Neighborhood Rough Sets for Dynamic Data Mining

Junbo Zhang,[1,†] Tianrui Li,[1,*] Da Ruan,[2,3,‡] Dun Liu[4,§]
[1]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, People's Republic of China*
[2]*Belgian Nuclear Research Centre (SCK•CEN), Boeretang 200, 2400 Mol, Belgium*
[3]*Department of Applied Mathematics & Computer Science, Ghent University, 9000 Gent, Belgium*
[4]*School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, People's Republic of China*

Approximations of a concept in rough set theory induce rules and need to update for dynamic data mining and related tasks. Most existing incremental methods based on the classical rough set model can only be used to deal with the categorical data. This paper presents a new dynamic method for incrementally updating approximations of a concept under neighborhood rough sets to deal with numerical data. A comparison of the proposed incremental method with a nonincremental method of dynamic maintenance of rough set approximations is conducted by an extensive experimental evaluation on different data sets from UCI. Experimental results show that the proposed method effectively updates approximations of a concept in practice. © 2012 Wiley Periodicals, Inc.

1. INTRODUCTION

Data mining is defined by the process of discovering significant and potentially valuable patterns, associations, trends, sequences, and dependencies in large volumes of data.[1] Many data mining approaches have been proposed such as association rule mining,[2,3] sequential pattern mining,[4] text mining,[5] and temporal data mining.[6]

Rough set theory, introduced by Pawlak in 1982, is a powerful mathematical tool for dealing with inconsistent information in decision situations[7–9]. Nowadays, many rough sets based approaches have been successfully applied in data

*Author to whom all correspondence should be addressed: e-mail: trli@swjtu.edu.cn.
†e mail: JunboZhang86@163.com, jbzhang86@gmail.com.
‡e-mail: druan@sckcen.be, da.ruan@ugent.be.
§e-mail: newton83@163.com.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 27, 317–342 (2012)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.21523

mining.[11–18] For instance, a conceptually simple and easy to implement method based on the rough sets was to construct neighborhood-based attribute reduction technique and classifiers.[11] A theoretic framework based on rough set theory, named the positive approximation, was to accelerate algorithms of heuristic attribute reduction.[18]

The idea of rough set theory is based on the indiscernibility relation, and any set of all indiscernible objects is called an elementary set, which forms a basic granule of knowledge in a universe.[19] However, the classical rough set model employs equivalence relations to partition the universe and generate mutually exclusive equivalence classes as elemental concepts. This is just applicable to data with nominal attributes whereas in many cases we have to deal with numerical attributes since practical data are always numerical.[20] Therefore, neighborhood relations have been proposed as an extended rough set model in the information system and have successfully applied in many fields.[11–13,21,22]

Nowadays, the volume of data is growing at an unprecedented rate. The information system evolves over time. Discovering knowledge is an expensive operation. Running data mining algorithms from scratch, each time there is a change in data, is obviously, not an efficient strategy.[23] Using previously discovered knowledge along with new data to maintain discovered knowledge could solve many problems. Therefore, approaches for updating knowledge incrementally in dynamic data mining become more and more popular. An information system consists of the attributes (features), the objects (instances), and the domain of attributes values. We briefly review the previous work of these three aspects for updating knowledge by rough set theory.

1. Knowledge Updating by the Variation of the Object Sets. The variation of object sets has been considered widely in dynamic data mining. Shan and Ziarko presented a discernibility-matrix based incremental methodology to find all maximally generalized rules.[24] Bang et al. proposed another incremental inductive learning algorithm to find a minimal set of rules for a decision table without recalculating all the set of instances when another instance is added into the universe.[25] Tong and An developed an algorithm based on the ∂-decision matrix for incremental learning rules. They listed seven cases that would happen when a new sample enters the system.[26] Liu et al. presented an incremental model and approach as well as its algorithm for inducing interesting knowledge when the object set varies over time.[14] Furthermore, in business intelligent information systems, Liu et al. proposed an optimization incremental approach as well as its algorithm for inducing interesting knowledge.[15] Jerzy et al. discussed the incremental induction of decision rules from dominance-based rough approximations to select the most interesting representatives in the final set of rules.[27] Zheng and Wang proposed an effective incremental algorithm, named as RRIA, for knowledge acquisition based on the rule tree.[28] In addition, Hu et al. constructed a novel incremental attribute reduction algorithm when new objects are added into a decision information system.[29] Li et al. proposed an approach for dynamically updating approximations in dominance-based rough sets when adding or deleting objects in the universe.[30]
2. Knowledge Updating by the Variation of the Attribute Sets. Chan noticed the variation of the attribute sets in 1998 and proposed an incremental mining algorithm for learning classification rules efficiently when an attribute set in the information system evolves over time.[31] Li et al. presented a method for updating approximations of a concept in an incomplete information system under the characteristic relation when an attribute set

varies over time.[32] Cheng proposed an incremental model for fast computing the rough fuzzy approximations.[33] In addition, Zhang and Liu et al. discussed the strategies and propositions under variable precision rough sets when attributes are changed.[34,35]

3. Knowledge Updating by the Variation of the Attribute Values. Recently, some researchers began to care about the variation of the attribute values in data mining. Chen et al. proposed an incremental algorithm for updating the approximations of a concept under coarsening or refining of attributes' values.[36] Liu et al. discussed some propositions of the interesting knowledge in consistent (inconsistent) systems and proposed several incremental learning strategies and an algorithm for inducing the interesting knowledge when attributes values are changed.[37,38]

In this paper, we focus on the first case, namely, knowledge updating under the variation of the object set. The variation of the objects may affect the granularity of the knowledge space. Decision rules induced by lower and upper approximations may also alter. Instead of the previous work on incremental learning mainly concerned the situation where only one single object enters into the information system or only categorical data in the information system, we investigate the method for incremental updating knowledge based on the neighborhood rough sets, which can also deal with numerical data in the information system when multiple objects enter into and get out of the information system simultaneously.

The remaining of the paper is organized as follows. Section 2 includes the elementary background introduction to neighborhood rough sets. The approaches and algorithms for updating approximations of a concept in the neighborhood system are presented in Section 3. Section 4 shows the experimental analysis. Experimental evaluations with nine data sets from UCI are given in Section 5. The paper ends with conclusions in Section 6.

2. PRELIMINARIES

We briefly review some basic concepts, notations, and results of rough sets as well as their extensions.[8,10–12]

Given a pair $\mathscr{K} = (U, R)$, where U is a finite and nonempty set called the universe, and $R \subseteq U \times U$ is an equivalence relation on U. The pair $\mathscr{K} = (U, R)$ is called an approximation space. The equivalence relation R partitions the set U into several disjoint subsets. This partition of the universe form a quotient set induced by R, denoted by U/R. If two elements $x, y \in U, (x \neq y)$ are indistinguishable under R, we say x and y belong to the same equivalence class, the equivalence class including x is denoted by $[x]_R$.

An approximation space $\mathscr{K} = (U, R)$ is characterized by an information system $S = (U, A, V, f)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a nonempty finite set of objects, called a universe. $A = \{a_1, a_2, \dots, a_m\}$ is a nonempty finite set of attributes (features). Specifically, $S = (U, A, V, f)$ is called a decision table if $A = C \cup D$, where C is a set of condition attributes and D is output, also called decision, $C \cap D = \emptyset$. $V = \bigcup_{a \in A} V_a$, V_a is a domain of the attribute a. $f : U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. $f(x_i, a_j)$ denotes the value of object x_i on the attribute a_j.

DEFINITION 1. *Let $B \subseteq C$ be a subset of attributes, $x \in U$. The neighborhood $\delta_B(x)$ of x in the feature space B is defined as*

$$\delta_B(x) = \{y \in U | \Delta_B(x, y) \leq \delta\} \tag{1}$$

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

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

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

$$\Delta_P(x, y) = \left(\sum_{j=1}^{m} \left|f(x, a_j) - f(y, a_j)\right|^P\right)^{1/P} \tag{2}$$

where (2) is called: (a). Manhattan distance Δ_1 if $P = 1$; (b). Euclidean distance Δ_2 if $P = 2$; (c). Chebychev distance if $P = \infty$.[39]

DEFINITION 2. *Given a pair $\mathscr{K}_{\mathscr{N}} = (U, N)$, where U is a set of samples, N is a neighborhood relation on U, $\{\delta(x) \mid x \in U\}$ is the family of neighborhood granules. Then we call the pair $\mathscr{K}_{\mathscr{N}} = (U, N)$ a neighborhood approximation space.*

DEFINITION 3. *Given a neighborhood approximation space $\mathscr{K}_{\mathscr{N}} = (U, N)$, for $\forall X \subseteq U$, two subsets of objects, called lower and upper approximations of X in terms of relation N, are defined as*

$$\underline{N}(X) = \{x \in U \mid \delta(x) \subseteq X\} \tag{3}$$

$$\overline{N}(X) = \{x \in U \mid \delta(x) \cap X \neq \emptyset\} \tag{4}$$

The boundary region of X in the approximation space is formulated as

$$BN(X) = \overline{N}(X) - \underline{N}(X) \tag{5}$$

The size of the boundary region reflects the degree of roughness of the set X in the approximation space. Assuming that X is the sample subset with a decision label, usually we hope that the boundary region of the decision is as little as possible for decreasing uncertainty in the decision procedure. Clearly, the sizes of the boundary regions depend on X, attributes B to describe U, and the threshold δ.

