

Incremental learning optimization on knowledge discovery in dynamic business intelligent systems

Dun Liu · Tianrui Li · Da Ruan · Junbo Zhang

Received: 1 April 2010 / Accepted: 31 August 2010 / Published online: 22 September 2010
© Springer Science+Business Media, LLC. 2010

Abstract As business information quickly varies with time, the extraction of knowledge from the related dynamically changing database is vital for business decision making. For an incremental learning optimization on knowledge discovery, a new incremental matrix describes the changes of the system. An optimization incremental algorithm induces interesting knowledge when the object set varies over time. Experimental results validate the feasibility of the incremental learning optimization.

Keywords Rough set theory · Incremental learning · Accuracy · Coverage · Interesting knowledge · Business information · Optimization

D. Liu (✉)
School of Economics and Management, Southwest Jiaotong University,
610031 Chengdu, People's Republic of China
e-mail: newton83@163.com

T. Li · J. Zhang
School of Information Science and Technology, Southwest Jiaotong University,
610031 Chengdu, People's Republic of China

T. Li
e-mail: trli@swjtu.edu.cn

J. Zhang
e-mail: JunboZhang86@163.com

D. Ruan
Belgian Nuclear Research Centre (SCK•CEN), Boeretang 200, 2400 Mol, Belgium
e-mail: druan@sckcen.be

D. Ruan
Department of Applied Mathematics & Computer Science, Ghent University,
9000 Ghent, Belgium
e-mail: da.ruan@ugent.be

1 Introduction

Business intelligence (BI) refers to skills, processes, technologies, applications and practices for business decision making. It is also a broad category of technologies and applications for gathering, storing, analyzing, and providing access to data to help enterprise users make better choices. Despite its many successful applications in different business domains, such as, E-government, E-business, E-commerce, E-market, E-finance, and E-learning systems with the corresponding technologies of web research, web mining, and web-based support systems [9, 19–21, 28, 37], BI has still many critical challenges, including fast and reliable decision making and extraction of relevant knowledge from dynamically changing massive databases [7, 26].

With the insights gained from these opinions, we focus on two aspects of BI decision process. The first is the strategies for knowledge discovery. Considering the business database in an enterprise, a natural strategy of knowledge acquisition is to use the associate rules to induce the corresponding interesting decision criteria. The second is to develop efficient and optimization algorithms for knowledge discovery in updating business systems. As the information is changing all the time and the database is changing at an unprecedented rate in real business decision problems, it is crucial to develop the optimization algorithms for the needs of the business decision support system. A good incremental optimization algorithm helps company managers make a quicker and better choice.

Incremental learning approaches in BI have already received much attention for nearly decades, both in theories and applications [2, 3, 8, 12, 22, 40]. The two simple and commonly operators named Zooming-in and Zooming-out [34], have been used to describe the dynamic character when the business system is changing. The Zooming-in operator allows us to refine the granules of the universe, like decomposing a granule into many granules; the Zooming-out operator allows us to coarsen the granules of the universe by omitting some details of the problem, such as combining many granules to form a new granule [34, 35]. With the insightful ideas from coarsening or refining process, the previous studies on incremental learning approaches and strategies for an updating BI information system mainly focused on three aspects: (1) The coarsening or refining of object set while the attribute set remains constant; (2) The coarsening or refining of attribute set while the object set remains constant; (3) The coarsening or refining of attribute values while the object set and attribute set remain constant. In the first case, Shan and Ziarko presented a discernibility-matrix based incremental methodology to find all maximally generalized rules [27]. Bang and Bien 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 [1]. 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 [29]. Furthermore, Liu et al. proposed an incremental model and approach as well as its algorithm for inducing interesting knowledge when the object set varies over time [15]. Zheng and Wang developed a rough set and rule tree based incremental knowledge acquisition algorithm RRIA, to get new knowledge learnt more quickly [38]. Hu et al. [11] constructed a novel incremental attribute reduction algorithm when new objects are added into a decision information system. In addition, Błaszczyński and Slowinski discussed the incremental induction of decision rules from dominance-based rough approximations to select the most interesting representatives in the final set of rules [13]. In the second case, Chan proposed an incrementally mining algorithm for learning classification rules efficiently when an attribute set in the information system evolves over time [4]. Li et al. [14] presented a method for updating approximations of a concept in an incomplete information system

through characteristic relations when an attribute set varies over time. Followed by Chan and Li's work, Liu et al. [18] discussed the strategies and propositions in a probabilistic approximate space when attributes are changed. In the third case, Liu et al. [16] 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 [17]. Chen et al. [5,6] developed an approach for dynamically updating approximations when attribute values are coarsened or refined.

We believe the first case of the object set, which evolves over time, seems more important in BI systems because the objects in business database are changing all the time. It is necessary to design some optimization algorithms for knowledge discovery from updating BI systems. Unfortunately, the former studies have mainly focused on the detailed change process when objects enter or get out of the system, but ignored considering the efficiency of the incremental strategies. Especially, in Liu's work [15], the change objects should be estimated to the proper classification each time, the ideas of their approach are reasonable but the heuristic algorithm of their work needs further improvements. Following their work, we induce the incremental matrix and present a new optimization approach for knowledge discovery from BI systems when the object set evolves with time.

The rest of the paper is organized as follows: we provide basic concepts of rough sets and interesting knowledge in Sect. 2. We introduce incremental matrix to illuminate the proposed approach, and present a new optimization model as well as its algorithm for incrementally learning interesting knowledge in dynamic BI systems in Sect. 3. We give an illustrated example and compare the simulation results of the proposed algorithm with other methods to validate the new model in Sect. 4. We conclude the paper with further research topics in Sect. 5.

2 Preliminaries

Basic concepts, notations and results of information systems as well as their extensions are briefly reviewed in this section [16,17,23–25,33,36,39].

For an approximation space $\mathcal{K} = (U, R)$, let U be a finite and non-empty set called the universe and $R \subseteq U \times U$ an equivalence on U . The equivalence relation R partitions the set U into several disjoint subsets, and the partitions of the universe form a quotient set induced by R , denoted by U/R . Suppose there are 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 $\mathcal{K} = (U, R)$ is characterized by an information system $S = (U, C \cup D, V, f)$ with $C \cap D = \emptyset$, where C denotes the condition attribute set and D denotes the decision attribute set. $V = \cup_{a \in A} V_a$, V_a is a domain of the attribute a and $A = C \cup D$. $f: U \times A \rightarrow V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. In addition, each non-empty subset $B \subseteq C$ determines an equivalence relation as follows.

$$R_B = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \forall a \in B\}$$

This equivalence relation R_B partitions U into some equivalence classes given by:

$$U/R_B = \{[x]_B \mid x \in U\}$$

where $[x]_B$ denotes the equivalence class determined by x with respect to B , $[x]_B = \{y \in U \mid (x, y) \in R_B\}$. For simplicity, U/R_B will be replaced by U/B .

We now review the concepts of interesting knowledge followed by [15]’s introductions. In the earlier studies, Wong and Ziarko used two measures, namely, the confidence and resolution factors for inductive learning [32]. However, Han and Kamber suggested that the interesting knowledge is usually induced by certain patterns, which represent in the form of association rules. The rule support and rule confidence are introduced in [10] to measure the rule interestingness. Furthermore, the rule support reflects the usefulness of discovered rules and the rule confidence reflects the certainty of discovered rules. In addition, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold [10]. In addition, Tsumoto argued the accuracy and coverage measure of the degree of sufficiency and necessity, respectively, and the simplest probabilistic model is the one that only uses classification rules that have high accuracy and high coverage [30, 31]. Motivated by these ideas, the three parameters: support, accuracy and coverage can be used to discover the interesting knowledge. Since “support” can be expressed by accuracy and coverage, we follow Tsumoto’s idea and choose the latter two factors to describe interesting knowledge in this paper.

Definition 1 [15] Suppose an information system $S = (U, C \cup D, V, f)$ with $C \cap D = \emptyset$. $U/C = \{X_1, X_2, \dots, X_m\}$ is a partition of objects under the condition attributes of C , where $X_i (i = 1, 2, \dots, m)$ is a condition equivalence class; $U/D = \{D_1, D_2, \dots, D_n\}$ is a partition of objects under the decision attribute of D , where $D_j (j = 1, 2, \dots, n)$ is a decision equivalence class. $\forall X_i \in U/C, \forall D_j \in U/D$, the support, accuracy and coverage of $X_i \rightarrow D_j$ are defined respectively as follows:

Support of $X_i \rightarrow D_j$: $Supp(D_j|X_i) = |X_i \cap D_j|$;

Accuracy of $X_i \rightarrow D_j$: $Acc(D_j|X_i) = |X_i \cap D_j|/|X_i|$;

Coverage of $X_i \rightarrow D_j$: $Cov(D_j|X_i) = |X_i \cap D_j|/|D_j|$.

where $|X_i|$ and $|D_j|$ denote the cardinality of set X_i and D_j , respectively.

