, 0/x_{10}, 0/x_{11}, 0/x_{12}, 0.2/x_{13}\}$,

$\overline{R}^{(t+1)}\tilde{X} = \{0.7/x_1, 0.7/x_2, 0.8/x_3, 0.8/x_4, 0.8/x_5, 0.7/x_6, 0.7/x_7, 0.7/x_8, 0/x_9, 0.6/x_{10}, 0/x_{11}, 0.6/x_{12}, 0.7/x_{13}\}$.

Input: (1) FDS=$(U, C \cup D, V, f)$ at time t; (2) $\underline{R^{(t)}}\tilde{X}, \overline{R^{(t)}}\tilde{X}$.
Output: $\underline{R^{(t+1)}}\tilde{X}, \overline{R^{(t+1)}}\tilde{X}$ at time $t+1$.
Find the partitions $\{E_{l_1}, ..., E_{l_p}\}$,$\{E^+_{k_1}, ..., E^+_{k_q}\}$,$\{E^+_{g_1}, ..., E^+_{g_r}\}$;
for *each* $E_{l_i} \in \{E_{l_1}, ..., E_{l_p}\}$ **do**
  for *each* $x_j \in E_{l_i}$ **do**
    $\underline{R^{(t+1)}}\tilde{X}(x_j) = \underline{R^{(t)}}\tilde{X}(x_j)$, $\overline{R^{(t+1)}}\tilde{X}(x_j) = \overline{R^{(t)}}\tilde{X}(x_j)$;
  end
end
$\overline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup E^+_{g_1} \cup ... \cup E^+_{g_r}$;
for *each* $E^+_{g_i} \in \{E^+_{g_1}, ..., E^+_{g_r}\}$ **do**
  for *each* $x_j \in E^+_{g_i}$ **do**
    $\underline{R^{(t+1)}}\tilde{X}(x_j) = \inf\{\tilde{X}(y)|y \in E^+_{g_i}\}$, $\overline{R^{(t+1)}}\tilde{X}(x_j) = \sup\{\tilde{X}(y)|y \in E^+_{g_i}\}$;
  end
end
for *each* $E^+_{k_i} \in \{E^+_{k_1}, ..., E^+_{k_q}\}$ **do**
  if $E_{k_i} \subseteq X$ *and* $E^+_{k_i} \subseteq X$ **then**
    $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup \{x^+_{k_{i1}}, ..., x^+_{k_{is}}\}$;//$x^+_{k_{i1}}, ..., x^+_{k_{is}}$ are new objects
    for *each* $x_j \in E^+_{k_i}$ **do**
      $\underline{R^{(t+1)}}\tilde{X}(x_j) = \underline{R^{(t)}}\tilde{X}(x_j) \wedge \inf\{\tilde{X}(x^+_{k_{i1}}), ..., \tilde{X}(x^+_{k_{is}})\}$;
    end
  end
  if $E_{k_i} \subseteq X$ *and* $E^+_{k_i} \nsubseteq X$ **then**
    $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X - E_{k_i}$;
    for *each* $x_j \in E^+_{k_i}$ **do**
      $\underline{R^{(t+1)}}\tilde{X}(x_j) = 0$;
    end
  end
  if $E_{k_i} \cap X \neq \emptyset$ **then**
    $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X \cup \{x^+_{k_{i1}}, ..., x^+_{k_{is}}\}$;//$x^+_{k_{i1}}, ..., x^+_{k_{is}}$ are the new objects
    for *each* $x_j \in E^+_{k_i}$ **do**
      $\overline{R^{(t+1)}}\tilde{X}(x_j) = \overline{R^{(t)}}\tilde{X}(x_j) \wedge \inf\{\tilde{X}(x^+_{k_{i1}}), ..., \tilde{X}(x^+_{k_{is}})\}$;
    end
  end
  if $E_{k_i} \cap X = \emptyset$ *and* $E^+_{k_i} \cap X \neq \emptyset$ **then**
    $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X \cup E^+_{k_i}$;
    for *each* $x_j \in E^+_{k_i}$ **do**
      $\overline{R^{(t+1)}}\tilde{X}(x_j) = \sup\{\tilde{X}(y)|y \in E^+_{k_i}\}$;
    end
  end
end

Algorithm 2: The DMARFS-MI algorithm.

The algorithm for Dynamic Maintenance of Approximations based on RFS when Multiple objects Immigrating (DMARFS-MI) is outlined in Algorithm 2.

Let l be the number of objects in the unchanged classes (E_l), g be the number of objects in new classes (E_g), k be the number of objects in the changing classes (E_k), and s be the number of new objects in the changing classes. $l + g + k = n, n = |U|$. In Algorithm 2, the time complexity of finding the partitions is $O(|C|(g + s))$. Because there are $2l$ times assignment operations to deal with the unchanged classes, the time complexity is $O(l)$. There are one sup function and one inf function to compute the membership of objects in new classes, and the time complexity is $O(g\log g)$. To objects in the changing classes, there are two steps to computing the lower and upper approximations. Step 1 is to obtain the objects included in the lower and upper approximations of X. There are $2k$ times assignment operations. Step 2 is to compute the memberships. There are one sup function and one inf function in $s+1$ objects, and the time complexity is $O(s\log s)$. Therefore, the whole time complexity of Algorithm 2 is $O(|C|(g+s)+g\log g+s\log s+l+k)$. When $n \approx l+k >> g+s$, it means that there are only a little part of objects entering into the system, and the time complexity is $O(n)$. When $n \approx g+s >> (l+k)$, it means that there are a great part of objects entering into the system, and the time complexity is $O(|C|(g+s)+g\log g+s\log s) \approx O(n\log n)$.

3.4. The emigration of objects

Suppose there are some objects U_{em} that emigrate from the FDS at time $t+1$. So $U' = U - U_{em}$, $\forall B \subseteq C$, $U'/R = \{E_{l_1},...,E_{l_p},E^-_{k_1},...,E^-_{k_q}\}$. $E_{l_1},...,E_{l_p}$ denote the unchanged equivalence classes at time $t+1$ in which no object has been deleted. $E_{k_1},...,E_{k_q}$ denote the emigration classes before emigrating, $E^-_{k_1},...,E^-_{k_q}$ denote the emigration classes at time $t+1$, respectively. For example, $E^-_{k_1} = E_{k_1} - \{x^-_{k_{j1}},...,x^-_{k_{js}}\}, E^-_{k_1} \neq \emptyset$, where $x^-_{k_{j1}},...,x^-_{k_{js}}$ emigrate from E_{k_1} at time $t+1$. There are some equivalence classes $\{E_{e_1},...,E_{e_r}\}$ in U/R where all objects are deleted at time $t+1$ (shortly, $E^-_{e_i} = \emptyset$).