An information system is called a neighborhood system if the attributes generate a neighborhood relation over the universe, defined as $NIS = (U, A, V, f)$, where A is a real-valued attribute set, f is an information function, $f : U \times A \to R$. More specially, a neighborhood information system is also called a neighborhood decision system $NDT = (U, C \cup D, V, f)$ if $A = C \cup D$. where C denotes condition attributes and D denotes decision attribute, respectively.

DEFINITION 4. *Given a neighborhood decision table $NDT = (U, C \cup D, V, f)$, suppose $U/D = \{D_1, D_2, \ldots, D_N\}$, $\delta_B(x)$ is the neighborhood information granules including x and generated by attributes $B \subseteq C$. Then the lower and upper approximations of the decision D with respect to attributes B are defined as*

$$\underline{N}_B(D) = \cup_{i=1}^{N} \underline{N}_B(D_i) \tag{6}$$

$$\overline{N}_B(D) = \cup_{i=1}^{N} \overline{N}_B(D_i) \tag{7}$$

where

$$\underline{N}_B(D_i) = \{x \in U \mid \delta_B(x) \subseteq D_i\}$$

$$\overline{N}_B(D_i) = \{x \in U \mid \delta_B(x) \cap D_i \neq \emptyset\}$$

The decision boundary region of D with respect to attributes B is defined as

$$BN(D) = \overline{N}_B(D) - \underline{N}_B(D) \tag{8}$$

The detailed theorems and properties on neighborhood rough sets can be found in Refs. 11 and 12.

3. INCREMENTAL METHOD FOR UPDATING APPROXIMATIONS BASED ON NEIGHBORHOOD ROUGH SETS

We discuss the change of approximations in dynamic neighborhood decision systems when the object set evolves over time while the attribute set remains constant. To illustrate this method clearly, we divide the work into three parts: we present several assumptions and the basic structure of the approach in Section 3.1; we discuss the immigration and the emigration of single object and design the corresponding algorithms in Section 3.2; we discuss the immigration and the emigration of multiple objects and design the corresponding algorithms in Section 3.3.

3.1. Assumptions and Basic Structure for the Incremental Approach

We assume the incrementally learning process lasts from time t to time $t+1$. Followed by Liu's introduction, the updating of the objects is divided into two parts: the immigration and the emigration of the objects.[14,15] The former case means the objects enter into the decision table at time $t+1$, and the universe is expanding; the latter case means the objects get out of the decision table, and the universe is contracting at time $t+1$. Both the immigration and emigration processes reflect the coarsening or refining of knowledge in neighborhood decision table.

To describe a dynamic neighborhood decision table, we denote a neighborhood decision table at time t as $NDT = (U, C \cup D, V, f)$, suppose $U/D = \{D_1, D_2, \ldots, D_N\}$ denotes N decision classes. At time $t+1$, some objects enter the decision table while some go out. Hence, the original neighborhood decision table NDT will be changed into $NDT' = (U', C' \cup D', V', f')$. Similarly, the neighborhood information granules including x and generated by attributes $B \subseteq C$ at time t denoted as $\delta_B(x)$, the one as $\delta'_B(x)$ at time $t+1$. According to Definition 4, the lower and upper approximations of the decision D with respect to attributes B at time t are denoted as $\underline{N}_B(D_i)$ and $\overline{N}_B(D_i)$, respectively, and the ones at time $t+1$ as $\underline{N}_B(D_i')$ and $\overline{N}_B(D_i')$, respectively.

With these stipulations, we focus on the algorithms for updating approximations of the decision D when (1) one object enters into the neighborhood decision table at time $t+1$; (2) one object gets out of the neighborhood decision table at time $t+1$; (3) the object set enters into the neighborhood decision table at time $t+1$; (4) the object set gets out of the neighborhood decision table at time $t+1$.

First, we design a naive algorithm for updating approximations based on neighborhood rough sets (NAUANRS), which can deal with the above four mentioned variations of the object set. The detailed process is outlined in Algorithm 1.

3.2. Immigration and the Emigration of Single Object

Given a neighborhood decision table $NDT = (U, C \cup D, V, f)$ at time t, where U is a nonempty set, $C \bigcap D = \emptyset$, C is a set of condition attributes and $D = \{d\}$ is a decision attribute. D divide the universe U into N parts, which denote as: $U/D = \{D_1, D_2, \ldots, D_N\}$.

Suppose there is one object that enters into or gets out of the neighborhood decision table at time $t+1$. It will change the original information granules, the union of decision classes, and their lower, upper approximations in the neighborhood decision table.

3.2.1. Immigration of One New Object

Suppose there is one object $\bar{x}$ that enters into the neighborhood decision table at time $t+1$. So $U' = U \cup \{\bar{x}\}$.

PROPOSITION 1. *When a new object $\bar{x}$ enters into the neighborhood decision table $NDT = (U, C \cup D, V, f)$, one of the following results holds:*

Algorithm 1: A Naive Algorithm for Updating Approximations based on Neighborhood Rough Sets (NAUANRS)

Input:
(1). $NDT = (U, C \cup D, V, f)$ at time t, the threshold δ;
(2). The approximations of decision classes at time t: $\overline{N}(D_i)$, $\underline{N}(D_i)$;
(3). The new immigrational object set U_{im}, the emigrational object set U_{em}, where $U \cap U_{im} = \emptyset$ and $U_{em} \subseteq U$.
Output:
The updated approximations of decision classes at time $t+1$: $\overline{N}(D_i')$, $\underline{N}(D_i')$.

begin
  $U' = U \cup U_{im}$;
  $U' = U - U_{em}$;
  compute each D_i';
  for *each* $x \in U'$ **do**
    compute $\delta'(x)$;
  end
  for *each* D_i' **do**
    $\overline{N}(D_i') = \{x \in U' | \delta'(x) \cap D_i' \neq \emptyset\}$;
    $\underline{N}(D_i') = \{x \in U' | \delta'(x) \subseteq D_i'\}$;
  end
  return $\overline{N}(D_i)$ and $\underline{N}(D_i)$.
end

I. *If* $\exists x \in U$, $f(\bar{x}, d) = f(x, d)$, *then* $\bar{x}$ *cannot form a new decision class;*
II. *If* $\forall x \in U$, $f(\bar{x}, d) \neq f(x, d)$, *then* $\bar{x}$ *can form a new decision class.*

THEOREM 1. *If* $\bar{x}$ *cannot form a new decision class, suppose* $U'/D' = \{D_1', D_2', \ldots, D_N'\}$ *at time* $t+1$. *Then*

$$D_i' = \begin{cases} D_i \cup \{\bar{x}\} : i = f(\bar{x}, d) \\ D_i : i \neq f(\bar{x}, d) \end{cases} \tag{9}$$

where $i \in \{1, 2, \ldots, N\}$.

THEOREM 2. *If* $\bar{x}$ *can form a new decision class with the decision* $N+1$, *suppose* $U'/D' = \{D_1', D_2', \ldots, D_N', D_{N+1}'\}$ *at time* $t+1$. *Then*

$$D_i' = \begin{cases} \{\bar{x}\} : i = N+1 \\ D_i : i \neq N+1 \end{cases} \tag{10}$$

where $i \in \{1, 2, \ldots, N+1\}$.

For the new object $\bar{x}$, by Definition 1, we have $\delta'(\bar{x}) = \{y \in U' | \Delta(\bar{x}, y) \leq \delta\}$.