By considering the massive data sets in real business database, we use the matrices to simplify the problem. Hence, the support matrix, the accuracy matrix, the coverage matrix as well as their propositions are defined as follows [15].

$$Supp(D|X) = \begin{pmatrix} Supp(D_1|X_1) & Supp(D_2|X_1) & \cdots & Supp(D_n|X_1) \\ Supp(D_1|X_2) & Supp(D_2|X_2) & \cdots & Supp(D_n|X_2) \\ \vdots & \vdots & \ddots & \vdots \\ Supp(D_1|X_m) & Supp(D_2|X_m) & \cdots & Supp(D_n|X_m) \end{pmatrix} \quad (1)$$

$$Acc(D|X) = \begin{pmatrix} Acc(D_1|X_1) & Acc(D_2|X_1) & \cdots & Acc(D_n|X_1) \\ Acc(D_1|X_2) & Acc(D_2|X_2) & \cdots & Acc(D_n|X_2) \\ \vdots & \vdots & \ddots & \vdots \\ Acc(D_1|X_m) & Acc(D_2|X_m) & \cdots & Acc(D_n|X_m) \end{pmatrix} \quad (2)$$

$$Cov(D|X) = \begin{pmatrix} Cov(D_1|X_1) & Cov(D_2|X_1) & \cdots & Cov(D_n|X_1) \\ Cov(D_1|X_2) & Cov(D_2|X_2) & \cdots & Cov(D_n|X_2) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(D_1|X_m) & Cov(D_2|X_m) & \cdots & Cov(D_n|X_m) \end{pmatrix} \quad (3)$$

Proposition 1 $Supp(D_j|X_i) \geq 0, \forall X_i \in U/C, \forall D_j \in U/D, i = 1, 2, \dots, m, j = 1, 2, \dots, n$.

Proposition 2 $0 \leq Acc(D_j|X_i) \leq 1$ and $\sum_{j=1}^n Acc(D_j|X_i) = 1, \forall X_i \in U/C, i = 1, 2, \dots, m$.

Proposition 3 $0 \leq Cov(D_j|X_i) \leq 1$ and $\sum_{i=1}^m Cov(D_j|X_i) = 1, \forall D_j \in U/D, j = 1, 2, \dots, n$.

With respect to the above discussions, we obtain $Acc(D_j|X_i) = \frac{Supp(D_j|X_i)}{\sum_{j=1}^n Supp(D_j|X_i)}$, $Cov(D_j|X_i) = \frac{Supp(D_j|X_i)}{\sum_{i=1}^m Supp(D_j|X_i)}$, which illustrate the relations among the three matrices.

The support matrix, accuracy matrix and coverage matrix help us extract the useful rules from the BI database, and the definition of interesting knowledge is displayed as follows.

Definition 2 [15] For $\forall X_i (i = 1, 2, \dots, m), \forall D_j (j = 1, 2, \dots, n)$, if $Acc(D_j|X_i) \geq \alpha$ and $Cov(D_j|X_i) \geq \beta$ hold, we call the rule $X_i \rightarrow D_j$ a kind of interesting knowledge where $\alpha \in (0.5, 1)$ and $\beta \in (0, 1)$.

The threshold $\alpha > 0.5$ is inspired by the criterion of a simple majority rule, that is, the accuracy of the rule which is more than 0.5 can be executed. In general, we usually choose the interesting rules with high accuracy and high coverage.

3 A new approach for incrementally mining interesting knowledge

We discuss the change of interesting knowledge in dynamic BI information systems when the object set evolves over time while the attribute set remains constant. To illustrate the method clearly, we divide the works into three parts. We give some assumptions and the basic structure of our approach are given in Sect. 3.1; we proposed an incremental model as well as its analysis process in Sect. 3.2; we provide the optimization algorithm when multiple objects enter and go out of the system in Sect. 3.3.

3.1 Assumptions and basic structure for the incrementally approach

In our discussion, 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 [15]. The former case means the objects enter into the system at time $t + 1$, and the universe is expanding; the latter case means the objects get out of the system, and the universe is contracting at time $t + 1$. Both the immigration and emigration processes reflect the coarsening or refining of the object set in BI systems.

To describe a business dynamic information system, we denote a complete information system at time t as $S = (U, C \cup D, V, f)$, with a condition class set of objects $U/C = \{X_1, X_2, \dots, X_m\}$ and a decision class set of objects $U/D = \{D_1, D_2, \dots, D_n\}$, where U is a non-empty finite set of objects at time t . At time $t + 1$, some objects enter the system while some go out, so the original business information system S will be changed into $S' = (U', C' \cup D', V', f')$. Similarly, the accuracy matrix and coverage matrix at time t are denoted as $Acc^{(t)}(D|X)$ and $Cov^{(t)}(D|X)$, respectively, and the ones at time $t + 1$ as $Acc^{(t+1)}(D'|X')$ and $Cov^{(t+1)}(D'|X')$, respectively. According to Definition 1, the rule $X_i \rightarrow D_j$ is interesting if $Acc^{(t)}(D_j|X_i) \geq \alpha$ and $Cov^{(t)}(D_j|X_i) \geq \beta$ at time t ; the rule $X'_i \rightarrow D'_j$ is interesting if $Acc^{(t+1)}(D'_j|X'_i) \geq \alpha$ and $Cov^{(t+1)}(D'_j|X'_i) \geq \beta$ at time $t + 1$ according to Definition 2.

With these stipulations, the following work focuses on two aspects: (1) constructing the optimization model for learning interesting knowledge based on the model in [15]; (2)

analyzing the implementation approach of the proposed model and designing its corresponding optimization algorithm.

3.2 The incremental model for learning interesting knowledge

Suppose there are N objects that enter the system and M objects that get out of the system at time $t + 1$. We denote the set of immigrating N objects as \mathbf{N} and the emigrating set of M objects as \mathbf{M} . When a new object \bar{x} enters into the information system, $\forall \bar{x} \in \mathbf{N}$, four possible cases may happen.

- (I). \bar{x} forms a new conditional equivalence class and a new decision equivalence class, and it satisfies: $\forall x \in U, \forall a \in C, f(\bar{x}, a) \neq f(x, a)$ and $\forall x \in U, \forall d \in D, f(\bar{x}, d) \neq f(x, d)$.
- (II). \bar{x} only forms a new conditional equivalence class, and it satisfies: $\forall x \in U, \forall a \in C, f(\bar{x}, a) \neq f(x, a)$ and $\exists x \in U, \forall d \in D, f(\bar{x}, d) = f(x, d)$.
- (III). \bar{x} only forms a new decision equivalence class, and it satisfies: $\exists x \in U, \forall a \in C, f(\bar{x}, a) = f(x, a)$ and $\forall x \in U, \forall d \in D, f(\bar{x}, d) \neq f(x, d)$.
- (IV). \bar{x} neither generates a new conditional equivalence class nor a new decision equivalence class, and it satisfies: $\exists x \in U, \forall a \in C, f(\bar{x}, a) = f(x, a)$ and $\exists x \in U, \forall d \in D, f(\bar{x}, d) = f(x, d)$.

In the same way, when an object \tilde{x} gets out of the information system, $\forall \tilde{x} \in \mathbf{M}$, there holds only one case, that is, the accuracy and coverage of the rule induced by \tilde{x} may change [15].

Based on the above discussion, the cases (I) and (II) generate new conditional equivalence classes, the cases (I) and (III) generate new decision equivalence classes. We assume the N new objects will form l new conditional classes $X'_{m+1}, X'_{m+2}, \dots, X'_{m+l}$ and r new decision classes $D'_{n+1}, D'_{n+2}, \dots, D'_{n+r}$. Then, $\forall \bar{x} \in \mathbf{N}$, we can count the cardinal number of N_i , which means the objects belong to conditional equivalence class X_i ($i = 1, 2, \dots, m + l$). In the same way, $\forall \tilde{x} \in \mathbf{M}$, we can count the cardinal number of M_i , which means the objects get out of conditional equivalence class X_i ($i = 1, 2, \dots, m$). The detail process of objects immigration and emigration are shown in Fig. 1. Thus we have,

$$N_i = \sum_{j=1}^{n+r} N_{ij}, \quad N = \sum_{i=1}^{m+l}, \quad N_i = \sum_{i=1}^{m+l} \sum_{j=1}^{n+r} N_{ij};$$

$$M_i = \sum_{j=1}^n M_{ij}, \quad M = \sum_{i=1}^m, \quad M_i = \sum_{i=1}^m \sum_{j=1}^n M_{ij}.$$

where N_{ij} represents that there are N_i objects that enter the conditional class X_i and N_{ij} objects of them result in the decision class D_j ; M_{ij} represents that there are M_i objects that get out of the conditional equivalence class X_i and in which M_{ij} objects get out from the decision equivalence class D_j .