Proposition 7. $\forall E_{l_j} \in \{E_{l_1},...,E_{l_p}\}, \forall x_i \in E_{l_j}$, the following results hold:
(1) $\underline{R^{(t+1)}}\tilde{X}(x_i) = \underline{R^{(t)}}\tilde{X}(x_i)$;
(2) $\overline{R^{(t+1)}}\tilde{X}(x_i) = \overline{R^{(t)}}\tilde{X}(x_i)$.

Proposition 8. When some objects $x^-_1,...,x^-_m$ get out of $FDS = (U, C \cup D, V, f)$, $\forall B \subseteq C$. One of the following results holds:

(1) If $\exists x^- \in \{x^-_1,...,x^-_m\}, x^- \in E_k, v = f(x^-,B), |E^-_k| = 0$. It means the emigration of $\{x^-_1,...,x^-_m\}$ eliminates an equivalence class with v;

(2) If $\forall x^- \in \{x^-_1,...,x^-_m\}, x^- \in E_k$, $v = f(x^-,B), |E^-_k| > 0$. It means the emigration of $\{x^-_1,...,x^-_m\}$ does not eliminate equivalence classes.

Proposition 9. $\forall E^-_{k_j} \in \{E^-_{k_1},...,E^-_{k_p}\}$, let $x^-_{k_{j1}},...,x^-_{k_{js}}$ be the objects emigrating from E_{k_j}, and $E^-_{k_j} = E_{k_j} - \{x^-_{k_{j1}},...,x^-_{k_{js}}\}, E^-_{k_j} \neq \emptyset$. The following results hold:

$$\underline{R^{(t+1)}}X = \begin{cases} \underline{R^{(t)}}X \cup E^-_{k_j}, & E_{k_j} \nsubseteq X and\, E^-_{k_j} \subseteq X \\ \underline{R^{(t)}}X - \{x^-_{k_{j1}},...,x^-_{k_{js}}\}, & E_{k_j} \subseteq X and\, E^-_{k_j} \subseteq X \\ \underline{R^{(t)}}X, & otherwise, \end{cases} \tag{14}$$

$$\underline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \inf\{\tilde{X}(y)|y \in E^-_{k_j}\}, & x_i \in E^-_{k_j}\ and\ x_i \in \underline{R^{(t+1)}}X \\ \underline{R^{(t)}}\tilde{X}(x_i), & otherwise. \end{cases}$$

Proof:

(1) If $E_{k_j} \nsubseteq X, E^-_{k_j} \subseteq X$, it is clear that $E_{k_j} - \{x^-_{k_{j1}},...,x^-_{k_{js}}\} \nsubseteq \underline{R^{(t)}}X,\ E_{k_j} - \{x^-_{k_{j1}},...,x^-_{k_{js}}\} \subseteq \underline{R^{(t+1)}}X$. Therefore, $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup E_{k_j} - \{x^-_{k_{j1}},...,x^-_{k_{js}}\} = \underline{R^{(t)}}X \cup E^-_{k_j}$.

If $E^-_{k_j} \subseteq X\ and\ E_{k_j} \subseteq X,\ E_{k_j} \subseteq \underline{R^{(t)}}X,\ E^-_{k_j} \subseteq \underline{R^{(t+1)}}X$. It is clear that $E^-_{k_j} \subseteq \underline{R^{(t)}}X$ and $\{x^-_{k_{j1}},...,x^-_{k_{js}}\} \nsubseteq \underline{R^{(t+1)}}X$. Therefore, $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X - \{x^-_{k_{j1}},...,x^-_{k_{js}}\}$.

If $E^-_{k_j} \nsubseteq X$, it is clear that $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X$.

(2) $\forall x_i \in E^-_{k_j}$, if $x_i \notin \underline{R^{(t+1)}}X$, then $x_i \notin \underline{R^{(t)}}X$. It is clear that $\underline{R^{(t+1)}}\tilde{X}(x_i) = \underline{R^{(t)}}\tilde{X}(x_i) = 0$. If $x_i \in \underline{R^{(t+1)}}X$, according to Definition 2, $\underline{R^{(t+1)}}\tilde{X}(x_i) = \inf\{\tilde{X}(y)|y \in E^-_{k_j}\}$. □

Proposition 10. $\forall E^-_{k_j} \in \{E^-_{k_1}, ..., E^-_{k_p}\}$, let $x^-_{k_{j1}}, ..., x^-_{k_{js}}$ be the emigration objects, and $E^-_{k_j} = E_{k_j} - \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\}, E^-_{k_j} \neq \emptyset$. The following results hold:

$$\overline{R^{(t+1)}}X = \begin{cases} \overline{R^{(t)}}X - E_{k_j}, & E_{k_j} \cap X \neq \emptyset \; and \; E^-_{k_j} \cap X = \emptyset \\ \overline{R^{(t)}}X - \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\}, & E_{k_j} \cap X \neq \emptyset \; and \; E^-_{k_j} \cap X \neq \emptyset \\ \overline{R^{(t)}}X, & otherwise, \end{cases} \tag{15}$$

$$\overline{R^{(t+1)}}\tilde{X}(x_i) = \begin{cases} \sup\{\tilde{X}(y)|y \in E^-_{k_j}\}, & \\ & x_i \in E^-_{k_j} \; and \; \sup\{\tilde{X}(x^-_{k_{j1}}), ..., \tilde{X}(x^-_{k_{js}})\} \geq \overline{R^{(t)}}\tilde{X}(x_i) \\ 0, & x_i \in E^-_{k_j} \; and \; x_i \notin \overline{R^{(t+1)}}X \\ \overline{R^{(t)}}\tilde{X}(x_i), & otherwise. \end{cases}$$

Proof:

(1) If $E^-_{k_j} \cap X = \emptyset, E_{k_j} \cap X \neq \emptyset$, it is clear that $E_{k_j} - \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\} \cap \overline{R^{(t+1)}}X = \emptyset$, $E_{k_j} \cap \overline{R^{(t+1)}}X = \emptyset, E_{k_j} \subseteq \overline{R^{(t)}}X$. Therefore, $\overline{R^{(t+1)}}\tilde{X} = \overline{R^{(t)}}X - E_{k_j}$.

