Knowledge-Based Systems 24 (2011) 1341–1347

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A dynamic attribute reduction algorithm based on 0-1 integer programming

Yitian Xu [a], Laisheng Wang [a,*], Ruiyan Zhang [b]

[a] College of Science, China Agricultural University, 100083 Beijing, China
[b] Science China Press, 100717 Beijing, China

ARTICLE INFO

Article history:
Received 4 January 2011
Received in revised form 10 June 2011
Accepted 14 June 2011
Available online 22 June 2011

Keywords:
Rough set theory
Attribute reduction
Dynamic reduct
0-1 Integer programming
Constraint preprocessing

ABSTRACT

Attribute reduction is an important research concept in rough set theory. Many attribute reduction algorithms were designed for the static information system in the past years. However, many real-world data are generated dynamically. Then a new dynamic attribute reduction algorithm based on a 0-1 integer programming is proposed to deal with the dynamic data in this paper. When multiple objects in the information system evolve over time, instead of treating the changed information table as a new one and finding the reduct again like rough set reduction algorithm does, the proposed algorithm just updates the original reduct. Therefore, its computational speed improves greatly. In addition, an approach of constraint preprocessing is also presented in this paper. Numerical experiments on twelve benchmark datasets testify the feasibility and validity of the proposed algorithm.

1. Introduction

Rough set theory has been introduced by Pawlak [1,2] to deal with imprecise or vague concepts. It can effectively eliminate the redundant information without reducing the classification performance [3]. At present, it has been successfully applied in various aspects ranging from knowledge acquisition, intelligent information processing, pattern recognition, data mining, and so on. Many researchers in these fields are very interested in this new research topic since it offers opportunities to discover useful knowledge in information systems.

Attribute reduction is an important concept in the rough set theory. However, the current researches mainly focus on the static information systems, i.e., the objects and attributes in a certain information system remain constant, e.g., discernibility matrix method [4], which reduces attributes by Boolean function in the conjunctive and disjunctive normal form, and information entropy method [5,6], which chooses the important attributes by changing of the information entropy. Unfortunately, finding minimal reduct turns out to be an NP-hard problem [7,8]. Then, many heuristic methods for finding one reduct have been investigated in [9–12]. But the above heuristic methods are inadequate at finding minimal reduct, then, reformulating the rough set reduction task in a propositional satisfiability framework was presented in [13]. In reality, real data sources possess dynamic characteristics and the data volume grows both in the dimensions of attributes and objects at an unprecedented rate. To maintain the effectiveness of knowledge from the dynamical data, it is necessary to develop an incremental strategy for the updating knowledge.

Nowadays, the incremental learning approaches based on rough set theory have received much attention. They mainly focus on these two cases [14]: (1) The object set in the information system evolves over time while the attribute set remains constant. (2) The attribute set in the information system evolves over time while the object set remains constant. In the first case, the incremental updating algorithms for attribute reduction based on the discernibility matrix or improved discernibility matrix were mainly discussed, e.g., Liu proposed an incremental arithmetic for the minimal reduct in [15], but it only discussed the information system without the decision attribute. Wang proposed an incremental reduction algorithm based on Skowron discernibility matrix in [16], but it cannot ensure to find the minimal reduct. Moreover, it cannot deal with the inconsistent system. An incremental updating algorithm for attribute reduction based on the improved discernibility matrix was presented in [17,18]. When the original reduct is not the candidate set, we need to reduce the attributes by heuristic method from the core set. Although it can dynamically update the reduct, it cannot ensure to find the minimal reduct. Moreover, it cannot deal with the case of multiple newly increased objects. Shan and Ziarko [19] presented an incremental methodology based on the discernibility matrix introduced by Skowron and Rauszer [4] for finding all maximally generalized rules. In the second case, Chan [20] proposed the incremental mining algorithms for learning classification rules when an attribute set in the information system evolves over time. Li et al. [21] presented a method for updating approximations of a concept in an incomplete information system via characteristic relations when an attribute set

* Corresponding author. Tel.: +86 10 62737077; fax: +86 10 62736777.
E-mail addresses: xytshuxue@126.com (Y. Xu), wanglaish@126.com (L. Wang).

0950-7051/$ - see front matter Crown Copyright © 2011 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.06.007

varies over time. Incremental learning with respect to the new input attributes was also researched in [22]. In this paper, we only focus on the first case, namely, when the object set in the information system evolves over time while the attribute set remains constant, how to dynamically reduce attributes.

To our best knowledge, previous work on incremental learning is mainly concerned with the situation where only one single object enters the information system. But in most real problems, multiple objects will enter the system constantly. In this paper, we consider the later case. We know that the process of attribute reduction is also to delete some unimportant attributes and reserve the important ones. In the spirit of the attribute reduction, a new attribute reduction algorithm based on 0-1 integer programming is proposed in this paper, which is applicable to the multiple objects entering the information system. The proposed algorithm converts the attribute reduction problem into a 0-1 integer programming problem. It just updates the old reduct based on the newly increased objects. Therefore, its efficiency is improved greatly.

