IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 22, 2019, accepted April 13, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913312

Knowledge Acquisition Approach Based on Incremental Objects From Data With Missing Values

WENHAO SHU[1], WENBIN QIAN[2,3], AND YONGHONG XIE[3]
[1]School of Information Engineering, East China Jiaotong University, Nanchang 330013, China
[2]School of Software, Jiangxi Agricultural University, Nanchang 330045, China
[3]Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Wenbin Qian (qianwenbin1027@126.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61662023 and Grant 61502213, and in part by the Science and Technology Research Project of Education Department of Jiangxi Province under Grant GJJ180200.

ABSTRACT Knowledge acquisition is the process of extracting useful knowledge from data sets to analyze data in areas of data mining and knowledge discovery. Most current knowledge acquisition work mainly focuses on static data. However, due to the dynamic characteristics of data, the objects grow at an unprecedented rate in real-world data sets. The incremental objects with a dynamic environment significantly affect knowledge updating. To maintain the effectiveness of knowledge from the dynamic data, it is necessary to update the knowledge timely. So far, there are relatively few studies on knowledge acquisition for the data with missing feature values, i.e., incomplete data. To handle with this issue, an incremental updating manner of the accuracy matrix and coverage matrix are first proposed on the basis of the computations of the tolerance classes in incomplete data, which plays an important role in the knowledge acquisition process. Then, an incremental knowledge acquisition algorithm is proposed when some new objects added to the data with missing values. Finally, some numerical experiments are conducted to evaluate the efficiency of the proposed algorithm.

INDEX TERMS Knowledge acquisition, incremental objects, incomplete decision system, granular computing, tolerance relation.

I. INTRODUCTION

As one of data analysis techniques, rough sets-based methods have been successfully applied in data mining and knowledge discovery during last decades [1]–[3], and particularly useful for rule acquisition [4] and feature selection [5] . Rough set theory introduced by Pawlak [6], [7] is a knowledge acquisition tool that can be used to help induce logical patterns hidden in big data. This knowledge can then be presented to the decision-maker as convenient decision rules. Its strength lies in its ability to deal with imperfect data and to classify. Knowledge hidden in information systems may be unraveled and expressed in the form of decision rules [8], [29]–[34]. The extracted rules can be used for making predictions in different domains. Often, the knowledge acquisition algorithms established on Pawlak's rough set model assumed that all feature values are complete.

The associate editor coordinating the review of this manuscript and approving it for publication was Eunil Park.

However, missing data in real-world applications often appears [10]–[16], [21], [22]. For example, in the warning management system of citys traffic, it may arise out of the problem of information transmission and congestion, the fault of sensor, or errors made by the human. The classical rough set theory based on the equivalence relation cannot handle effectively such situation. The main reason is that the equivalence relation is limited in dealing with the missing feature values. One possible way to solve this problem is to fill up the feature values, but it may cause information loss [9]. Another feasible way to overcome this shortcoming is to extend the classical rough set theory [10]–[12]. A pioneering work on dealing with incomplete data was proposed by Kryszkiewicz [11] by defining a tolerance relation. In the tolerance-rough set, a tolerance relation characterizes the relationship between two objects instead of the equivalence relation used in the classical rough set. To date, many research results have been obtained in the field of tolerance-rough set [13], [14]. Note that the extended rough set is beneficial to the implementation of knowledge acquisition for incomplete data [15], [30]. For example, Leung *et al.* [15] used the lower and upper approximations to form the mining of certain and association rules in static incomplete data sets. Wu *et al.* [30] proposed a rough set approach to knowledge discovery in incomplete multi-scale decision tables from the perspective of granular computing. Knowledge acquisition in incomplete data is of interest because such datasets are frequently encountered in the real world. However, with the rapid growth of data sets in real-world applications, an object set in the incomplete data may change with time when new information arrives. The fast increase of objects in the data with missing values brings a new challenge to quickly extract useful information with data mining techniques. Therefore, this paper mainly focuses on this issue of knowledge acquisition in dynamic incomplete data with the arrival of new objects.

At present, the knowledge acquisition approaches based on rough set theory have received much attention. Li *et al.* [16] proposed an approach for extracting non-redundant approximate decision rules from an incomplete decision context. In addition, Shao *et al.* [17] formulated an approach to extract "if-then" rule from formal decision contexts by using formal concept analysis. Shi *et al.* [18] employed rough sets and association rule mining to generate knowledge which describes the relationship between the critical form features and the corresponding KANSEI adjectives. Feng *et al.* [2] proposed a vague-rough set approach for extracting knowledge in a vague decision information system. Du *et al.* [19] proposed a neighborhood covering reduction based approach to extract rules from numerical data. Tan *et al.* [20] proposed a fast approach to knowledge acquisition in covering information systems using matrix operations. Prado *et al.* [22] proposed a knowledge acquisition method in fuzzy-rule-based systems with particle-swarm optimization. Dai *et al.* [23] proposed a rough set approach for rule induction based on classification consistency rate in inconsistent data. Li *et al.* [24] investigated the relationship between multi-granulation rough sets and concept lattices via rule acquisition. Zhang *et al.* [25] presented parallel large-scale rough set based methods for knowledge acquisition using MapReduce. Zhang *et al.* [28] proposed a confidence-preserved attribute reduction approach to extract compact decision rules from an interval-valued decision system.

To mine knowledge from very large data sets based on rough sets, incremental techniques are employed to improve the computational efficiency. Liu *et al.* [26] gave an incremental model and approach as well as its algorithm for inducing interesting knowledge when the object set varies over time in the complete information system. Fan *et al.* [27] proposed an incremental rule-extraction algorithm based on the previous rule-extraction algorithm when a new object is added to an information system. As an efficient data analysis's technique, the incremental approaches have become one of the hot topics on knowledge acquisition from the dynamic data sets. Applying the incremental method, it is unnecessary to recompute the new knowledge from the beginning, which only update the new knowledge by partially modifying the original knowledge, such that the computational efficiency is improved. Therefore, this paper utilizes the incremental strategy to extract useful knowledge from data with missing values based on incremental objects.

The main contributions of this paper are summarized as follows: (1) Upon the arrival of new objects, the accuracy matrix and coverage matrix are incrementally computed without re-scanning the dynamic system. (2) An incremental knowledge acquisition algorithm is proposed to update the interesting knowledge when some new objects add into the data with missing values, rather than to compute the whole new system from scratch. (3) The time efficiency of the proposed algorithms against the non-incremental algorithm is validated on different UCI data sets.

This paper is organized as follows. In Section 2, we review some basic concepts referred in this paper. In Section 3, the computation of tolerance classes in incomplete data is presented, which will be used in a later subsection. In Section 4, the incremental computations of the accuracy matrix and coverage matrix for data with missing values at the arrival of new objects are analyzed. On this basis, an incremental knowledge acquisition algorithm is developed to update the interesting knowledge. Experimental analysis is given in Section 5. The paper ends with conclusions in Section 6.

II. PRELIMINARIES

In this section, we firstly review some basic concepts such as incomplete decision system and tolerance relation [11], [15]. Then we introduce some necessary concepts of knowledge acquisition, such as accuracy and coverage of rules [26], [27].

Data sets are usually given as the form of tables, we call a data table as an information system, formulated as $IS = < U, A, V, f >$, where,

(1) U is a set of nonempty and finite objects, called the universe;

(2) A is a set of features characterizing the objects;

(3) V is the union of feature domains, i.e., $V = \cup_{a \in A} V_a$, where V_a is the value set of feature a, called the domain of a;

(4) $f : U \times A \to V$ is an information function, which assigns feature values to objects such as $\forall a \in A$, $x \in U$, and $f(x, a) \in V_a$, where $f(x, a)$ denotes the value of feature a for object x.

Each subset of features $B \subseteq A$ determines a binary indiscernibility relation $IND(B)$ on U as follows: $IND(B) = \{(x, y) \in U \times U | \forall a \in B, f(x, a) = f(y, a)\}$. It can be easily shown that $IND(B)$ is an equivalence relation and it constructs a partition of U, denoted by $U/IND(B) = \{[x]_B | x \in U\}$, where $[x]_B$ denotes the equivalence class containing x. The elements in $[x]_B$ are indiscernible or equivalent with respect to B, i.e., $[x]_B = \{y \in U | (x, y) \in IND(B)\}$.

If the feature set is divided into condition feature set C and decision feature set D, the information system is called a decision system. If there exist $x \in U$ and $a \in A$ such that $f(x, a)$ is equal to a missing value (a null or unknown value, denoted as "*"), i.e., $* \in V_a$, then the information system is an incomplete information system (IIS). Otherwise, it is a complete information system (CIS). If $* \notin V_D$ but $* \in V_C$, then the decision system is an incomplete decision system (IDS).

Generally speaking, the data with missing values is represented as an incomplete decision system. Given an incomplete information system $IIS =< U, C \cup D, V, f >$, for $\forall B \subseteq C$, a tolerance relation between objects that are possibly indiscernible in terms of B is defined by $TR(B) = \{(x, y)|\forall a \in B, f(x, a) = f(y, a) \vee f(x, a) = * \vee f(y, a) = *\}$. Obviously, $TR(B)$ is reflexive and symmetric but not transitive. It can be easily shown that $TR(B) = \cap_{a\in B} TR(\{a\})$. The tolerance class of object x with reference to a feature set B is denoted as $T_B(x) = \{y|(x, y) \in TR(B)\}$. Let $U/TR(B)$ denote the family set $\{T_B(x)|x \in U\}$, which is the classification induced by B. For $X \subseteq U$, the lower and upper approximation of X with respect to B can be defined as $\underline{B}(X) = \{x \in U|T_B(x) \subseteq X\}$ and $\overline{B}(X) = \{x \in U|T_B(x) \cap X \neq \varnothing\}$. X is called $B-$definable if and only if $\overline{B}(X) = \underline{B}(X)$. Otherwise, $\overline{B}(X) \neq \underline{B}(X)$ and X is rough.

Definition 1: Let $IDS =< U, C \cup D, V, f >$ be an incomplete decision system, $U = \{x_1, x_2, \ldots, x_m\}$, for $\forall x_i \in U(1 \leq i \leq m)$, $U/D = \{D_1, D_2, \ldots, D_n\}$, where $D_j(1 \leq j \leq n)$ is a decision class. The decision rule γ_{x_i} of object x_i is defined by: $des([x_i]_C) \rightarrow des(D_j)$.

For the decision rule γ_{x_i}, $des([x_i]_C) =_{\wedge c\in C} (c, f(x_i, c))$ is the descriptions of object x_i under the condition feature set C, $des(D_j) =_{\wedge d\in D} (d, f(x_i, d))$ is the descriptions of object x_i under the decision feature set D.

The remarkable rules extracted from the incomplete decision system are the interesting knowledge. Because accuracy and coverage are the two statistical measures for rule induction [26], [27], we focus on the accuracy and coverage as two important factors to describe the interesting knowledge in this paper.