If $E^-_{k_j} \cap X \neq \emptyset, E_{k_j} \cap X \neq \emptyset, E^-_{k_j} \subseteq \overline{R^{(t+1)}}X$, $E^-_{k_j} \subseteq \overline{R^{(t)}}X$. Because $E_{k_j} \subseteq \overline{R^{(t)}}X$ and $E_{k_j} = E^-_{k_j} \cup \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\}$. It is clear that $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X - \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\}$.

If $E_{k_j} \cap X = \emptyset \; and \; E^-_{k_j} \cap X = \emptyset$, it is clear that $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X$.

(2) If $x_i \in E^-_k \; and \; x_i \notin \overline{R^{(t+1)}}X$, according to Definition 2, $\overline{R^{(t+1)}}\tilde{X}(x_i) = 0$.

According to Definition 2, $\forall x_i \in E_{k_j}, E_{k_j} \cap X \neq \emptyset$, $\overline{R^{(t)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E_{k_j}\} = \sup\{\tilde{X}(y)|y \in E^-_{k_j} \cup \{x^-_{k_{j1}}, ..., x^-_{k_{js}}\}\} = \sup\{\tilde{X}(y)|y \in E^-_{k_j}\} \vee \sup\{\tilde{X}(x^-_{k_{j1}}), ..., \tilde{X}(x^-_{k_{js}})\}$. If $\sup\{\tilde{X}(x^-_{k_{j1}}), ..., \tilde{X}(x^-_{k_{js}})\} \geq \sup\{\tilde{X}(y)|y \in E^-_{k_j}\}$, $\overline{R^{(t)}}\tilde{X}(x_i) = \sup\{\tilde{X}(x^-_{k_{j1}}), ..., \tilde{X}(x^-_{k_{js}})\}$. Otherwise, $\overline{R^{(t)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E^-_{k_j}\}$. When $\overline{R^{(t)}}\tilde{X}(x_i) = \sup\{\tilde{X}(x^-_{k_{j1}}), ..., \tilde{X}(x^-_{k_{js}})\}$ and $x^-_{k_{j1}}, \dots,$ $x^-_{k_{js}}$ are deleted from the FDS, $\overline{R^{(t+1)}}\tilde{X}(x_i)$ needs to be computed again. Therefore, $\overline{R^{(t+1)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E^-_{k_j}\}$. When $\overline{R^{(t)}}\tilde{X}(x_i) = \sup\{\tilde{X}(y)|y \in E^-_{k_j}\}$, the emigration of $x^-_{k_{j1}}, ..., x^-_{k_{js}}$ does not affect the results of $E^-_{k_j}$. Therefore, $\overline{R^{(t+1)}}\tilde{X}(x_i) = \overline{R^{(t)}}\tilde{X}(x_i)$. □

Example 3.4. Shown as Table 5, three objects $x^-_1 = x_2, x^-_2 = x_{10}, x^-_3 = x_{13}$ emigrate from the FDS based on Example 3.3. We have, $U'/R = \{E^-_1, E_2, E^-_3, E_4, E^-_5\} = \{\{x_1\}, \{x_3, x_4, x_5\}, \{x_6, x_7, x_8\}, \{x_9, x_{11}\}, \{x_{12}\}\}$. There are three classes are changed but no class is deleted (E_1, E_3, E_5 are changed to E^-_1, E^-_3, E^-_5, respectively).

Because $x_1, x_2 \in E_1$, $E_1 \nsubseteq X$, and $E^-_1 \subseteq X$, according to Proposition 9, $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup \{x_1\}$ and $\underline{R^{(t+1)}}\tilde{X}(x_1) = \tilde{X}(x_1) = 0.7$. According to Proposition 10, $\overline{R^{(t+1)}}\tilde{X} = \overline{R^{(t)}}\tilde{X} - \{x_2\}$. Because $\tilde{X}(x_1) = 0.7 = \overline{R^{(t)}}\tilde{X}(x_1) = 0.7$, $\overline{R^{(t+1)}}\tilde{X}(x_1) = \tilde{X}(x_1) = 0.7$.

Table 5. The emigration does not eliminate classes.

U	**Headache(a)**	**Muscle Pain(b)**	**Temperature(c)**	**Fuzzy attribute d**		
				Flu	**No Flu**	
x_1	Heavy	Yes	High	0.7	0.3	
x_2	Heavy	Yes	High	0	1	×
x_3	Moderate	Yes	High	0.7	0.4	
x_4	Moderate	Yes	Normal	0.8	0.2	
x_5	Moderate	No	Normal	0	1	
x_6	No	No	Normal	0.5	0.7	
x_7	No	No	Normal	0.2	0.8	
x_8	No	Yes	High	0.6	0.4	
x_9	Slight	Yes	High	0	0.5	
x_{10}	Very Slight	Yes	Moderate	0.6	0.5	
x_{11}	Slight	Yes	High	0	0.3	
x_{12}	Very Slight	Yes	Moderate	0	0.5	×
x_{13}	No	Yes	Moderate	0.7	0.5	×

Because $x_{10}, x_{12} \in E_5$, and $E_5, E_5^- \nsubseteq X$, according to Proposition 9, x_{12} should not be included into the lower approximations too. Because $E_5 \cap X \neq \emptyset$, and $E_5^- \cap X = \emptyset$, according to Proposition 10, $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X - \{x_{12}\}$. $\tilde{X}(x_{12}) = 0 < \overline{R^{(t)}}\tilde{X}(E_5) = 0.6$. Therefore, $\overline{R^{(t)}}\tilde{X}(x_{10}) = 0.6$.

