An incremental attribute reduction algorithm based on improved binary distinguishable matrix

Weibing Feng¹, Yunyun Ma²*

¹College of science, Xi’an University of Science and Technology, Xi’an, Shaanxi, ZIP code:710000, China
²Applied mathematics, Xi’an University of Science and Technology, Xi’an, Shaanxi, ZIP code:710000, China
*Corresponding author’s e-mail: 894445231@qq.com

Abstract. The essence of attribute reduction is to delete unnecessary or unimportant attributes in the knowledge base while keeping the classification ability of the information system. Most of the existing reduction algorithms are designed for static decision tables. However, when the data set changes dynamically, the time consumption of static reduction algorithms is huge and the efficiency is not high. At present, there are few studies on the reduction algorithms of dynamic decision tables. In this paper, an existing binary distinguishable matrix reduction algorithm is optimized by using the method of binary distinguishable matrix when the objects are dynamically increased. In the case of static data set, the algorithm is improved to deal with dynamic data set by judging the relationship between new data and original data set. Examples show that the algorithm is simple, efficient, accurate, and has certain practicability and completeness.

1. Introduction

Rough set theory [1, 2] is a mathematical tool proposed by Polish scientist Pawlak in 1982 to deal with fuzzy, imprecise knowledge and incomplete information. Rough set theory does not need to provide any required prior knowledge beyond the data set, and has been widely used in machine learning, decision analysis, process control, pattern recognition, data mining, and other fields. Attribute reduction is one of the core contents of rough set theory. How to quickly solve all attribute reductions of decision tables has been proved to be an NP-hard problem. At present, the main methods for reducing the dimensionality of big data based on rough set theory include the method based on positive domains proposed by Pawlak [1], the method based on information entropy proposed by Miao [3] and the method based on distinct matrix proposed by Skowron [4]. In the past one decade, researchers have proposed many incremental attribute reduction algorithms. For instance, Zhi and Miao [11] proposed an attribute reduction algorithm based on binary distinguishable matrix, while the binary distinguishable matrices created by the algorithm result in the generation of some unnecessary element terms. Meanwhile, the algorithm cannot identify the optimal reduction form when there are multiple reductions. An incremental attribute reduction method of decision tables based on improved distinguishable matrix proposed by Liu et al. [5] takes the positive domain as the reduction criterion, and obtains attribute reduction by using the idea of greedy algorithm. Nevertheless, the new method can not effectively calculate all attribute reductions, and does not give an incremental algorithm when attributes and samples increase simultaneously. Based the idea, Shen et al. [6] proposed an incremental
attribute reduction algorithm based on the relatively positive domain. This algorithm only provides a reduction algorithm when a single object is added, and does not analyze the dynamic processing when multiple samples are added. Ge [7] proposed an incremental reduction algorithm based on a relational matrix decision table, while the algorithm only discusses the case when the objects are added, not containing the case when the attributes are added. Cheng [8] proposed an incremental reduction algorithm based on the relationship change, and obtain the final reduction on the basis of knowledge granularity, but the knowledge dimension and the time consumption are relatively large. Recently, based on rough set theory, Gao et al. [9] proposed an incremental attribute reduction algorithm, i.e. A complex problem is divided into several simple small problems, and the final attribute reduction is obtained by using the divided reduction set identification matrix. However, the time consumption is large in calculating the difference function of the identification matrix. Besides, under the framework of heuristic algorithm and relative differentiation algorithm, Yang et al. [10] analyzed and compared the time complexity of the non-incremental case and the incremental case when a single sample and multiple samples increased, and demonstrated that the incremental algorithm has less time consumption and higher efficiency.

In this paper, an improved incremental reduction algorithm is proposed for the common problems of the existing binary algorithms, such as not being able to adapt to the common problems of dynamic databases and high time and space complexity. Moreover, the accuracy of the proposed algorithm is proved by two typical instances.

2. Basic concepts
Definition 1 [12] Information systems(S) are usually described in form as a quaternion S = \( (U, A, V, f) \), where \( U = \{x_1, x_2, \ldots, x_n\} \) represents the set of all samples (objects), \( A = \{a_1, a_2, \ldots, a_n\} \) represents the set of attributes (features), \( V = \bigcup_{a \in A} V_a \) represents the set of attribute values, and \( f \) represents an information function, which establishes a corresponding mapping relationship \( U \times A \rightarrow V \) among samples, attributes and attribute values, and it can be described as \( \forall x \in U, \forall a \in A, \exists f(x, a) \in V_a \). The attribute set is divided into two parts: conditional attributes \( C \) and decision attributes \( D \), that is \( A = C \cup D \), and \( C \cap D = \emptyset, D \neq \emptyset \). At this time, the information system is called as a decision system.