THEOREM 3 . *For each $x \in U$, then*

$$\delta'(x) = \begin{cases} \delta(x) \cup \{\bar{x}\} : x \in \delta'(\bar{x}) \\ \delta(x) : x \notin \delta'(\bar{x}) \end{cases} \tag{11}$$

According to the properties of the immigration of one new object, we present the updating methodology of approximations of the decision classes when one new object enters into the neighborhood decision table as follows.

Given a neighborhood decision table $NDT = (U, C \cup D, V, f)$. Neighborhoods of all objects in U generated by all condition attributes C are denoted by $NC_U = \{\delta_C(x_1), \delta_C(x_2), \ldots, \delta_C(x_{|U|})\}$. Decision classes generated by the attribute D are denoted by $U/D = \{D_1, D_2, \ldots, D_N\}$, $1 \leq N \leq |U|$. After the phase of immigrating one new object $\bar{x}$, $U' = U \cup \{\bar{x}\}$, the neighborhood of the new object $\bar{x}$ is generated, and the neighborhoods of the object in U may change. Let $NC_{U'} = \{\delta_C(x_1), \delta_C(x_2), \ldots, \delta_C(x_{|U|}), \delta_C(x_{|U|+1})\}$ be the updated neighborhoods. If the new decision class is not be generated, then $U'/D' = \{D'_1, D'_2, \ldots, D'_N\}$ denotes the updated decision classes; Otherwise $U'/D' = \{D'_1, D'_2, \ldots, D'_N, D'_{N+1}\}$ denotes the updated decision classes.

In our method, the decision classes are updated at first. Then, the neighborhoods of the immigrational object $\bar{x}$ is computed, and the neighborhoods of the object in U are updated. Finally, the upper and lower approximations of the decision classes are updated. The detail process is outlined in Algorithm 2.

3.2.2. Emigration of One Object

Suppose there is one object $\tilde{x}$ that gets out of the neighborhood decision table at time $t + 1$. So $U' = U - \{\tilde{x}\}$.

PROPOSITION 2. *When an object $\tilde{x}$ gets out of the neighborhood decision table $NDT = (U, C \cup D, V, f)$, one of the following results holds:*

I. *If $v = f(\tilde{x}, d)$, $|D_v| = 1$ (it means the size of the set D_v is equal to one), then $\tilde{x}$ can eliminate a decision class with the decision v;*
II. *If $v = f(\tilde{x}, d)$, $|D_v| > 1$ (it means the size of the set D_v is more than one), then $\tilde{x}$ cannot eliminate a decision class.*

THEOREM 4 . *If $\tilde{x}$ can delete a decision class with the decision v, suppose $U'/D' = \{D'_1, D'_2, \ldots, D'_{v-1}, D'_{v+1}, \ldots, D'_N\}$ at time $t + 1$. Then*

$$D'_i = D_i \tag{12}$$

where $i \in \{1, 2, \ldots, v - 1, v + 1, \ldots, N\}$.

Algorithm 2: A Fast Algorithm for Updating Approximations based on Neighborhood Rough Sets when Single Object Immigrating (FAUANRS-SI)

Input:
(1). $NDT = (U, C \cup D, V, f)$ at time t, the threshold δ;
(2). The object subsets with decision 1 to N: $U/D = \{D_i | i = 1, 2, \ldots, N\}$; the neighborhood of x in the feature space C: $\{\delta(x) | x \in U\}$; the approximations of decisions classes: $\overline{N}(D_i)$, $\underline{N}(D_i)$ at time t;
(3). The new immigrational object: $\bar{x}$.
Output:
The updated approximations of decision classes at time $t + 1$: $\overline{N}(D_i')$, $\underline{N}(D_i')$.

begin
  $U' = U \cup \bar{x}$;
  for *each* D_i **do** $D_i' \leftarrow D_i$;
  let $v = f(\bar{x}, d)$;
  if $v \in \{1, 2, \ldots, N\}$ **then** $D_v' \leftarrow D_v \cup \{\bar{x}\}$;
  else $D_{N+1}' = \{\bar{x}\}$; //new decision class
  compute $\delta'(\bar{x})$;
  for *each* $x \in U$ **do**
    if $x \in \delta'(\bar{x})$ **then** $\delta'(x) \leftarrow \delta(x) \cup \{\bar{x}\}$;
    else $\delta'(x) \leftarrow \delta(x)$;
  end
  for *each* D_i' **do**
    $\overline{N}(D_i') = \{x \in U' | \delta(x) \cap D_i' \neq \emptyset\}$;
    $\underline{N}(D_i') = \{x \in U' | \delta(x) \subseteq D_i'\}$;
  end
  return $\overline{N}(D_i')$ and $\underline{N}(D_i')$.
end

THEOREM 5. *If $\tilde{x}$ cannot delete a decision class, suppose $U'/D' = \{D_1', D_2', \ldots, D_N'\}$ at time $t + 1$. Then*

$$D_i' = \begin{cases} D_i - \{\tilde{x}\} : i = f(\tilde{x}, d) \\ D_i : i \neq f(\tilde{x}, d) \end{cases} \tag{13}$$

where $i \in \{1, 2, \ldots, N\}$.

For the object $\tilde{x}$, $\delta(\tilde{x})$ has been given at time t.

THEOREM 6. *For each $x \in U - \{\tilde{x}\}$, then*

$$\delta'(x) = \begin{cases} \delta(x) - \{\tilde{x}\} : x \in \delta(\tilde{x}) \\ \delta(x) : x \notin \delta(\tilde{x}) \end{cases} \tag{14}$$

By the properties of the emigration of one object, we present the updating methodology of approximations of the decision classes when one object gets out of the neighborhood decision table as follows.

Given a neighborhood decision table $NDT = (U, C \cup D, V, f)$. Neighborhoods of all objects in U generated by all condition attributes C are denoted by $NC_U = \{\delta_C(x_1), \delta_C(x_2), \ldots, \delta_C(x_{|U|})\}$. Decision classes generated by the attribute D are denoted by $U/D = \{D_1, D_2, \ldots, D_N\}$, $1 \leq N \leq |U|$. After the phase of emigrating one object $\tilde{x}$, $U' = U - \{\tilde{x}\}$, the neighborhood of the object $\tilde{x}$ is emigration, and the neighborhoods of the object in U may change. Let $NC_{U'} = \{\delta_C(x_1), \delta_C(x_2), \ldots, \delta_C(x_{|U'|})\}$ be the updated neighborhoods. If the decision class D_v is deleted, then $U'/D' = \{D'_1, D'_2, \ldots, D'_{v-1}, D'_{v+1}, \ldots, D'_N\}$ denotes the updated decision classes; Otherwise, $U'/D' = \{D'_1, D'_2, \ldots, D'_N\}$ denotes the updated decision classes.

In our method, the decision classes are updated at first. Then, the neighborhood of the emigrational object $\tilde{x}$ is deleted from NC_U, and the neighborhoods of the object in $U - \tilde{x}$ are updated. Finally, the upper and lower approximations of the decision classes are updated. The detailed process is outlined in Algorithm 3.

Algorithm 3: A Fast Algorithm for Updating Approximations based on Neighborhood Rough Sets when Single Object Emigrating (FAUANRS-SE)

Input:
(1). $NDT = (U, C \cup D, V, f)$ at time t, the threshold δ;
(2). The object subsets with decisions 1 to N: $U/D = \{D_i | i = 1, 2, \ldots, N\}$; the neighborhood of x in the feature space C: $\{\delta(x) | x \in U\}$; the approximations of decision classes: $\overline{N}(D_i)$, $\underline{N}(D_i)$ at time t;
(3). The new emigrational object: $\tilde{x}$.
Output:
The updated approximations of decision classes at time $t+1$: $\overline{N}(D'_i)$, $\underline{N}(D'_i)$.

begin
  $U' = U - \tilde{x}$;
  for *each* D_i **do** $D'_i \leftarrow D_i$;
  let $v = f(\tilde{x}, d)$;
  $D'_v \leftarrow D_v - \{\tilde{x}\}$;
  if $|D'_v| = 0$ **then delete** D'_v from $\{D'_1, D'_2, \ldots, D'_N\}$;
  for *each* $x \in U - \{\tilde{x}\}$ **do**
    if $x \in \delta'(\tilde{x})$ **then** $\delta'(x) \leftarrow \delta(x) - \{\tilde{x}\}$;
    else $\delta'(x) \leftarrow \delta(x)$;
  end
  delete $\delta(\tilde{x})$ from $\{\delta(x_1), \delta(x_2), \ldots, \delta(x_n)\}$;
  for *each* D'_i **do**
    $\overline{N}(D'_i) = \{x \in U' | \delta(x) \cap D'_i \neq \emptyset\}$;
    $\underline{N}(D'_i) = \{x \in U' | \delta(x) \subseteq D'_i\}$;
  end
  return $\overline{N}(D'_i)$ and $\underline{N}(D'_i)$.
end

3.3. Immigration and the Emigration of Multiple Objects

In Section 3.2, we first discussed the immigration and the emigration of single object in the neighborhood system. Then we designed the algorithms FAUANRS-SI and FAUANRS-SE. These two algorithms are useful when one object enters into or gets out of the neighborhood system. However, if there are multiple objects that enter into or get out of neighborhood system, these two algorithms may also be inefficient since the two algorithms only deal with the object one by one naively. It means there are $O_{im} + O_{em}$ operations if O_{im} objects enter into and O_{em} objects get out of the neighborhood system. Therefore, we discuss the immigration and the emigration of multiple objects in the neighborhood system in this section.