Definition 2: Let $IDS =< U, C \cup D, V, f >$ be an incomplete decision system, $U = \{x_1, x_2, ..., x_m\}$, $U/TR(C) = \{T_C(x_1), T_C(x_2), ..., T_C(x_m)\}$ is the classification induced by C, where $T_C(x_i)(1 \leq i \leq m)$ is a tolerance class, $U/IND(D) = \{D_1, D_2, \ldots, D_n\}$, where $D_j(1 \leq j \leq n)$ is a decision class. The accuracy, and coverage of γ_{x_i} are defined respectively by:

Accuracy of $des([x_i]_C) \rightarrow des(D_j)$:

$$Acc(D_j|T_C(x_i)) = \frac{|T_C(x_i) \cap D_j|}{|T_C(x_i)|};$$

Coverage of $des([x_i]_C) \rightarrow des(D_j)$:

$$Cov(D_j|T_C(x_i)) = \frac{|T_C(x_i) \cap D_j|}{|D_j|}.$$

Then the accuracy matrix of the incomplete decision system is constructed by, $Acc(D|U)$, as shown at the bottom of this page, and the coverage matrix of the incomplete decision system is constructed by, $Cov(D|U)$, as shown at the bottom of this page.

By this definition, we can easily obtain the value range of the accuracy and the coverage of γ_{x_i}, respectively.

Proposition 1: $0 \leq Acc(D_j|T_C(x_i)) \leq 1$, $0 \leq Cov(D_j|T_C(x_i)) \leq 1$, $\forall x_i \in U(1 \leq i \leq m)$, $D_j \in U/IND(D)(1 \leq j \leq n)$.

According to two measures of rule quality, the interesting knowledge can be considered as follows if the association rule satisfies both a minimum threshold. Such threshold can be set by users or domain experts.

Definition 3: If $Acc(D_j|T_C(x_i)) \geq \alpha$ and $Cov(D_j|T_C(x_i)) \geq \gamma$, $\forall x_i \in U(1 \leq i \leq m)$, $D_j \in U/IND(D)$ $(1 \leq j \leq n)$, then the rule $des([x_i]_C) \rightarrow des(D_j)$ of object x_i is an interesting knowledge where $\alpha \in [0, 1]$ and $\gamma \in [0, 1]$.

Since the accuracy and coverage factors of rule induction are two statistical measures, a classification error parameter $\beta = 1 - |T_C(x_i) \cap [x_i]_D|/|T_C(x_i)|(|T_C(x_i)| > 0)$ proposed in [26], [27] is used to obtain the accuracy value, we get $\alpha = 1 - \beta$. From Definition 3, the interesting knowledge can be generated with high accuracy and high coverage from the incomplete decision system.

III. COMPUTATION OF TOLERANCE CLASSES FOR DATA WITH MISSING VALUES

In this section, we introduce the computation of tolerance classes [33] for the incomplete decision system to classify objects in the knowledge acquisition process.

The classical method of computing tolerance classes has to compare the feature values of objects by pairwise under the whole condition feature set. The time complexity is $O(|U|^2|C|)$, where U is the number of objects from a given universe and C is the number of conditional features in the

$$Acc(D|U) = \begin{pmatrix} Acc(D_1|T_C(x_1)) & Acc(D_2|T_C(x_1)) & \ldots & Acc(D_n|T_C(x_1)) \\ Acc(D_1|T_C(x_2)) & Acc(D_2|T_C(x_2)) & \ldots & Acc(D_n|T_C(x_2)) \\ \ldots & \ldots & \ldots & \ldots \\ Acc(D_1|T_C(x_m)) & Acc(D_2|T_C(x_m)) & \ldots & Acc(D_n|T_C(x_m)) \end{pmatrix}$$

$$Cov(D|U) = \begin{pmatrix} Cov(D_1|T_C(x_1)) & Cov(D_2|T_C(x_1)) & \ldots & Cov(D_n|T_C(x_1)) \\ Cov(D_1|T_C(x_2)) & Cov(D_2|T_C(x_2)) & \ldots & Cov(D_n|T_C(x_2)) \\ \ldots & \ldots & \ldots & \ldots \\ Cov(D_1|T_C(x_m)) & Cov(D_2|T_C(x_m)) & \ldots & Cov(D_n|T_C(x_m)) \end{pmatrix}$$

incomplete decision system. As a result, it is often time-consuming to use this method to satisfy the requirement for large volumes of data. In order to enhance the time efficiency, we just first compute all the blocks (elementary sets) for the objects containing no missing feature values under each condition feature, then the tolerance class of each object is the residual object set through sequentially minus some blocks that from the whole set of objects.

In what follows, an example is given to illustrate this computation with a small incomplete decision system shown in Table 1.

TABLE 1. An incomplete decision system.

Car	P	M	S	X	Acceleration
x_1	High	High	Full	Low	Good
x_2	Low	*	Full	Low	Good
x_3	*	*	Compact	Low	Poor
x_4	High	*	Full	High	Good
x_5	*	*	Full	High	Excellent
x_6	Low	High	Full	*	Good

An incomplete decision system from [11], [13] is shown in Table 1, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. For convenience, in the sequel, P,M,S and X will stand for Price, Mileage, Size, Max-speed, respectively. $C = \{P, M, S, X\}$ and $D = \{\text{Acceleration}\}$. By the method of [33], the elementary sets of each condition feature are computed as $[(P, High)] = \{x_1, x_4\}$, $[(P, Low)] = \{x_2, x_6\}$; $[(M, High)] = \{x_1, x_6\}$; $[(S, Full)] = \{x_1, x_2, x_4, x_5, x_6\}$, $[(S, Compact)] = \{x_3\}$; $[(X, Low)] = \{x_1, x_2, x_3\}$, $[(X, High)] = \{x_4, x_5\}$, then, the tolerance classes of each object are computed as follows:

$$T_C(x_1) = U - \{x_2, x_6\} - \{x_3\} - \{x_4, x_5\} = \{x_1\},$$
$$T_C(x_2) = U - \{x_1, x_4\} - \{x_3\} - \{x_4, x_5\} = \{x_2, x_6\},$$
$$T_C(x_3) = U - \{x_1, x_2, x_4, x_5, x_6\} - \{x_4, x_5\} = \{x_3\},$$
$$T_C(x_4) = U - \{x_2, x_6\} - \{x_3\} - \{x_1, x_2, x_3\} = \{x_4, x_5\},$$
$$T_C(x_5) = U - \{x_3\} - \{x_1, x_2, x_3\} = \{x_4, x_5, x_6\},$$
$$T_C(x_6) = U - \{x_1, x_4\} - \{x_3\} = \{x_2, x_5, x_6\}.$$

IV. THE NON-INCREMENTAL KNOWLEDGE ACQUISITION APPROACH FOR DATA WITH MISSING VALUES AT THE ARRIVAL OF NEW OBJECTS

In practice, data processing tools have been developed rapidly in recent years. Thus the incomplete decision systems may increase quickly in objects with time in real-life applications. Suppose that many objects are added to the system, the general (non-incremental) algorithm needs to compute the accuracy matrix and coverage matrix repeatedly on the incomplete decision system, which may be inefficient. Given a dynamic incomplete decision system, the non-incremental knowledge acquisition approach updates the knowledge from the scratch. The detailed process procedure is presented in Algorithm 1, which is denoted by Algorithm NKAC.

Algorithm 1 The Non-Incremental Knowledge Acquisition Algorithm for Data With Missing Values at the Arrival of New Objects (Algorithm NKAC)

Input: An incomplete decision system $IDS = < U, C \cup D, V, f >$ at time t, where $U = \{x_1, x_2, \ldots, x_m\}$, a new adding object x at time $t+1$, two thresholds α and β;
Output: Interesting knowledge at time $t+1$.
Begin

1) % Compute the new accuracy matrix and new coverage matrix at time $t+1$ from the scratch, and output the interesting knowledge
2) Let $x_{m+1} = x$ and $U' = U \cup \{x\} = \{x_1, x_2, \ldots, x_m, x_{m+1}\}$;
3) Compute the tolerance classes $U'/TR(C) = \{T'_C(x_1), T'_C(x_2), \ldots, T'_C(x_m), T'_C(x_{m+1})\}$ by Theorem 1;
4) Compute decision classes $U'/IND(D) = \{D'_1, D'_2, \ldots, D'_n\}$;
5) for $i = 1$ to $m+1$ do
6) for $j = 1$ to n do
7) recompute the accuracy matrix $Acc^{(t+1)}(D'_j|T'_C(x_i))$, and the coverage matrix $Cov^{(t+1)}(D'_j|T'_C(x_i))$ at time $t+1$;
8) for $i = 1$ to $m+1$ do
9) for $j = 1$ to n do
10) if $Acc^{(t+1)}(D'_j|T'_C(x_i)) \geq \alpha$ and $Cov^{(t+1)}(D'_j|T'_C(x_i)) \geq \beta$, Output the interesting knowledge $des[x_i]_C \rightarrow des(D'_j)$;

End

From Algorithm NKAC, it treats the dynamic data by the adding objects as absolutely new data without using any incremental strategy and directly compute the interesting knowledge, which does not take into consideration the previous results thus may be time-consuming. When confronting a new adding object, Steps 3-4 are to compute the tolerance classes and decision classes respectively, the results will be prerequisite for computing the accuracy and coverage of each object. Steps 5-7 are to compute the accuracy matrix and coverage matrix from the scratch. Steps 8-10 are to update the interesting knowledge by Definition 3.

In what follows, an illustrative example is employed to analyze the knowledge acquisition process by the algorithm NKAC.

We illustrate the proposed algorithm NKAC using the incomplete decision system shown in Table 1, Table 1 is taken as the original incomplete decision system. Now, a new object x_7 is added into the system, where $x_7 = \{$High, High, Full, High, Excellent$\}$ is added into the system. Suppose $\alpha = 0.6$, $\beta = 0.4$, the new knowledge acquisition process is shown as follows.

As that in Example 1 for Table 1, let $x_7 = x$, $U' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$. Compute the tolerance classes by Theorem 1 as follows: $T'_C(x_1) = \{x_1\}$, $T'_C(x_2) = \{x_2, x_6\}$, $T'_C(x_3) = \{x_3\}$, $T'_C(x_4) = \{x_4, x_5, x_7\}$, $T'_C(x_5) = \{x_4, x_5, x_6, x_7\}$, $T'_C(x_6) = \{x_2, x_5, x_6\}$ and $T'_C(x_7) = \{x_4, x_5, x_7\}$.