Definition 2 [12] The quaternion \( S = (U, A, V, f) \) represents a decision information system. If \( P \subseteq A \), the equivalent relationship determined by \( P \) is:
\[
\text{IND}(P) = \{(x, y) \in U \times U | \forall a \in P, f(x, a) = f(y, a)\}
\]

Here, the relationship is recorded as \( U/\text{IND}(P) \) or \( U/P \). If \( U/\text{IND}(P) = \{x_1, x_2, \ldots, x_k\} \), where \( x_i \in U, x_j \in U, x_i \cap x_j = \emptyset (i \neq j) \), it indicates that the universe \( U \) is divided by \( P \) into \( k \) divisions, where each division information particle \( x_i \) is an equivalent class of any sample \( x \), and is expressed as \( \left[x \right]_{\text{IND}(P)} = \{y | (x, y) \in \text{IND}(P)\} \).

Definition 3 [12] In the decision information system \( s = (U, C \cup D, V, f) \), \( B \subseteq C \), the positive region of \( D \) is recorded as \( \text{POS}_B(D) \), and it is defined as:
\[
\text{POS}_B(D) = \bigcup_{X \in D/B} P(X).
\]

Definition 4 [12] Assuming \( R \) is an equivalence relation cluster, then \( r \in R \), if \( \text{IND}(R) = \text{IND}(R - \{r\}) \), then attribute \( r \) is unnecessary to \( R \); otherwise, attribute \( r \) is necessary. If for every attribute of \( r \in R \) is necessary, it is said \( R \) to be independent; otherwise it is said \( R \) to be dependent.

Definition 5 [12] In a decision information system \( s = (U, C \cup D, V, f) \), \( a \in C \), if
In the decision table $S = (U, C \cup D, V, f)$, $\forall B \subseteq C$, $\forall x \in U$, $U/D = \{D_1, D_2, \ldots, D_k\}$, defines

$$
\mu_B(x) = \left( \begin{array}{c}
\frac{|D_1 \cap \{x\}|}{|x|}, \frac{|D_2 \cap \{x\}|}{|x|}, \ldots, \frac{|D_k \cap \{x\}|}{|x|}
\end{array} \right),
$$

then called $\mu_B(x)$ is the probability distribution function of $B$ on $D$.

Definition 6[13] In the decision table $S = (U, C \cup D, V, f)$, $U/C = \{x_1', x_2', \ldots, x_m'\}$, and call $s' = (U', C \cup D, V, f)$ is the decision table a simplified decision table.

Definition 7[13] In the decision table $S = (U, C \cup D, V, f)$, $U/C = \{x_1', x_2', \ldots, x_m'\}$, and call $s' = (U', C \cup D, V, f)$ is a simplified decision table, the simplified binary difference matrix is defined as: $m = (m(i, j), k)$, and its elements are defined as follows:

$$
m = (m(i, j), k) = \begin{cases} 1, & C_k \in C, f(x_i', C_k) \neq f(x_j', C_k) \land \mu_{C_k}(x_i') \neq \mu_{C_k}(x_j') \\ 0, & \text{else} \end{cases}
$$

Definition 8[12] Assuming $M' = (m(i, j), k)$ is simplified binary difference matrix of decision table $s = (U, C \cup D, V, f)$, $B \subseteq C$ if $B$ satisfies: (1) in a sub-matrix composed of columns corresponding to all attributes in $B$, the number of rows that are not all 0 is equal to the number of rows with incomplete 0; (2) for $\forall B' \subseteq B$ are not satisfied (1), it is said that $B$ to be an attribute reduction of $D$.

3.2. Simplified method of binary distinguishable matrix

In the rough set theory, the general method of calculating the equivalence class of the decision table is: for all objects in the universe of discourse $U$, compare them one by one to calculate whether their values are equal under the conditional attribute $\forall c_i \in C$, and finally according to whether their decision attributes are equal, they are classified as an equivalent class. The time complexity of the traditional algorithm is $O(\frac{|U|^2|C|}$ and the efficiency is low. It can be seen that simplifying the calculation of equivalent classes of decision tables is the first step to improve the efficiency of algorithm.