There are two contributions in this paper. One is that we propose a dynamic attribute reduction algorithm based on 0-1 integer programming for the multiple objects added into the system. The other is that we provide an approach of constraint preprocessing. Large numbers of redundant constraints will be deleted by this approach so that it can make the optimization problem resolved easily.

The paper is organized as follows. Section 2 outlines some preliminary knowledge. A new attribute reduction algorithm based on 0-1 integer programming for the static information system is presented in Section 3. An approach of constraint preprocessing is shown in Section 4. A dynamic reduction algorithm based on 0-1 integer programming is proposed in Section 5. Computational complexity is analyzed in Section 6. Numerical experiments are conducted in Section 7. The last section deals with conclusions.

2. Preliminaries

2.1. Rough set theory

Rough set theory deals with information represented by a table called the information system, which consists of objects (or cases) and attributes.

Definition 1 (*Information system*). An information system is composed of a 4-tuple as follows,

$$S = \langle U, A, V, f \rangle, \tag{1}$$

where U is the universe, a finite set of n objects $\{x_1, x_2, \ldots, x_n\}$, $A = C \cup D$, where C is a set of condition attributes and D is a set of decision attributes, respectively. V is a set of values of attributes. $f : U \times A \to V$ is the total decision function called the information function.

Let $S = \langle U, A, V, f \rangle$ be an information system, and every $P \subseteq A$ generates an indiscernibility relation $Ind(P)$ on U.

Definition 2 (*An indiscernibility relation*). Given a subset of attributes $P \subseteq A$, an indiscernibility relation is defined by

$$Ind(P) = \{(x,y) \in U \times U : f(x,\alpha) = f(y,\alpha), \forall \alpha \in P\}. \tag{2}$$

$U/Ind(P) = \{C_1, C_2, \ldots, C_k\}$ is a partition of U by P, where every C_i is an equivalence class.

Upper and lower approximations are the important concepts in rough set theory.

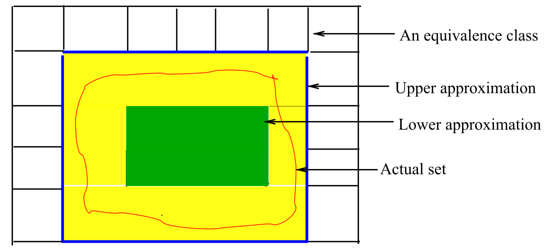

Fig. 1. Lower and upper approximations of a set in rough set theory.

Definition 3 (*Upper approximation*). For a given subset $R \subseteq A$ and subset $X \subseteq U$ (a concept X), the R upper approximation $\overline{R}X$ of set X is defined as follows,

$$\overline{R}X = \{x \in U : [x]_R \cap X \neq \emptyset\}. \tag{3}$$

$\overline{R}X$ is the set of objects of U which can be possibly classified as elements of X using the set of attributes R, where $[x]_R$ denotes an equivalence class of $Ind(R)$ which contains x (called the indiscernibility relation).

Definition 4 (*Lower approximation*). For given subsets $R \subseteq A$ and $X \subseteq U$ (a concept X), the R-lower approximation $\underline{R}X$ of set X is defined by

$$\underline{R}X = \{x \in U : [x]_R \subseteq X\}. \tag{4}$$

$\underline{R}X$ is the set of all objects from U which can be classified certainly as elements of X employing the set of attributes R.

The illustration of rough set theory is shown in Fig. 1.

Definition 5 (*Positive region*). The C-positive region of D is the set of all objects from the universe U which can be classified with certainty to classes of U/D employing attributes from C, i.e.,

$$pos_C(D) = \bigcup_{X \in U/ind(D)} \underline{C}X, \tag{5}$$

where $\underline{C}X$ denotes the lower approximation of the set X with respect to C, namely, the set of all objects from U that can be classified with certainty as elements of X based on attributes from C.

Definition 6 (*Negative region*). The C-negative region of D is the set of all objects from the universe U which for sure cannot be classified to classes of U/D employing attributes from C, namely,

$$neg_C(D) = U - \bigcup_{X \in U/ind(D)} \overline{C}X. \tag{6}$$

2.2. Discernibility matrix and discernibility function

In [4], Skowron and Rauszer proposed the notion of a discernibility matrix for storing sets of attributes which can discern all pairs of objects. Both the rows and columns of the matrix correspond to the objects. An element of the matrix is the set of all attributes that distinguish the corresponding object pair. Two objects are discernible if their values are different on at least one attribute.

Definition 7 (*Discernibility matrix*). Given an information table S, its discernibility matrix $M = (M(x,y))$ is a $|U| \times |U|$ matrix, in which the element $M(x,y)$ for an object pair (x,y) is defined by

$$m_{ij} = \{\alpha \in A | \alpha(x_i) \neq \alpha(x_j)\}, \quad for \quad i,j = 1,2,\ldots,n. \tag{7}$$

The physical meaning of the matrix element $M(x,y)$ is that objects x and y can be distinguished by any attribute in $M(x,y)$. An object pair (x,y) can be discerned if $M(x,y) \neq \emptyset$. A discernibility matrix M is a symmetric, i.e., $M(x,y) = M(y,x)$ and $M(x,x) = \emptyset$. Therefore, it is sufficient to consider only the lower triangle or the upper triangle of the matrix.