Compute the decision classes $U'/IND(D) = \{D'_1, D'_2, \ldots, D'_n\}$, where $D'_1 = \{x_1, x_2, x_4, x_6\}$, $D'_2 = \{x_3\}$, $D'_3 = \{x_5, x_7\}$. Then, the new accuracy matrix and new coverage matrix are recomputed as follows:

$$Acc^{(t+1)}(D'_j|T'_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}$$

$$Cov^{(t+1)}(D'_j|T'_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & 1 \\ \frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & 1 \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$Acc^{(t+1)}(D'_1|T'_C(x_2)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_2)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.4;$$
$$Acc^{(t+1)}(D'_3|T'_C(x_4)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_3|T'_C(x_4)) = 1 \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_6)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_6)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_3|T'_C(x_7)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_3|T'_C(x_7)) = 1 \geq 0.4.$$

we can find out the interesting knowledge as follows:

$$des[x_2]_C \to des(D'_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \wedge (X, Low) \to (Acceleration, Good);$$
$$des[x_3]_C \to des(D'_2) : (P, *) \wedge (M, *) \wedge (S, Compact) \wedge (X, Low) \to (Acceleration, Poor);$$
$$des[x_4]_C \to des(D'_3) : (P, High) \wedge (M, *) \wedge (S, Full) \wedge (X, High) \to (Acceleration, Excellent);$$
$$des[x_6]_C \to des(D'_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \wedge (X, *) \to (Acceleration, Good);$$
$$des[x_7]_C \to des(D'_3) : (P, High) \wedge (M, High) \wedge (S, Full) \wedge (X, High) \to (Acceleration, Excellent).$$

V. INCREMENTAL KNOWLEDGE ACQUISITION ALGORITHM FOR DATA WITH MISSING VALUES AT THE ARRIVAL OF NEW OBJECTS

The above analysis shows that recalculating the interesting knowledge is not a wise method when confronting the dynamic incomplete data with new adding objects, since it needs to compute repeatedly and consume a large amount of computational time, while incremental methods attracted by many scholars are efficient to deal with such data because they can directly run the computation by using the previous results [5], [26]. In this section, we first analyze the incremental computation of accuracy and coverage of each object to acquire new knowledge when new objects are added into the incomplete data. Then we give an illustrative example to explain this incremental manner. Finally, we design an incremental knowledge acquisition algorithm for inducing interesting knowledge.

To describe a dynamic incomplete data, we denote an incomplete decision system at time t as $IDS = <U, C \cup D, V, f>$, $U = \{x_1, x_2, \ldots, x_m\}$, with the classification $U/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T_C(x_m)\}$, where $T_C(x_i)(1 \leq i \leq m)$ is a tolerance class; $U/IND(D) = \{D_1, D_2, \cdots, D_n\}$, where $D_j(1 \leq j \leq n)$ is a decision class. At time $t+1$, some objects are added into the system, the original incomplete decision system IDS will become $IDS' =< U', C \cup D, V', f' >$. At time t, we denote the accuracy and coverage of each object as $Acc^{(t)}(D_j|T_C(x_i))$ and $Cov^{(t)}(D_j|T_C(x_i))$ at time t, the decision rule $\gamma_{x_i} : des([x_i]_C) \to des(D_j)$ is interesting if $Acc^{(t)}(D_j|T_C(x_i)) \geq \alpha$ and $Cov^{(t)}(D_j|T_C(x_i)) \geq \gamma$. In a similar way, the accuracy and coverage will become $Acc^{(t+1)}(D'_j|T'_C(x_i))$ and $Cov^{(t+1)}(D'_j|T'_C(x_i))$ at time $t+1$, the decision rule $\gamma'_{x_i} : des([x_i]'_C) \to des(D'_j)$ is interesting if $Acc^{(t+1)}(D'_j|T'_C(x_i)) \geq \alpha$ and $Cov^{(t+1)}(D'_j|T'_C(x_i)) \geq \gamma$ at time $t+1$.

A. INCREMENTAL COMPUTATIONS OF THE NEW ACCURACY MATRIX AND COVERAGE MATRIX

Since the adding of multiple objects can be regarded as the composition of a single object, we only consider the case of a single adding object. In the following, the calculation process for the new accuracy and coverage of each object is introduced when adding a new object x into the incomplete decision system at time $t+1$. On the basis of the updated accuracy matrix and coverage matrix, the interesting knowledge can be easily induced. There are four cases with regard to the classification of x on C and D as follows. The flowchart of the incremental updating of accuracy matrix and coverage matrix is shown in Fig.1.

If a new object x satisfying Case 1 is added into the system, i.e., $T_C(x_{m+1}) = \{x\}$ and $D_{n+1} = \{x\}$, modify the last

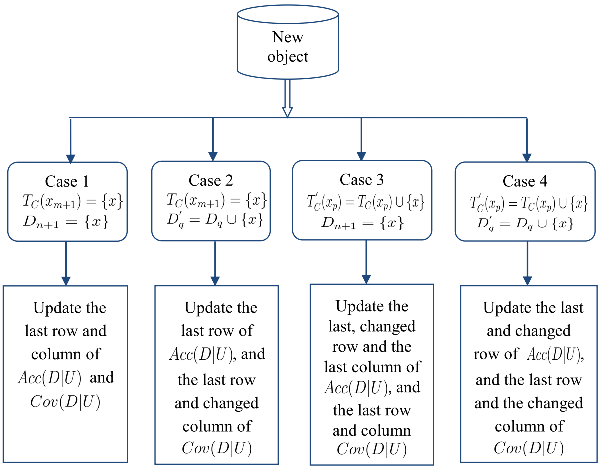

FIGURE 1. **Incremental update of the new accuracy matrix and coverage matrix.**

row of accuracy matrix, and the last column of coverage matrix needs to be updated; if a new object is in Case 2, i.e., $T_C(x_{m+1}) = \{x\}$ and $D'_q = D_q \cup \{x\}$, update the last row of accuracy matrix, and modify the changed column and last row of coverage matrix; if a new object satisfying Case 3 is added, i.e., $T'_C(x_p) = T_C(x_p) \cup \{x\}$ and $D_{n+1} = \{x\}$, the last row, changed row and last column of accuracy matrix, the last row and column of coverage matrix all need to be updated; when the new object is in Case 4, i.e., $T'_C(x_p) = T_C(x_p) \cup \{x\}$ and $D'_q = D_q \cup \{x\}$, the changed column and last row of accuracy matrix, and the changed column and last row of coverage matrix all need to be modified.

Case 1: Forming a new tolerance class and a new decision class.

In this case, $U'/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T_C(x_m), T_C(x_{m+1})\}$, where $T_C(x_{m+1}) = \{x\}$. In addition, $U'/IND(D) = \{D_1, D_2, \cdots, D_n\}$, where $D_{n+1} = \{x\}$.

At this time $t+1$, $Acc^{(t+1)}(D_{n+1}|T_C(x_{m+1})) = 1$ and $Acc^{(t+1)}(D_j|T_C(x_{m+1})) = 0$, $Cov^{(t+1)}(D_{n+1}|T_C(x_{m+1})) = 1$ and $Cov^{(t+1)}(D_j|T_C(x_{m+1})) = 0$ for $1 \le j \le n$.

$Acc^{(t+1)}(D'_j|T'_C(x_i)) = Acc^{(t)}(D_j|T_C(x_i))$ and $Cov^{(t+1)}(D'_j|T'_C(x_i)) = Cov^{(t)}(D_j|T_C(x_i))$ for $1 \le i \le m$, $1 \le j \le n$.

The new accuracy matrix is updated by, $Acc^{(t+1)}(D|U)$, as shown at the bottom of this page, and the new coverage matrix is updated by, $Cov^{(t+1)}(D|U)$, as shown at the bottom of this page.

Let Table 1 be the original incomplete decision system, a new object $x = \{$High, Low, Compact, High, VeryGood$\}$ is added into the system. The updating computations of the new accuracy matrix and coverage matrix are shown as follows.

As Table 1 shown, the classification induced by C is $U/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T_C(x_6)$, where $T_C(x_1) = \{x_1\}$,$T_C(x_2) = \{x_2, x_6\}$, $T_C(x_3) = \{x_3\}$, $T_C(x_4) = \{x_4, x_5\}$, $T_C(x_5) = \{x_4, x_5, x_6\}$, $T_C(x_6) = \{x_2, x_5, x_6\}$; the classification induced by D is $U/IND(D) = \{D_1, D_2, D_3\}$, where $D_1 = \{x_1, x_2, x_4, x_6\}$, $D_2 = \{x_3\}$, $D_3 = \{x_5\}$. By Definition 2, the accuracy matrix and the coverage matrix at time t are computed, respectively, as follows.

$$Acc^{(t)}(D_j|T_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}$$

$$Cov^{(t)}(D_j|T_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & 1 \\ \frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & 1 \end{pmatrix}$$

When the new object x is added into the system, it forms a new tolerance class $T_C(x_7) = \{x\}$ and a new decision class $D_4 = \{x\}$. As analyzed above, we only need to compute the last row and the last column of two matrices $Acc^{(t+1)}(D_4|T_C(x_7)) = 1$, $Acc^{(t+1)}(D_j|T_C(x_7)) = 0(1 \le j \le 3)$, $Cov^{(t+1)}(D_4|T_C(x_7)) = 1$, $Cov^{(t+1)}(D_j|T_C(x_7)) = 0(1 \le j \le 3)$. The computations of accuracy and coverage for other objects are unchanged. Therefore, the new accuracy matrix and the coverage matrix at time $t+1$ are shown, respectively,

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} Acc(D_1|T_C(x_1)) & Acc(D_2|T_C(x_1)) & \ldots & Acc(D_n|T_C(x_1)) & 0 \\ Acc(D_1|T_C(x_2)) & Acc(D_2|T_C(x_2)) & \ldots & Acc(D_n|T_C(x_2)) & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ Acc(D_1|T_C(x_m)) & Acc(D_2|T_C(x_m)) & \ldots & Acc(D_n|T_C(x_m)) & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}$$

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} Cov(D_1|T_C(x_1)) & Cov(D_2|T_C(x_1)) & \ldots & Cov(D_n|T_C(x_1)) & 0 \\ Cov(D_1|T_C(x_2)) & Cov(D_2|T_C(x_2)) & \ldots & Cov(D_n|T_C(x_2)) & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ Cov(D_1|T_C(x_m)) & Cov(D_2|T_C(x_m)) & \ldots & Cov(D_n|T_C(x_m)) & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}$$

as follows.

$$Acc^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 & \mathbf{0} \\ 1 & 0 & 0 & \mathbf{0} \\ 0 & 1 & 0 & \mathbf{0} \\ \frac{1}{2} & 0 & \frac{1}{2} & \mathbf{0} \\ \frac{2}{3} & 0 & \frac{1}{3} & \mathbf{0} \\ \frac{2}{3} & 0 & \frac{1}{3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

$$Cov^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & 0 & \mathbf{0} \\ \frac{1}{2} & 0 & 0 & \mathbf{0} \\ 0 & 1 & 0 & \mathbf{0} \\ \frac{1}{4} & 0 & 1 & \mathbf{0} \\ \frac{1}{2} & 0 & 1 & \mathbf{0} \\ \frac{1}{2} & 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$Acc^{(t+1)}(D'_1|T'_C(x_2)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_2)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_5)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_5)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_6)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_6)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D_4|T_C(x_7)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D_4|T_C(x_7)) = 1 \geq 0.4.$$

we can find out the interesting knowledge as follows:

$$des[x_2]_C \rightarrow des(D_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \wedge (X, Low) \rightarrow (Acceleration, Good);$$
$$des[x_3]_C \rightarrow des(D_2) : (P, *) \wedge (M, *) \wedge (S, Compact) \wedge (X, Low) \rightarrow (Acceleration, Poor);$$
$$des[x_5]_C \rightarrow des(D_1) : (P, *) \wedge (M, *) \wedge (S, Full) \wedge (X, High) \rightarrow (Acceleration, Good);$$
$$des[x_6]_C \rightarrow des(D_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \wedge (X, *) \rightarrow (Acceleration, Good);$$
$$des[x_7]_C \rightarrow des(D_4) : (P, High) \wedge (M, Low) \wedge (S, Compact) \wedge (X, High) \rightarrow (Acceleration, VeryGood).$$

Case 2: Only forming a new tolerance class.