By using pointers to store objects, those duplicate objects in the universe of discourse $U$ are deleted, thereby greatly reducing the time complexity of solving equivalent classes. The specific steps are as follows: firstly, record the maximum and minimum values of each $c_i$, by the method of pointers to store objects, assign objects to queues by pointers, and then collect pointers by modifying pointers to
recombine object lists to obtain equivalent class of universe of discourse \( U \) under each \( c_i \), and get a simplified decision table \( U' = U'_{\text{pos}} \cup U'_{\text{neg}} \). In the worst case, the time complexity of the simplified decision table is \( O(|U||C| + \sum_{i=1}^n (M_i - m_i + 1)) \). When the value range is not intensively scattered, the time complexity is \( O(|U||C|) \) and the space complexity is \( O(|U|) \).

### 3.3. An improved binary distinguishable matrix incremental reduction algorithm

#### Definition 10

In a binary distinguishable matrix BM, the attribute frequency function of a conditional attribute is defined as follows:

\[
f(a_k) = \sum_{i=1}^{N} \left( m(i, j) \right) \left( \sum_{i=1}^{C'} m(i, j) \right)^{-1}, \quad a_k \in C'
\]

Where \( N \) is the number of element items in the binary distinguishable matrix BM \( (0 \leq N \leq |U'|(|U'| - 1/2)) \), and \( C' \) is the number of medium conditional attributes.

According to the definition of binary distinguishable matrix:

**Theorem 1.** If there is only one 1 in a row of the binary distinguishable matrix and the other are all 0, the attribute corresponding to the element with the attribute value 1 in this row is the only attribute that can distinguish this pair of objects, then the attribute is indispensable, that is the attribute is a nuclear attribute.

The conditional attribute frequency function in definition 10 is extended from the definition of traditional attribute importance based on difference matrix. The reason why an attribute is important to the system is that it plays a certain role in distinguishing some objects. If it is necessary to recalculate the remaining attributes.

Most of the existing attribute reduction algorithms based on binary distinguishable matrix need to recalculate all the existing data when the database changes dynamically, which results in high time and space complexity. This paper proposes an improved binary distinguishing matrix attribute reduction algorithm based on the simplified decision table. Under the condition that a simplified decision table is obtained, the algorithm uses a definition to calculate the distinguishable matrix of the simplified decision table, obtains a kernel from the matrix, determines the type of the new object, and uses the attribute frequency function to judge the importance of attributes to get the final reduction. The existing binary-based distinguishable algorithms are highly efficient in dealing with static database data, but when the database changes dynamically, the algorithm needs to recalculate all known data. At this time, the time consumption is huge, resulting in a large amount of space waste. The improved algorithm 1 can effectively avoid the secondary operations and greatly improve the efficiency.

Under the condition that the simplified decision table is obtained, the distinguishable matrix of the simplified decision table is calculated by definition, and the kernel is obtained from the matrix. The final reduction is obtained by judging the type of the added object and the importance of the attribute by using the attribute frequency function.

#### Algorithm 1: An improved binary distinguishable incremental reduction algorithm

```
Input: Information system \( S = (U, A, V, f) \), where \( U \) is universe of discourse, \( A \) is attribute set written as \( A = C \cup D, \; C \cap D = \phi, \; C \) is the set of conditional attributes, and \( D \) is the set of decision attributes.
Import: attribute reduction.
Step 1: \( Red = \phi, \; Core = \phi \);
Step 2: To obtain the simplified decision table \( S' = (U', A, V, f) \) by calculating \( U/A \) based on the idea of simplified decision table;
Step 3: Let the number of new objects is \( r \geq 0 \). If \( r = 0 \), go to Step 5, otherwise go to Step 4;
```
Step 4: Assume the new objects are \( \bar{x} \).