3.3.1. Immigration of Multiple Objects

Suppose there are multiple objects U_{im} that enter into the neighborhood decision table at time $t+1$. So $U' = U \cup U_{im}$. Let $U_{im}/D' = \{D'_{l_1}, \ldots, D'_{l_p}, D'_{g_1}, \ldots, D'_{g_q}\}$ be the object subsets with decisions $\{l_1, \ldots, l_p, g_1, \ldots, g_q\}$, where $\{l_1, \ldots, l_p\} \subseteq \{1, 2, \ldots, N\}$, and $\{g_1, \ldots, g_q\} \cap \{1, 2, \ldots, N\} = \emptyset$. So the decision classes $\{D'_{g_1}, \ldots, D'_{g_q}\}$ are generated at time $t+1$, and $U'/D' = \{D'_1, D'_2, \ldots, D'_N, D'_{g_1}, \ldots, D'_{g_q}\}$.

THEOREM 7. *Let $i \in T$, $T = \{1, 2, \ldots, N, g_1, \ldots, g_q\}$. For each $D'_i \in U'/D'$, there are*

$$D'_i = \begin{cases} D_i \cup D'_i : i \in \{l_1, \ldots, l_p\} \\ D'_i : i \in \{g_1, \ldots, g_q\} \\ D_i : else \end{cases} \tag{15}$$

For each $\bar{x} \in U_{im}$, its neighborhood is $\delta'(\bar{x}) = \{y \in U' | \Delta(\bar{x}, y) \le \delta\}$.

THEOREM 8. *For each $x \in U$, each $\bar{x} \in U_{im}$ then*

$$\delta'(x) = \begin{cases} \delta(x) \cup \{\bar{x}\} : x \in \delta'(\bar{x}) \\ \delta(x) : x \notin \delta'(\bar{x}) \end{cases} \tag{16}$$

By the properties of the immigration of multiple objects, we present the updating methodology of approximations of the decision classes when multiple objects enter into the neighborhood decision table. The detailed process is outlined in Algorithm 4.

3.3.2. Emigration of Multiple Objects

Suppose there are multiple objects U_{em} that get out of the neighborhood decision table at time $t+1$. So $U' = U - U_{em}$. Let $U_{em}/D' = \{D'_{l_1}, \ldots, D'_{l_p}\}$ be the object subsets with decisions $\{l_1, \ldots, l_p\}$, Obviously, $\{l_1, \ldots, l_p\} \subseteq \{1, 2, \ldots, N\}$.

Algorithm 4: A Fast Algorithm for Updating Approximations based on Neighborhood Rough Sets when Multiple Objects Immigrating (FAUANRS-MI)

Input:
(1). $NDT = (U, C \cup D, V, f)$ at time t, the threshold δ;
(2). The object subsets with decisions 1 to N: $U/D = \{D_i | i = 1, 2, \ldots, N\}$; the neighborhood of x in the feature space C: $\{\delta(x) | x \in U\}$; the approximations of decision classes: $\overline{N}(D_i)$, $\underline{N}(D_i)$ at time t;
(3). The new immigrant object set: U_{im}, where $U \cap U_{im} = \emptyset$.
Output:
The updated approximations of decision classes at time $t+1$: $\overline{N}(D'_i)$, $\underline{N}(D'_i)$.

begin
  $U' = U \cup U_{im}$;
  for *each* D_i **do** $D'_i \leftarrow D_i$;
  compute $U_{im}/D' = \{D'_{k_j} | j = 1, \ldots, q\}$;
  for *each* $D'_{k_j} \in U_{im}/D'$ **do**
    if $k_j \in \{1, 2, \ldots, N\}$ **then** $D'_{k_j} \leftarrow D_{k_j} \cup D'_{k_j}$;
    else $D'_{k_j} \leftarrow D'_{k_j}$; //new decision class
  end
  for $\bar{x} \in U_{im}$ **do compute** $\delta'(\bar{x})$;
  for *each* $x \in U$ **do**
    $\delta'(x) \leftarrow \delta(x)$;
    for *each* $\bar{x} \in U_{im}$ **do**
      if $x \in \delta'(\bar{x})$ **then**
        $\delta'(x) \leftarrow \delta(x) \cup \{\bar{x}\}$;
      end
    end
  end
  for *each* D'_i **do**
    $\overline{N}(D'_i) = \{x \in U' | \delta(x) \cap D'_i \neq \emptyset\}$;
    $\underline{N}(D'_i) = \{x \in U' | \delta(x) \subseteq D'_i\}$;
  end
  return $\overline{N}(D'_i)$ and $\underline{N}(D'_i)$.
end

THEOREM 9. *Let $i \in T$, $T = \{1, 2, \ldots, N\}$. For each $D'_i \in U'/D'$, there are*

$$D'_i = \begin{cases} D_i - D'_i : i \in \{l_1, \ldots, l_p\} \\ D_i : else \end{cases} \tag{17}$$

THEOREM 10. *For each $x \in U - U_{em}$, each $\tilde{x} \in U_{em}$ then*

$$\delta'(x) = \begin{cases} \delta(x) - \{\tilde{x}\} : x \in \delta'(\bar{x}) \\ \delta(x) : x \notin \delta'(\bar{x}) \end{cases} \tag{18}$$

According to the properties of the emigration of multiple objects, we present the updating methodology of approximations of the decision classes when multiple objects get out of the neighborhood decision table. The detailed process is outlined in Algorithm 5.

Algorithm 5: A Fast Algorithm for Updating Approximations based on Neighborhood Rough Sets when Multiple objects Emigrating (FAUANRS-ME)

Input:
(1). $NDT = (U, C \cup D, V, f)$ at time t, the threshold δ;
(2). The object subsets with decisions 1 to N: $U/D = \{D_i | i = 1, 2, \ldots, N\}$; the neighborhood of x in the feature space C: $\{\delta(x) | x \in U\}$; the approximations of decision classes: $\overline{N}(D_i)$, $\underline{N}(D_i)$ at time t;
(3). The emigrant object set: U_{em}, where $U_{em} \subseteq U$.
Output:
The updated approximations of decision classes at time $t+1$: $\overline{N}(D'_i)$, $\underline{N}(D'_i)$.

begin
  $U' = U - U_{em}$;
  for *each* D_i **do** $D'_i \leftarrow D_i$;
  compute $U_{em}/D' = \{D'_{k_j} | j = 1, \ldots, p\}$;
  for *each* $D'_{k_j} \in U_{em}/D'$ **do**
    $D'_{k_j} \leftarrow D_{k_j} - D'_{k_j}$;
  end
  for $\tilde{x} \in U_{em}$ **do** $\delta'(\tilde{x}) = \delta(\tilde{x})$;
  for *each* $x \in U - U_{em}$ **do**
    $\delta'(x) \leftarrow \delta(x)$;
    for *each* $\tilde{x} \in U_{em}$ **do**
      if $x \in \delta'(\tilde{x})$ **then**
        $\delta'(x) \leftarrow \delta'(x) - \{\tilde{x}\}$;
      end
    end
  end
  for *each* D'_i **do**
    $\overline{N}(D'_i) = \{x \in U' | \delta(x) \cap D'_i \neq \emptyset\}$;
    $\underline{N}(D'_i) = \{x \in U' | \delta(x) \subseteq D'_i\}$;
  end
  return $\overline{N}(D'_i)$ and $\underline{N}(D'_i)$.
end

4. AN ILLUSTRATION

We now illuminate the above approaches and algorithms to update the interesting knowledge dynamically. An neighborhood decision table is given in Table I.

Table I. A neighborhood decision table.