In this case, $U'/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T_C(x_m), T_C(x_{m+1})\}$, where $T_C(x_{m+1}) = \{x\}$. In addition, $U'/IND(D) = \{D_1, D_2, \ldots, D'_q, \ldots, D_n\}$, where $D'_q = D_q \cup \{x\}$ $(1 \leq q \leq n)$.

At this time $t+1$, $Acc^{(t+1)}(D'_q|T_C(x_{m+1})) = 1$, $Cov^{(t+1)}(D'_q|\, T_C(x_{m+1})) = \frac{1}{|D_q|+1}$;

$Acc^{(t+1)}(D'_k|T_C(x_{m+1})) = 0$, $Cov^{(t+1)}(D'_k|T_C(x_{m+1})) = 0$ since $[x]_C \cap D_k = \varnothing (1 \leq k \neq q \leq n)$;

$Acc^{(t+1)}(D'_j|T'_C(x_i)) = Acc^{(t)}(D_j|T_C(x_i))$ and $Cov^{(t+1)}(D'_j|T'_C(x_i)) = Cov^{(t)}(D_j\,|T_C(x_i))$ $(1 \leq i \leq m, 1 \leq j \leq n, j \neq q)$;

$Acc^{(t+1)}(D'_j|T'_C(x_i)) = Acc^{(t)}(D_j|T_C(x_i))$ and $Cov^{(t+1)}(D'_j|T'_C(x_i)) = \frac{|T_C(x_i) \cap D_q|}{|D_q|+1}$ $(1 \leq i \leq m, j = q)$.

The new accuracy matrix is updated by, $Acc^{(t+1)}(D|U)$, as shown at the bottom of this page, and the new coverage matrix is updated by, $Cov^{(t+1)}(D|U)$, as shown at the bottom of this page.

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} Acc(D_1|T_C(x_1)) & \ldots & Acc(D_q|T_C(x_1))\ldots & Acc(D_n|T_C(x_1)) \\ Acc(D_1|T_C(x_2)) & \ldots & Acc(D_q|T_C(x_2))\ldots & Acc(D_n|T_C(x_2)) \\ \ldots & \ldots & \ldots & \ldots \\ Acc(D_1|T_C(x_m)) & \ldots & Acc(D_q|T_C(x_m))\ldots & Acc(D_n|T_C(x_m)) \\ 0 & \ldots & 1 & 0 \end{pmatrix}$$

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} Cov(D_1|T_C(x_1)) & \ldots & \frac{|T_C(x_1)) \cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_1)) \\ Cov(D_1|T_C(x_2)) & \ldots & \frac{|T_C(x_2)) \cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_2)) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ Cov(D_1|T_C(x_m)) & \ldots & \frac{|T_C(x_m) \cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_m)) \\ 0 & \ldots & \frac{1}{|D_q|+1} & \ldots & 0 \end{pmatrix}$$

Let Table 1 be the original incomplete decision system, a new object $x = \{\text{High, Low, Compact, High, Excellent}\}$ is added into the system. The updating computations of the new accuracy and coverage for each object are shown as follows.

When the new object x is added into the system, it only forms a new tolerance class $T_C(x_7) = \{x\}$, and the changed decision class $D'_3 = \{x, x_5\}$. As analyzed above, we need to compute the accuracy of the new object $Acc^{(t+1)}(D'_3|T_C(x_7)) = 1$, $Acc^{(t+1)}(D'_k|T_C(x_7)) = 0$ $(k = 1, 2)$. The computations of accuracy for other objects are unchanged. In addition, the coverage of the new object $Cov^{(t+1)}(D'_3|T_C(x_7)) = \frac{1}{|D_3|+1} = \frac{1}{2}$, $Cov^{(t+1)}(D_k|T_C(x_7)) = 0(k = 1, 2)$, and $Cov^{(t+1)}(D'_3|T_C(x_1)) = \frac{|T_C(x_1)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_2)) = \frac{|T_C(x_2)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_3)) = \frac{|T_C(x_3)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_4)) = \frac{|T_C(x_4)\cap D_3|}{|D_3|+1} = \frac{1}{2}$, $Cov^{(t+1)}(D'_3|T_C(x_5)) = \frac{|T_C(x_5)\cap D_3|}{|D_3|+1} = \frac{1}{2}$, $Cov^{(t+1)}(D'_3|T_C(x_6)) = \frac{|T_C(x_6)\cap D_3|}{|D_3|+1} = \frac{1}{2}$. The computations of coverage for other objects are unchanged.

The new accuracy matrix and the coverage matrix at time $t + 1$ are shown, respectively, as follows.

$$Acc^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

$$Cov^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & \mathbf{0} \\ \frac{1}{2} & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \frac{1}{4} & 0 & \mathbf{\frac{1}{2}} \\ \frac{1}{2} & 0 & \mathbf{\frac{1}{2}} \\ \frac{1}{2} & 0 & \mathbf{\frac{1}{2}} \\ \mathbf{0} & \mathbf{0} & \mathbf{\frac{1}{2}} \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$Acc^{(t+1)}(D'_1|T'_C(x_2)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_2)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_5)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_5)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_6)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_6)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_3|T'_C(x_7)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_3|T'_C(x_7)) = \frac{1}{2} \geq 0.4.$$

we can find out the interesting knowledge as follows:

$$des[x_2]_C \rightarrow des(D_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \wedge (X, Low) \rightarrow (Acceleration, Good);$$
$$des[x_3]_C \rightarrow des(D_2) : (P, *) \wedge (M, *) \wedge (S, Compact) \wedge (X, Low) \rightarrow (Acceleration, Poor);$$
$$des[x_5]_C \rightarrow des(D_1) : (P, *) \wedge (M, *) \wedge (S, Full) \wedge (X, High) \rightarrow (Acceleration, Good);$$
$$des[x_6]_C \rightarrow des(D_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \wedge (X, *) \rightarrow (Acceleration, Good);$$
$$des[x]_C \rightarrow des(D_3) : (P, High) \wedge (M, Low) \wedge (S, Compact) \wedge (X, High) \rightarrow (Acceleration, Excellent).$$

Case 3: Only forming a new decision class.

In this case, $U'/TR(B) = \{T_C(x_1), T_C(x_2), \ldots, T'_C(x_p), \ldots, T_C(x_m), T_C(x_{m+1})\}$, where $T'_C(x_p) = T_C(x_p) \cup \{x\}$ and $T_C(x_{m+1}) = \{x\} \cup \{x_p\}(1 \leq p \leq m)$. In addition, $U'/IND(D) = \{D_1, D_2, \cdots, D_n, D_{n+1} = [x]_D\}$, where $D_{n+1} = \{x\}$.

At this time $t + 1$, $Acc^{(t+1)}(D_{n+1}|T_C(x_{m+1})) = \frac{1}{|T_C(x_{m+1})|}$, $Cov^{(t+1)}(D_{n+1}|\ T_C(x_{m+1})) = 1$;

$$Acc^{(t+1)}(D'_j|T_C(x_{m+1})) = \frac{|T_C(x_{m+1}) \cap D_j|}{|T_C(x_{m+1})|},$$
$$Cov^{(t+1)}(D'_j|T_C(x_{m+1})) = 0(1 \leq j \leq n);$$
$$Acc^{(t+1)}(D_{n+1}|T'_C(x_p)) = \frac{1}{|T_C(x_p)| + 1},$$
$$Cov^{(t+1)}(D_{n+1}|T'_C(x_p)) = 1;$$
$$Acc^{(t+1)}(D'_j|T'_C(x_p)) = \frac{|T_C(x_p) \cap D_j|}{|T_C(x_p)| + 1},$$
$$Cov^{(t+1)}(D'_j|T'_C(x_p)) = Cov^{(t+1)}(D_j|T_C(x_p))(1 \leq j \leq n);$$
$$Acc^{(t+1)}(D'_j|T'_C(x_i)) = Acc^{(t)}(D_j|T_C(x_i)),$$
$$Cov^{(t+1)}(D'_j|T'_C(x_i)) = Cov^{(t)}(D_j|T_C(x_i)), \quad (1 \leq i \leq m, i \neq p, 1 \leq j \leq n);$$
$$Acc^{(t+1)}(D_{n+1}|T'_C(x_i)) = Cov^{(t+1)}(D_{n+1}|T'_C(x_i)) = 0(1 \leq i \leq m, i \neq p).$$

The new accuracy matrix is updated by

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} Acc(D_1|T_C(x_1)) & \ldots & Acc(D_n|T_C(x_1)) & 0 \\ Acc(D_1|T_C(x_2)) & \ldots & Acc(D_n|T_C(x_2)) & 0 \\ \ldots & \ldots & \ldots & \ldots \\ \frac{|T_C(x_p)\cap D_1|}{|T_C(x_p)|+1} & \ldots & \frac{|T_C(x_p)\cap D_n|}{|T_C(x_p)|+1} & \frac{1}{|T_C(x_p)|+1} \\ \ldots & \ldots & \ldots & \ldots \\ Acc(D_1|T_C(x_m)) & \ldots & Acc(D_n|T_C(x_m)) & 0 \\ \frac{|T_C(x_{m+1})\cap D_1|}{|T_C(x_{m+1})|} & \ldots & \frac{|T_C(x_{m+1})\cap D_n|}{|T_C(x_{m+1})} & \frac{1}{|T_C(x_{m+1})|} \end{pmatrix}$$

and the new coverage matrix is updated by

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} Cov(D_1|T_C(x_1)) & \ldots & Cov(D_n|T_C(x_1)) & 0 \\ Cov(D_1|T_C(x_2)) & \ldots & Cov(D_n|T_C(x_2)) & 0 \\ \ldots & \ldots & \ldots & \ldots \\ Cov(D_1|T_C(x_p)) & \ldots & Cov(D_n|T_C(x_p)) & 1 \\ \ldots & \ldots & \ldots & \ldots \\ Cov(D_1|T_C(x_m)) & \ldots & Cov(D_n|T_C(x_m)) & 0 \\ 0 & \ldots & 0 & 1 \end{pmatrix}$$

Let Table 1 be the original incomplete decision system, a new object x = {High, High, Full, High, VeryGood} is added into the system. The updating computations of the new accuracy and coverage for each object are shown as follows.