A natural strategy of incremental learning is directly using the above judgment criteria to the immigrating set \mathbf{N} and emigrating set \mathbf{M} , which is detailed discussed in [15]. Since the immigration (emigration) of the object sets \mathbf{N} and \mathbf{M} can be regarded as the composition of single object immigrates (emigrates), a simply way to deal with the problem is gradually updating the actuary matrix and coverage matrix for every object in the immigrating set \mathbf{N} and the emigrating set \mathbf{M} , we can get the new actuary matrix and coverage matrix at time $t + 1$ after the last updating [15].

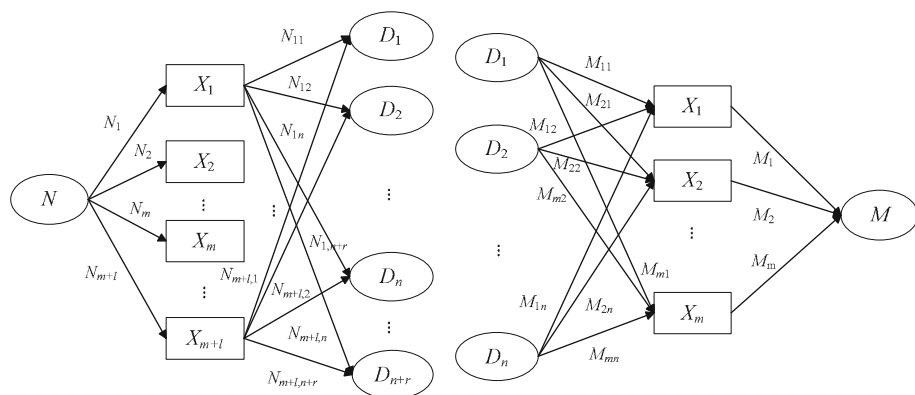


Fig. 1 The immigration and emigration of the object set [15]

However, the approach in [15] is not efficient enough because the updating process is step by step calculated for every changed object. Instead of this, a new model and approach are proposed in this section. From Fig. 1, we have the relationship of condition equivalence classes and decision equivalence classes between time t and time $t+1$. As already mentioned previously, at time t , we have a condition equivalence class set $U/C = \{X_1, X_2, \dots, X_m\}$ and a decision equivalence class set $U/D = \{D_1, D_2, \dots, D_n\}$. At time $t+1$, the conditional equivalence set is marked as $U'/C' = \{X'_1, X'_2, \dots, X'_m, \dots, X'_{m+l}\}$ and the decision equivalence set $U'/D' = \{D'_1, D'_2, \dots, D'_n, \dots, D'_{n+r}\}$. Note that X_i and X'_i actually describe the same conditional equivalence class. The only difference of them is that their cardinalities are different, of which the relationships list as follows, and this is also true for D_j and D'_j .

$$|X'_i| = \begin{cases} |X_i| + N_i - M_i; & i \in \{1, 2, \dots, m\} \\ N_i; & i \in \{m+1, m+2, \dots, m+l\} \end{cases} \quad (4)$$

$$|D'_j| = \begin{cases} |D_j| + \sum_{i=1}^{m+l} N_{ij} - \sum_{i=1}^m M_{ij}; & j \in \{1, 2, \dots, n\} \\ \sum_{i=1}^{m+l} N_{ij}; & j \in \{n+1, n+2, \dots, n+r\} \end{cases} \quad (5)$$

We also note that $|X'_i|$ and $|D'_j|$ can be equal to zero, because we do include the consideration of M objects emigrated from condition equivalence classes and decision equivalence classes. If $|X'_{i^*}| = 0$, $i^* \in \{1, 2, \dots, m\}$, all the objects related to X'_{i^*} have disappeared from time t to time $t+1$, without any incoming object that belongs to X'_{i^*} . According to the definitions of $Acc(D'_j|X'_{i^*})$ and $Cov(D'_j|X'_{i^*})$, the i^* th row in the matrix $Acc^{(t+1)}(D'|X')$ and the matrix $Cov^{(t+1)}(D'|X')$ are all zero. Similarly, if $|D'_{j^*}| = 0$, $j^* \in \{1, 2, \dots, n\}$, all the objects related to D'_{j^*} have emigrated from the system and no new object result in D'_{j^*} . In this case, we have the j^* th column in the matrix $Acc^{(t+1)}(D'|X')$ and the matrix $Cov^{(t+1)}(D'|X')$ are all zero.

According to Fig. 1 and the above analysis, we can use the following matrices to describe the updating process.

At time t , the support matrix describes the cardinal number distribution for $X_i \rightarrow D_j$. To simplify the model, we expand the original $m \times n$ matrix into $(m+l) \times (n+r)$ matrix, and

the row from $m + 1$ to $m + l$, the column from $n + 1$ to $n + r$ are all set as zero. The support matrix at time t can be rewritten as:

$$Supp^{(t)}(D|X) = \begin{pmatrix} |X_1 \cap D_1| & |X_1 \cap D_2| & \cdots & |X_1 \cap D_n| & 0 & \cdots & 0 \\ |X_2 \cap D_1| & |X_2 \cap D_2| & \cdots & |X_2 \cap D_n| & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ |X_m \cap D_1| & |X_m \cap D_2| & \cdots & |X_m \cap D_n| & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (6)$$

At time $t + 1$, N new objects enter into the system and M objects get out of the system. We define an incremental matrix to show the changes in Fig. 1 as follows:

$$INC(D|X) = \begin{pmatrix} N_{11} - M_{11} & \cdots & N_{1,n} - M_{1,n} & N_{1,n+1} & \cdots & N_{1,n+r} \\ N_{21} - M_{21} & \cdots & N_{2,n} - M_{2,n} & N_{2,n+1} & \cdots & N_{2,n+r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{m1} - M_{m1} & \cdots & N_{m,n} - M_{m,n} & N_{m,n+1} & \cdots & N_{m,n+r} \\ N_{m+1,1} - M_{m+1,1} & \cdots & N_{m+1,n} - M_{m+1,n} & N_{m+1,n+1} & \cdots & N_{m+1,n+r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{m+l,1} & \cdots & N_{m+l,n} & N_{m+l,n+1} & \cdots & N_{m+l,n+r} \end{pmatrix} \quad (7)$$

The support matrix at time $t + 1$ can be then generated from (6) to (7) and displayed as follows.

$$Supp^{(t+1)}(D'|X') = Supp^{(t)}(D|X) + INC(D|X) \\ = \begin{pmatrix} |X_1 \cap D_1| + N_{11} - M_{11} & \cdots & |X_1 \cap D_n| + N_{1,n} - M_{1,n} & N_{1,n+1} & \cdots & N_{1,n+r} \\ |X_2 \cap D_1| + N_{21} - M_{21} & \cdots & |X_2 \cap D_n| + N_{2,n} - M_{2,n} & N_{2,n+1} & \cdots & N_{2,n+r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ |X_m \cap D_1| + N_{m1} - M_{m1} & \cdots & |X_m \cap D_n| + N_{m,n} - M_{m,n} & N_{m,n+1} & \cdots & N_{m,n+r} \\ N_{m+1,1} - M_{m+1,1} & \cdots & N_{m+1,n} - M_{m+1,n} & N_{m+1,n+1} & \cdots & N_{m+1,n+r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{m+l,1} & \cdots & N_{m+l,n} & N_{m+l,n+1} & \cdots & N_{m+l,n+r} \end{pmatrix} \quad (8)$$

The matrix (8) gives an intuitive interpretation of the changing objects. We can easily calculate the accuracy and coverage at time $t + 1$ by using (4), (5) and (8) according to Definition 1 as follows, respectively.