U	a_1	a_2	d
x_1	0.1	0.1	Y
x_2	0.1	0.2	Y
x_3	0.1	0.3	N
x_4	0.2	0.2	Y
x_5	0.3	0.2	N
x_6	0.3	0.3	N

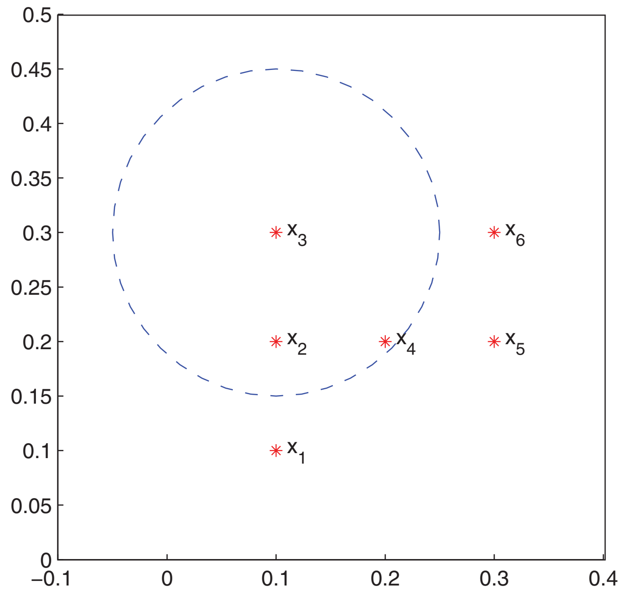

Figure 1. Original data in the numerical feature space.

Let $U = \{x_1, x_2, \ldots, x_6\}$, $C = \{a_1, a_2\}$, $D = \{d\}$, we set the threshold $\delta = 0.15$. Here, we use Euclidean distance as our measure. Then, $U/D = \{D_Y, D_N\} = \{Y, N\}$, where $D_Y = \{x_1, x_2, x_4\}$, $D_N = \{x_3, x_5, x_6\}$.

Figure 1 shows the original data in the numerical feature space. We obtain $\delta(x_1) = \{x_1, x_2, x_4\}$, $\delta(x_2) = \{x_1, x_2, x_3, x_4\}$, $\delta(x_3) = \{x_2, x_3, x_4\}$, $\delta(x_4) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\delta(x_5) = \{x_4, x_5, x_6\}$, $\delta(x_6) = \{x_4, x_5, x_6\}$. Hence, $\overline{N}(D_Y) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\underline{N}(D_Y) = \{x_1\}$, $\overline{N}(D_N) = \{x_2, x_3, x_4, x_5, x_6\}$, $\underline{N}(D_N) = \emptyset$.

4.1. New Object Set Is Immigrated

Table II shows the data after immigrating the new object set $U_{im} = \{x_7, x_8\}$. Figure 2 shows the data in the numerical feature space.

According to Algorithm 4, we update approximations dynamically as follows. Since

$$f(x_7, d) = N$$

$$f(x_8, d) = Y$$

then

$$D_Y = D_Y \cup \{x_8\} = \{x_1, x_2, x_4, x_8\}$$

$$D_N = D_N \cup \{x_7\} = \{x_3, x_5, x_6, x_7\}.$$

Table II. A neighborhood decision table after immigrating multiple objects.

U	a_1	a_2	d
x_1	0.1	0.1	Y
x_2	0.1	0.2	Y
x_3	0.1	0.3	N
x_4	0.2	0.2	Y
x_5	0.3	0.2	N
x_6	0.3	0.3	N
x_7	0.2	0.1	N
x_8	0.2	0.3	Y

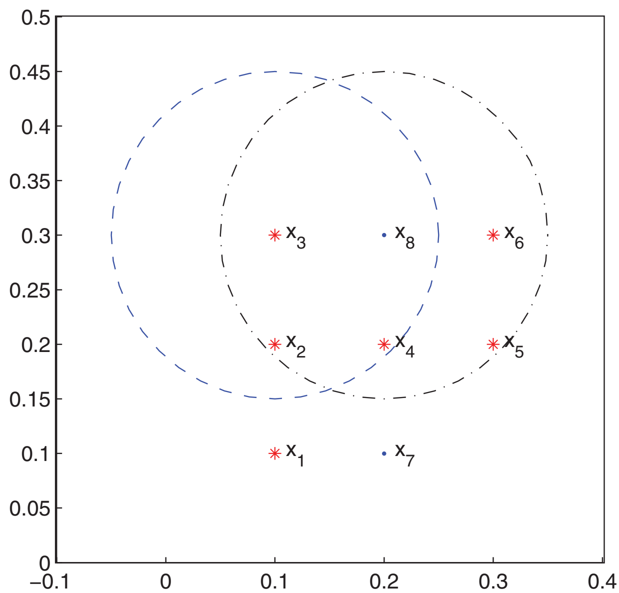

Figure 2. Data in the numerical feature space after immigrating multiple objects.

Since

$$\delta(x_7) = \{x_1, x_2, x_4, x_5, x_7\}$$

$$\delta(x_8) = \{x_2, x_3, x_4, x_5, x_6, x_8\}$$

then

$$\delta(x_1) = \delta(x_1) \cup \{x_7\} = \{x_1, x_2, x_4, x_7\}$$

$$\delta(x_2) = \delta(x_2) \cup \{x_7\} \cup \{x_8\} = \{x_1, x_2, x_3, x_4, x_7, x_8\}$$

$$\delta(x_3) = \delta(x_3) \cup \{x_8\} = \{x_2, x_3, x_4, x_8\}$$

$$\delta(x_4) = \delta(x_4) \cup \{x_7\} \cup \{x_8\} = \{x_1, x_2, \ldots, x_8\}$$

$$\delta(x_5) = \delta(x_5) \cup \{x_7\} \cup \{x_8\} = \{x_4, x_5, x_6, x_7, x_8\}$$

$$\delta(x_6) = \delta(x_3) \cup \{x_8\} = \{x_4, x_5, x_6, x_8\}$$

Because of

$$\delta(x_7) \cap D_Y \neq \emptyset \qquad \delta(x_8) \cap D_Y \neq \emptyset$$

$$\delta(x_7) \cap D_N \neq \emptyset \qquad \delta(x_8) \cap D_N \neq \emptyset$$

$$\delta(x_7) \not\subseteq D_Y \qquad \delta(x_8) \not\subseteq D_Y$$

$$\delta(x_7) \not\subseteq D_N \qquad \delta(x_8) \not\subseteq D_N$$

Specially, $\delta(x_1) \not\subseteq D_Y$. Therefore

$$\overline{N}(D_Y) = \overline{N}(D_Y) \cup \{x_7\} \cup \{x_8\} = \{x_1, x_2, \ldots, x_8\}$$

$$\overline{N}(D_N) = \overline{N}(D_N) \cup \{x_7\} \cup \{x_8\} = \{x_1, x_2, \ldots, x_8\}$$

$$\underline{N}(D_Y) = \underline{N}(D_Y) - \{x_1\} = \emptyset$$

$$\underline{N}(D_N) = \underline{N}(D_N) = \emptyset.$$

4.2. Object Set is Emigrated

Table III shows the data after emigrating the object set $U_{em} = \{x_4, x_5\}$. Figure 3 shows the data in the numerical feature space.

Table III. A neighborhood decision table after emigrating multiple objects.

U	a_1	a_2	d
x_1	0.1	0.1	Y
x_2	0.1	0.2	Y
x_3	0.1	0.3	N
x_6	0.3	0.3	N

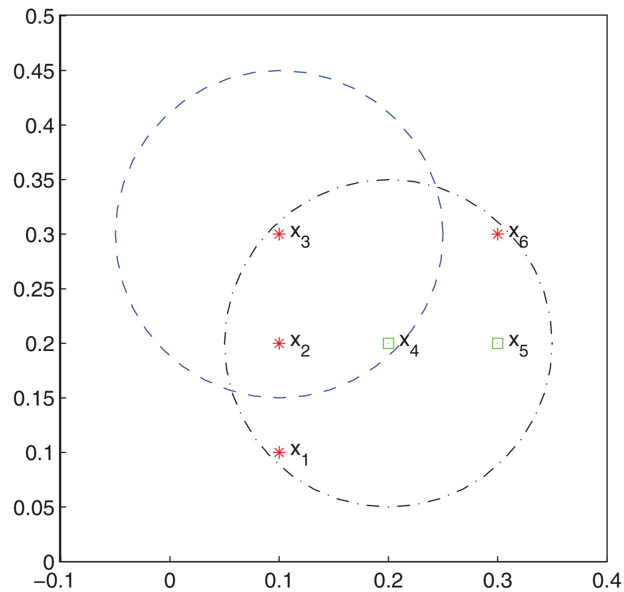

Figure 3. Data in the numerical feature space after emigrating multiple objects.