When the new object x is added into the system, it only forms a new tolerance class $D_4 = [x]_D = \{x\}$, the changed tolerance class $T'_C(x_4) = \{x, x_4, x_5\}$, $T'_C(x_5) = \{x, x_4, x_5, x_6\}$ and $T_C(x_7) = \{x, x_4, x_5\}$. As analyzed above, we need to compute the accuracy of the new object

$$Acc^{(t+1)}(D_4|T_C(x_7)) = \frac{1}{|T_C(x_7)|} = \frac{1}{3},$$
$$Acc^{(t+1)}(D_1|T_C(x_7)) = \frac{1}{|T_C(x_7)|} = \frac{1}{3},$$
$$Acc^{(t+1)}(D_2|T_C(x_7)) = 0, \quad Acc^{(t+1)}(D_3|T_C(x_7)) = \frac{1}{3},$$
$$Acc^{(t+1)}(D_1|T'_C(x_4)) = \frac{1}{|T_C(x_4)|+1} = \frac{1}{3},$$
$$Acc^{(t+1)}(D_2|T'_C(x_4)) = 0,$$
$$Acc^{(t+1)}(D_3|T'_C(x_4)) = \frac{1}{|T_C(x_4)|+1} = \frac{1}{3},$$
$$Acc^{(t+1)}(D_1|T'_C(x_5)) = \frac{2}{|T_C(x_5)|+1} = \frac{1}{2},$$
$$Acc^{(t+1)}(D_2|T'_C(x_5)) = 0,$$
$$Acc^{(t+1)}(D_3|T'_C(x_5)) = \frac{1}{|T_C(x_5)|+1} = \frac{1}{4},$$
$$Acc^{(t+1)}(D_4|T_C(x_i)) = 0 (i = 1, 2, 3, 6).$$

The computations of accuracy for other objects are unchanged. In addition, the coverage of the new object $Cov^{(t+1)}(D_4|T_C(x_7)) = 1$, $Cov^{(t+1)}(D_j|T_C(x_7)) = 0 (1 \leq j \leq 3)$, $Cov^{t+1}(D_4|T'_C(x_4)) = 1$, and $Cov^{(t+1)}(D'_4|T'_C(x_5)) = 1$, $Cov^{(t+1)}(D'_4|T'_C(x_i)) = 0 (i = 1, 2, 3, 6)$. The computations of accuracy and coverage for other objects are unchanged.

The new accuracy matrix and the coverage matrix at time $t+1$ are shown, respectively, as follows.

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} 1 & 0 & 0 & \mathbf{0} \\ 1 & 0 & 0 & \mathbf{0} \\ 0 & 1 & 0 & \mathbf{0} \\ \frac{1}{3} & 0 & \frac{1}{3} & \mathbf{\frac{1}{3}} \\ \frac{1}{2} & 0 & \frac{1}{4} & \mathbf{\frac{1}{4}} \\ \frac{2}{3} & 0 & \frac{1}{3} & \mathbf{0} \\ \mathbf{\frac{1}{3}} & \mathbf{0} & \mathbf{\frac{1}{3}} & \mathbf{\frac{1}{3}} \end{pmatrix}$$

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} \frac{1}{4} & 0 & 0 & \mathbf{0} \\ \frac{1}{2} & 0 & 0 & \mathbf{0} \\ 0 & 1 & 0 & \mathbf{0} \\ \frac{1}{4} & 0 & 1 & \mathbf{1} \\ \frac{1}{2} & 0 & 1 & \mathbf{1} \\ \frac{1}{2} & 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$Acc^{(t+1)}(D'_1|T'_C(x_2)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_2)) = \frac{1}{2} \geq 0.4;$$
$$Acc^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.6,$$
$$Cov^{(t+1)}(D'_2|T'_C(x_3)) = 1 \geq 0.4;$$
$$Acc^{(t+1)}(D'_1|T'_C(x_6)) = \frac{2}{3} \geq 0.6,$$
$$Cov^{(t+1)}(D'_1|T'_C(x_6)) = \frac{1}{2} \geq 0.4;$$

we can find out the interesting knowledge as follows:

$$des[x_2]_C \rightarrow des(D_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \wedge (X, Low) \rightarrow (Acceleration, Good);$$
$$des[x_3]_C \rightarrow des(D_2) : (P, *) \wedge (M, *) \wedge (S, Compact) \wedge (X, Low) \rightarrow (Acceleration, Poor);$$
$$des[x_6]_C \rightarrow des(D_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \wedge (X, *) \rightarrow (Acceleration, Good).$$

Case 4: Neither generating a new tolerance class nor a new decision class.

In this case, $U'/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T'_C(x_p), \ldots, T_C(x_m), T_C(x_{m+1})\}$, where $T'_C(x_p) = T_C(x_p) \cup \{x\}$ and $T_C(x_{m+1}) = \{x\} \cup \{x_p\} (1 \leq p \leq m)$, In addition, $U'/IND(D) = \{D_1, D_2, \ldots, D'_q, \ldots, D_n\}$, where $D'_q = D_q \cup \{x\}$ $(1 \leq q \leq n)$.

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} Acc(D_1|T_C(x_1)) & \ldots & Acc(D_q|T_C(x_1)) & \ldots & Acc(D_n|T_C(x_1)) \\ Acc(D_1|T_C(x_2)) & \ldots & Acc(D_q|T_C(x_2)) & \ldots & Acc(D_n|T_C(x_2)) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \frac{|T_C(x_p)\cap D_1|}{|T_C(x_p)|+1} & \ldots & \frac{|T_C(x_p)\cap D_q|}{|T_C(x_p)|+1} & \ldots & \frac{|T_C(x_p)\cap D_n|}{|T_C(x_p)|+1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ Acc(D_1|T_C(x_m)) & \ldots & Acc(D_q|T_C(x_m)) & \ldots & Acc(D_n|T_C(x_m)) \\ \frac{|T_C(x_{m+1})\cap D_1|}{|T_C(x_{m+1})|} & \ldots & \frac{|T_C(x_{m+1})\cap D_q|}{|C(x_{m+1})|} & \ldots & \frac{|T_C(x_{m+1})\cap D_n|}{|T_C(x_{m+1})|} \end{pmatrix}$$

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} Cov(D_1|T_C(x_1)) & \ldots & \frac{|T_C(x_1)\cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_1)) \\ Cov(D_1|T_C(x_2)) & \ldots & \frac{|T_C(x_2)\cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_2)) \\ \ldots & \ldots & & \ldots & \\ Cov(D_1|T_C(x_p)) & \ldots & \frac{|T_C(x_p)\cap D_q|}{|D_q|+1} & \ldots & Cov(D_n|T_C(x_p)) \\ \ldots & \ldots & \ldots & \ldots & \\ Cov(D_1|T_C(x_m)) & \ldots & Cov(D_q|T_C(x_m)) & \ldots & Cov(D_n|T_C(x_m)) \\ \frac{|T_C(x_{m+1})\cap D_1|}{|D_1|} & \ldots & \frac{|T_C(x_{m+1})\cap D_q|}{|D_q|+1} & \ldots & \frac{|T_C(x_{m+1})\cap D_n|}{|D_n|} \end{pmatrix}$$

At this time $t+1$, $Acc^{(t+1)}(D'_q|T'_C(x_p)) = \frac{|T_C(x_p)\cap D_q|}{|T_C(x_p)|+1}$, $Cov^{(t+1)}(D'_q|T'_C(x_p)) = \frac{|T_C(x_p)\cap D_q|}{|D_q|+1}$;

$$Acc^{(t+1)}(D'_j|T'_C(x_p)) = \frac{|T_C(x_p)\cap D_j|}{|T_C(x_p)|+1},$$

$$Cov^{(t+1)}(D'_j|T'_C(x_p)) = Cov^{(t)}(D_j|T_C(x_p)), \quad (1\le j\le n).$$

$$Acc^{(t+1)}(D'_j|T'_C(x_i)) = Acc^{(t)}(D_j|T_C(x_i)) \quad \text{and}$$

$$Cov^{(t+1)}(D'_j|T'_C(x_i)) = Cov^{(t)}(D_j|T_C(x_i)), \quad 1\le i\le m, i\ne p, \quad 1\le j\le n.$$

$$Acc^{(t+1)}(D'_q|T'_C(x_i)) = Acc^{(t)}(D_q|T_C(x_i))$$

and $$Cov^{(t+1)}(D'_q|T'_C(x_i)) = \frac{|T_C(x_i)\cap D_q|}{|D_q|+1}, \quad (1\le i\le m), \quad i\ne p.$$

$$Acc^{(t+1)}(D'_j|T_C(x_{m+1})) = \frac{|T_C(x_{m+1})\cap D_j|}{|T_C(x_{m+1})|},$$

$$Cov^{(t+1)}(D'_j|T_C(x_{m+1})) = \frac{|T_C(x_{m+1})\cap D_j|}{|D_j|}, \quad (1\le j\le n).$$

The new accuracy matrix is updated by, $Acc^{(t+1)}(D|U)$, as shown at the top of this page, and the new coverage matrix is updated by, $Cov^{(t+1)}(D|U)$, as shown at the top of this page.

Let Table 1 be the original incomplete decision system, a new object x = {High, High, Full, High, Excellent} is added into the system. The updating computations of the new accuracy and coverage for each object are shown as follows.