$$Acc^{(t+1)}(D'_j|X'_i) = \frac{|X'_i \cap D'_j|}{|X'_i|} = \begin{cases} \frac{|X_i \cap D_j| + N_{ij} - M_{ij}}{|X_i| + N_i - M_i} & i \in \{1, 2, \dots, m\}, \\ & j \in \{1, 2, \dots, n\} \\ \frac{N_{ij}}{|X_i| + N_i - M_i} & i \in \{1, 2, \dots, m\}, \\ & j \in \{n + 1, n + 2, \dots, n + r\} \\ \frac{N_{ij}}{N_i} & i \in \{m + 1, m + 2, \dots, m + l\}, \\ & j \in \{1, 2, \dots, n + r\} \end{cases} \quad (9)$$

$$Cov^{(t+1)}(D'_j|X'_i) = \frac{|X'_i \cap D'_j|}{|D'_j|} = \begin{cases} \frac{|X_i \cap D_j| + N_{ij} - M_{ij}}{|D_j| + \sum_{i=1}^{m+l} N_{ij} - \sum_{i=1}^m M_{ij}} & i \in \{1, 2, \dots, m\}, \\ & j \in \{1, 2, \dots, n\} \\ \frac{N_{ij}}{|D_j| + \sum_{i=1}^{m+l} N_{ij} - \sum_{i=1}^m M_{ij}} & i \in \{m+1, m+2, \dots, m+l\}, \\ & j \in \{1, 2, \dots, n\} \\ \frac{N_{ij}}{\sum_{i=1}^{m+l} N_{ij}} & i \in \{1, 2, \dots, m+l\}, \\ & j \in \{n+1, n+2, \dots, n+r\} \end{cases} \quad (10)$$

The two mathematical equations can be used to construct the accuracy matrix and coverage matrix at time $t + 1$, respectively. It also displays the relationship of accuracy and coverage between time t and time $t + 1$, which reveals the inner connection of the two pairs of matrices. Using this model, the following work focuses on the implementation algorithm of our model.

3.3 The incremental algorithm for dynamic knowledge discovery

The incremental algorithm for dynamic knowledge discovery is based on the model described in Sect. 3.2. Compared with the algorithm proposed in [15], their approach mainly focuses on the detailed change process of each object. It should judge the possible situation for every object when it enters into or gets out of the system at first, then its corresponding row and column of accuracy and coverage matrix should be recalculated again and again at time $t + 1$ [15]. However, here we simply use one incremental matrix to present the change process of objects, and the updating matrices can directly calculate by the incremental matrix.

Therefore, the most important task of updating interesting knowledge has broken down to find out nonzero N_{ij} and M_{ij} in the incremental matrix. An smart and optimization way to compute the accuracy matrix and coverage matrix after the system change is only updating those impacted objects in the two matrices. Motivated by these opinions, the $(m+l) \times (n+r)$ incremental matrix is setting as all zero at time t . Then, at time $t + 1$, if the objects enter into the system, we check whether to add rows (columns) or not. In the same way, if the objects get out of the system, we also check what change will happen in their corresponding row or column. Furthermore, the nonzero parameters should be found out to generate the incremental matrix. For instance, if $N_{ab} \neq 0$ and $M_{cd} \neq 0$ for $a \in \{1, 2, \dots, m+l\}$, $b \in \{1, 2, \dots, n+r\}$, $c \in \{1, 2, \dots, m\}$, $d \in \{1, 2, \dots, n\}$, we only update the corresponding row $Acc^{(t+1)}(D'_j|X'_a)$, $Cov^{(t+1)}(D'_j|X'_a)$, $Acc^{(t+1)}(D'_j|X'_c)$, $Cov^{(t+1)}(D'_j|X'_c)$ and corresponding column $Acc^{(t+1)}(D'_b|X'_i)$, $Cov^{(t+1)}(D'_b|X'_i)$, $Acc^{(t+1)}(D'_d|X'_i)$, $Cov^{(t+1)}(D'_d|X'_i)$ using (4), (5), (9) and (10). Again, as discussed in Sect. 3.1, if $|X'_a| = 0$ or $|D'_b| = 0$, we set the corresponding row or column as zero.

The concrete steps of the incremental approach for updating interesting knowledge are displayed as follows.

- Step 1: Calculating the support matrix, the accuracy matrix and the coverage matrix at time t according to Definition 1 as $Supp^{(t)}(D|X) = (Supp^{(t)}(D_j|X_i))_{m \times n}$, $Acc^{(t)}(D|X) = (Acc^{(t)}(D_j|X_i))_{m \times n}$ and $Cov^{(t)}(D|X) = (Cov^{(t)}(D_j|X_i))_{m \times n}$, respectively.
- Step 2: Constructing the incremental matrix and calculate the support matrix $Supp^{(t+1)}(D'|X') = (Supp^{(t+1)}(D'_j|X'_i))_{(m+l) \times (n+r)}$ at time $t + 1$. Then, calculating the accuracy matrix $Acc^{(t+1)}(D'|X') = (Acc^{(t+1)}(D'_j|X'_i))_{(m+l) \times (n+r)}$ and the

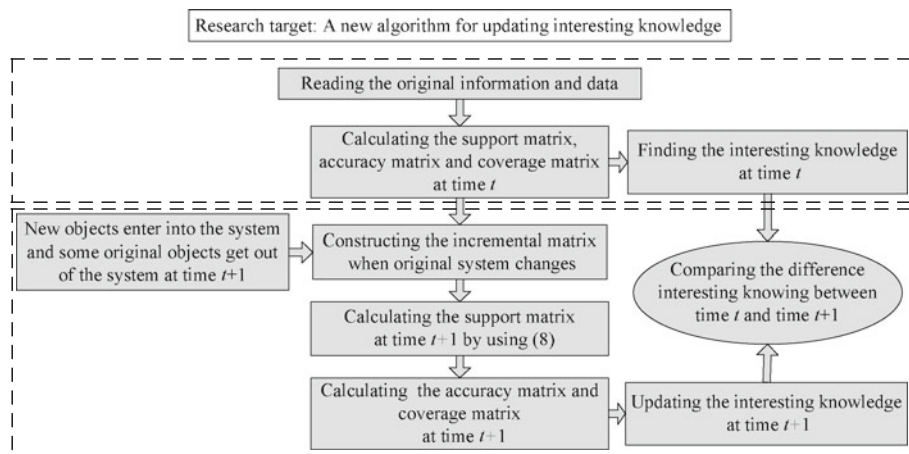


Fig. 2 The flowchat of the new incremental algorithm

coverage matrix $Cov^{(t+1)}(D'|X') = (Cov^{(t+1)}(D'_j|X'_i))_{(m+l) \times (n+r)}$ at time $t+1$ by using Algorithm 1, which shows the detail steps of knowledge updating after the system changes.

Step 3: Constructing two 2-dimensional attribute value pairs for every rule $X_i \rightarrow D_j$ at time t and $X'_i \rightarrow D'_j$ at time $t+1$. Therefore, we may trace the change of interesting knowledge at different times according to the variation of these tables.

The simple flowchat of the optimization incremental algorithm for updating interesting knowledge is shown as follows, and the detailed algorithm is listed in the “Appendix” (Fig. 2).

4 An illustration

In this section, we discuss our incremental algorithm and the algorithm proposed in [15] by using an example of custom behavior at first. Then, we compare our algorithm with other existed algorithms by using the real databases, and present the experimental results at the end of this section.

4.1 An example to update the business intelligent system dynamically

As the same illustration in [15], considering a BI complete information system $S = (U, A, V, f)$ at time t given in Table 1. $U = \{x_1, x_2, \dots, x_{12}\}$ stands for the 12 shopping-behavior customers, we choose shopping experience, brand and price as condition attributes and described by the set $C = \{a_1, a_2, a_3\}$, the purchase intention is considered as the decision attribute $D = \{d\}$. Then, $U/C = \{X_1, X_2, X_3, X_4, X_5, X_6\}$, $U/D = \{D_1, D_2, D_3, D_4\} = \{0, 1, 2, 3\}$, where $X_1 = \{x_1\}$, $X_2 = \{x_2, x_3\}$, $X_3 = \{x_4, x_5\}$, $\{X_4\} = \{x_6\}$, $X_5 = \{x_7, x_8, x_9\}$, $X_6 = \{x_{10}, x_{11}, x_{12}\}$, $D_1 = \{x_1, x_2, x_7\}$, $D_2 = \{x_3, x_4, x_8, x_{10}\}$, $D_3 = \{x_6, x_9, x_{11}\}$, $D_4 = \{x_5, x_{12}\}$. And Num stands for the cardinality of objects.

The meaning of the values for every attribute is shown as follows.

Shopping experience (a_1): 0 = Inexperience; 1 = Some experience; 2 = Rich experience.

Brand (a_2): 0 = Unknown; 1 = Average; 2 = Famous.