Definition 8 (*The discernibility function*). The discernibility function of a discernibility matrix is defined by

$$f(M) = \wedge\{\vee(M(x,y))|(x,y) \in U, M(x,y) \neq \emptyset\}. \tag{8}$$

The expression $\vee(M(x,y))$ is the disjunction of all attributes in $M(x,y)$, indicating that the object pair (x,y) can be distinguished by any attribute in $M(x,y)$. The expression $\wedge\{\vee(M(x,y))\}$ is the conjunction of $\vee(M(x,y))$ indicating that the family of discernible object pairs can be distinguished by a set of attributes satisfying $\wedge\{\vee(M(x,y))\}$.

Research on the discernibility matrix has received much attention. Some improved algorithms based on it were proposed in [23–27]. However, the above algorithms cannot ensure to find the minimal reduct, and a counterexample was given by Pang in [28]. Pawlak introduced the notion of a minimum discernibility matrix in [29]. Given a reduct, a minimum discernibility matrix of the reduct is a matrix whose elements are either the empty set or singleton subsets. The union of all elements in the minimum matrix is a reduct. Yao proposed a reduct construction method based on discernibility matrix simplification in [30], where he employed elementary matrix simplification operations. The elements of a minimum discernibility matrix are either the empty set or singleton subsets, and their union derives a reduct.

2.3. Incremental updating algorithm based on the improved discernibility matrix

An incremental updating algorithm for attribute reduction based on an improved discernibility matrix was proposed by Yang in [17].

Given an information system S, the element m_{ij} of the improved discernibility matrix was defined as follows,

$$\begin{cases} \{\alpha \in C : f(x_i,\alpha) \neq f(x_j,\alpha)\}, \ \textit{when}\ f(x_i,D) \neq f(x_j,D), \\ \qquad x_i, x_j \in U_1, \\ \{\alpha \in C : f(x_i,\alpha) \neq f(x_j,\alpha)\}, \ \textit{when}\ x_i \in U_1, x_j \in U_2', \\ \phi, \ \textit{else}, \end{cases} \tag{9}$$

where $U_1 = pos_C(D)$, $U_2 = U - U_1$. $U_2' = delrep(U_2)$, which denotes a set of deleting the repetitive objects in U_2.

When a new object enters the information system, we only need to update the discernibility matrix and judge whether the original reduct is the candidate set of dynamic reduct or not. If the original reduct is not one, we need to reduce attributes by the heuristic method from the core set. Although it can dynamically update the reduct, it cannot ensure to find the minimal reduct. Moreover, this algorithm cannot deal with the case of multiple objects added into the system. In order to improve the efficiency of attributes reduction and get the reliable reduction results, a new discernibility matrix based on conditional entropy was presented in [27]. It combined the advantages of discernibility matrix and conditional entropy.

3. Attribute reduction algorithm based on 0-1 integer programming

Attribute reduction not only can eliminate the irrelevant and redundant attributes but reserve the important attributes which play essential roles in the classification problem. If the reserved attributes are labeled "1" and the deleted ones are labeled "0", then, the attribute reduction problem can be converted into a 0-1 integer programming problem, which can be resolved easily.

3.1. Attribute reduction algorithm based on 0-1 integer programming

Suppose S is a consistent system. The flowchart of the attribute reduction algorithm based on 0-1 integer programming is described in Algorithm 1.

Algorithm 1

Input: Given an information system $S = \langle U, C \cup D, V, f \rangle$.
Output: Output the reduct of information system S.

1. Compute the deference $h_k(i,j)$ of the object pair (U_i,U_j) belonging to the different decision classes
$$h_k(i,j) = \begin{cases} 1, & v(i,k) \neq v(j,k) \wedge D(i) \neq D(j), \\ 0, & v(i,k) = v(j,k) \wedge D(i) \neq D(j), \end{cases} \tag{10}$$
where $v(i,k)$ denotes the value of the ith object on condition attribute C_k, and $v(j,k)$ denotes the corresponding attribute value of the jth object. The case of $h_k(i,j) = 1$ implies that the attribute C_k can distinguish the ith object from the jth object. Otherwise, it can not discern the object pair (U_i,U_j).
2. Construct the following constraints
$$\begin{cases} \sum_{k=1}^{n} h_k(i,j)x_k \geqslant 1, & \forall i,j, \\ x_k = 0,1, & k = 1,2,\ldots,n, \end{cases} \tag{11}$$
where n denotes the number of condition attributes. At least one condition attribute is necessary to discern the object pair (U_i,U_j) in a consistent system. Therefore, the condition $\sum_{k=1}^{n} h_k(i,j)x_k \geqslant 1$ is satisfied. The case of $x_k = 1$ implies that the attribute C_k is necessary and should be reserved in the final reduct. Otherwise, it is redundant and should be deleted.
3. Construct the following optimal objective function
$$\min \sum_{i=1}^{n} x_i. \tag{12}$$
We know that a reduct is any minimal subset of condition attributes, which can discern all pairs of objects from the different decision classes and owns the same classification performance as the original attribute sets. The value $\sum_{i=1}^{n} x_i$ denotes the number of condition attributes reserved in the reduct.
4. Resolve the following 0-1 integer programming
$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i, \\ \text{s.t.} \quad & \sum_{k=1}^{n} h_k(i,j)x_k \geqslant 1, \quad \forall i,j, \\ & x_k = 0,1, \quad k = 1,2,\ldots,n \end{aligned} \tag{13}$$
and achieve its optimal solution $(x_1^*, x_2^*, \ldots, x_n^*)$.
5. Construct a reduct according to the solution $(x_1^*, x_2^*, \ldots, x_n^*)$, which is composed of the condition attributes corresponding to variables $x_k^* \neq 0$ $(k = 1,2,\ldots,n)$.