Step 4.1: if \( \exists x_i \in U'_\neg \land \forall C_j \in C \Rightarrow f(x_i, C_j) = f(x_j, C_j) \), \( U' \) and \( \text{Red}(C) \) maintain constant;

Step 4.2: if \( \exists x_i \in U'_\pos \land \forall C_j \in C \Rightarrow f(x_i, C_j) = f(x_i, C_j) \land f(x_i, D) = f(x_i, D) \), \( U' \) and \( \text{Red}(C) \) maintain constant;

Step 4.3: if \( \exists x_i \in U'_\pos \land \forall C_j \in C \Rightarrow f(x_i, C_j) = f(x_i, C_j) \land f(x_i, D) \neq f(x_i, D) \)

Step 4.3.1: \( \forall x_i \in U'_\neg, \ \text{BM} \ x_i \ x_i \ \text{BM} \ x_i \ \forall x_k \in U'_\pos \)

Step 4.3.2: \( U'_\pos = U'_\pos - \{ x_i \}, \ U'_\neg = U'_\neg + \{ x_i \} \);

Step 4.4: if \( \exists x_i \in U'_\pos \land \forall C_j \in C \Rightarrow f(x_i, C_j) \neq f(x_i, C_j) \)

Step 4.4.1: Add \( \bar{x} \) and \( \forall x_k \in U' \) the row corresponding to the matrix element generated by \( \text{BM} \);

Step 4.4.2: \( U'_\pos = U'_\pos \cup \{ \bar{x} \} \);

Step 5: Delete all rows of 1 and 0 according to definition 8 to obtain the distinguishable matrix \( \text{BM} \);

Step 6: According to definition 1 to obtain Core, \( \text{Red} = \text{Core} \);

Step 7: \( \text{BM} \ m((i, j), \ k) \ U' \)

Step 8: \( \text{BM} \ f(a_i), (i = 1, 2, \ldots, |C - \text{Red}|) \)

Step 9: while \( (\text{BM} \neq \varnothing) \)

\{ 

Step 9.1: \( a = \max \{ f(a_i) \} \ \ \max \{ f(a_i) \} \ a_i \ a = \max \text{DAGL}(S') \)

\( S' = \left( U', (C - \text{Red}) \cup D, V, f \right) \)

Step 9.2: \( \text{Red} = \text{Red} \cup a \)

Step 9.3: \( \text{BM} \ a \ m((i, j), \ k) \ a \ a \ m((i, j), \ k) \ U'' \)

Step 9.4: Based on definition 10, calculate \( f(a_i) = 0, (i = 1, 2, \ldots, |C - \text{Red}|) \) in \( \text{BM} \).

\} // end 

Step 10: for \( j = 1 \) to \( |\text{Red} - \text{Core}| \) do

\[ \text{if } \left( \text{POS}_{C'}(D) = \text{POS}_{\text{Red}-a_j}(D) \right) \]

then \( \text{Red} = \text{Red} - \{ a_j \} \);

Step 11: Import Red;

Step 12: End.

The main time consumption of Algorithm 1 is in the process of calculating the binary distinguishable matrix. In the worst case, the complexity is \( \max \left\{ O(|U| |C|), O\left( |U|^2 |C|^2 \right) \right\} \) and the space complexity is \( O\left( |U|^2 |C| \right) \). When more than one attribute that meets the highest attribute frequency appears in this algorithm, the following algorithm 2 uses positive domain judgment to accurately add the attributes with high importance to the reduction set, thereby avoiding a large number of calculations caused by blindly adding attributes.
According to Algorithm 1, we can obtain:

\[ x_\alpha, \{\} \]

\[ \text{Case analysis} \]

Example 1. Given decision table \( U = \{x_1, x_2, \ldots, x_9\} \), conditional attribute \( C = \{a, b, c, d, e\} \), decision attribute \( D \). According to Algorithm 1, we can obtain \( U_{pos} = \{x_1, x_2\} \), \( U_{neg} = \{x_3, x_4, x_5, x_6, x_7, x_8, x_9\} \), \( U/C = \{\{1, 10\}, \{2, 9\}, \{3, 5\}, \{4, 13\}, \{6, 11\}, \{7, 16\}, \{8, 15\}, \{12, 14\}\} \), \( U/D = \{\{1, 7, 10, 16\}, \{2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15\}\} \), and get following simplified reduction of decision table:

| \( U' \) | a | b | c | d | E | D | \( \mu_\alpha(*) \) |
|-----|----|----|----|----|----|----|------------------|
| \( x_1 \) | 1  | 0  | 0  | 1  | 1  | 1  | (1/2, 0)          |
| \( x_2 \) | 0  | 1  | 0  | 1  | 1  | 0  | (0, 1/6)          |
| \( x_3 \) | 0  | 0  | 1  | 1  | 0  | 0  | (0, 1/6)          |
| \( x_4 \) | 1  | 0  | 1  | 1  | 0  | 0  | (0, 1/6)          |
| \( x_5 \) | 1  | 0  | 0  | 0  | 0  | 0  | (0, 1/6)          |
| \( x_6 \) | 1  | 1  | 1  | 1  | 0  | 0  | (0, 1/6)          |
| \( x_7 \) | 0  | 1  | 1  | 1  | 0  | 0  | (0, 1/6)          |
| \( x_8 \) | 0  | 1  | 1  | 1  | 0  | 0  | (0, 1/6)          |

