Research Article

Indistinguishable Element-Pair Attribute Reduction and Its Incremental Approach

Baohua Liang,1,2,3 Haiqi Zhang,1,2 Zhengyu Lu,1,2 and Zhengjin Zhang1,3

1Institute of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, Guangxi, China
2Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, Guangxi Normal University, Guilin 541004, China
3School of Information Engineering, Chaohu University, Hefei 238000, Anhui, China

Correspondence should be addressed to Zhengjin Zhang; 054029@chu.edu.cn

Received 24 May 2022; Accepted 8 August 2022; Published 29 September 2022

1. Introduction

Rough sets theory (RST) is a valid mathematical tool, which was proposed by Pawlak and Skowron in 1982, for dealing with inaccurate, incomplete, and vague information [1]. RST has been widely used in many fields such as machine learning [2], data mining [3], decision supporting [4], expert system [5], pattern recognition [6], and music emotions annotation [7]. Attribute reduction is one of the hot research focuses in RST [8], which aims to delete redundant data, while keeping the distinguishing power of the original data in information systems. For the convenience of the following description, Table 1 summarizes the list of abbreviations in the article. In the last two decades, many heuristic attribute reduction approaches have been developed based on the positive region [9], discernibility matrix [10, 11], information entropy [12], fuzzy rough [13, 14], m-polar fuzzy [15, 16], and knowledge granularity [17].

Among the abovementioned approaches, DMA is a typical reduction model. Since DMA consumes a lot of space to store distinguishable information, it cannot reduce large data sets. In order to effectively express the distinguishable information among samples, Hu and Cercone [18] proposed a concise definition of a discernibility matrix. Ye and Chen [19] proposed a discernibility matrix-elements that retains all bases of 1. Yang and Sun [20] use the sample comparison of the upper and lower approximation to obtain the discernibility matrix. Dong et al. [21] proposed a fast algorithm of attribute reduction for covering the decision system with minimal elements in discernibility matrix. Wei et al. [22] proposed two discernibility matrices in the sense of entropies. However, these approaches only consider how to improve the distinguishing ability of samples, and do not consider the space consumption. In order to reduce the space computation, Jiang [10] proposed a minimal element selection tree. Li et al. [23] proposed a simple object-attribute discernibility matrix approach. Although scholars have improved the discernibility matrix, the space consumption problem has not been fundamentally solved. To overcome...
this deficiency, this paper proposes a method based on IEP without discernibility matrix. Firstly, we divide the data set according to conditional attributes and decision attributes and calculate the number of indistinguishable element pairs. Then, select the conditional attribute with the smallest values and calculate the number of indistinguishable element pairs. Finally, repeat the abovementioned two steps until the value is 0.

With the rapid development of communication and network techniques, the actual data may change over time. However, the IEP method is only suitable for static data sets. Hence, it is desired to design an incremental attribute reduction algorithm with IEP to deal with dynamic decision systems.

Incremental learning is an efficient approach making full use of the precious results of the original decision system, which can obtain the efficient reduced results by recomputing the updated part of the dynamic data set. Many incremental algorithms have been proposed with different models for dynamic data. Yang proposed an incremental algorithm for updating an object or attribute [24]. Ge et al. developed an incremental attribute reduction based on a simplified discernibility matrix, which is equivalent to attribute reduction based on a positive region [25]. Liu et al. proposed a strong discernibility matrix method for incremental attribute reduction on fuzzy decision tables [26]. In literature [27], Wei proposed three new types of discernibility matrices by compacting a decision table. Zhang et al. proposed a method based on a relation matrix under the change attribute reduction in set-valued information systems [28]. Ma et al. [29] proposed a compressed binary discernibility matrix to process the group dynamic data. Obviously, the abovementioned matrix methods mainly focus on updating the elements of discernibility matrix. These approaches are ineffective in obtaining the reduction results with large-scale decision systems due to the limited memory space. Hence, we incorporate the incremental update mechanism into the IEP approach. Verifies the feasibility and efficiency of proposed algorithm through extensive experiments on UCI data sets.

### 2. Preliminaries

In this section, we review some basic concepts about rough set, discernibility matrix, and indistinguishable element-pair.