Table 1 A BI complete information system for customers

| U | a_1 | a_2 | a_3 | d | Num |
|----------|-------|-------|-------|-----|-------|
| x_1 | 0 | 0 | 0 | 0 | 10 |
| x_2 | 0 | 1 | 0 | 0 | 15 |
| x_3 | 0 | 1 | 0 | 1 | 5 |
| x_4 | 0 | 1 | 1 | 1 | 25 |
| x_5 | 0 | 1 | 1 | 3 | 3 |
| x_6 | 1 | 1 | 2 | 2 | 42 |
| x_7 | 1 | 2 | 2 | 0 | 3 |
| x_8 | 1 | 2 | 2 | 1 | 20 |
| x_9 | 1 | 2 | 2 | 2 | 47 |
| x_{10} | 2 | 2 | 2 | 1 | 5 |
| x_{11} | 2 | 2 | 2 | 2 | 15 |
| x_{12} | 2 | 2 | 2 | 3 | 30 |

Price (a_3): 0 = Cheap; 1 = Average; 2 = Expensive.

Purchase intention (d): 0 = No intention; 1 = Balance; 2 = Possible intention; 3 = Strong intention.

First, we construct the 6×4 support matrix for $X_i \rightarrow D_j$ at time t as follows.

$$Supp^{(t)}(D|X) = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 15 & 5 & 0 & 0 \\ 0 & 25 & 0 & 3 \\ 0 & 0 & 42 & 0 \\ 3 & 20 & 47 & 0 \\ 0 & 5 & 15 & 30 \end{pmatrix} \quad (11)$$

Second, the accuracy matrix and the coverage matrix at time t are computed respectively according to Definition (1) in the following.

$$Acc^{(t)}(D|X) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.893 & 0 & 0.107 \\ 0 & 0 & 1 & 0 \\ 0.043 & 0.286 & 0.671 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \end{pmatrix} \quad (12)$$

$$Cov^{(t)}(D|X) = \begin{pmatrix} 0.357 & 0 & 0 & 0 \\ 0.536 & 0.091 & 0 & 0 \\ 0 & 0.454 & 0 & 0.091 \\ 0 & 0 & 0.404 & 0 \\ 0.107 & 0.364 & 0.452 & 0 \\ 0 & 0.091 & 0.144 & 0.909 \end{pmatrix} \quad (13)$$

At time $t + 1$, some objects will enter or get out of the system. Suppose three cases may happen: (I). 10 customers with the behavior of “ $a_1 = 0, a_2 = 0, a_3 = 0, d = 0$ ” will get out of the system; (II). 10 customers with the behavior of “ $a_1 = 1, a_2 = 2, a_3 = 2, d = 2$ ” will enter into the system; (III). 20 customers with the behavior of “ $a_1 = 2, a_2 = 3, a_3 = 2, d = 3$ ” will enter into the system.

In Liu's approach, the updating process is divided into three parts, which corresponds the three cases (I), (II) and (III) [15].

For the case (I), the changes affect the first row and the first column in (12)–(13), and we have:

$$Acc^{(t+1)}(D'_j|X'_1) = Cov^{(t+1)}_1(D'_j|X'_1) = 0.$$

$$\text{And for } X'_u \rightarrow D'_1 (u \neq 1),$$

$$Acc^{(t+1)}(D'_1|X'_u) = Acc^{(t)}(D_1|X_u),$$

$$Cov^{(t+1)}(D'_1|X'_u) = (|X_u \cap D_1|)/(|D_1| - 10).$$

The accuracy matrix and the coverage matrix are changed as:

$$Acc^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.893 & 0 & 0.107 \\ 0 & 0 & 1 & 0 \\ 0.043 & 0.286 & 0.671 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \end{pmatrix} \quad (14)$$

$$Cov^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.833 & 0.091 & 0 & 0 \\ 0 & 0.454 & 0 & 0.091 \\ 0 & 0 & 0.404 & 0 \\ 0.167 & 0.364 & 0.452 & 0 \\ 0 & 0.091 & 0.144 & 0.909 \end{pmatrix} \quad (15)$$

We use (14)–(15) to update the (12)–(13) as the two metrics at time t . Then, for the case (II), the changes affect the fifth row and the third column in (14)–(15), and we have:

$$Acc^{(t+1)}(D'_3|X'_5) = (|X_5 \cap D_3| + 10)/(|X_5| + 10), Cov^{(t+1)}(D'_3|X'_5) = (|X_5 \cap D_3| + 10)/(|D_3| + 10) \text{ for } X'_5 \rightarrow D'_3.$$

$$Acc^{(t+1)}(D'_k|X'_5) = (|X_5 \cap D_k|)/(|X_5| + 10), Cov^{(t+1)}(D'_k|X'_5) = Cov^{(t)}(D'_k|X'_5) \text{ for } X'_5 \rightarrow D'_k (k \neq 3).$$

$$Acc^{(t+1)}(D'_3|X'_u) = Acc^{(t)}(D_3|X_u), Cov^{(t+1)}(D'_3|X'_u) = (|X_u \cap D_3|)/(|D_3| + 10) \text{ for } X'_u \rightarrow D'_3 (u \neq 5).$$

The accuracy matrix and the coverage matrix are changed as:

$$Acc^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.893 & 0 & 0.107 \\ 0 & 0 & 1 & 0 \\ 0.038 & 0.25 & 0.712 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \end{pmatrix} \quad (16)$$

$$Cov^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.833 & 0.091 & 0 & 0 \\ 0 & 0.454 & 0 & 0.091 \\ 0 & 0 & 0.368 & 0 \\ 0.167 & 0.364 & 0.5 & 0 \\ 0 & 0.091 & 0.132 & 0.909 \end{pmatrix} \quad (17)$$

We again use (16)–(17) to update (14)–(15) as the two metrics at time t . Then, for the case (III), the changes generate a new conditional equivalent class, and the two matrices of (16) and (17) should add a new row named the seventh row. So, the changes affect the seventh row and the forth column in (16)–(17), and we have:

$Acc^{(t+1)}(D'_4|X'_7) = 1$, $Cov^{(t+1)}(D'_4|X'_7) = 20/(|D_4| + 20)$ for $X'_7 \rightarrow D'_4$.

$Acc^{(t+1)}(D'_k|X'_7) = Cov^{(t+1)}(D'_k|X'_7) = 0$ for $X'_7 \rightarrow D'_k$ ($k \neq 4$).

$Acc^{(t+1)}(D'_4|X'_u) = Acc^{(t)}(D_4|X_u)$, $Cov^{(t+1)}(D'_4|X'_u) = |X_u \cap D_4|/(|D_4| + 20)$ for $X'_u \rightarrow D'_4$ ($u \neq 7$).

$$Acc^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.893 & 0 & 0.107 \\ 0 & 0 & 1 & 0 \\ 0.038 & 0.25 & 0.712 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (18)$$

$$Cov^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.833 & 0.091 & 0 & 0 \\ 0 & 0.454 & 0 & 0.057 \\ 0 & 0 & 0.368 & 0 \\ 0.167 & 0.364 & 0.5 & 0 \\ 0 & 0.091 & 0.132 & 0.566 \\ 0 & 0 & 0 & 0.377 \end{pmatrix} \quad (19)$$

Compared with the above method in [15], the new approach in this paper gives a simple and clear way to solve the same problem. According to the algorithm proposed in Sect. 3, for (I), $M_{11} = 10$; for (II), $N_{53} = 10$; for (III), there will add one row X_7 and $N_{74} = 20$. So, the incremental matrix can be written as:

$$INC(D|X) = \begin{pmatrix} -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 \end{pmatrix} \quad (20)$$

Hence, the support matrix for $X_i \rightarrow D_j$ at time $t + 1$ can be calculated as follows.

$$\begin{aligned} Supp^{(t+1)}(D'|X') &= Supp^{(t)}(D|X) + INC(D|X) \\ &= \begin{pmatrix} 10 & 0 & 0 & 0 \\ 15 & 5 & 0 & 0 \\ 0 & 25 & 0 & 3 \\ 0 & 0 & 42 & 0 \\ 3 & 20 & 47 & 0 \\ 0 & 5 & 15 & 30 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 15 & 5 & 0 & 0 \\ 0 & 25 & 0 & 3 \\ 0 & 0 & 42 & 0 \\ 3 & 20 & 57 & 0 \\ 0 & 5 & 15 & 30 \\ 0 & 0 & 0 & 20 \end{pmatrix} \quad (21) \end{aligned}$$