3.2. The relation between the optimal solution and the reduct

In Algorithm 1, the case of $x_k = 1$ means that C_k is an important attribute and should be reserved in the reduct. Otherwise, it is a redundant attribute and should be deleted. The relation between the optimal solution and the reduct is discussed in this subsection.

Proposition 1. *The optimal value of the objective function in 0-1 integer programming (13) is equal to the number of the attributes in the reduct.*

Proof. The variables x_i $(i = 1,2,\dots,n)$ in objective function $\sum_{i=1}^{n} x_i$ satisfy the condition $x_i = 0$ or 1. The case of $x_k = 1$ means that attribute C_k is necessary and should be reserved in the reduct. Therefore, the value of objective function $\sum_{i=1}^{n} x_i$ is equal to the number of condition attributes in the reduct. □

Proposition 2. *There exists at least one solution in a 0-1 integer programming problem (13) , which implies that there exists at least one reduct in the consistent information system.*

Proof. We know that there exists at least one attribute which can discern an object pair (U_i,U_j) belonging to the deferent decision classes in a consistent information system. Hence, there exists one variable k satisfying the condition $h_k(i,j) = 1$, or it will be an inconsistent information system.

At least one variable x_k is equal to 1 if the inequality constraint $\sum_{k=1}^{n} h_k(i,j)x_k \geqslant 1$ is satisfied. At the same time, we have inequality constraint $\sum_{k=1}^{n} x_k \geqslant 1$. According to Proposition 1, the case of $x_i = 1$ $(i = 1,2,\dots,n)$ implies that all the constraints in (11) will be satisfied and the optimal value of the objective function is equal to n. Therefore, there exists a solution $(x_1,x_2,\dots,x_n)$ in the optimization problem (13). □

Proposition 3. *The solution of the integer programming problem (13) is not unique, which means the reduct is not unique in a consistent information system.*

4. An approach of constraint preprocessing

We know that a 0-1 integer programming can be resolved easily for the small-scale dataset. Otherwise, it will cause much trouble to the solution. In order to improve its speed, an approach of constraint preprocessing is presented in this section. For convenience's sake, we take the three classes case for example. Suppose there are l_1 objects labeled "1", l_2 objects labeled "2" and l_3 objects labeled "3". It gives rise to $l_1 \times (l_2 + l_3) + l_2 \times l_3$ inequality constraints in the optimization problem (13). So the number of constraints will increase greatly with the increasing of the objects so that it causes much trouble to the solution. Therefore, we must firstly simplify the constraints before resolving it.

Proposition 4. *Suppose there exists one inequality constraint $x_k \geqslant 1$ $(k = 1,2,\dots,n)$ in (11) . Then, condition attribute C_k $(k = 1,2, \dots,n)$ should be reserved in the reduct.*

Proof. Suppose there exists one inequality constraint $x_k \geqslant 1$ $(k = 1,2,\dots,n)$ in (11), which means condition attribute C_k $(k = 1,2,\dots,n)$ is the only one which can discern a pair of objects (U_i,U_j). Then it should be reserved in the final reduct. □

Proposition 5. *Suppose there exists one constraint $x_k \geqslant 1$ $(k = 1,2, \dots,n)$ in (11), then the constraints including variable x_k $(k = 1,2,\dots, n)$ are redundant to the solution of optimization problem (13).*

Proof. Suppose there exists one inequality constraint $x_1 \geqslant 1$. According to Proposition 4, the condition attribute C_1 is necessary and should be reserved in the reduct. If there exists another constraint $x_1 + x_2 \geqslant 1$ in (11), which means both attributes C_1 and C_2 can discern an object pair (U_i,U_j). Then, attribute C_2 is redundant and the inequality constraint $x_1 + x_2 \geqslant 1$ can be deleted. □

Table 1
Information system table.