According to Algorithm 5, we update approximations dynamically as follows: Since

$$f(x_4, d) = Y$$

$$f(x_5, d) = N$$

then

$$D_Y = D_Y - \{x_4\} = \{x_1, x_2\}$$

$$D_N = D_N - \{x_5\} = \{x_3, x_6\}.$$

Since

$$\delta(x_4) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

$$\delta(x_5) = \{x_4, x_5, x_6\}$$

then

$$\delta(x_1) = \delta(x_1) - \{x_4\} = \{x_1, x_2\}$$

$$\delta(x_2) = \delta(x_2) - \{x_4\} = \{x_1, x_2, x_3\}$$

$$\delta(x_3) = \delta(x_3) - \{x_4\} = \{x_2, x_3\}$$

$$\delta(x_6) = \delta(x_6) - \{x_4\} - \{x_5\} = \{x_6\}$$

Because of

$$\delta(x_1) \cap D_Y \neq \emptyset \qquad \delta(x_1) \cap D_N = \emptyset$$

$$\delta(x_2) \cap D_Y \neq \emptyset \qquad \delta(x_2) \cap D_N \neq \emptyset$$

$$\delta(x_3) \cap D_Y \neq \emptyset \qquad \delta(x_3) \cap D_N \neq \emptyset$$

$$\delta(x_6) \cap D_Y = \emptyset \qquad \delta(x_6) \cap D_N \neq \emptyset$$

$$\delta(x_1) \subseteq D_Y \qquad \delta(x_1) \nsubseteq D_N$$

$$\delta(x_2) \nsubseteq D_Y \qquad \delta(x_2) \nsubseteq D_N$$

$$\delta(x_3) \nsubseteq D_Y \qquad \delta(x_3) \nsubseteq D_N$$

$$\delta(x_6) \nsubseteq D_Y \qquad \delta(x_6) \subseteq D_N$$

Therefore

$$\overline{N}(D_Y) = \{x_1, x_2, x_3\}$$

$$\overline{N}(D_N) = \{x_2, x_3, x_6\}$$

$$\underline{N}(D_Y) = \{x_1\}$$

$$\underline{N}(D_N) = \{x_6\}.$$

To sum up, the two incremental algorithms give an intuitive understanding after the object set immigrate or emigrate, which help us to easily extract relevant knowledge from dynamically changing large databases.

5. EXPERIMENTAL EVALUATIONS

We compare the computational time of the naive algorithm for updating approximations based on the neighborhood rough sets algorithm (NAUANRS) and the above two fast algorithms for updating approximations based on neighborhood rough sets when multiple objects immigrating or emigrating (FAUANRS-MI and FAUANRS-ME) on different data sets.

The effectiveness of incremental updating methods based on rough sets in the discrete feature space has already been discussed in the literature.[14] Here, we

Table IV. A description of data sets.

	Data sets	Samples	Features	Classes
1	Cardiotocography	2126	23	3
2	Credit	690	14	2
3	Ionosphere	351	34	2
4	Iris	150	4	3
5	Letter-recognition	20000	16	26
6	MAGIC-gamma-telescope	19020	11	2
7	Sonar-Mines-vs-Rocks	208	60	2
8	WDBC	569	31	2
9	Wine	178	13	3

mainly show that the incremental updating model based on neighborhood rough sets is applicable to the data with numerical or heterogeneous attributes in this work. To test the proposed two fast algorithms FAUANRS-MI and FAUANRS-ME, we utilize some standard data sets from the machine learning data repository, University of California at Irvine.[40] The basic information of data sets is outlined in Table IV. There are nine sets including numerical or heterogeneous features. To distinguish the computational times, we divide each of these nine data sets into ten parts of equal size. The first part is regarded as the first data set, the combination of the first part and the second part is viewed as the second data set, the combination of the second data set, and the third part is regarded as the third data set, and so on, the combination of all ten parts is viewed as the tenth data set as shown in Ref. 18. These data sets can be used to calculate time used by the above three algorithms and show it vis à vis the sizes of universe. The computations are conducted on a Dell PowerEdge T410 with Windows server 2003 and Inter(R) Core(TM)4 CPU E5504 and 4 GB memory. The algorithms are coded by C++ and running in Microsoft Visual C++ 6.0

5.1. NAUANRS and FAUANRS-MI

We compare NAUANRS with FAUANRS-MI on the nine data sets shown in Table IV. The experimental results of these nine data sets are shown in Table V and Figure 4. In each figure part of Figure 4, the x-coordinate pertains to the size of the data set (the 10 data sets starting from the smallest one), whereas the y-coordinate concerns the computational time. Table V shows the comparisons of computational time with the algorithm NAUANRS and the fast algorithm FAUANRS-MI on the nine data sets when multiple objects immigrating, whereas Figure 4 displays more detailed change trend of each of two algorithms with the increasing size of the data set. Here, we assume that the ratio of the number of immigrant data and original data is equal to 5%.

From Table V and Figure 4, the computational time of each of these two algorithms increases with the increase of the size of data. As one of the important advantages of algorithm FAUANRS-MI, as shown in Table V and Figure 4, we see

Table V. Computational time of the algorithms NAUANRS and FAUANRS-MI.

	Data sets	NAUANRS algorithm	FAUANRS-MI algorithm
1	Cardiotocography	1.0039	0.0894
2	Credit	0.1301	0.0137
3	Ionosphere	0.4819	0.0466
4	Iris	0.0069	0.0013
5	Letter-recognition	120.8800	16.5473
6	MAGIC-gamma-telescope	118.1020	13.6587
7	Sonar-Mines-vs-Rocks	0.0181	0.0022
8	WDBC	0.1059	0.0108
9	Wine	0.0092	0.0012

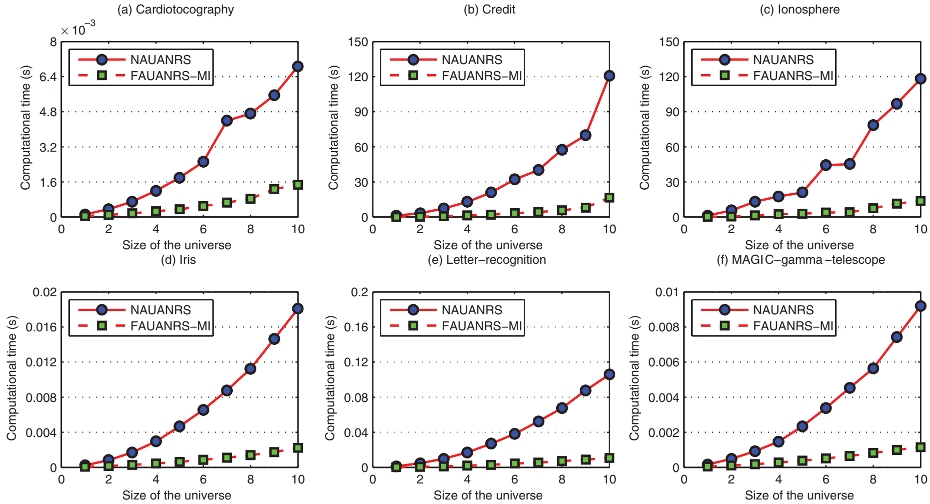

Figure 4. Times of NAUANRS and FAUANRS-MI versus the data size.

that algorithm FAUANRS-MI is much faster than algorithm NAUANRS on updating approximations when the same objects enter into the neighborhood decision table.