Because $x_{13} \in E_3$, and $E_3, E_3^- \subset X$, according to Proposition 9, x_{13} should be deleted from the lower approximations. $\tilde{X}(x_{13}) = 0.7 \geq \underline{R^{(t)}}\tilde{X}(E_3) = 0.2$. To the ojects x_6, x_7, x_8 in E_3^-, the lower approximations should not be changed. According to Proposition 10, $\tilde{X}(x_{13}) = 0.7 \geq \overline{R^{(t)}}\tilde{X}(E_3) = 0.7$. Therefore, $\overline{R^{(t+1)}}\tilde{X}(x_6) = \overline{R^{(t+1)}}\tilde{X}(x_7) = \overline{R^{(t+1)}}\tilde{X}(x_8) = \sup\{\tilde{X}(x_6), \tilde{X}(x_7), \tilde{X}(x_8)\} = 0.6$.

$$\underline{R^{(t+1)}}\tilde{X} = \{0.7/x_1, 0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8, 0/x_9, 0.6/x_{10}, 0/x_{11}\};$$
$$\overline{R^{(t+1)}}\tilde{X} = \{0.7/x_1, 0.8/x_3, 0.8/x_4, 0.8/x_5, 0.6/x_6, 0.6/x_7, 0.6/x_8, 0/x_9, 0.6/x_{10}, 0/x_{11}\}.$$

To the classes which have been deleted, according to Proposition 11, their elements should be deleted from the lower and upper approximations.

Proposition 11. $\forall E_{e_j} \in \{E_{e_1}, ..., E_{e_r}\}$, the following results hold:

(1) $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X - E_{e_j}$, $\underline{R^{(t+1)}}\tilde{X} = \underline{R^{(t)}}\tilde{X} - A_{e_j}$,

(2) $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X - E_{e_j}$, $\overline{R^{(t+1)}}\tilde{X} = \overline{R^{(t)}}\tilde{X} - A_{e_j}$,

where $A_{e_j} = \{\tilde{X}(x_1)/x_1, ..., \tilde{X}(x_m)/x_m\}, x_1, ..., x_m \in E_{e_j}$ and m is the number of objects in E_{e_j}.

Example 3.5. In Table 6, three objects $x_1^- = x_1, x_2^- = x_{10}, x_3^- = x_{11}$ emigrate from the FDS based on Example 3.4. $U'/R = \{E_1 - \{x_1\}, E_2, E_3, E_4 - \{x_{10}\}, E_5 - \{x_{11}\}\} = \{E_2, E_3\} = \{\{x_3, x_4, x_5\}, \{x_6, x_7, x_8\}\}$. Therefore, there are three classes are deleted, e.g., E_1, E_4, E_5.

Table 6. The emigration can eliminate classes.

U	**Headache(a)**	**Muscle Pain(b)**	**Temperature(c)**	**Fuzzy attribute d**		
				Flu	**No Flu**	
x_1	Heavy	Yes	High	0.7	0.3	×
x_3	Moderate	Yes	High	0.7	0.4	
x_4	Moderate	Yes	Normal	0.8	0.2	
x_5	Moderate	No	Normal	0	1	
x_6	No	No	Normal	0.5	0.7	
x_7	No	No	Normal	0.2	0.8	
x_8	No	Yes	High	0.6	0.4	
x_9	Slight	Yes	High	0	0.5	
x_{10}	Very Slight	Yes	Moderate	0.6	0.5	×
x_{11}	Slight	Yes	High	0	0.3	×

According to Propositions 11, it is clear that x_1, x_{10}, x_{11} should not be in the lower and upper approximations. We have,

$$\underline{R^{(t+1)}}\tilde{X} = \{0/x_3, 0/x_4, 0/x_5, 0.2/x_6, 0.2/x_7, 0.2/x_8, 0/x_9\};$$
$$\overline{R^{(t+1)}}\tilde{X} = \{0.8/x_3, 0.8/x_4, 0.8/x_5, 0.6/x_6, 0.6/x_7, 0.6/x_8, 0/x_9\}.$$

The algorithm for Dynamic Maintenance of Approximations based on RFS when Multiple objects Emigrating (DMARFS-ME) is outlined in Algorithm 3.

Let l be the number of objects in the unchanged classes (E_l), e be the number of objects in deleted classes (E_e), k be the number of objects in the emigrating classes (E_k) and s be the number of deleted objects from emigrating classes. $l+e+k+s = n, n = |U|$. The time complexity of finding the partitions is $O(e + s)$. Because there are $2l$ times assignment operations to deal with the unchanged classes, the time complexity is $O(l)$. Since there are $2e$ times deleting operations to delete the emigrating objects from the deleted classes, the time complexity is $O(e)$. To the emigrating classes, there are two steps to compute the lower and upper approximations. Step 1 is to obtain the lower and upper approximations of X. There are $2s$ times deleting operations. Step 2 is to compute the membership. The time complexity is $O(k\text{log}k)$. Therefore, the whole time complexity of Algorithm 3 is $O(e + s + l{+}k\text{log}k)$. When $n \approx l >> (k+e+s)$, it means that there are only a little part of objects emigrating from the system, and the time complexity is $O(n)$. When $n \approx e >> (l + k)$, it means that there are a great part of objects and classes emigrating from the system, and the time complexity is $O(n)$. When $n \approx k >> (l+e)$, it means that there are a great part of emigrating classes, but little classes are deleted, and the time complexity is $O(k\text{log}k)$. Then the time complexity is $O(n\text{log}n)$.

4. Experimental evaluation

We compare the computational time of the non-incremental algorithm (NDMARFS) and the two incremental algorithms (DMARFS-MI and DMARFS-ME) on different data sets.