U	C_1	C_2	C_3	C_4	D
1	1	1	1	1	1
2	1	0	1	0	1
3	0	1	1	0	2
4	1	0	1	1	2
5	0	1	1	1	3
6	0	0	1	1	3

Example 1. Given an information system as Table 1.

According to Algorithm 1, we can construct a 0-1 integer programming including $2 \times (2 + 2) + 2 \times 2 = 12$ inequality constraints. After constraint preprocessing, the constructed optimization problem can be simplified to the following formulation (14),

$$\begin{array}{ll} \min & \sum_{i=1}^{4} x_i, \\ \text{s.t.} & x_1 \geqslant 1, \\ & x_2 \geqslant 1, \\ & x_4 \geqslant 1, \\ & x_1, x_2, x_3, x_4 = 0, 1. \end{array} \tag{14}$$

It includes only three inequality constraints. However, there are twelve inequality constraints in the original optimization problem, and the number of constraints reduces 66.7%. Hence, we can resolve it easily and achieve its solution $(x_1,x_2,x_3,x_4) = (1,1,0,1)$. According to the above propositions, we can find the reduct of Table 1 as follows,

$$R_1 = \{C_1, C_2, C_4\}. \tag{15}$$

5. Dynamic attribute reduction based on 0-1 integer programming

Algorithm 1 only suits the static information system. We know that an information system changes over time, and the database has not been set up from the beginning. The new objects enter the system continuously, so the new attribute reduction algorithm for the dynamic information system should be constructed.

For the convenience of description, we also take three decision classes for example. Suppose l'_1 newly increased objects labeled "1", l'_2 newly increased objects labeled "2" and l'_3 newly increased objects labeled "3" are added into the system. It will increase $l'_1 \times (l_2 + l_3) + l'_2 \times (l_1 + l_3) + l'_3 \times (l_1 + l_2)$ inequality constraints. How to dynamically update the reduct is the task confronting us. Motivated by the above studies, a dynamic attribute reduction algorithm is presented in the following.

Algorithm 2

Input: Given an information system $S = S_1 \cup S_2$, where S_1 denotes an original information system and S_2 denotes the newly increased part, respectively. $R_1 = \{C_1,C_2,\dots,C_k\}$ is a reduct of the original information system S_1.

Output: Output the reduct of the dynamic information system S.

1. Construct constraints for the newly increased objects in the dynamic system, and it produces $l'_1 \times (l_2 + l_3) + l'_2 \times (l_1 + l_3) + l'_3 \times (l_1 + l_2)$ constraints in all.

2. Preprocess the newly increased constraints according to Propositions 4 and 5. Suppose the constraints after preprocessing are signed as T.
3. Resolve the following 0-1 integer programming,

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i, \\ \text{s.t.} \quad & T \geqslant 1, \\ & x_i = 0, 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{16}$$

Suppose its solution is $x_{k+1} = 1, \ldots, x_{k+h} = 1$, and other variables are equal to 0. According to the relation between the optimal solution and the reduct, we can achieve the reduct $R_2 = \{C_{k+1}, C_{k+2}, \ldots, C_{k+h}\}$ easily.

4. Update the reduct of the dynamic system S,

$$R = R_1 \cup R_2 = \{C_1, \ldots, C_k, C_{k+1}, \ldots, C_{k+h}\}. \tag{17}$$

Proposition 6. *The reduct R of the dynamic system is composed of R_1 and R_2, i.e., $R = R_1 \cup R_2$, where R_1 is a reduct of S_1 and R_2 is the reduct of S_2, respectively.*

Proof. Suppose system S is comprised of two parts, i.e., $S = S_1 \cup S_2$. Suppose $R_1 = \{C_1, C_2, \ldots, C_k\}$ is the reduct of S_1. In the solution of (16), variables x_i ($i = 1, 2, \ldots, k$) are equal to 0 since constraints T do not include variable x_i ($i = 1, 2, \ldots, k$).

The case of $T = \emptyset$ means that the original reduct R_1 can also discern the newly increased objects. Otherwise, it cannot discern them. Moreover $R_1 \cap R_2 = \emptyset$. Therefore, they are all necessary for the dynamic information system. We can get $R = R_1 \cup R_2$. □

Example 2. Given an information system as Table 1, the attribute values of the three newly increased objects are shown in Table 2.

When an object U_7 labeled "1" is compared with all the objects labeled "2" and "3" in S_1, it will increase the following four inequality constraints,

$$\begin{aligned} & x_1 \geqslant 1, \\ & x_2 + x_4 \geqslant 1, \\ & x_1 + x_4 \geqslant 1, \\ & x_1 + x_2 + x_4 \geqslant 1. \end{aligned} \tag{18}$$

Similarly, it will increase the following four inequality constraints as the object U_8 enters the system,

$$\begin{aligned} & x_1 + x_3 + x_4 \geqslant 1, \\ & x_1 + x_2 + x_3 \geqslant 1, \\ & x_3 + x_4 \geqslant 1, \\ & x_2 + x_3 + x_4 \geqslant 1. \end{aligned} \tag{19}$$

For the object U_9, it will increase the following four inequality constraints,

$$\begin{aligned} & x_3 + x_4 \geqslant 1, \\ & x_2 + x_3 \geqslant 1, \\ & x_1 + x_3 \geqslant 1, \\ & x_2 + x_3 + x_4 \geqslant 1. \end{aligned} \tag{20}$$

In addition, the following three constraints are necessary to discern the newly increased object pairs (U_7, U_8), (U_7, U_9) and (U_8, U_9).

$$\begin{aligned} & x_1 + x_3 \geqslant 1, \\ & x_3 \geqslant 1, \\ & x_1 \geqslant 1. \end{aligned} \tag{21}$$

Table 2
The attribute values of the newly increased objects.