Its binary distinguishable \( BM \) is:

| \( m(x_\alpha, x_\beta) \) | a | b | c | d | e |
|--------------------------|----|----|----|----|----|
| \( (1, 2) \)             | 1  | 1  | 0  | 0  | 0  |
| \( (1, 3) \)             | 1  | 1  | 0  | 0  | 1  |
| \( (1, 4) \)             | 0  | 0  | 1  | 0  | 1  |
| \( (1, 6) \)             | 0  | 0  | 0  | 0  | 1  |
| \( (1, 8) \)             | 1  | 1  | 1  | 1  | 0  |
| \( (2, 7) \)             | 1  | 1  | 1  | 1  | 0  |
| \( (4, 7) \)             | 1  | 0  | 1  | 1  | 0  |
| \( (6, 7) \)             | 0  | 0  | 1  | 1  | 0  |
| \( (7, 8) \)             | 1  | 1  | 0  | 0  | 1  |
| \( (7, 12) \)            | 1  | 1  | 0  | 0  | 1  |
| \( (8, 12) \)            | 0  | 0  | 0  | 0  | 1  |

From the distinguishable matrix \( BM \), we can obtain: \( Red = Core = \{e\} \). Then, according to definition 10, calculate the distinguishable matrix \( BM' \) of binary distinguishable matrix \( BM \) after attribute \( e \) is removed, as shown in Table 3, calculate the attribute frequency function of attribute \( a \),

\[ \text{Algo} }
Due to \( f(a) = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{2} = \frac{3}{2} \), \( f(b) = \frac{3}{2} \), \( f(c) = \frac{1}{2} \), \( f(d) = \frac{1}{2} \). Due to \( f(a) = f(b) \) and \( f(c) = f(d) \), in Table 2, delete column \( e \) of attribute and the row corresponding to attribute value 1, and Table 4 is as follows:

| \( m(x_j, x_j) \) | \( a \) | \( b \) | \( C \) | \( d \) |
|----------------|-----|-----|-----|-----|
| (1,2)          | 1   | 1   | 0   | 0   |
| (1,8)          | 1   | 1   | 1   | 1   |
| (2,7)          | 1   | 1   | 1   | 1   |
| (7,8)          | 1   | 1   | 0   | 0   |

It is necessary to select the attribute with the maximum classification ability (maxDAGL) according to Algorithm 2. According to Table 3, we can obtain: \( U/a = \{ (x_1, x_7), (x_2, x_8) \} = U/b \), \( U/c = \{ (x_1, x_2), (x_7, x_8) \} = U/d \), \( U/D = \{ (x_1, x_7), (x_2, x_8) \} \), thus \( POS_a(D) = POS_b(D) = \{ x_1, x_2, x_7, x_8 \} \), \( POS_c(D) = POS_d(D) = \phi \). Therefore, it is the same to select attribute \( a \) or \( b \), add \( a \) to reduction set here, that is \( \text{Red} = \{ a, e \} \) or \( \text{Red} = \{ b, e \} \). At this time, \( BM' = \phi \), and \( POS_{\text{Red-}[a]}(D) \neq POS_{\text{Red-}[b]}(D) \neq POS_{\text{Red-}[c]}(D) \). Therefore, optimal reduction form is \( \text{Red} = \{ a, e \} \) or \( \text{Red} = \{ b, e \} \).

Example 2. The accuracy of the improved Algorithm 1 is verified by adding the objects in example 1, and the effectiveness of the improved Algorithm 1 is verified when the simplified decision table \( S' \) and attribute reduction \( \text{Red}(C) \) are updated dynamically.