In the same way, the accuracy matrix and the coverage matrix at time $t + 1$ can be calculated by (21) according to Definition 1 as follows, and we find the two matrices have the

Table 2 The 2-dimensional table of knowledge at time t and $t + 1$

| Time | Class | $D_1 (D'_1)$ | $D_2 (D'_2)$ | $D_3 (D'_3)$ | $D_4 (D'_4)$ |
|---------|--------|----------------|----------------|----------------|----------------|
| t | X_1 | (1, 0.357) | (0, 0) | (0, 0) | (0, 0) |
| $t + 1$ | X'_1 | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| t | X_2 | (0.75, 0.536) | (0.25, 0.091) | (0, 0) | (0, 0) |
| $t + 1$ | X'_2 | (0.75, 0.833) | (0.25, 0.091) | (0, 0) | (0, 0) |
| t | X_3 | (0, 0) | (0.893, 0.454) | (0, 0) | (0.107, 0.091) |
| $t + 1$ | X'_3 | (0, 0) | (0.893, 0.454) | (0, 0) | (0.107, 0.057) |
| t | X_4 | (0, 0) | (0, 0) | (1, 0.404) | (0, 0) |
| $t + 1$ | X'_4 | (0, 0) | (0, 0) | (1, 0.368) | (0, 0) |
| t | X_5 | (0.043, 0.107) | (0.286, 0.364) | (0.671, 0.452) | (0, 0) |
| $t + 1$ | X'_5 | (0.038, 0.167) | (0.25, 0.364) | (0.712, 0.5) | (0, 0) |
| t | X_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.144) | (0.6, 0.909) |
| $t + 1$ | X'_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.132) | (0.6, 0.566) |
| t | X_7 | – | – | – | – |
| $t + 1$ | X'_7 | (0, 0) | (0, 0) | (0, 0) | (1, 0.377) |

same results as (18) and (19).

$$Acc^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.893 & 0 & 0.107 \\ 0 & 0 & 1 & 0 \\ 0.038 & 0.25 & 0.712 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (22)$$

$$Cov^{(t+1)}(D'|X') = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.833 & 0.091 & 0 & 0 \\ 0 & 0.454 & 0 & 0.057 \\ 0 & 0 & 0.368 & 0 \\ 0.167 & 0.364 & 0.5 & 0 \\ 0 & 0.091 & 0.132 & 0.566 \\ 0 & 0 & 0 & 0.377 \end{pmatrix} \quad (23)$$

Followed by the incremental algorithm in this paper, (12) and (13) are used to construct the 2-dimensional value pairs $(Acc^{(t)}(D_j|X_i), Cov^{(t)}(D_j|X_i))$ at time t . (22) and (23) are used to construct the another 2-dimensional value pairs $(Acc^{(t+1)}(D'_j|X'_i), Cov^{(t+1)}(D'_j|X'_i))$ at time $t + 1$. The 2-dimensional table of knowledge at time t and $t + 1$ is shown in Table 2.

In Table 2, the first place of the value pair (\bullet, \bullet) means the accuracy value and the second place means the coverage values, the sign “–” stands for the non-existence pair. The table gives an intuitive impression for interesting knowledge discover. If we fix the value pair (α, β) , we obtain the interesting knowledge at time t and $t + 1$ directly and quickly from Table 2, i.e., if we set $\alpha = 0.6$ and $\beta = 0.4$, the rules of $X_2 \rightarrow D_1$, $X_3 \rightarrow D_2$, $X_4 \rightarrow D_3$, $X_5 \rightarrow D_3$ and $X_6 \rightarrow D_4$ are interesting customer behavior at time t by using the thresholds requirement in Definition 2. Similarly, the rules of $X'_2 \rightarrow D'_1$, $X'_3 \rightarrow D'_2$, $X'_5 \rightarrow D'_3$ and $X'_6 \rightarrow D'_4$ are interesting customer behavior at time $t + 1$, and the rule $X'_4 \rightarrow D'_3$ is no longer interesting because of the changes of objects. Furthermore, decision makers can choose the property α and β to generate the interesting knowledge according to their own experience [15].

Table 3 The 2-dimensional table of knowledge from time t to time $t + 3$

| Time | Class | $D_1(D'_1, D''_1, D'''_1)$ | $D_2(D'_2, D''_2, D'''_2)$ | $D_3(D'_3, D''_3, D'''_3)$ | $D_4(D'_4, D''_4, D'''_4)$ |
|---------|----------|----------------------------|----------------------------|----------------------------|----------------------------|
| t | X_1 | (1, 0.357) | (0, 0) | (0, 0) | (0, 0) |
| $t + 1$ | X'_1 | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| $t + 2$ | X''_1 | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| $t + 3$ | X'''_1 | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| t | X_2 | (0.75, 0.536)* | (0.25, 0.091) | (0, 0) | (0, 0) |
| $t + 1$ | X'_2 | (0.75, 0.833)* | (0.25, 0.091) | (0, 0) | (0, 0) |
| $t + 2$ | X''_2 | (0.75, 0.833)* | (0.25, 0.091) | (0, 0) | (0, 0) |
| $t + 3$ | X'''_2 | (0.75, 0.833)* | (0.25, 0.091) | (0, 0) | (0, 0) |
| t | X_3 | (0, 0) | (0.893, 0.454)* | (0, 0) | (0.107, 0.091) |
| $t + 1$ | X'_3 | (0, 0) | (0.893, 0.454)* | (0, 0) | (0.107, 0.091) |
| $t + 2$ | X''_3 | (0, 0) | (0.893, 0.454)* | (0, 0) | (0.107, 0.091) |
| $t + 3$ | X'''_3 | (0, 0) | (0.893, 0.454)* | (0, 0) | (0.107, 0.057) |
| t | X_4 | (0, 0) | (0, 0) | (1, 0.404)* | (0, 0) |
| $t + 1$ | X'_4 | (0, 0) | (0, 0) | (1, 0.404)* | (0, 0) |
| $t + 2$ | X''_4 | (0, 0) | (0, 0) | (1, 0.368) | (0, 0) |
| $t + 3$ | X'''_4 | (0, 0) | (0, 0) | (1, 0.368) | (0, 0) |
| t | X_5 | (0.043, 0.107) | (0.286, 0.364) | (0.671, 0.452)* | (0, 0) |
| $t + 1$ | X'_5 | (0.043, 0.167) | (0.286, 0.364) | (0.671, 0.452)* | (0, 0) |
| $t + 2$ | X''_5 | (0.038, 0.167) | (0.25, 0.364) | (0.712, 0.5)* | (0, 0) |
| $t + 3$ | X'''_5 | (0.038, 0.167) | (0.25, 0.364) | (0.712, 0.5)* | (0, 0) |
| t | X_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.144) | (0.6, 0.909)* |
| $t + 1$ | X'_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.144) | (0.6, 0.909)* |
| $t + 2$ | X''_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.132) | (0.6, 0.909)* |
| $t + 3$ | X'''_6 | (0, 0) | (0.1, 0.091) | (0.3, 0.132) | (0.6, 0.566)* |
| t | X_7 | – | – | – | – |
| $t + 1$ | X'_7 | – | – | – | – |
| $t + 2$ | X''_7 | – | – | – | – |
| $t + 3$ | X'''_7 | (0, 0) | (0, 0) | (0, 0) | (1, 0.377) |

At last, we note that the change process may last for many different times because the BI system many change every day. In our approach, the whole change process can be divided into several small parts, i.e., from time t to time $t + 1$, from time $t + 1$ to time $t + 2$, etc. Simply, we use the new updating data set in the next time to replace the old data, and generating new support matrix, accuracy matrix and coverage matrix to replace old ones. Specially, supposed the change process in Table 1 last for three different times, that is, the case (I) happens at time $t + 1$, the case (II) happens at time $t + 2$ and the case (III) happens at time $t + 3$. So, (14)–(15) are used to replace (12)–(13) after the first change at time $t + 1$; (16)–(17) are used to replace (14)–(15) after the second change at time $t + 2$; (18)–(19) are used to replace (16)–(17) after the third change at time $t + 3$. The updating processes of the two matrices from time t to time $t + 3$ are shown in Table 3 by using the algorithm in [15].

In Table 3, X_i, X'_i, X''_i, X'''_i stand for the i th conditional equivalence class at time $t, t + 1, t + 2$ and $t + 3$, respectively. D_j, D'_j, D''_j, D'''_j stand for the j th conditional equivalence class at time $t, t + 1, t + 2$ and $t + 3$, respectively. As stated above, when we set $\alpha = 0.6$ and $\beta = 0.4$, the value pair which labels the sign “*” on the top right corner induce the interesting

knowledge in different periods of time. Compared with Tables 2, 3, the computing results and the knowledge discover process are clearly displayed. In addition, to validate and analyze the proposed approach in this paper, we will introduce some real databases to explain our method in the following discussions.