#### 2.1. Basic Concepts

**Definition 1** (see [1]). Given the decision system is a quadruple tuple $S = (U, A, V, f)$, where $U$ is a finite nonempty object set and $A$ is a finite nonempty attribute set, $V = \cup_{a \in A} V_a$, $V_a$ is a set of its values, and $f: U \times A \rightarrow V$ is an information function with $f(x, a) = V_a$ for each $a \in A$ and $x \in U$. If $A = C \cup D$, where $C$ is the conditional attribute set, and $D$ is the decision attribute set. For every subset $P \subseteq A$, an indiscernibility relation $\text{IND}(P)$ is defined as follows:

\[
\text{IND}(P) = \{(x, y) \in U \times U | \forall a \in P, f(x, a) = f(y, a)\}.
\]

Obviously, if $\text{IND}(P)$ denotes as $U/P$, $U/P$ is an equivalence relation. We assume includes $x$, the equivalence relation $x$ is defined as:

\[
[x]_P = \{y | \forall a \in P, f(x, a) = f(y, a)\}.
\]

**Definition 2** (see [1]). Given the decision system $S = (U, A, V, f)$ for every subset $Y \subseteq U$ and indiscernibility relation $\text{IND}(P)$, the upper approximation set and the lower approximation set of $Y$ can be defined by the basic set of $P$ as follows:

\[
\overline{P}(Y) = \{x \in U | [x]_P \subseteq Y\},
\]
\[
\underline{P}(Y) = \{x \in U | [x]_P \cap Y = \emptyset\}.
\]

The universe $U$ is partitioned into three disjoint regions by these two approximations $\overline{P}(Y)$ and $\underline{P}(Y)$: the positive region $\text{POS}_p(Y)$, the negative region $\text{NEG}_p(Y)$, and the boundary region $\text{BND}_p(Y)$. Then the three different regions are defined as following, respectively:

\[
\text{NEG}_p(Y) = U - \overline{P}(Y),
\]
\[
\text{BND}_p(Y) = \overline{P}(Y) - \underline{P}(Y),
\]
\[
\text{POS}_p(Y) = \underline{P}(Y).
\]

**Definition 3** (see [18]). Let $DT^* = (U, C \cup D)$ be a decision table, $C$ be the condition attribute set, and $D$ be the decision attribute. The discernibility matrix in all samples is defined as $M_{DT}^P = \{m_{ij}^P\}$, where:

\[
m_{ij}^P = \begin{cases} 
   \{c \in C: f(x_i, c) \neq f(x_j, c)\}, & f(x_i, d) \neq f(x_j, d), \\
   \emptyset, & \text{otherwise}.
\end{cases}
\]

**Definition 4** (see [31]). Let $DT^* = (U, C \cup D)$ be a decision table, $C$ be the condition attribute set, and $D$ be the decision attribute. In terms of a positive region, the discernibility matrix is defined as $M_{DT}^P = \{m_{ij}^P\}$, where:
\[ m_{ij}^E = \begin{cases} 
\{ c \in C: f(x_i, c) \neq f(x_j, c) \}, 
\{ c \in C: f(x_i, c) \neq f(x_j, c) \}, 
\{ c \in C: f(x_i, c) \neq f(x_j, c) \}, 
\emptyset, 
\end{cases} \]

\[ f(x_i, d) \neq f(x_j, d) \text{ and } x_i, x_j \in U_1, \]
\[ x_i \in U_1, x_j \in U_2, \]
\[ x_i, x_j \in U_2, \]
\[ \text{otherwise}. \]

\[ \text{count2} = \sum |D_i|^2 \text{ where } x_i \in U_1, x_j \in U_2. \]

\[ \text{count3} = |D_{\text{neg}}| \cdot (|D_{\text{neg}}| - 1)/2. \]

Overall, the total number (TotalCount) of comparisons in discernibility matrix based on Definition 5 is as follows:
The division on decision attribute $A$ is a conditional attribute set. \( A = \{ X_1, X_2, \ldots, X_n \} \) is the division of data set $U$ on decision attribute $D$. \( U/D = \{ Y_1, Y_2, \ldots, Y_m, Y_{neg} \} \) is the division on decision attribute $D$. \( Y_{neg} \) is the division of inconsistent samples in decision attributes $A$.

**Definition 6.** Suppose $U$ is a universe that is nonempty finite data set, $A$ is a conditional attribute set. $U/A = \{ X_1, X_2, \ldots, X_n \}$ is the division of data set $U$ on decision attribute $D$. \( U/D = \{ Y_1, Y_2, \ldots, Y_m, Y_{neg} \} \) is the division on decision attribute $D$. $Y_{neg}$ is the division of inconsistent samples in decision attributes $A$.

\[
\text{Total count} = \text{Count1} + \text{Count2} + \text{Count3} = \frac{\left| D_{pos} \right|^2 - \sum_{i=1}^{n} \left| D_i \right|^2}{2} + \left| D_{pos} \right| \cdot \left| D_{neg} \right| + \left| D_{neg} \right|
\]

\[
\left( \frac{\left| D_{neg} \right| - 1}{2} \right) = \left| U \right|^2 - \left| U \right| - \sum_{i=1}^{n} \left( \left| D_i \right|^2 - \left| D_i \right| \right)
\]

\[
= C_2^2 - \sum_{i=1}^{n} C_2^2 [D_i]
\]

\[
= C_2^2 - \sum_{i=1}^{n} C_2^2 [D_i].
\]

Obviously, when a data set is given, the total number of comparisons is only related to the division of decision attributes, and irrelevant to conditional attributes. 

\[\Box\]

**2.2. The Presentation of the Indistinguishable Element-Pair.**

The discernibility matrix algorithm records the differences between samples by different values of conditional attributes between them, and the amount of distinguishable information measures the importance of the attributes. The larger the value, the more important the attribute. For the discernibility matrix proposed in literature [24] and literature [22], the number of comparisons between samples in the discernibility matrix is determined when the data set is given. Among the samples to be compared, the value of certain condition attributes is either the same or different. The same values of one conditional attribute mean being indistinguishable, while different means being distinguishable. If the amount of distinguishable information is larger, the amount of indistinguishable information is smaller when the total number of comparisons does not change. Here, we use the amount of indistinguishable information to measure the importance of conditional attributes.