Input: (1) FDS=$(U, C \cup D, V, f)$ at time t; (2) $\underline{R^{(t)}}\tilde{X}, \overline{R^{(t)}}\tilde{X}$.
Output: $\underline{R^{(t+1)}}\tilde{X}, \overline{R^{(t+1)}}\tilde{X}$ at time $t+1$.
Find the partitions $\{E_{l_1}, ..., E_{l_p}\}$,$\{E^-_{k_1}, ..., E^-_{k_q}\}$,$\{E_{e_1}, ..., E_{e_r}\}$.
for *each* $E_{l_i} \in \{E_{l_1}, ..., E_{l_p}\}$ **do**
  for *each* $x_j \in E_{l_i}$ **do**
    $\underline{R^{(t+1)}}\tilde{X}(x_j) = \underline{R^{(t)}}\tilde{X}(x_j)$, $\overline{R^{(t+1)}}\tilde{X}(x_j) = \overline{R^{(t)}}\tilde{X}(x_j)$;
  end
end
for *each* $E_{e_i} \in \{E_{e_1}, ..., E_{e_r}\}$ **do**
  $\underline{R^{(t+1)}}\tilde{X} = \underline{R^{(t)}}\tilde{X} - A_{e_j}$;//$A_{e_j} = \{\tilde{X}(x_1)/x_1, ..., \tilde{X}(x_m)/x_m\}, x_1, ..., x_m \in E_{e_j}$.
  $\overline{R^{(t+1)}}\tilde{X} = \overline{R^{(t)}}\tilde{X} - A_{e_j}$;
end
for *each* $E^-_{k_i} \in \{E^-_{k_1}, ..., E^-_{k_q}\}$ **do**
  // deal with the changing partitions
  if $E_{k_i} \subseteq X$ *and* $E^-_{k_i} \subseteq X$ **then**
    $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X - \{x^-_{k_{i1}}, ..., x_{k^-_{is}}\}$;//$x^-_{k_{i1}}, ..., x^-_{k_{is}}$ are emigration objects
    for *each* $x_j \in E^-_{k_i}$ **do**
      if $\inf\{\tilde{X}(x^-_{k_{i1}}), ..., \tilde{X}(x^-_{k_{is}})\} > \underline{R^{(t)}}\tilde{X}(x_j)$ **then**
        $\underline{R^{(t+1)}}\tilde{X}(x_j) = \underline{R^{(t)}}\tilde{X}(x_j)$;
      end
      else
        $\underline{R^{(t+1)}}\tilde{X}(x_j) = \inf\{\tilde{X}(y)|y \in E^-_{k_i}\}$;
      end
    end
  end
  if $E_{k_i} \not\subseteq X$ *and* $E^-_{k_i} \subseteq X$ **then**
    $\underline{R^{(t+1)}}X = \underline{R^{(t)}}X \cup E^-_{k_i}$;
    $\underline{R^{(t+1)}}\tilde{X}(x_j) = \inf\{\tilde{X}(y)|y \in E^-_{k_i}\}$;
  end
  if $E^-_{k_i} \cap X \neq \emptyset$ **then**
    $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X - \{x^-_{k_{i1}}, ..., x^-_{k_{is}}\}$;//$x^-_{k_{i1}}, ..., x^-_{k_{is}}$ are emigration objects
    if $\sup\{\tilde{X}x^-_{k_{i1}}), ..., \tilde{X}(x^-_{k_{is}})\} \geq \overline{R^{(t)}}\tilde{X}(E_{k_i})$ **then**
      for *each* $x_j \in E^-_{k_i}$ **do**
        $\overline{R^{(t+1)}}\tilde{X}(x_j) = \sup\{\tilde{X}(y)|y \in E^-_{k_i}\}$;
      end
    end
    else
      $\overline{R^{(t+1)}}\tilde{X}(x_j) = \overline{R^{(t)}}\tilde{X}(x_j)$;
    end
  end
  if $E_{k_i} \cap X \neq \emptyset$ *and* $E^-_{k_i} \cap X = \emptyset$ **then**
    $\overline{R^{(t+1)}}X = \overline{R^{(t)}}X - E_{k_i}$;
    for *each* $x_j \in E^-_{k_i}$ **do**
      $\overline{R^{(t+1)}}\tilde{X}(x_j) = 0$;
    end
  end
end

Algorithm 3: The DMARFS-ME algorithm.

Experiments are performed on a 2.2 GHz, 4 cores and 8 threads Server with 4 GB of memory, running Win 7. Algorithms are coded in SQL Server 2005. Six data sets used here come from the UCI Machine Learning Repository (http://www.ics.uci.edu/)(shown as Table 7). Categorical decision attributes are redesigned to use the following steps to define the fuzzy membership: (1) Compute the mean of the objects in each decision class A_i as the class center c_i; (2) Compute the Euclidean distance $d(u, c_i)$ between all objects and each class center; (3) Compute each objects' membership $A_i(u) = 1 - \frac{d(u,c_i)}{max(d(u,c_1),...,d(u,c_m))}$, where m is the number of decision values. For example, the large nursery evaluation database consists of 17960 objects. The number of attributes is 8, all of which have been normalized. Using the membership functions, 5 fuzzy decision classes (namely, fuzzy concepts) can be obtained, namely, not_recom, $recommend$, $very\_recom$, $priority$ and $spec\_prior$. In our experiments, the 5 classes are denoted as A_1, A_2, A_3, A_4 and A_5, respectively. The time unit of incremental updating is second. The experiments calculate the average elapsed time by repeating the computing process for 100 times under the same conditions.

4.1. A comparison of NDMARFS and DMARFS-MI

We compare NDMARFS with DMARFS-MI on the six data sets shown in Table 7. The experimental results are presented in Figures 1-2.