As we know, the new objects will enter into the neighborhood decision system over time. Let R_{im} be the ratio of the number of immigrant data and original data, called immigrant ratio. Every time, the number of immigrant data is different. It

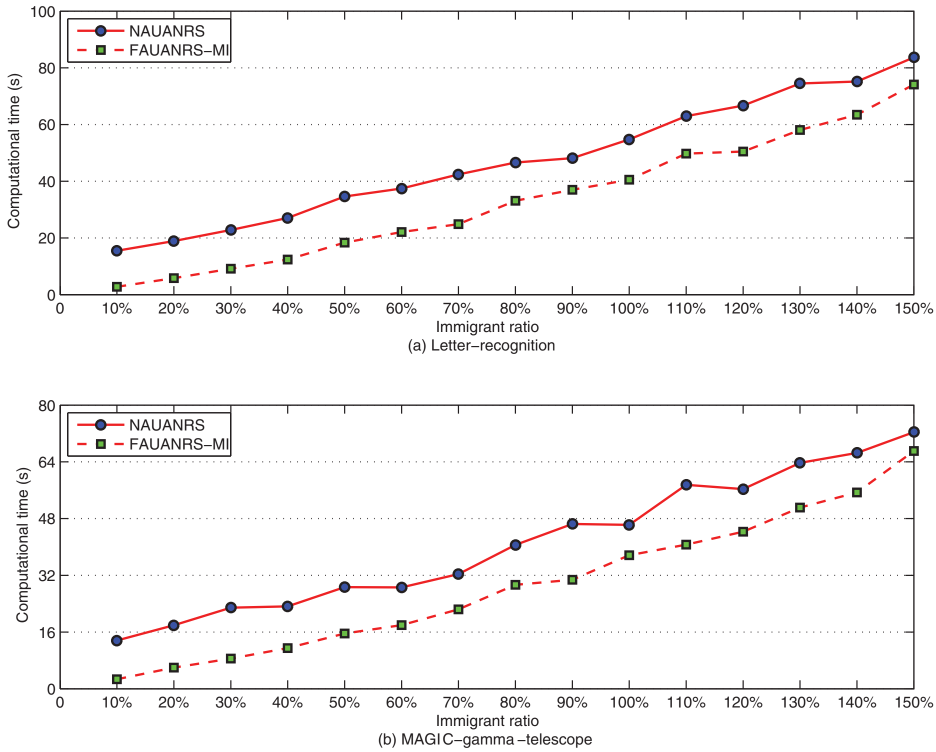

Figure 5. Times of Algorithms NAUANRS and FAUANRS-MI versus the immigrant ratio of data.

means R_{im} is different. We test the two algorithms NAUANRS and FAUANRS-MI with the changing of R_{im} on two large data sets Letter-recognition and MAGIC-gamma-telescope shown in Table IV. The experimental results of these two data sets are shown in Figure 5.

In each figure part of Figure 5, the x-coordinate pertains to the immigrant ratio of the data, whereas the y-coordinate concerns the computational time. In Figure 5a, we find algorithm FAUANRS-ME is always much faster than NAUANRS algorithm when $R_{im} < 150\%$. In Figure 5b, we also find algorithm FAUANRS-ME is always much faster than NAUANRS algorithm when $R_{im} < 150\%$.

5.2. NAUANRS and FAUANRS-ME

We compare NAUANRS with FAUANRS-ME on the nine real-world data sets shown in Table IV. The experimental results of these nine data sets are shown in Table VI and Figure 6. In Figure 6, we display more detailed change trend line of each of two algorithms with the increasing size of the data set. Table VI shows the comparisons of computational time with the algorithm NAUANRS and the

Table VI. Computational time of the algorithms NAUANRS and FAUANRS-ME.

	Data sets	NAUANRS algorithm	FAUANRS-ME algorithm
1	Cardiotocography	1.0566	0.0043
2	Credit	0.1182	0.0017
3	Ionosphere	0.4525	0.0034
4	Iris	0.0064	0.0007
5	Letter-recognition	114.7880	0.7584
6	MAGIC-gamma-telescope	97.8761	0.6440
7	Sonar-Mines-vs-Rocks	0.0166	0.0004
8	WDBC	0.0950	0.0012
9	Wine	0.0086	0.0004

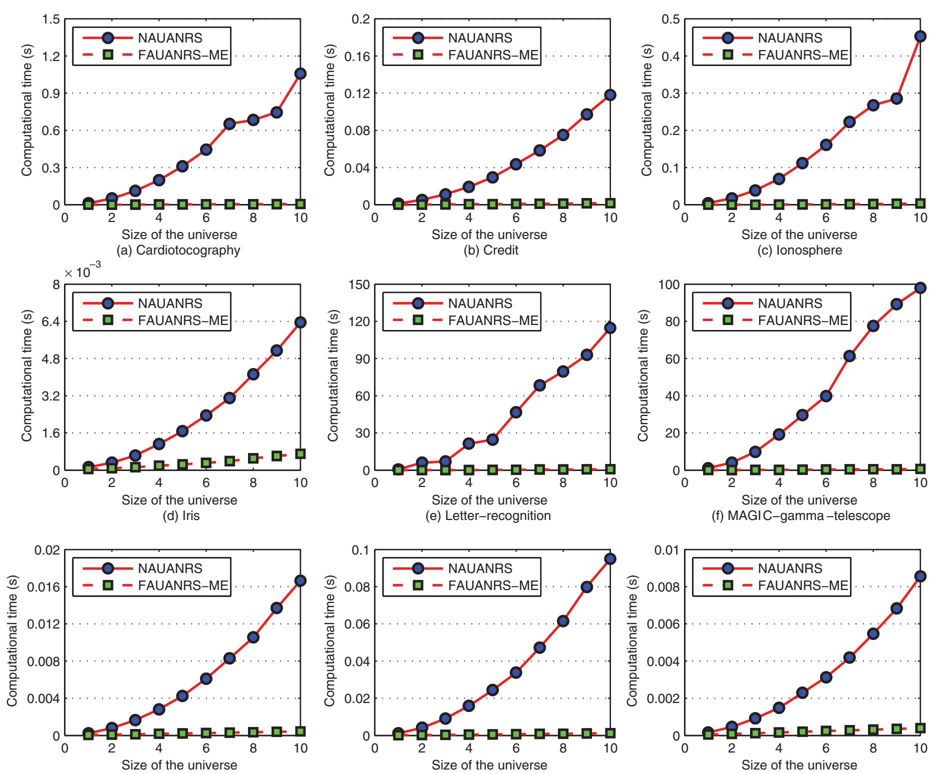

Figure 6. Times of NAUANRS and FAUANRS-ME versus the data size.

fast algorithm FAUANRS-ME on nine data sets when multiple objects emigrating. Similarly, in each of these Figures 6a–6i, the x-coordinate pertains to the size of the data set (the 10 data sets starting from the smallest one), whereas the y-coordinate concerns the computing time. Here, we assume that the ratio of the number of emigrant data and original data is equal to 5%.

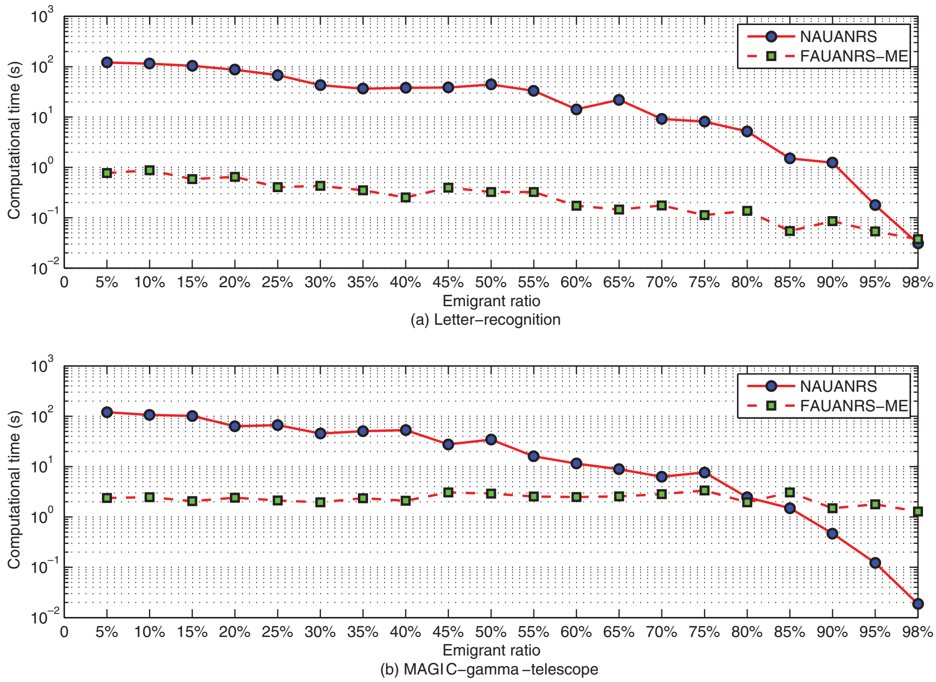

Figure 7. Times of NAUANRS and FAUANRS-ME versus the emigrant ratio of data.

From Table VI and Figure 6, the computational time of each of these two algorithms increases with the increase of the size of data. As one of the important advantages of the FAUANRS-ME, as shown in Table VI and Figure 6, we see that algorithm FAUANRS-ME is much faster than algorithm NAUANRS on updating approximations when the same objects gets out of the neighborhood decision table.