U	C_1	C_2	C_3	C_4	D
7	1	1	1	0	1
8	0	1	0	0	2
9	1	1	0	0	3

Finally, it increases fifteen inequality constraints in all as three objects enter the system. However, after constraint preprocessing, only one inequality constraint $x_3 \geqslant 1$ is necessary, and other constraints are redundant. Hence, we only need to resolve the following optimization problem,

$$\begin{aligned} \min \quad & \sum_{i=1}^{4} x_i, \\ \text{s.t.} \quad & x_3 \geqslant 1, \\ & x_1, x_2, x_3, x_4 = 0, 1. \end{aligned} \tag{22}$$

We can obtain its solution $(x_1, x_2, x_3, x_4) = (0, 0, 1, 0)$. Then its corresponding reduct is $R_2 = \{C_3\}$. The final dynamic reduct is

$$R = R1 \cup R_2 = \{C_1, C_2, C_4\} \cup \{C_3\} = \{C_1, C_2, C_3, C_4\}.$$

6. Algorithm complexity analysis

In [16], Wang proposed an attribute reduction algorithm based on attributes significance in the discernibility matrix. Its time complexity was $O(|C|^2|U|^2)$ and the space complexity was $O(|C||U|^2)$ (where $|C|$ denotes the number of attributes and $|U|$ denotes the cardinality of universe, respectively).

In the incremental updating algorithm based on improved discernibility matrix [17], the space complexity was $O(|C||U|^2)$ in the worst case, namely, there was not an inconsistent object when the new objects entered the information system. Its time complexity was $O(|C| \times |U_1|) \times (|U_1| + |U_2'|)$ (where $|C|$ denotes the number of attributes in reduct, $|U_1|$ denotes the cardinality of positive region, $|U_2'|$ denotes the cardinality of negative region after deleting the repetitive objects), and time complexity was $O(|C||U|^2)$ in the worst case [17]. In addition, it only discussed the case of one object entering the system. However, multiple objects are added into the system simultaneity. We need to modify the discernibility matrix repeatedly so that it reduces the efficiency of algorithm.

In the proposed algorithm, we take the three decision classes case for example. Suppose the number of objects in three classes is approximately equal to $|U|/3$, and its space complexity and time complexity are both $O(|C||U|^2/3)$. In the process of dynamic reduction, although multiple objects enter the information simultaneously, only a few constraints are reserved after constraint preprocessing. Therefore, the 0-1 programming problem can be resolved easily.

7. Numerical experiments

The effectiveness of the proposed dynamic reduction algorithm was tested on a collection of twelve benchmark datasets from UCI machine learning repository[1]; Breast cancer$_1$, Pima, Heart, Ionosphere, Sonar, Spectf, Valley, Eighthr, Breast cancer$_2$, Iris, Wine, and Hayes-Roth. The first nine datasets are two decision classes problems, and the rest datasets are three decision classes problems. Before employing rough set theory, the real value data must be discrete, so K-means discrete method is employed in our experiments. The number of clusters is equal to the number of classes, i.e., the

[1] http://archive.ics.uci.edu/ml/datasets.html

Table 3
The performance comparisons of before and after constraint preprocessing in the original information system.

Data	Class	Objects	Attr	$Objects_1$	$Const_1$	$Const_2$	$Prop_1$ (%)
$Breast_1$	2	683	9	546	69,972	38,829	48.34
Pima	2	768	8	615	85,814	13,465	15.69
Heart	2	270	13	220	11,904	11,578	97.3
Ionosphere	2	351	34	300	20,099	18,706	93.0
Sonar	2	208	60	168	7040	6114	86.8
Spectf	2	269	44	200	2775	510	18.38
Valley	2	864	100	541	73,080	5030	6.80
Eighthr	2	1163	72	1027	111,192	10,744	9.70
$Breast_2$	2	194	33	144	3959	1467	36.80
Iris	3	150	4	120	2700	1946	72.07
Wine	3	178	13	118	3081	1857	60.3
Hayes-roth	3	132	5	67	1456	1430	98.2

Table 4
The performance comparisons of before and after constraint preprocessing in the dynamic information system.