4.2 Experimental evaluation

In this subsection, we chose four databases, from the well-known machine learning web (<http://www.cs.waikato.ac.nz/ml/weka/>), named “IRIS”, “CPU”, “Bank-data” and “Segment”, as benchmarks for the performance tests. The Table 4 shows the basic information of the four databases. Meanwhile, experiments have been performed on a 1.8G MHz Pentium Server with 2G of memory, running windows XP, algorithms are coded in VC++. The experiment illuminates the efficiency of our algorithm by comparing with other existing algorithms.

The strategies of the experiments are to delete the incomplete items of the database since the algorithm is based on complete information systems. Then, we use the 10-cross validation in the experimental evaluation. Specifically, we randomly choose 90% data from the original database as the training data at time t . The another 10% data to be considered as the immigration objects will enter into the system at time $t + 1$. We also randomly choose 5% data from the original data at time t as the emigration objects which will get out of the system at time $t + 1$.

Furthermore, to validate this new optimization algorithm, we induce two more algorithms into this research. The first one is called “Algorithm 1”, which means we treat the data at time $t + 1$ as absolutely new data without using any incremental strategy. The second one is called “Algorithm 2” which was originally proposed by Liu et al. in [15], every object that enters or gets out of the system may affect its corresponding row and column, and the change times of the matrices are depended on the total numbers of the immigration and emigration objects. In the optimization algorithm, the incremental matrix is directly used to describe the changes between time t and $t + 1$, and we can immediately calculate the accuracy and coverage matrix by using (9) and (10).

Following the discussion in Sect. 3.1, we suppose there are K objects at time t , and form m conditional classes and n decision classes ($m \leq K$, $n \leq K$). At time $t + 1$, N objects

Table 4 The basic information of the four databases

| Name | IRIS | CPU | Bank-data | Segment |
|-------------------------------------|------|-----|-----------|---------|
| Numbers of the objects | 150 | 209 | 600 | 1500 |
| Numbers of the condition attributes | 4 | 6 | 10 | 19 |
| Numbers of the decision attributes | 1 | 1 | 1 | 1 |

Table 5 The average elapsed times among the three algorithms (seconds)

| Name | Algorithm 1 | Algorithm 2 | Algorithm 3 |
|-----------|-------------|-------------|-------------|
| IRIS | 0.0283 | 0.0105 | 0.0086 |
| CPU | 0.1067 | 0.0391 | 0.0174 |
| Bank-data | 5.0063 | 2.5888 | 0.2001 |
| Segment | 30.8741 | 4.7238 | 0.5133 |

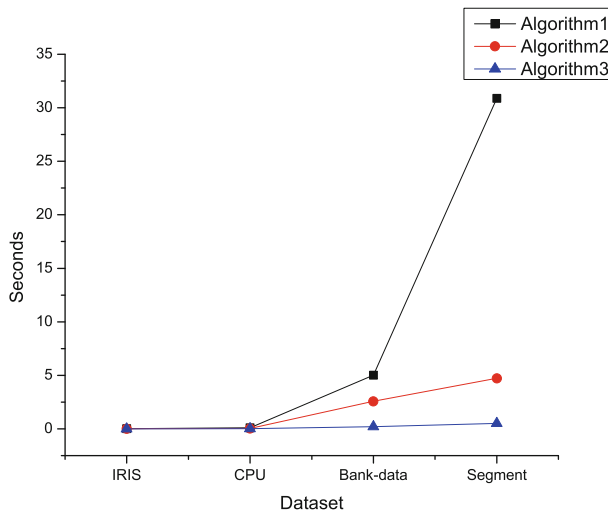


Fig. 3 The average elapsed times among the three algorithms

enter the system and M objects get out of the system at time $t + 1$, and the N new objects form l new conditional classes and r new decision classes ($l \leq N$, $r \leq M$). The “Algorithm 1” calculates $(K + N + M) \times (m + n + l + r)$ times, the “Algorithm 2” calculates $(N + M) \times (m + n + l + r - 1)$ times, and the new algorithm only calculates $(N + M)$ times. For convenience, here we name the new optimization algorithm as “Algorithm 3”.

In addition, we use the “10-cross validation” strategy to estimate the elapsed times for the four databases by using the three algorithms. The threshold values of α and β are fixed firstly. Then, we calculate the average elapsed times by repeating the computing process for 100 times. The experimental results are shown in Table 5 and Fig. 3.

From Table 5 and Fig. 3, we find Algorithm 3 is effective for the dynamic information system, especially for the complex and massive database. Since the objects in BI systems are updating all the time, our approach helps decision makers to make a quicker and better choice.

5 Conclusions

The varieties of environments lead to the changes of objects, namely, the immigration and emigration of objects result in the coarsening and refining of the universe. In this paper, we discussed some new optimization strategies and mechanisms for incremental learning knowledge in BI systems when objects are changed. Following the approach proposed in [15], we induced an incremental matrix to describe the object changes between time t and time $t + 1$. The incremental matrix helps us directly to calculate the support matrix, accuracy matrix and coverage matrix for generating the corresponding interesting rules. Finally, we provide a case study and the experiments to validate the rationality and validity of the proposed optimization methods. The study here is nevertheless based on the complete information system and the equivalence relation, and our future research will focus on extensions of the current approach to incomplete information systems and other generalized rough set models.

Acknowledgments This work is partially supported by the National Science Foundation of China (No. 60873108), the Doctoral Innovation Foundation of Southwest Jiaotong University (No. 200907) and the Scientific Research Foundation of Graduate School of Southwest Jiaotong University (No. 2009LD), China. The authors also thank Dong Han, Xiaodong Wang and Zhijie Chen for their assistance in preparing the manuscript.

Appendix: A new algorithm for updating interesting knowledge incrementally

Data: An information system $S = (U, C \cup D, V, F)$, two thresholds α and β .

Result: Support matrix, Accuracy matrix, Coverage matrix, Interesting knowledge at time t and $t + 1$, respectively.

Calculate the support matrix $Supp^{(t)}(D|X)$, accuracy matrix $Acc^{(t)}(D|X)$ and coverage matrix $Cov^{(t)}(D|X)$ at time t and output interesting knowledge.

```

for  $i = 1$  to  $m$  do
  for  $j = 1$  to  $n$  do
    calculate support  $Supp^{(t)}(D_j|X_i)$ , accuracy  $Acc^{(t)}(D_j|X_i)$  and coverage  $Cov^{(t)}(D_j|X_i)$  for every rule  $X_i \rightarrow D_j$ 
  end
end
for  $i = 1$  to  $m$  do
  for  $j = 1$  to  $n$  do
    if  $Acc^{(t)}(D_j|X_i) \geq \alpha$  and  $Cov_{i,j}^{(t)} \geq \beta$  then
      output the rule  $X_i \rightarrow D_j$ 
    end
  end
end

```

Construct the incremental matrix.

Consider \tilde{x} ($\forall \tilde{x} \in \mathbf{N}$) enters system at time $t + 1$, find nonzero N_{ij} .

```

for every  $\tilde{x}$  do
  for  $i = 1$  to  $m + l$ ,  $l = 0$  do
    for  $j = 1$  to  $n + r$ ,  $r = 0$  do
      if  $x \in X_i == false$  then
        if  $x \in D_j == false$  then
          obtain a new condition class and a new decision class, update  $N_{ij}$  then  $l++$ ,  $r++$ ;
        else
          obtain a new condition class, update  $N_{ij}$  then  $l++$ ;
        end
      else
        if  $x \in D_j == false$  then
          obtain a new decision class, update  $N_{ij}$  then  $r++$ ;
        else
          update  $N_{ij}$ .
        end
      end
    end
  end
end

```

Consider \tilde{x} ($\forall \tilde{x} \in \mathbf{M}$) gets out of system at time $t + 1$, find nonzero M_{ij} .

```

for every  $\tilde{x}$  do
  for  $i = 1$  to  $m$  do
    for  $j = 1$  to  $n$  do
      update  $M_{ij}$ 
    end
  end
end

```

Update the incremental matrix and calculate the support matrix $Supp^{(t+1)}(D'|X')$.