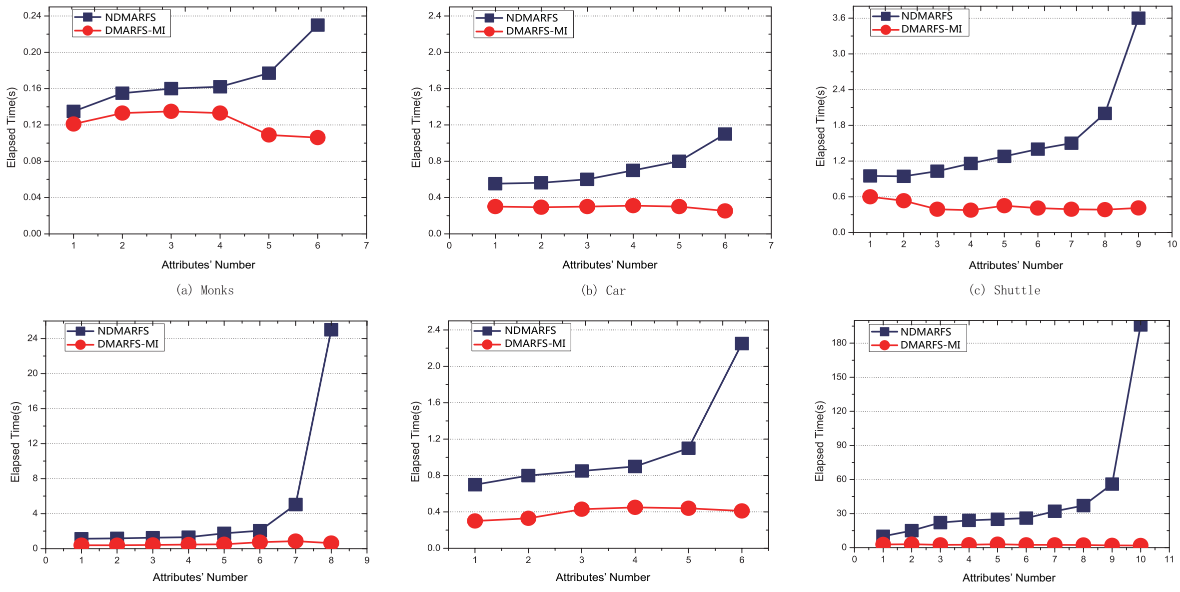
Table 7. A description of data sets.

ID	Data	Samples	Features	Decision Class
1	Monks	432	6	2
2	Car	1928	6	4
3	Shuttle	14500	9	7
4	Nursery	17960	8	5
5	Chess(King-Rook vs. King)	28056	6	18
6	Poker	1000000	10	10

In each sub-figure of Figure 1, the x-coordinate pertains to the attributes number of the data set (the 6 data sets starting from the smallest one), while the y-coordinate concerns the computational time. Here, we randomly select 10% objects of the data from the data set as the immigrant data, and the left 90% data as the original data. R_{im} denotes the ratio of the number of immigrant data and the whole data set, called the immigrant ratio [24]. Let $R_{im} = 10\%$. Figure 1 displays more detailed change trend of each of the two algorithms with the increasing number of attributes.

Shown as Figure 1, the elapsed time of DMARFS-MI is much smaller than NDMARFS on updating approximations when the same objects enter into the FDS. With the increasing number of attributes, the elapsed time is relatively stable in the incremental method (DMARFS-MI), but the elapsed time increases drastically in the non-incremental updating method (NDMARFS).

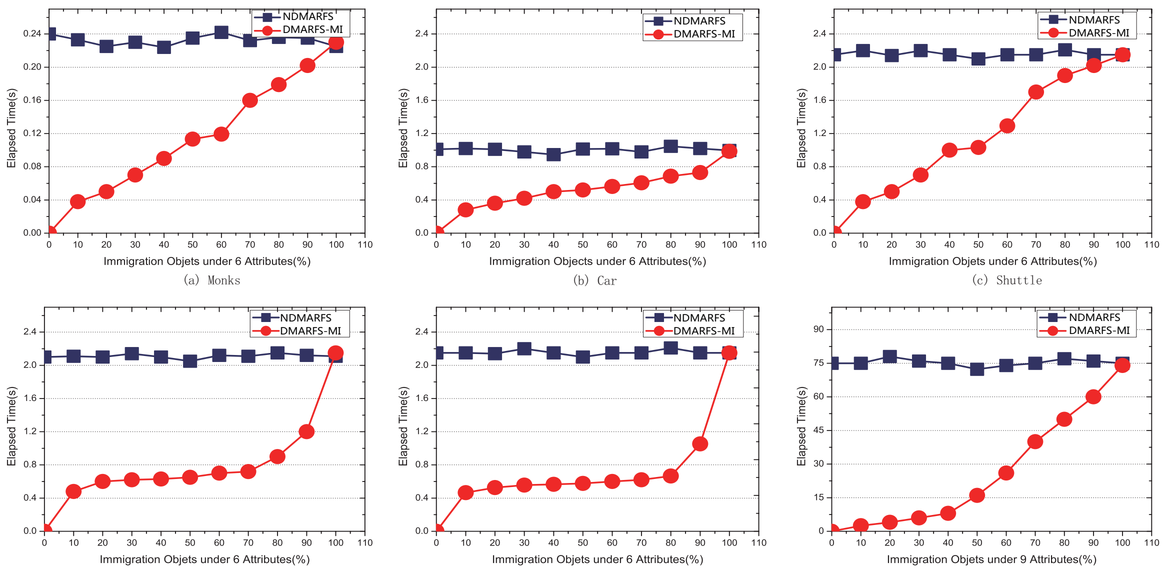
In each sub-figure of Figure 2, the x-coordinate pertains to the immigrant ratio R_{im}, while the y-coordinate concerns the elapsed time. Figure 2 displays more detailed change trend of each of two algorithms with the increasing value of R_{im}.



(a) Monks (b) Car (c) Shuttle

(d) Nursery (e) Chess(King-Rook vs. King) (f) Poker

Figure 1. Elapsed time of NDMARFS and DMARFS-MI versus the attributes' number.



(a) Monks (b) Car (c) Shuttle

(d) Nursery (e) Chess(King-Rook vs. King) (f) Poker

Figure 2. Elapsed time of NDMARFS and DMARFS-MI versus the immigrant ratio R_{im}.

Shown as Figure 2, the elapsed time of DMARFS-MI is smaller than NDMARFS. The elapsed time of the DMARFS-MI increases with the increasing value of R_{im}. With the increase of R_{im}, the elapsed time of the DMARFS-MI is more and more close to that of NDMARFS.

4.2. A comparison of NDMARFS and DMARFS-ME

We compare NDMARFS with DMARFS-ME on the six data sets shown in Table 7. The experimental results of these six data sets are shown in Figures 3-4.

In each sub-figure of Figure 3, the x-coordinate pertains to the attributes number of the data set, while the y-coordinate concerns the computational time. R_{em} denotes the ratio of the number of emigrant data and the whole data set, called the emigrant ratio [24]. Let $R_{em} = 10\%$. Figure 3 displays more detailed change trend of each of two algorithms with the increasing number of attributes.