Data	$Objects_2$	$Const_3$	$Const_4$	$Prop_2$ (%)
$Breast_1$	137	36,144	0	0
Pima	153	48,186	0	0
Heart	50	6096	7	0.1
Ionosphere	51	8251	31	0.4
Sonar	40	3727	10	0.3
Spectf	69	8995	55	0.6
Valley	323	113,463	1	0
Eighthr	136	21,288	126	0.59
$Breast_2$	50	2849	196	6.87
Iris	60	4800	41	0.85
Wine	60	3940	92	2.3
Hayes-roth	65	4205	0	0

number of clusters is equal to 2 when the dataset is two classes problem. Our Algorithm and Pawlak reduction are implemented in Matlab 7.0 environment.

The comparison results of the original system are summarized in Table 3, where "Class" denotes the number of classes, "Objects" stands for the number of objects, "Attr" denotes the number of attributes, "$Objects_1$" denotes the number of objects in the original system, "$Const_1$" stands for the number of constraints in the original system, "$Const_2$" denotes the number of constraints after preprocessing in the original system, and "$Prop_1$" represents the proportion of constraints after and before preprocessing in the original system.

The performance comparisons of constraint preprocessing are shown in Table 4, where "$Objects_2$" denotes the number of newly increased objects, "$Const_3$" stands for the amount of newly increased constraints, "$Const_4$" stands for the number of newly increased constraints after preprocessing, and "$Prop_2$" denotes the fraction of constraints after and before preprocessing. It shows that the number of constraints increases less although multiple newly increased objects enter the original information system, namely, we can achieve the dynamic reduct just by updating a few constraints, instead of treating the changed information system as a new one and finding the reduct again like rough set reduction algorithm does.

The reduction results of the dynamic system by our algorithm are shown in Table 5, where "$Attr_1$" stands for the cardinality of reduct in the original information system, "MN" denotes the cardinality of minimal reduct. "$Time_1$" denotes the reduction time of the original system, and "$Time_2$" stands for the increasing time of the dynamic system. Meanwhile, we also give the reduction result by Pawlak algorithm. From the Table 5, we can find that the dynamic reduct changes little when multiple new objects enter the

Table 5
The result comparisons of our algorithm with Pawlak reduction.

Data	Our algorithm					Pawlak
	Attr	$Attr_1$	MN	$Time_1$	$Time_2$	MN
$Breast_1$	9	9	9	168.66	29.15	9
Pima	8	6	6	310.78	28.30	6
Heart	13	10	11	46.11	0.937	11
Ionosphere	34	20	23	209.9	5.781	23
Sonar	60	21	24	36.61	1.515	24
Spectf	44	10	11	17.391	3.172	11
Valley	100	7	9	2765	2326	8
Eighthr	72	21	22	4745	100	22
$Breast_2$	33	15	19	7.782	0.984	19
Iris	4	3	4	1.11	0.08	4
Wine	13	9	10	2.55	0.31	10
Hayes-roth	5	5	5	1.02	0.07	5

Table 6
The performance comparisons of different algorithms.

Data	Attr	Our	Pawlak	IUAARI	IUAARS
$Breast_1$	9	9	9	6	6
Pima	8	6	6	5	5
Car evaluation	6	6	6	5	5
Chess	36	29	29	30	29
Mushroom	22	4	3	5	4

information system constantly. That is to say, we can find the dynamic reduct only by modifying the reduct of the original information system. Moreover, we can find that the reduction result obtained by our algorithm is nearly the same as Pawlak reduction, which verifies the validity of our proposed algorithm.

In order to further demonstrate the feasibility and validity of our algorithm, we also compare our algorithm with other two incremental algorithms, i.e., IUAARI [18] and IUAARS [31]. Their performance comparisons are shown in Table 6, where the reduction results of IUAARI and IUAARS came from [31].

From Table 6, we can learn that our proposed algorithm yields nearly the same reduction result as Pawlak reduction, but it is not the case for algorithms IUAARI and IUAARS, which shows the validity of our proposed algorithm. In addition, from Table 5, we can find that our proposed algorithm costs more time when the scale of data exceeds 800, and its main reason lies in the limitations of function "bintprog" used in matlab. That is to say, there is no more suitable tool to resolve the large-scale 0-1 inter programming problems at present.

8. Conclusion

Aiming at the case of multiple objects constantly entering the information system, a dynamic attribute reduction algorithm based on 0-1 programming is proposed in this paper. Firstly, we construct a 0-1 integer programming for the original information system S_1 and achieve its reduct R_1. Secondly, we construct a 0-1 integer programming for the newly increased objects and obtain its reduct R_2. Finally, we achieve the reduct $R = R_1 \cup R_2$ of the dynamic information system S. Before resolving the 0-1 integer programming, we can preprocess the constraints so that it can reduce the number of constraints and bring convenience for the solution. The proposed dynamic reduction algorithm just updates the old reduct set based on the newly increased objects, instead of treating the changed information table as a new one and finding the reduct. Therefore, the efficiency of attribute reduction is improved greatly. Certainly, researches on the dynamic attribute reduction in concept lattices [32], fuzzy concept lattices [33] and vague set [34] and their relations are our future works.