1. When the newly added object \( x' = (0,1,0,1,1,1) \), there are \( \exists x_2 = U'_{\text{neg}} \), \( \forall c_j \in C \Rightarrow f(x', c_j) = f(x_2, c_j) \), then \( U' \) and \( \text{Red}(C) \) remain unchanged, and the results are consistent with those in [5];

2. When a new object \( x' = (1,0,1,0,1,1) \) is added, \( \exists x_7 = U'_{\text{pos}} \), so that \( f(x', c_j) = f(x_7, c_j) \wedge f(x, D) = f(x_7, D) \), then \( U' \) and \( \text{Red}(C) \) remain unchanged, and the results are consistent with those in [5];

3. When the newly added object \( x' = (1,0,0,1,1,0) \), \( \exists x_1 = U'_{\text{pos}} \), so that \( f(x', c_j) = f(x_1, c_j) \wedge f(x, D) \neq f(x_1, D) \), incremental update according to Step 5 of Algorithm 4: \( U'_{\text{pos}} = \{ x_7 \} \) and \( U'_{\text{neg}} = \{ x_1, x_2, x_3, x_4, x_5, x_8, x_{12} \} \); in the binary distinguishable matrix, delete the row corresponding to \( x_1, x_2, x_3, x_4, x_5, x_8, x_{12} \), and add the row generated by \( x_i \) and \( x_j \), as shown in Table 5:
After removing \( m(x_1, x_2), m(x_3, x_4), m(x_4, x_5), m(x_5, x_6), m(x_1, x_6), m(x_7, x_12) \) from \( BM \), according to Algorithm 1 and Algorithm 2, \( \text{Red}(C) \) reduction remains unchanged, and the results is consistent with that of [11].

After adding \( m(x_1, x_7) \) from \( BM \), \( \text{Red}(C)=\{a, c, e\} \) or \( \text{Red}(C)=\{b, d, e\} \) can be reduced according to Algorithm 1 and Algorithm 2, and the results are consistent with those in [11].

(4) When the newly added object \( x=(1,1,0,0,0,0) \), these are \( x^t=U^t_{pos}, \exists \exists C_j \in C \), we have \( f(x', C_j) \neq f(x, C_j) \), incremental update according to Step 4 of Algorithm 1: \( U^t_{pos} = \{x_1, x_7, x'\} \); adding generated row in \( x' \) and \( x_2, x_3, x_4, x_6, x_8, x_{12} \), as shown in Table 6:

| \( m(x_8, x_9) \) | a | b | c | d | e |
|-------------------|---|---|---|---|---|
| \((1.2)\)          | 1 | 1 | 0 | 0 | 0 |
| \((1.3)\)          | 1 | 1 | 0 | 0 | 1 |
| \((1.4)\)          | 0 | 0 | 1 | 0 | 1 |
| \((1.6)\)          | 0 | 0 | 1 | 1 | 1 |
| \((1.8)\)          | 1 | 1 | 1 | 1 | 0 |
| \((1.12)\)         | 1 | 1 | 1 | 1 | 0 |

Adding to \( m(x_1, x_7), m(x_1, x_3), m(x_1, x_4), m(x_1, x_5), m(x_1, x_6), m(x'_1, x_12) \) from \( BM \), according to Algorithm 1 and 2, we can obtain: \( \text{Red}(C)=\{a, b, c\} \) or \( \text{Red}(C)=\{b, c, e\} \). \( POS_{\text{Red}=\{a\}}(D)=POS_C(D) \) and \( POS_{\text{Red}=\{e\}}(D)=POS_C(D) \), thus, the optimal reduction form is: \( \text{Red}(C)=\{b, e\} \).

| Table 7. Comparing of reduction results in case 4. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| Current algorithm | Algorithm in [5] | Traditional algorithm based on positive domain the |
| Reduction Red | \( \{b, e\} \) | \( \{a, d\}, \{b, c, d\}, \{a, c, d\} \) | \( \{a, b, e\}, \{a, c, e\} \) | \( \{b, e\} \) |
| Efficiency | No | Yes | No |
| Optimal reduction | Yes | No | Yes |
| Time complexity | \( O(|C||U'|) \) | \( O(|C||U'|) \) | \( O(|C||U'|^2) \) |

The reduction results of (4) is \( \text{Red}(C)=\{a, d\} \), \( \text{Red}(C)=\{b, c, d\} \), \( \text{Red}(C)=\{a, c, d\} \), \( \text{Red}(C)=\{a, b, c\} \), and \( \text{Red}(C)=\{a, c, e\} \). But some are not reduction, such as \( POS_{\{b, c, d\}}(D)=POS_C(D) \), thus, \( \text{Red}(C)=\{b, c, d\} \) is not reduction.