When the new object x is added into the system, the changed tolerance class $T'_C(x_4) = \{x, x_4, x_5\}$, $T'_C(x_5) = \{x, x_4, x_5, x_6\}$ and $T_C(x_7) = \{x, x_4, x_5\}$, and the changed decision class $D'_3 = \{x, x_5\}$. As analyzed above, we need to compute the accuracy of the new object $Acc^{(t+1)}(D'_3|T_C(x_7)) = \frac{|\{x,x_5\}|}{|\{x,x_4,x_5\}|} = \frac{2}{3}$, $Acc^{(t+1)}(D'_1|T_C(x_7)) = \frac{|\{x_4\}|}{|\{x,x_4,x_5\}|} = \frac{1}{3}$, $Acc^{(t+1)}(D'_2|T_C(x_7)) = 0$, $Acc^{(t+1)}(D'_1|T'_C(x_4)) = \frac{1}{3}$, $Acc^{(t+1)}(D'_2|T'_C(x_4)) = 0$, $Acc^{(t+1)}(D'_3|T'_C(x_4)) = \frac{2}{3}$, $Acc^{(t+1)}(D'_1|T'_C(x_5)) = \frac{|\{x_4,x_6\}|}{3+1} = \frac{1}{2}$, $Acc^{(t+1)}(D'_2|T'_C(x_5)) = 0$, $Acc^{(t+1)}(D'_3|T'_C(x_5)) = \frac{|\{x,x_5\}|}{3+1} = \frac{1}{2}$. The computations of accuracy for other objects are unchanged. In addition, the coverage of the new object $Cov^{(t+1)}(D'_3|T_C(x_7)) = \frac{2}{|D_3|+1} = 1$, $Cov^{(t+1)}(D'_3|T_C(x_1)) = \frac{|T_C(x_1)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_2)) = \frac{|T_C(x_2)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_3)) = \frac{|T_C(x_3)\cap D_3|}{|D_3|+1} = 0$, $Cov^{(t+1)}(D'_3|T_C(x_4)) = \frac{|T_C(x_4)\cap D_3|}{|D_3|+1} = 1$, $Cov^{(t+1)}(D'_3|T_C(x_5)) = \frac{|T_C(x_5)\cap D_3|}{|D_3|+1} = 1$, $Cov^{(t+1)}(D'_3|T_C(x_6)) = \frac{|T_C(x_6)\cap D_3|}{|D_3|+1} = \frac{1}{2}$ and $Cov^{(t+1)}(D'_1|T_C(x_7)) = \frac{1}{4}$, $Cov^{(t+1)}(D'_2|T_C(x_7)) = 0$. The computations of accuracy and coverage for other objects are unchanged. The new accuracy matrix and the coverage matrix at time $t+1$ are shown, respectively, as follows.

$$Acc^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{\frac{1}{3}} & \mathbf{0} & \mathbf{\frac{2}{3}} \\ \mathbf{\frac{1}{2}} & \mathbf{0} & \mathbf{\frac{1}{2}} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \mathbf{\frac{1}{3}} & \mathbf{0} & \mathbf{\frac{2}{3}} \end{pmatrix}$$

$$Cov^{(t+1)}(D_j|T_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & \mathbf{0} \\ \frac{1}{2} & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \frac{1}{4} & 0 & \mathbf{1} \\ \frac{1}{2} & 0 & \mathbf{1} \\ \frac{1}{2} & 0 & \mathbf{\frac{1}{2}} \\ \mathbf{\frac{1}{4}} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$\begin{aligned} &Acc^{(t+1)}(D_1'|T_C'(x_2)) = 1 \geq 0.6, \\ &Cov^{(t+1)}(D_1'|T_C'(x_2)) = \frac{1}{2} \geq 0.4; \\ &Acc^{(t+1)}(D_2'|T_C'(x_3)) = 1 \geq 0.6, \\ &Cov^{(t+1)}(D_2'|T_C'(x_3)) = 1 \geq 0.4; \\ &Acc^{(t+1)}(D_3'|T_C'(x_4)) = \frac{2}{3} \geq 0.6, \\ &Cov^{(t+1)}(D_3'|T_C'(x_4)) = 1 \geq 0.4; \\ &Acc^{(t+1)}(D_1'|T_C'(x_6)) = \frac{2}{3} \geq 0.6, \\ &Cov^{(t+1)}(D_1'|T_C'(x_6)) = \frac{1}{2} \geq 0.4; \\ &Acc^{(t+1)}(D_3'|T_C(x_7)) = \frac{2}{3} \geq 0.6, \\ &Cov^{(t+1)}(D_3'|T_C(x_7)) = 1 \geq 0.4; \end{aligned}$$

we can find out the interesting knowledge as follows:

$$\begin{aligned} des[x_2]_C &\rightarrow des(D_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \\ &\quad \wedge(X, Low) \rightarrow (Acceleration, Good); \\ des[x_3]_C &\rightarrow des(D_2) : (P, *) \wedge (M, *) \wedge (S, Compact) \\ &\quad \wedge(X, Low) \rightarrow (Acceleration, Poor); \\ des[x_4]_C &\rightarrow des(D_3') : (P, High) \wedge (M, *) \wedge (S, Full) \\ &\quad \wedge(X, High) \rightarrow (Acceleration, Excellent); \\ des[x_6]_C &\rightarrow des(D_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \\ &\quad \wedge(X, *) \rightarrow (Acceleration, Good); \\ des[x]_C &\rightarrow des(D_3) : (P, High) \wedge (M, High) \wedge (S, Full) \\ &\quad \wedge(X, High) \rightarrow (Acceleration, Excellent). \end{aligned}$$

Based on the aforementioned results, the new accuracy matrix and coverage matrix can be obtained by computing the changed tolerance class. If adding a new object, the key problem of computing the new accuracy matrix and coverage matrix is to update the tolerance class according to four kinds of cases. Cases (1)-(4) presents four updating formulas of the accuracy matrix and coverage matrix, which plays in important role in the knowledge acquisition process.

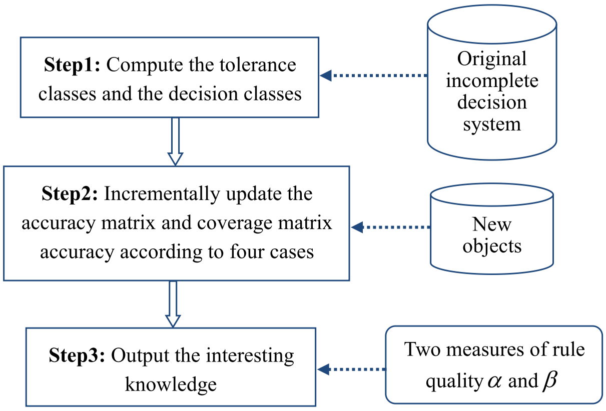

FIGURE 2. The process of incremental knowledge acquisition algorithm for data with missing values at the arrival of new objects.

B. INCREMENTAL KNOWLEDGE ACQUISITION ALGORITHM FOR DATA WITH MISSING VALUES AT THE ARRIVAL OF NEW OBJECTS

In the following, based on the incremental computations of the new accuracy matrix and coverage matrix for the incremental objects in incomplete data, we will develop an incremental knowledge acquisition algorithm to acquire the new interesting knowledge. The flowchart of the proposed knowledge acquisition algorithm is shown in Fig.2.

According to Fig.2, the detailed description of the knowledge acquisition algorithm is given as follows. The proposed algorithm IKAC is mainly made up of three parts: 1) compute the tolerance classes and decision classes of the original incomplete decision system; 2) incrementally update the new accuracy matrix and new coverage matrix according to the discussed four cases; 3) induce the interesting knowledge according to two measures. The intelligence of the proposed knowledge acquisition algorithm mainly stems from the updating computations of the new accuracy matrix and new coverage matrix. The updating strategy of the accuracy matrix and coverage matrix directly affect the computational efficiency of the knowledge acquisition.

As Algorithm IKAC shown, it mainly includes the following process: Steps 2-3 are to compute the tolerance classes and decision classes of the original incomplete decision system, respectively; Steps 5-16 are to determine the new object is in which case, are to update the new accuracy matrix and new coverage matrix incrementally according to the discussed four cases in the above subsection 5.1. Steps 17-19 are to induce the interesting knowledge at time $t+1$. according to two parameters α and β. When a new object adds into the incomplete decision system, it can be known that we only need to compute the changed tolerance classes for updating the local computations of the new accuracy matrix and coverage matrix which is more efficient than that of computing two matrices from the scratch. Therefore, this incremental updating knowledge acquisition algorithm is an effective way to maintain knowledge dynamically, to avoid unnecessary computations by utilizing the previous results.

Algorithm 2 Incremental Knowledge Acquisition Algorithm for Data With Missing Values at the Arrival of New Objects (Algorithm IKAC)

Input: An incomplete decision system $IDS =< U, C \cup D, V, f >$, where $U = \{x_1, x_2, \ldots, x_m\}$, the original accuracy matrix $Acc^{(t)}(D|U)$, the original coverage matrix $Cov^{(t)}(D|U)$ at time t, a new object x at time $t+1$, two thresholds α and β;
Output: Interesting knowledge at time t and $t+1$, respectively.
Begin
1) % Compute the new accuracy matrix and new coverage matrix and induce the interesting knowledge at time $t+1$
2) Compute the tolerance classes $U/TR(C) = \{T_C(x_1), T_C(x_2), \ldots, T_C(x_m)\}$ by Theorem 1;
3) Compute decision classes $U/IND(D) = \{D_1, D_2, \ldots, D_n\}$;
4) Let $x_{m+1} = x$ and $U' = U \cup \{x\} = \{x_1, x_2, \ldots, x_m, x_{m+1}\}$;
5) for $i = 1$ to m do
6) for $j = 1$ to n do
7) if $x \in T_C(x_i) == false$
8) if $x \in D_j == false$ then
9) obtain a new tolerance class $T_C(x_{m+1}) = \{x\}$ and a new decision class $D_{n+1} = [x]_D = \{x\}$, do Case 1, update $Acc^{(t+1)}(D|U)$ and $Cov^{(t+1)}(D|U)$ incrementally;
10) else
11) obtain a new tolerance class $T_C(x_{m+1}) = \{x\}$, and the changed decision class $D_j = D_j \cup [x]$, do Case 2, update $Acc^{(t+1)}(D|U)$ and $Cov^{(t+1)}(D|U)$ incrementally;
12) else // $x \in T_C(x_i) == true$
13) if $x \in D_j == false$ then
14) obtain the changed tolerance class $T_C(x_i) = T_C(x_i) \cup \{x\}$, $T_C(x_{m+1}) = \{x\} \cup \{x_i\}$, and a new decision class $D_{n+1} = [x]_D = \{x\}$, do Case 3, update $Acc^{(t+1)}(D|U)$ and $Cov^{(t+1)}(D|U)$ incrementally;
15) else
16) obtain the changed tolerance class $T_C(x_i) = T_C(x_i) \cup \{x\}$, $T_C(x_{m+1}) = \{x\} \cup \{x_i\}$, and the changed decision class $D_j = D_j \cup [x]$, do Case 4, update $Acc^{(t+1)}(D|U)$ and $Cov^{(t+1)}(D|U)$ incrementally;
17) for $i = 1$ to $m+1$ do
18) for $j = 1$ to $n+1$ do
19) if $Acc^{(t+1)}(D_j|T_C(x_i)) \geq \alpha$ and $Cov^{(t+1)}(D_j|T_C(x_i)) \geq \beta$, output the interesting knowledge $des[x_i]_C \to des(D_j)$ at time $t+1$;
End