Acknowledgements

This work was supported by Chinese Universities Scientific Found (No. 2010JS043) and National Natural Science Foundation of China (No. 10771213). The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have greatly improved the presentation.

References

[1] Z. Pawlak, Rough sets, International Journal of Information and Computer Sciences 11 (5) (1982) 341–356.
[2] Z. Pawlak, Rough sets and intelligent data analysis, Information Sciences 147 (2002) 1–12.
[3] Roman W. Swiniarski, Andrzej Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (2003) 833–849.
[4] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: Slowinski (Ed.), Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic, Dordrecht, 1992, pp. 331–362.
[5] G. Wang, Rough sets theory and knowledge acquisition, Xi'an Jiaotong University Press, 2001.
[6] G. Wang, Decision table reduction based on conditional information entropy, Chinese Journal of computers (7) (2002) 759–766.
[7] S.K.M. Wong, W. Ziarko, On optimal decision rules in decision tables, Bulletin of Polish Academy of Sciences 33 (1985) 693–696.
[8] S. Tan, H. Xu, J. Gu, Efficient algorithm for attributes reduction problem, International Journal of Innovative Computing, Information and Control1 (4) (2005) 767–777.
[9] D. Miao, G. Hu, A heuristic algorithm for reduction of knowledge, Journal of computer research and development (6) (1999) 681–684.
[10] F. Hu, G.Y. Wang, Quick reduction algorithm based on attribute order, Chinese Journal of Computers 30 (2007) 1429–1435.
[11] W. Wu, M. Zhang, H.Z. Li, J.S. Mi, Knowledge reduction in random information systems via Dempster–Shafer theory of evidence, Information Sciences 174 (3) (2005) 143–164.
[12] K. Zhao, J. Wang, A reduction algorithm meeting users' requirements, Journal of Computer Science and Technology 17 (2002) 578–593.
[13] R. Jensen, Q. Shen, A. Tuson, Finding rough set reducts with SAT, in: Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI, vol. 3641, 2005, pp. 194–203.
[14] D. Liu, T. Li, D. Ruan, W. Zou, An incremental approach for inducing knowledge from dynamic information systems, Fundamenta Informaticae 94 (2) (2009) 245–260.
[15] Z. Liu, An incremental arithmetic for the smallest reduction of attributes, Chinese Journal of Electronics 27 (11) (1999) 96–98.
[16] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix: The ordered attributes method, Journal of Computer Science and Technology 16 (6) (2001) 489–504.
[17] M. Yang, An incremental updating algorithm of the computation of a core based on the improved discernibility matrix, Chinese Journal of Computers 29 (3) (2006) 407–413.
[18] M. Yang, An incremental updating algorithm for attribute reduction based on improved discernibility matrix, Chinese Journal of Computers 30 (5) (2007) 815–822.
[19] N. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, Computational Intelligence 11 (2) (1995) 357–370.
[20] C. Chan, A rough set approach to attribute generalization in data mining, Information Science 107 (1–4) (1998) 177–194.
[21] T. Li, D. Ruan, et al., A rough sets based characteristic relation approach for dynamic attribute generalization in data mining, Knowledge-Based Systems 20 (5) (2007) 485–494.
[22] S. Guan, S. Li, Incremental learning with respect to new incoming input attributes, Neural Processing Letters 14 (3) (2001) 241–260.
[23] X. Hu, N. Cercone, Learning in relational databases: A rough set approach, Computational Intelligence 11 (2) (1995) 323–338.
[24] D. Ye, Z. Chen, A new discernibility matrix and the computation of a core, Acta Electronica Sinica 30 (7) (2002) 1086–1088.
[25] Q. Liu, Rough set and Rough Reasoning, Science Press, 2001.
[26] X. Gao, J. Ding, An attribute reduction algorithm based on simple discernibility matrix, Systems Engineering Theory and Practice (6) (2006) 101–107.
[27] X. Tan, An incremental conditional attribute reduction algorithm based on improved discernibility matrix, System Engineering Theory and Practice 30 (9) (2002) 1684–1694.
[28] Y. Pang, K. Liu, The simplify algorithm of calculating the reduced differential matrix is untenable, Systems Engineering Theory and Practice (2) (2004) 142–144.
[29] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Dordrecht, MA, 1991.
[30] Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, Information Sciences 179 (7) (2009) 867–882.
[31] L. Guan, G. Wang, An incremental updating algorithm for attribute reduction set of decision tables, Journal of Frontiers of Computer Science and Technology 4 (5) (2010) 436–444.
[32] L. Wei, J.J. Qi, Relation between concept lattice reduction and rough set reduction, Knowledge-Based Systems 23 (8) (2010) 934–938.
[33] L. Li, J. Zhang, Attribute reduction in fuzzy concept lattices based on the T implication, Knowledge-Based Systems 23 (6) (2010) 497–503.
[34] L. Feng, T. Li, D. Ruan, S. Gou, A vague-rough set approach for uncertain knowledge acquisition, Knowledge-Based Systems 24 (6) (2011) 837–843.