There are some shortcomings in [11]: when the algorithm obtains multiple reductions, it does not indicate which one is the optimal reduction; the reductions obtained are not consistent with the reduction results obtained by the positive domain algorithm, and some are not reductions; In this paper,
the time complexity of the dynamic update algorithm is, the space complexity is, and the time complexity of the algorithm in [11] is, and the space complexity is. Obviously, the algorithm in this paper greatly reduces time consumption and has high efficiency. Although the traditional positive-domain algorithm has obtained the most reduction results, the time complexity is higher than the algorithm in this paper. In the algorithm of this paper, when new objects are dynamically added to the decision table, there is no need to perform secondary operations on all the known data in the entire database. You only need to determine the situation of the new objects in Algorithm 1, and calculate the corresponding matrix elements. This not only saves a lot of computing time, but also reduces the storage space. The validity and accuracy of the algorithm in this paper is also verified in the example.

4. Conclusion
When dealing with actual problems, most of the data in the database is dynamically changed, and the algorithm for processing static data sets has some problems and deficiencies at this time. Based on the existing binary distinguishable matrix reduction algorithm, this paper proposes an incremental attribute reduction algorithm based on an improved binary distinguishable matrix. In the example, when a single object in the decision table dynamically increases, the improved algorithm can find the reduction of the new decision table in a shorter time than the algorithm in [11], and it is the optimal reduction, which overcomes the fact that the algorithm in the literature does not indicate the optimal when obtaining multiple reductions. The shortcomings of reduction also delete the inaccurate reduction, and at the same time improve the efficiency of the algorithm, so the accuracy and effectiveness of the algorithm have been verified. The next step of this article is to explore an incremental attribute reduction algorithm based on a binary discernible matrix when the attributes in the decision table change dynamically.

References
[1] Pawlak Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11(5):341-356.
[2] Wong SKM, Optimal decision rules in decision table [J]. Bulletin of Polish Academy of Sciences, 1985, 33(11-12), 693-696.
[3] Miao Duo-qian, Wang Jue. An information representation of the concepts and operations in rough set theory [J]. Journal of Software, 1999, 10(2):113-116.
[4] Skowron A, Rauszer C. The discernibility matrices and functions in information systems [M]. Intelligent Decision Support-handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publishers, 1992: 331-362.
[5] Liu Fengfeng, Mu Lianming. Incremental attribute reduction algorithm of decision table based on improved discernibility matrix [J]. Computer Engineering, 2010, 36 (20): 46-48.
[6] Shen Xuefen, Xie Yan. An incremental attribute reduction algorithm based on relative positive field. [J]. Journal of Guangxi Normal University: Natural Science Edition, 2013, 31 (3): 45-50.
[7] Ge Yunjing. An incremental attribute reduction algorithm based on relational matrix decision table [J]. Small Micro Computer System, 2015, 5 (5): 1069-1072.
[8] Cheng Ni. An incremental simplified algorithm based on attribute changes [J]. Journal of Shanxi Datong University: Natural Science Edition, 2016, 32 (1): 3-7.
[9] Gao Xiaohong, Li Xingqi. An incremental attribute reduction algorithm based on rough set theory [J]. Journal of Changchun University, 2018, 28 (12): 16-20.
[10] Yanyan Yang, Dedang Chen, Hui Wang. Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving [J]. Applied Soft Computing, 2016, 45:129-149.
[11] Zhi Tianyun, Miao Duoqian. Transformation of Binary Discernible Matrix and Construction of Efficient Attribute Reduction Algorithm [J]. Computer Science, 2002, 29 (2): 140-142.
[12] Miao Duoqian, Li Daoguo. Rough Set Theory, Algorithm and Application [M]. Beijing: Tsinghua University Press, 2008, 4.
[13] Zhangyan Xu, Wenbin Qian. Research on fast update algorithm of attribute reduction and kernel based on rough set [D]. Guangxi: Guangxi Normal University, 2007, 29-33.

[14] Ge Hao, Yang Chuanjian, Li Longyan. An improved attribute reduction algorithm based on binary distinguishable matrix [J]. Computer Technology and Development, 2008, 18 (8): 12-16.