Shown as Figure 3, DMARFS-ME is much faster than NDMARFS on updating approximations when the same objects enter into the FDS. With the increasing number of attributes, the elapsed time is relatively stable in DMARFS-ME, but that of NDMARFS increases drastically.

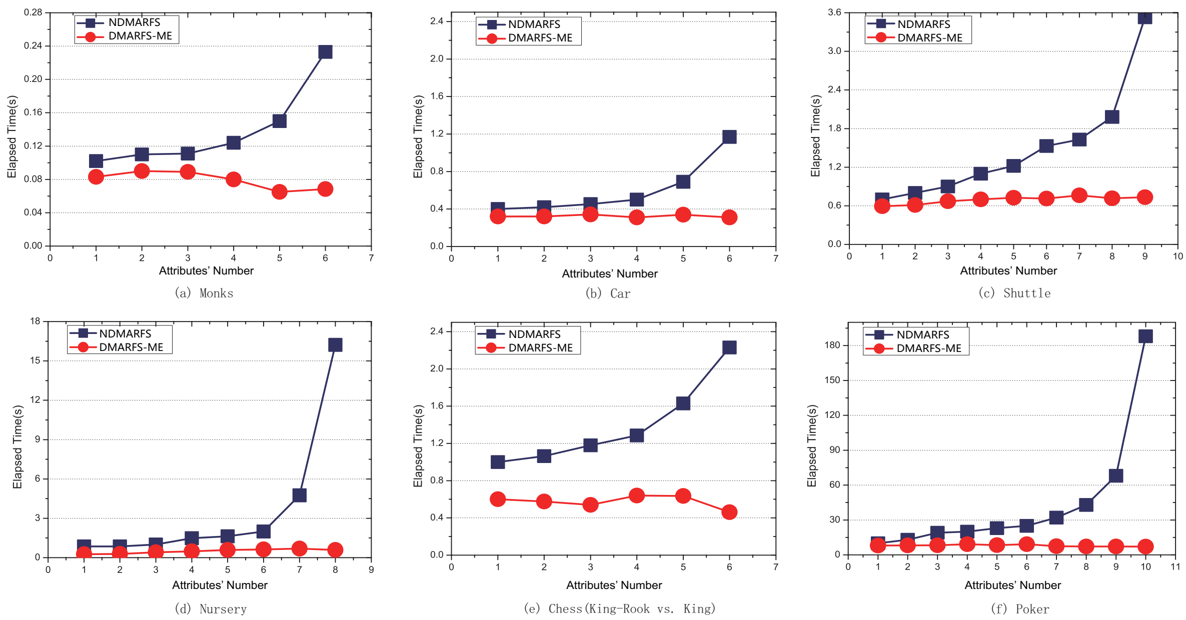


Figure 3. Elapsed time of NDMARFS and DMARFS-ME versus the attributes' number.

In each sub-figure of Figure 4, the x-coordinate pertains to the emigrant ratio R_{em}, while the y-coordinate concerns the elapsed time. Figure 4 displays more detailed change trend of each of two algorithms with the increase of R_{em}.

Shown as Figure 4, the elapsed time of DMARFS-ME increases with the increasing value of R_{em}, but that of NDMARFS decreases. Shown as the sub-figures (a), (b), (c), (d), (e), (f) in Figure 4, when R_{em} is smaller than 90%, 91%, 92%, 93%, 90%, 98%, respectively, DMARFS-ME is much faster than NDMARFS.

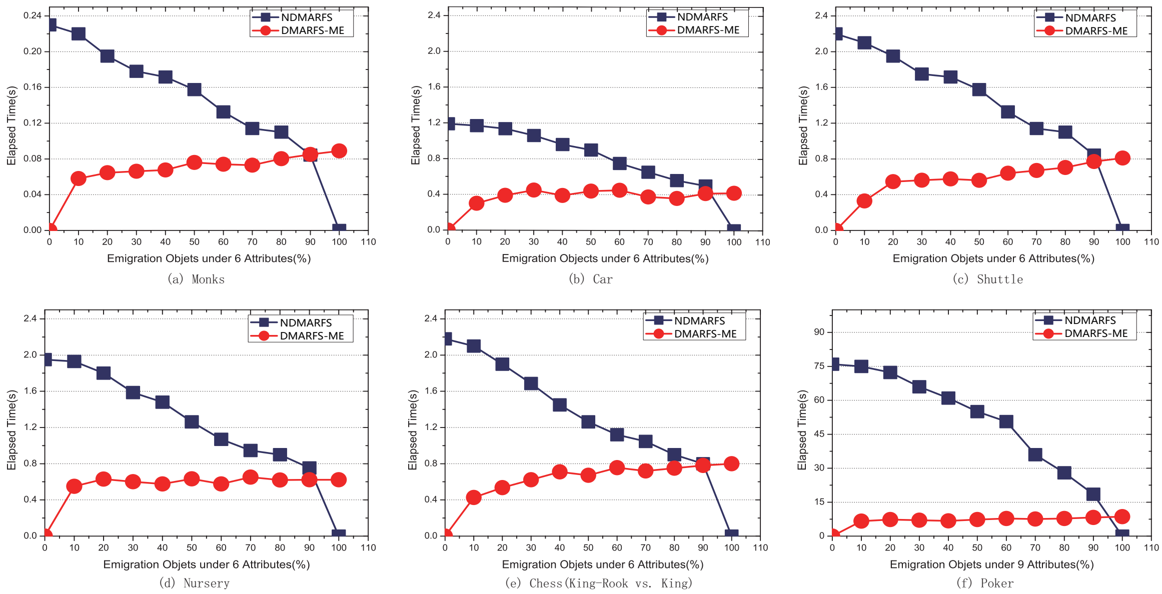


Figure 4. Elapsed time of NDMARFS and DMARFS-ME versus the emigrant ratio R_{em}.