Calculate the accuracy matrix $Acc^{(t+1)}(D'|X')$ and coverage matrix $Cov^{(t+1)}(D'|X')$ at time $t + 1$ and output interesting knowledge.

```

for  $i = 1$  to  $m + l$  do
  for  $j = 1$  to  $n + r$  do
    if  $N_{ij} \neq 0$  or  $M_{ij} \neq 0$  then
      if  $|X'_i| == 0$  then
        set row  $i$  in Matrix  $Acc^{(t+1)}(D'|X')$  and  $Cov^{(t+1)}(D'|X')$  as zero
      else
        if  $|D'_j| == 0$  then
          set column  $j$  in Matrix  $Acc^{(t+1)}(D'|X')$  and Matrix  $Cov^{(t+1)}(D'|X')$  as zero
        else
          update row  $i$  and column  $j$  in Matrix  $Acc^{(t+1)}(D'|X')$  and  $Cov^{(t+1)}(D'|X')$  using
            formulae (4), (5), (9), (10)
          end
        end
      end
    else
       $Acc^{(t+1)}(D'_j|X'_i) = Acc^{(t)}(D_j|X_i)$ ;  $Cov^{(t+1)}(D'_j|X'_i) = Cov^{(t)}(D_j|X_i)$ .
    end
  end
end
for  $i = 1$  to  $m + l$  do
  for  $j = 1$  to  $n + r$  do
    if  $Acc^{(t+1)}(D'_j|X'_i) \geq \alpha$  and  $Cov^{(t+1)}(D'_j|X'_i) \geq \beta$  then
      output the rule  $X'_i \rightarrow D'_j$ 
    end
  end
end
end

```

References

- Bang, W., Bien, Z.: New incremental learning algorithm in the framework of rough set theory. *Int. J. Fuzzy Syst.* **1**, 25–36 (1999)
- Baourakis, G., Conisescu, M., van Dijk, G., Pardalos, P.M., Zopounidis, C.: A multicriteria approach for rating the credit risk of financial institutions. *Comput. Manage. Sci.* **6**(3), 347–356 (2009)
- Campana, E., Fasano, G., Pinto, A.: Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *J. Glob. Optim.* (2010). doi:[10.1007/s10898-009-9493-0](https://doi.org/10.1007/s10898-009-9493-0)
- Chan, C.: A rough set approach to attribute generalization in data mining. *Inf. Sci.* **107**, 177–194 (1998)
- Chen, H., Li, T., Liu, W. Zou, W.: Research on the approach of dynamically maintenance of approximations in rough set theory while attribute values coarsening and refining. In: *Proceedings of 2009 IEEE International Conference on GrC*, pp. 45–48 (2009)
- Chen, H., Li, T., Qiao, S., Ruan, D.: A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values. *Int. J. Intell. Syst.* **25**(10), 1005–1026 (2010)
- Cody, W., Kreulen, J., Krishna, V., Spangler, W.: The integration of business intelligence and knowledge management. *IBM Syst. J.* **41**(4), 697–713 (2002)
- Floudas, C., Pardalos, P.M.: *Encyclopedia of Optimization*, 2nd ed., XXXIV, 4626 pp. Springer (2009)
- Goyal, M., Lu, J., Zhang, G.: Decision making in multi-issue e-market auction using fuzzy attitudes. *J. Theor. Appl. Electron. Commer. Res.* **3**, 97–110 (2008)
- Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Press, San Francisco (2006)
- Hu, F., Wang, G., Huang, H., Wu, Y.: Incremental attribute reduction based on elementary sets. In: *RSFDGrC2005, LNAI*, vol. **3641**, pp. 185–193 (2005)
- Hu, J., Chang, H., Fu, M., Marcus, S.: Dynamic sample budget allocation in model-based optimization. *J. Glob. Optim.* (2009). doi:[10.1007/s10898-009-9493-0](https://doi.org/10.1007/s10898-009-9493-0)
- Jerzy, B., Slowinski, R.: Incremental induction of decision rules from dominance-based rough approximations. *Electron. Notes Theor. Comput. Sci.* **82**, 40–51 (2003)
- Li, T., Ruan, D., Wets, G., Song, J., Xu, Y.: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. *Knowl. Based Syst.* **20**, 485–494 (2007)
- Liu, D., Li, T., Ruan, D., Zou, W.: An incremental approach for inducing knowledge from dynamic information systems. *Fundam. Inform.* **94**, 245–260 (2009)

16. Liu, D., Li, T., Chen, H., Ji, X.: Approaches to knowledge incremental learning based on the changes of attribute values. In: Proceedings of the 4th International Conference on Intelligent Systems and Knowledge Engineering, pp. 94–99 (2009)
17. Liu, D., Li, T., Liu, G., Hu, P.: An approach for inducing interesting incremental knowledge based on the change of attribute values. In: Proceedings of 2009 IEEE International Conference on Granular Computing, pp. 415–418 (2009)
18. Liu, D., Zhang, J., Li, T.: A probabilistic rough set approach for incremental learning knowledge on the change of attribute. In: Proceedings of 2010 International Conference on Foundations and Applications of Computational Intelligence, pp. 722–727 (2010)
19. Lu, J., Ruan, D., Zhang, G.: E-service intelligence-Methodology, Technologies and Applications, E-service Intelligence. pp. 1–33. Springer, (2007)
20. Lu, J., Bai, C., Zhang, G.: E-service cost benefit evaluation and analysis, E-Service Intelligence. pp. 389–409. Springer, New York (2007)
21. Lu, J., Bai, C., Zhang, G.: Cost-benefit factor analysis in e-services using bayesian networks. *Expert Syst. Appl.* **36**, 4617–4625 (2009)
22. Pardalos, P.M., Hansen, P.: Data Mining and Mathematical Programming. American CRM Proceedings & Lecture Notes, vol. **45**, 234 pp. American Mathematical Society (2008)
23. Pawlak, Z.: Rough sets. *Int. J. Comput. Inf. Sci.* **11**, 341–356 (1982)
24. Pawlak, Z.: Rough set theory and its application to data analysis. *Cybern. Syst.* **29**, 661–688 (1998)
25. Qian, Y., Liang, J., Pedrycz, Z., Dang, C.: Positive approximation: an accelerator for attribute reduction rough set theory. *Artif. Intell.* **174**, 597–618 (2010)
26. Shaku, A.: The top 10 critical challenges for business intelligence success. Retrieved from <http://www.computerworld.com/computerworld/records/images/BusIntellWPonline.pdf>
27. Shan, L., Ziarko, W.: Data-based acquisition and incremental modification of classification rules. *Comput. Intell.* **11**, 357–370 (1995)
28. Sanati, F., Lu, J.: Life-event modelling framework for E-government integration. *Electron. Gov Int. J.* **7**, 183–202 (2010)
29. Tong, L.: An: incremental learning of decision rules based on rough set theory. In: Proceedings of the World Congress on Intelligent Control and Automation (WCICA2002), pp. 420–425 (2002)
30. Tsumoto, S.: Extraction of experts' decision process from clinical databases using rough set model. In: Proceedings of PKDD 1997, pp. 58–67 (1997)
31. Tsumoto, S.: Accuracy and coverage in rough set rule induction. In: Alpigini, J. et al. (eds.) RSCTC 2002, LNAI, vol. **2475**, pp. 373–380. (2002).
32. Wong, S., Ziarko, W., Pawlak, A.: Algorithm for inductive learning. *Bull. Pol. Acad. Sci. Tech. Sci.* **34**, 271–276 (1986)
33. Yao, Y., Wong, S.: A decision theoretic framework for approximating concepts. *Int. J. Man Mach. Stud.* **37**(6), 793–809 (1992)
34. Yao, Y.: A partition model of granular computing. *Transactions on Rough Sets I*, pp. 232–253 (2004)
35. Yao, Y.: Integrative levels of granularity. In: Human-Centric Information Processing Through Granular Modelling, pp. 31–47 (2009)
36. Yao, Y.: Three-way decisions with probabilistic rough sets. *Inf. Sci.* **180**(3), 341–353 (2010)
37. Zhang, G., Lu, J.: Fuzzy bilevel programming with multiple objectives and cooperative multiple followers. *J. Glob. Optim.* (2008). doi:[10.1007/s10898-008-9365-z](https://doi.org/10.1007/s10898-008-9365-z)
38. Zheng, Z., Wang, G.: RRJA: a rough set and rule tree based incremental knowledge acquisition algorithm. *Fundam. Inform.* **59**, 299–313 (2004)
39. Ziarko, W.: Variable precision rough set model. *Comput. Syst. Sci.* **46**, 39–59 (1993)
40. Zopounidis, C., Pardalos, P.M.: Handbook of multicriteria analysis. *Appl. Optim.* **103**, XXV, 455 pp (2010)