As we know, the unvalued objects will get out of the neighborhood decision system over time. Let R_{em} be the ratio of the number of emigrant data and original data, called the emigrant ratio. In each time, the number of emigrant data is different, which means R_{em} is different. We test the two algorithms NAUANRS and FAUANRS-ME with the changing of R_{em} on two large data sets Letter-recognition and MAGIC-gamma-telescope shown in Table IV. The experimental results of these two data sets are shown in Figure 7.

In each of these figure parts of Figure 7, the x-coordinate pertains to the emigrant ratio of the data, whereas the y-coordinate concerns the computational time. In Figure 7a, we find algorithm FAUANRS-ME is much faster than algorithm NAUANRS when $R_{em} < 98\%$, but algorithm FAUANRS-ME is slower when $R_{em} > 98\%$. In Figure 7b, we find the algorithm FAUANRS-ME is much faster than algorithm NAUANRS when $R_{em} < 80\%$, but FAUANRS-ME algorithm is slower when $R_{em} > 80\%$.

6. CONCLUSIONS

The incremental methods based on the classical rough set model is widely discussed by many researchers. However, this model can just deal with nominal data. In this paper, we discussed the change of approximations in neighborhood decision systems when the object set evolves over time. We proposed a method of incrementally updating approximations based on neighborhood rough sets in dynamic data mining. We designed a naive algorithm for updating approximations (NAUANRS) and two fast algorithms for updating approximations (FAUANRS-MI and FAUANRS-ME) when multiple objects enter into or get out of the neighborhood decision table. The experimental results show that the proposed method can deal with both categorical attributes and numerical variables without discretization. Experimental studies pertaining to nine UCI data sets show that the proposed algorithms significantly reduce computational time of updating approximations.

Acknowledgments

This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630127), the Fundamental Research Funds for the Central Universities (No. SWJTU11ZT08), and the Doctoral Innovation Foundation of Southwest Jiaotong University (No. 2012ZJB), China.

References

1. Han J, Kamber M. Data mining: Concepts and techniques, second ed.; San Francisco; CA: Morgan Kaufman; 2006.
2. Hsu PY, Chen YL, Ling CC. Algorithms for mining association rules in bag databases. Inf Sci 2004;166:31–47.
3. Yu JX, Chong Z, Lu H, Zhang Z, Zhou A. A false negative approach to mining frequent itemsets from high speed transactional data streams. Inf Sci 2006;176:1986–2015.
4. Hu YC, Tzeng GH, Chen CM. Deriving two-stage learning sequences from knowledge in fuzzy sequential pattern mining. Inf Sci 2004;159:69–86.
5. Lagus K, Kaski S, Kohonen T. Mining massive document collections by the WEBSOM method. Inf Sci 2004;163:135–156.
6. Li Y, Zhu S, Wang XS, Jajodia S. Looking into the seeds of time: discovering temporal patterns in large transaction sets. Inf Sci 2006;176:1003–1031.
7. Pawlak Z. Rough set theory and its application to data analysis. Cybern Syst 1998;29:661–688.
8. Pawlak Z. Rough sets: Theoretical aspects of reasoning about data. Boston, MA: Kluwer; 1991.
9. Pawlak Z, Skowron A. Rudiments of rough sets. Inf Sci 2007;177:3–27.
10. Pawlak Z, Skowron A. Rough sets: Some extensions. Inf Sci 2007;177:28–40.
11. Hu QH, Yu DR, Xie ZX. Neighborhood classifiers. Expert Syst Appl 2008;34:866–876.
12. Hu QH, Liu JF, Yu DR. Mixed feature selection based on granulation and approximation. Knowl-Based Syst 2008;21:294–304.
13. Hu QH, Yu DR, Liu JF, Wu CX. Neighborhood rough set based heterogeneous feature subset selection. Inf Sci 2008;178:3577–3594.

14. Liu D, Li TR, Ruan D, Zou WL. An incremental approach for inducing knowledge from dynamic information systems. Fundam Inf 2009;94:245–260.
15. Liu D, Li TR, Ruan D, Zhang JB. Incremental learning optimization on knowledge discovery in dynamic business intelligent systems. J Global Optim 2011;51:325–344.
16. Qian YH, Liang JY, Yao YY, Dang CY. MGRS: A multi-granulation rough set. Inf Sci 2010;180:949–970
17. Qian YH, Liang JY, Dang CY. Incomplete multigranulation rough set. IEEE Trans Syst Man Cybern, PART A 2010;40:420–431.
18. Qian YH, Liang JY, Witold P, Dang CY. Positive approximation: an accelerator for attribute reduction in rough set theory. Artif Intell 2010;174:597–618.
19. Yao YY. Information granulation and rough set approximation. Int J Intell Syst 2001;16:87–104.
20. Muni DP, Pal NR, Das J. Genetic programming for simultaneous feature selection and classifier design. IEEE Trans Syst Man Cyberne Part B: Cybernet 2006;36(1):106–117.
21. Jin W, Tung Anthony KH, Han J, Wang W. Ranking outliers using symmetric neighborhood relationship. In: PAKDD; 2006. pp 577–593.
22. Wang H. Nearest neighbors by neighborhood counting. IEEE Transa PAMI 2006;28:942–953.
23. Vijay R, Alaaeldin H. Dynamic data mining. Lect Notes Comput Sci 2000;1821:207–238.
24. Shan L, Ziarko W. Data-based acquisition and incremental modification of classification rules. Comput Intell 1995;11:357–370.
25. Bang W, Bien Z. New incremental learning algorithm in the framework of rough set theory. Int J Fuzzy Syst 1999;1:25–36.
26. Tong L, An L. Incremental learning of decision rules based on rough set theory. In: IWCICA2002; 2002. pp 420–425.
27. Jerzy B, Slowinski R. Incremental induction of decision rules from dominance-based rough approximations. Electr Notes Theor Comput Sci 2003;82:40–51.
28. Zheng Z, Wang G. RRIA: a rough set and rule tree based incremental knowledge acquisition algorithm. Fundam Inf 2004;59:299–313.
29. Hu F, Wang G, Huang H, Wu Y. Incremental attribute reduction based on elementary sets. In: RSFDGrC2005. LNAI 2005;3641:185–193.
30. Li SY, Li TR, Chen HM, Ji XL. An incremental approach for updating approximations in dominance-based rough sets. In: Proc 9th Int FLINS Conference, Chengdu, China; 2010. pp 728–733.
31. Chan CC. A rough set approach to attribute generalization in data mining. Inf Sci 1998;107:177–194.
32. Li TR, Ruan D, Geert W, Song J, Xu Y. A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowl-Based Syst 2007;20:485–494.
33. Cheng Y. The incremental method for fast computing the rough fuzzy approximations. Data Knowl Eng 2011;70:84–100.
34. Liu D, Zhang JB, Li TR. A probabilistic rough set approach for incremental learning knowledge on the change of attribute. In: Proc 2010 Int Conf on Foundations and Applications of Computational Intelligence; 2010. pp 722–727.
35. Zhang JB, Li TR, Liu D. An approach for incremental updating approximations in variable precision rough sets while attribute generalizing. In: Proc 2010 IEEE Int Conf on Intelligent Systems and Knowledge Engineering, Hangzhou, China; 2010. pp 77–81.
36. Chen HM, Li TR, Qiao SJ, Ruan D. A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values. Int J Intell Syst 2010;25:1005–1026.
37. Liu D, Li TR, Liu GR, Hu P. An incremental approach for inducing interesting knowledge based on the change of attribute values. In: Proc 2009 IEEE Int Conf on Granular Computing, Nanchang, China; 2009. pp 415–418.

38. Liu D, Li TR, Zhang JB. An incremental approach for rule induction under coarsening and refining of attribute values in e-business Systems. In: Proc 2010 Int Conf on Electronic-Business Intelligence; 2010. pp 541–547.
39. Randall Wilson D. Martinez TR. Improved heterogeneous distance functions. J Artif Intell Res 1997;6:1–34.
40. Newman DJ, Hettich S, Blake CL, Merz CJ. UCI repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA, 1998. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html. Accessed Oct 20, 2010.
41. Pawlak Z, Skowron A. Rough sets and boolean reasoning. Inf Sci 2007;177:41–73.
42. Ziarko W. Incremental learning and evaluation of structures of rough decision tables. Trans Rough Sets. 2005;IV:162–177.