We illustrate the proposed algorithm IKAC using the incomplete decision system shown in Table 1, Table 1 is taken as the original incomplete decision system. Now, a new object x is added into the system, where $x = \{\text{High, High, Full, High, Excellent}\}$ is added into the system. Suppose $\alpha = 0.6$, $\beta = 0.4$, the new knowledge acquisition process is shown as follows. The original accuracy matrix and coverage matrix are shown as follows:

$$Acc^{(t)}(D_j|T_C(x_i)) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}$$

$$Cov^{(t)}(D_j|T_C(x_i)) = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

As that in Example 1 for Table 1, the tolerance classes and decision classes of the original incomplete decision system are shown as follows: $T_C(x_1) = \{x_1\}$, $T_C(x_2) = \{x_2, x_6\}$, $T_C(x_3) = \{x_3\}$, $T_C(x_4) = \{x_4, x_5\}$, $T_C(x_5) = \{x_4, x_5, x_6\}$, $T_C(x_6) = \{x_2, x_5, x_6\}$; $D_1 = \{x_1, x_2, x_4, x_6\}$, $D_2 = \{x_3\}$, $D_3 = \{x_5\}$. When x_7 is added into the system, $U' = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, since $x \in T_C(x_i) == true$ ($i = 4, 5$) and $x \in D_j == true$ ($j = 3$), one can obtain that the changed tolerance class $T_C(x_4) = \{x, x_4, x_5\}$, $T_C(x_5) = \{x, x_4, x_5, x_6\}$ and $T_C(x_7) = \{x, x_4, x_5\}$, and the changed decision class $D_3 = \{x, x_5\}$. Do Case 4, we only need to update the last and changed row of the original accuracy matrix, and the last row and changed column of the original coverage matrix. Then, the new accuracy matrix and new coverage matrix are updated as follows:

$$Acc^{(t+1)}(D|U) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}$$

$$Cov^{(t+1)}(D|U) = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & 1 \\ \frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & 1 \end{pmatrix}$$

By the new accuracy matrix and new coverage matrix, one can obtain:

$$\begin{aligned} Acc^{(t+1)}(D_1|T_C(x_2)) &= 1 \geq 0.6, \\ Cov^{(t+1)}(D_1|T_C(x_2)) &= \frac{1}{2} \geq 0.4; \\ Acc^{(t+1)}(D_3|T_C(x_4)) &= \frac{2}{3} \geq 0.6, \\ Cov^{(t+1)}(D_3|T_C(x_4)) &= \frac{1}{2} \geq 0.4; \\ Acc^{(t+1)}(D_1|T_C(x_6)) &= \frac{2}{3} \geq 0.6, \\ Cov^{(t+1)}(D_1|T_C(x_6)) &= \frac{1}{2} \geq 0.4; \\ Acc^{(t+1)}(D_3|T_C(x_7)) &= \frac{2}{3} \geq 0.6, \\ Cov^{(t+1)}(D_3|T_C(x_7)) &= 1 \geq 0.4; \end{aligned}$$

we can find out the interesting knowledge as follows:

$$\begin{aligned} des[x_2]_C &\rightarrow des(D_1) : (P, Low) \wedge (M, *) \wedge (S, Full) \\ &\quad \wedge (X, Low) \rightarrow (Acceleration, Good); \\ des[x_4]_C &\rightarrow des(D_3) : (P, High) \wedge (M, *) \wedge (S, Full) \\ &\quad \wedge (X, High) \rightarrow (Acceleration, Excellent); \\ des[x_6]_C &\rightarrow des(D_1) : (P, Low) \wedge (M, High) \wedge (S, Full) \\ &\quad \wedge (X, *) \rightarrow (Acceleration, Good); \\ des[x]_C &\rightarrow des(D_3) : (P, High) \wedge (M, High) \wedge (S, Full) \\ &\quad \wedge (X, High) \rightarrow (Acceleration, Excellent). \end{aligned}$$

VI. EXPERIMENTAL ANALYSIS

To test the performance of the proposed knowledge acquisition algorithm, we conduct some experiments on a PC with Windows 7, Intel (R) Core(TM) Duo CPU 2.93 GHz and 4GB memory. Algorithms are coded in C++ and the software being used is Microsoft Visual Studio 2017.

It is obvious that different thresholds of accuracy and coverage lead to different knowledge acquired. Generally speaking, if thresholds vary, the acquired interesting knowledge also change. Such thresholds can be set by users or domain experts [26], [27], [30]. As the conclusions obtained in [26], the accuracy value of interesting knowledge is no less than 0.5, α is in [0.5, 1]. In the following, the parameters used for defining the interesting knowledge are fixed as follows: $\alpha = 0.6$, $\beta = 0.4$.

TABLE 2. A description of eight data sets.

Data sets	Objects	Features	Classes
Zoo	101	17	7
Soybean	307	35	19
Dernatology	366	34	6
Voting records	435	16	2
Credit	690	15	2
Vehicle	946	18	4
Car	1728	6	4
Chess kr-vs-kp	3196	36	2

We perform the experiments on eight real UCI data sets, which are downloaded from UCI Repository of machine learning databases in [35]. The characteristics of eight data sets are described in Table 2. For the complete data sets, we randomly change 5% of the known features values from each original data set into missing values to create incomplete data sets. For the numerical features, we use the data tool Rosetta (http://www.lcb.uu.se/tools/rosetta/index.php) to discretize them. For each data set shown in Table 2, 60% of the objects are taken as the original incomplete data sets, and the remaining 40% of the objects are taken as adding objects.

In what follows, to show the efficiency of the proposed algorithm, we choose the non-incremental knowledge acquisition algorithm NKAC as the reference algorithm. For the non-incremental algorithm NKAC, we view the dynamic incomplete data with the variation of object set as absolutely new data without using the incremental strategy. The accuracy matrix and the coverage matrix is computed from the scratch. The main difference between the incremental algorithm and non-incremental algorithm is the computations of accuracy matrix and coverage matrix.

TABLE 3. Comparison of the computational time between IKAC and NKAC.

Data sets	NKAC	IKAC
Zoo	1.1350	2.8617
Soybean	4.2792	14.0158
Dernatology	5.5061	23.1699
Voting records	7.4834	20.9725
Credit	124.9209	26.1842
Vehicle	728.3965	115.4713
Car	514.0812	73.2940
Chess kr-vs-kp	2710.6855	264.8136

When new objects are added into the incomplete data set, the knowledge acquisition algorithms NKAC and IKAC can induce the interesting knowledge from the dynamic data set. Table 3 records the computational time of two algorithms NKAC and IKAC for inducing the interesting knowledge. The computational time is expressed as seconds.

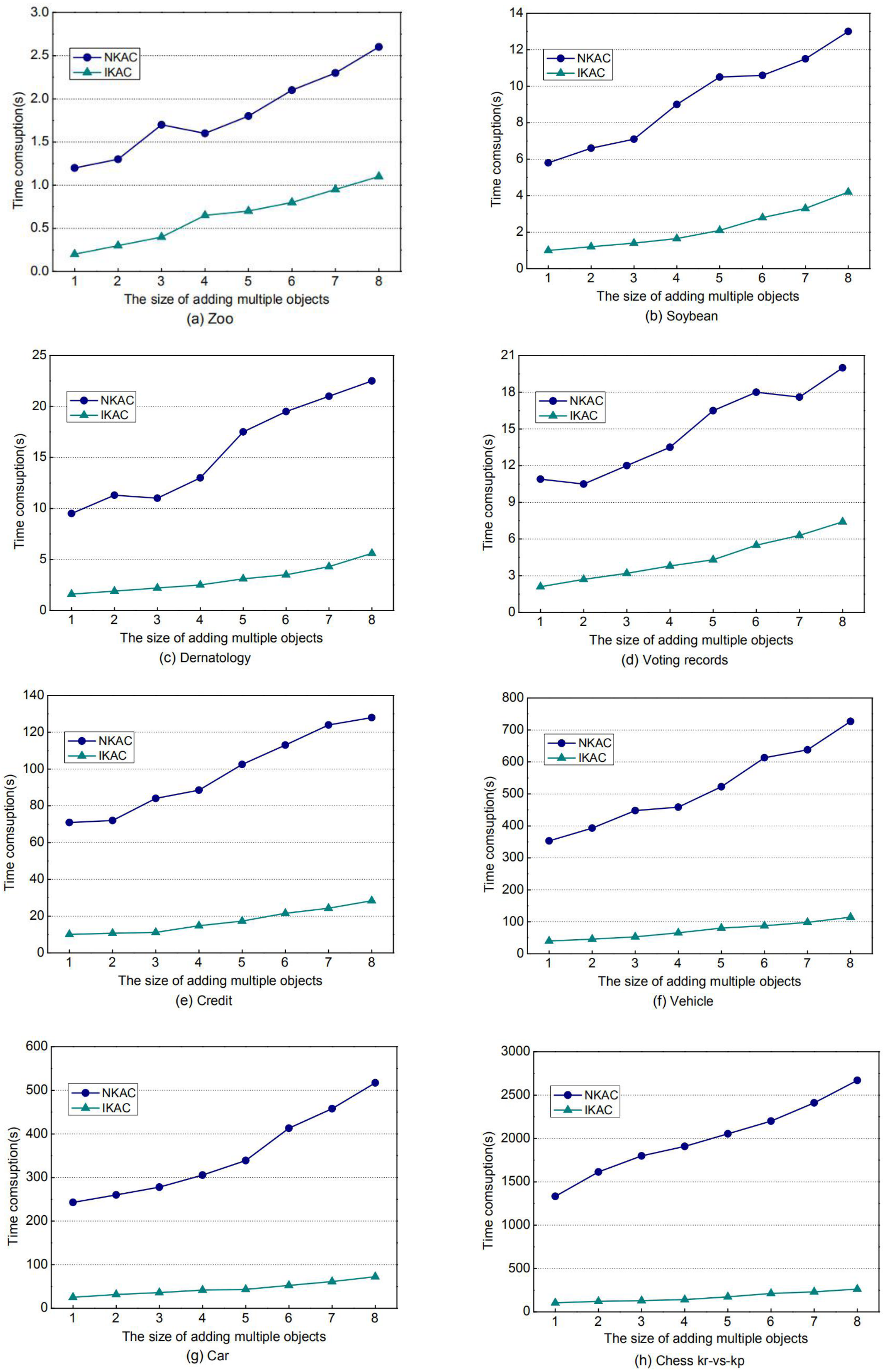

FIGURE 3. The computational time of Algorithms IKAC and NKAC versus different data sets with the arrival of incremental objects.

All the results reported in Table 3 establish the fact that the computational time of two algorithms increases as the size of data sets increases. However, the proposed algorithm IKAC has less time than NKAC to extract the interesting knowledge. For example, for the Car data set, NKAC takes 514.0812s to extract the decision rules. In contrast, IKAC takes about

73.2940s to find the decision rules. In addition, take the Chess kr-vs-kp data set as an example, IKAC needs 264.8136s to find the interesting knowledge, while the algorithm NKAC use more than ten times than that of IKAC. The similar behaviors also hold for other data sets. The advantage of IKAC over NKAC is clear, particularly for large-scale data sets. This is expected since the proposed algorithm uses the original data results upon the arrival of new objects. The non-incremental attribute reduction algorithm NKAC has to be run from scratch when new objects arrive so that it is often computationally time consuming and even for data sets with large objects.