5. Conclusions

RFS can deal with the two cases: (1) the decision attribute values are fuzzy; (2) each object belongs to multiple decision classes by different probabilities. In addition, the volume of data is growing at an unprecedented rate. The information system evolves over time. The traditional updating approximations algorithms are time-consuming. In this paper, we discussed the dynamic approach for maintenance of approximations under RFS when the object set evolves over time. First, two steps to update principles of lower and upper approximations under RFS were discussed: Step 1 is to update the objects of approximations incrementally using Pawlak's rough set; Step 2 is to update the memberships. Several examples were designed to illustrate the proposed methods. And then, the corresponding algorithms (DMARFS-MI and DMARFS-ME) for dynamic maintenance of approximations were presented. With 6 UCI data sets, a series of experiments were conducted for evaluating the proposed methods. The results indicated that (1) the proposed algorithms significantly reduce computational time of updating approximations; (2) the finer the granular size (the more attributes), the more elapsed time the proposed algorithms reduce; (3) the fewer the number of immigrant or emigrant objects, the more obvious advantages the proposed algorithms have.

Acknowledgement

This work is supported by the National Science Foundation of China (Nos. 61175047, 61100117) and NSAF (No. U1230117), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630 127), the Scientific Research Fundation of Sichuan Provincial Education Department

(No. 13ZB0210), the Fundamental Research Funds for the Central Universities (Nos. SWJTU11ZT08, SWJTU12CX117, SWJTU12CX091), and the 2013 Doctoral Innovation Funds of Southwest Jiaotong University.

References

[1] Z. Pawlak: Rough sets. International Journal of Computer and Information Sciences, 11(5): 341-356 (1982).

[2] D. Dubois, H. Prade: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems, 17(2-3): 191-209 (1990).

[3] Y. Yao: Combination of rough and fuzzy sets based on α-level sets. In: Proc. Rough Sets and Data Mining: Analysis for Imprecise Data, Lin, T.Y. and Cercone, N. (Eds.), Kluwer Academic Publishers, Boston, pp. 301-321 (1997).

[4] J. Zhang, T. Li, D. Ruan, Z. Gao, C. Zhao: A parallel method for computing rough set approximations, Information Sciences, 194: 209-223 (2012).

[5] J. Zhang, T. Li, Y. Pan: Parallel rough set based knowledge acquisition using mapreduce from big data. ACM SIGKDD12 Big Data Mining (BigMine'12) Workshop, Beijing, China, pp. 20-27 (2012).

[6] C. Chan: A rough set approach to attribute generalization in data mining, Information Sciences, 107: 177-194 (1998).

[7] T. Li, D. Ruan, G. Wets, J. Song, Y. Xu: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowledge-Based Systems, 20(5): 485-494 (2007).

[8] J. Zhang, T. Li, D. Liu: An approach for incremental updating approximations in variable precision rough sets while attribute generalizing. In: Proceedings of 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, China, pp. 17-81 (2010).

[9] S. Li , T. Li, D. Liu: Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set. Knowledge-Based Systems, 40: 17-26 (2013).

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

[11] Y. Cheng: The incremental method for fast computing the rough fuzzy approximations. Data and Knowledge Engineering, 70: 84-100 (2011).

[12] J. Zhang, T. Li, H. Chen: Composite rough sets for dynamic data mining. Information Sciences, 257: 81-100 (2014).

[13] H. Chen, T. Li, S. Qiao, D. Ruan: A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values. International Journal of Intelligent Systems, 25: 1005-1026 (2010).

[14] H. Chen, T. Li, D. Ruan: Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining, Knowledge-Based Systems, 31: 140-161 (2012).

[15] H. Chen, T. Li, D. Ruan , J. Lin, C. X. Hu: A rough-set based incremental approach for updating approximations under dynamic maintenance environments. IEEE Transactions on Knowledge and Data Engineering, 25(2): 274-284 (2013).

[16] D. Liu, J. Zhang, T. Li: A probabilistic rough set approach for incremental learning knowledge on the change of attribute. In: Proceedings of 2010 International Conference on Foundations and Applications of Computational Intelligence, EMei, China, pp. 722-727 (2010).

[17] D. Liu, T. Li, G. Liu, P. Hu: An incremental approach for inducing interesting knowledge based on the change of attribute values. In: Proceedings of 2009 IEEE International Conference on Granular Computing, Nanchang, China, pp. 415-418 (2009).

[18] L. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules. Computational Intelligence, 11: 357-370 (1995).

[19] W. Bang, Z. Bien, New incremental learning algorithm in the framework of rough set theory. International Journal of Fuzzy Systems, 1: 25-36 (1999).

[20] D. Liu, T. Li, D. Ruan, J. Zhang: Incremental learning optimization on knowledge discovery in dynamic business intelligent systems. Journal of Global Optimization, 51: 325-344 (2011).

[21] Z. Zheng, G. Wang: A rough set and rule tree based incremental knowledge acquisition algorithm. Fundamenta Informaticae, 59(2-3): 299-313 (2004).

[22] F. Hu, G. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets. In: Proceedings of 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI, 3641, Regina, Canada, pp. 185-193 (2005).

[23] L. Wang, Y. Wu, G. Wang: An incremental rule acquisition algorithm based on variable precision rough set model. Journal of Chongqing University of Posts and Telecommunications, 17(6): 709-713 (2005).

[24] J. Zhang, T. Li, D. Ruan, D. Liu: Neighborhood rough sets for dynamic data mining. International Journal of Intelligent Systems, 27: 317-342 (2012).

[25] S. Li , T. Li, D. Liu: Dynamic maintenance of approximations in dominance-based rough Set approach under the variation of the object Set. International Journal of Intelligent Systems, 28 (8): 729-751 (2013).

[26] C. Luo , T. Li, H. Chen, D. Liu: Incremental approaches for updating approximations in set-valued ordered information systems. Knowledge-Based Systems, 50: 218-233 (2013).

[27] J. Zhang, J.-S. Wong, T. Li, Y. Pan: A comparison of parallel large-scale knowledge acquisition using rough set theory on different mapReduce runtime systems. International Journal of Approximate Reasoning, 55(3): 896-907 (2014).

[28] D. Liu, T.R. Li, J.B. Zhang, An incremental approach for rule induction under coarsening and refining of attribute values in e-business systems. In: Proceedings of 2010 International Conference on Electronic-Business Intelligence, Kunming, China, pp. 541-547 (2010).

[29] Z. Xu, Z. Liu, B. Yang, W. Song: A quick attribute reduction algorithm with complexity of max(O($|C||U|$), O($|C|^2|U/C|$)). Chinese Journal of Computers, 29(3): 391-399 (2006) (in Chinese).