In addition, to validate the computational efficiency of the proposed algorithm, for each data set shown in Table 6, 60% of the objects are selected as the original incomplete data sets, and the remaining 40% of the objects are divided into eight parts of equal size. The first part is regarded as the first incremental dataset arriving, the combination of the first incremental object set and the second part is viewed as the second incremental object set, the combination of the second incremental object set and the third part is viewed as the third incremental object set, …., and the combinative of all eight parts is viewed as the eighth incremental object set. With the increase of data size, the experimental results of two knowledge acquisition algorithms are shown in Fig.3. This figure displays more detailed change trend of two algorithms in the computational time with the increasing size of the data set. In Fig.3, the x-coordinate pertains to the ith incremental data set arriving, while the y-coordinate concerns the computational time. The computational time is expressed in seconds.

From Fig.3, we can see that the computational time of two algorithms increases as the increasing size of data sets. However, Algorithm IKAC is faster than NKAC on knowledge updating in all eight data sets. Take the data set Credit as an example, the computational time of Algorithm IKAC is about 14.8s at the fourth data set, while NKAC takes about 88.5s to extract the rule at the fourth data set. The main reason attributed to the fact that IKAC can induce the interesting knowledge at a time. The accuracy matrix and coverage matrix in the algorithm IKAC avoid recalculating from the scratch, which only compute the changed tolerance classes using the previous results. However, NKAC retrains the dynamic data set as a new one, which needs to be executed repeatedly to induce the interesting knowledge. The effect is more obvious for large-scale data sets. For another example, for the data set Vehicle, NKAC takes about 448s to extract the rules for finding the knowledge at the third data set, while IKAC takes about 53.2s to induce the interesting knowledge. On the whole, the experimental results indicate that in comparison with the non-incremental algorithm NKAC, the proposed algorithm IKAC can induce the interesting knowledge in much shorter time from data with missing values at the arrival of new objects, especially for massive data sets.

Based on the aforementioned experimental results, we can conclude that the proposed algorithm gives an efficient way to knowledge acquisition from data with missing values at the arrival of new objects.

VII. CONCLUSION

Knowledge acquisition is to extract useful knowledge from the solicited domain so as to construct a knowledge-based system. However, knowledge acquisition from the data with missing values is a challenging problem, especially for dynamic environment. In this paper, we first give a fast approach for computing tolerance classes in the incomplete data, which play an important role in the knowledge acquisition process. Then, we develop an incremental knowledge acquisition algorithm to update the original knowledge when a single object adds into the incomplete decision system. Applying this algorithm, the new knowledge can be quickly obtained, not recomputing the knowledge from the very beginning, such that the computational efficiency is improved. Finally, experimental results demonstrate the effectiveness of the proposed algorithm. Our future research work will focus on the proposed algorithm can be extended to other generalized granular computing models.

REFERENCES

[1] J. B. Zhang, T. R. Li, and H. M. Chen, "Composite rough sets for dynamic data mining," *Inf. Sci.*, vol. 257, pp. 81–100, Feb. 2014

[2] L. Feng, T. R. Li, D. Ruan, and S. R. Gou, "A vague-rough set approach for uncertain knowledge acquisition," *Knowl.-Based Syst.*, vol. 24, no. 6, pp. 837–843, Aug. 2011.

[3] S. Trabelsi, Z. Elouedi, and P. Lingras, "Classification systems based on rough sets under the belief function framework," *Int. J. Approx. Reasoning*, vol. 52, no. 9, pp. 1409–1432, Dec. 2011.

[4] S.-H. Lin, C.-C. Huang, and Z.-X. Che, "Rule induction for hierarchical attributes using a rough set for the selection of a green fleet," *Appl. Soft Comput.*, vol. 37, pp. 456–466, Dec. 2015.

[5] X. J. Xie and X. L. Qin, "A novel incremental attribute reduction approach for dynamic incomplete decision systems," *Int. J. Approx. Reasoning*, vol. 93, pp. 443–462, Feb. 2018.

[6] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA, USA: Kluwer, 1991.

[7] Z. Pawlak and A. Skowron, "Rough sets and Boolean reasoning," *Inf. Sci.*, vol. 177, no. 1, pp. 41–73, Jan. 2007.

[8] M. Slota, J. Leite, and T. Swift, "On updates of hybrid knowledge bases composed of ontologies and rules," *Artif. Intell.*, vol. 229, pp. 33–104, Dec. 2015

[9] L. Y. Zhang, W. Lu, X. D. Liu, W. Pedrycz, and C. Q. Zhong, "Fuzzy C-Means clustering of incomplete data based on probabilistic information granules of missing values," *Knowl.-Based Syst.*, vol. 99, pp. 51–70, May 2016.

[10] C. Luo, T. R. Li, and Y. Yao, "Dynamic probabilistic rough sets with incomplete data," *Inf. Sci.*, vol. 417, pp. 39–54, Nov. 2017

[11] M. Kryszkiewicz, "Rules in incomplete information systems," *Inf. Sci.*, vol. 113, nos. 3–4, pp. 271–292, Feb. 1999

[12] W. H. Shu and H. Shen, "Multi-criteria feature selection on cost-sensitive data with missing values," *Pattern Recognit.*, vol. 51, pp. 268–280, Mar. 2016

[13] W. B. Qian and W. H. Shu, "Mutual information criterion for feature selection from incomplete data," *Neurocomputing*, vol. 168, pp. 210–220, Nov. 2015

[14] P. G. Clark, J. W. Grzymala-Busse, and W. Rzasa, "Consistency of incomplete data," *Inf. Sci.*, vol. 322, pp. 197–222, Nov. 2015.

[15] Y. Leung, W.-Z. Wu, and W.-X. Zhang, "Knowledge acquisition in incomplete information systems: A rough set approach," *Eur. J. Oper. Res.*, vol. 168, no. 1, pp. 164–180, Jan. 2006.

[16] J. Li, C. Mei, and Y. Lv, "Incomplete decision contexts: Approximate concept construction, rule acquisition and knowledge reduction," *Int. J. Approx. Reasoning*, vol. 54, no. 1, pp. 149–165, Jan. 2013.

[17] M.-W. Shao, Y. Leung, and W.-Z. Wu, "Rule acquisition and complexity reduction in formal decision contexts," *Int. J. Approx. Reasoning*, vol. 55, no. 1, pp. 259–274, Jan. 2014.
[18] F. Shi, S. Sun, and J. Xu, "Employing rough sets and association rule mining in KANSEI knowledge extraction," *Inf. Sci.*, vol. 196, pp. 118–128, Aug. 2012.
[19] Y. Du, Q. Hu, P. Zhu, and P. Ma, "Rule learning for classification based on neighborhood covering reduction," *Inf. Sci.*, vol. 181, no. 24, pp. 5457–5467, Dec. 2011.
[20] A. Tan, J. J. Li, G. P. Lin, and Y. J. Lin, "Fast approach to knowledge acquisition in covering information systems using matrix operations," *Knowl.-Based Syst.*, vol. 79, pp. 90–98, May 2015.
[21] R. Deb, and A. W.-C. Liew, "Missing value imputation for the analysis of incomplete traffic accident data," *Inf. Sci.*, vol. 339, pp. 274–289, Apr. 2016.
[22] R. P. Prado, S. G. Galán, J. E. M. Exposito, and A. J. Yuste, "Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 6, pp. 1083–1097, Dec. 2010.
[23] J. H. Dai, H. W. Tian, W. Wang, and L. Liu, "Decision rule mining using classification consistency rate," *Knowl.-Based Syst.*, vol. 43, pp. 95–102, May 2013.
[24] J. H. Li, Y. Ren, C. L. Mei, Y. H. Qian, and X. B. Yang, "A comparative study of multigranulation rough sets and concept lattices via rule acquisition," *Knowl.-Based Syst.*, vol. 91, pp. 152–164, Jan. 2016.
[25] J. B. Zhang, J.-S. Wong, T. R. Li, and Y. Pan, "A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems," *Int. J. Approx. Reasoning*, vol. 55, no. 3, pp. 896–907, Mar. 2014.
[26] D. Liu, T. Li, D. Ruan, and W. Zou, "An incremental approach for inducing knowledge from dynamic information systems," *Fundam. Inform.*, vol. 94, no. 2, pp. 245–260, Apr. 2009.
[27] Y. Fan, C.-C. Chern, and C.-C. Huang, "Rule induction based on an incremental rough set," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jun. 2008, pp. 1207–1214.
[28] X. Zhang, C. L. Mei, D. G. Chen, and J. H. Li, "Multi-confidence rule acquisition and confidence-preserved attribute reduction in interval-valued decision systems," *Int. J. Approx. Reasoning*, vol. 55, no. 8, pp. 1787–1804, 2014.
[29] S. Petrov, "Dynamics properties of knowledge acquisition," *Cogn. Syst. Res.*, vol. 47, pp. 12–15, Jan. 2018.
[30] W.-Z. Wu, Y. H. Qian, T.-J. Li, and S.-M. Gu, "On rule acquisition in incomplete multi-scale decision tables," *Inf. Sci.*, vol. 378, pp. 282–302, Feb. 2017.
[31] M. Ali *et al.*, "A data-driven knowledge acquisition system: An end-to-end knowledge engineering process for generating production rules," *IEEE Access*, vol. 6, pp. 15587–15607, 2018.
[32] F. Ganz, P. Barnaghi, and F. Carrez, "Automated semantic knowledge acquisition from sensor data," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1214–1225, Sep. 2016.
[33] W. H. Shu and H. Shen, "Incremental feature selection based on rough set in dynamic incomplete data," *Pattern Recognit.*, vol. 47, no. 12, pp. 3890–3906, Dec. 2014.
[34] K.-P. Lin, K.-C. Hung, and C.-L. Lin, "Rule generation based on novel kernel intuitionistic fuzzy rough set model," *IEEE Access*, vol. 6, pp. 11953–11958, 2018.
[35] *UCI Machine Learning Repository*. [Online]. Availble: http://www.ics.uci.edu/~mlearn/MLRepository.html

WENHAO SHU received the M.S. degree in computer science from the Key Laboratory of Data Processing and Intelligent Computing, Guangxi Normal University, in 2011, and the Ph.D. degree in computer science from the School of Computer and Information Technology, Beijing Jiaotong University, China, in 2015. She is currently a Lecturer with the School of Information Engineering, East China Jiaotong University, China. Her main research interests include knowledge discovery, granular computing, and data mining. She has authored some international journal papers in her research field, such as *Pattern Recognition*, *Knowledge-Based Systems*, and the *International Journal of Approximate Reasoning*.

WENBIN QIAN received the M.S. degree in computer science from the Key Laboratory of Data Processing and Intelligent Computing, Guangxi Normal University, in 2010, and the Ph.D. degree in computer science from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2014. He is currently an Associate Professor with the School of Software, Jiangxi Agricultural University, China. His current research interests include granular computing and machine learning. He has published more than 30 referred journal and conference papers in these areas.

YONGHONG XIE received the M.S. and Ph.D. degrees in computer science from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2001 and 2005, respectively. She is currently an Associate Professor with the Beijing Key Laboratory of Knowledge Engineering for Materials Science, and the School of Computer and Communication Engineering, University of Science and Technology Beijing. Her current research interests include data mining and intelligent systems.