**Definition 6.** Suppose $U$ is a universe that is nonempty finite data set, $A$ is a conditional attribute set. $U/A = \{ X_1, X_2, \ldots, X_n \}$ is the division of data set $U$ on conditional attribute $A$ and $U/D = \{ Y_1, Y_2, \ldots, Y_m, Y_{neg} \}$ is the division on decision attribute $D$. $Y_{neg}$ is the division of inconsistent samples in decision attributes $A$.

\[
\text{IEP}_U (D|A) = \sum_{i=1}^{n} \left( C_2^2 [X_i] - \sum_{k=i}^{m} C_2^2 [X_i, Y_k] \right).
\]

Where $C_2^2 [X_i] = |X_i| \cdot (|X_i| - 1)/2$.

In fact, all data objects among the subdivision $X_i$ are indistinguishable from each other. There are $C_2^2 [X_i]$ pairs. However, among these element pairs, some comparisons should be subtracted due to some data objects with the same decision attribute value. We have the definition of indistinguishable element-pair.

The asterisked (*) data objects belong to the negative region set

\[\text{Example 1. Suppose} U \text{ is a simplified decision table without repeated samples in Table 2. A is a conditional attribute and} D \text{ is a decision attribute. Let} A = a \cup b, \text{ the data objects} \{ x_9, x_{10} \} \text{ belong to the negative region, Let} U/D = \{ \{ x_1, x_2, x_3 \}, \{ x_4, x_5, x_6 \}, \{ x_7, x_8 \}, \{ x_9, x_{10} \} \}, \text{and} U/A = \{ \{ 1, 2, 5, 6 \}, \{ 3, 4, 9 \}, \{ 7, 8, 10 \} \}. \text{In subdivisions} U/A, \text{the underlined data objects have the same decision attribute value. Data set} U \text{ has three subdivisions} U_1 = \{ 1, 2, 5, 6 \}, U_2 = \{ 3, 4, 9 \} \text{ and} U_3 = \{ 7, 8, 10 \} \text{ on conditional attribute} A. \text{ According to Definition 6, we have indistinguishable information of the three subdivisions as follows:}

\[
\begin{align*}
\text{IEP}_{U_1} (D|A) &= C_2^2 - C_2^2 - C_2^2, \\
\text{IEP}_{U_2} (D|A) &= C_2^2, \\
\text{IEP}_{U_3} (D|A) &= C_3^2 - C_2^2, \\
\text{IEP}_U (D|A) &= \text{IEP}_{U_1} (D|A) + \text{IEP}_{U_2} (D|A) + \text{IEP}_{U_3} (D|A) = 9.
\end{align*}
\]

Theorem 2. The smaller the indistinguishable element pair, the stronger the distinguishing ability.

Proof. According to Theorem 1, if the data set is given, the number of data objects in the positive and negative regions is
Table 2: Simplified decision table.

<table>
<thead>
<tr>
<th>No</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>*</td>
</tr>
</tbody>
</table>

IEP_U(D|B) = \Delta + C^2_{[x_i \cup x_{i'}]} - \sum_{k=1}^{m} C^2_{[x_i \cap x_{i_k}]} \cap x_{i_k} = \Delta + \frac{(x+y)(x+y-1)}{2} - \frac{(ax+by)(ax+by-1)}{2} IEP_U(D|P)

= \Delta + \frac{x(x-1)}{2} + \frac{y(y-1)}{2} - \frac{ax(ax-1)}{2} \quad (12)

Since 0 \leq a, b \leq 1, IEP_U(D|B) - IEP_U(D|P) = C^2_{[x_i \cup x_{i'}]} - C^2_{[x_i \cap x_{i_k}]} \geq 0. Weave IEP_U(D|P) \leq IEP_U(D|B) \quad (13)

Theorem 4. Let S = (U, C \cup D) be a decision table and U is a data set without duplicate samples, then IEP_U(D|C) = 0.

Proof. Assume U/C = \{U_1, U_2, \ldots, U_n\}. Since U is a data set without duplicate samples, |U_1| = |U_2| = \ldots = |U_n| = 1, So IEP_U(D|C) = \sum_{i=1}^{n} D

Definition 7. Let S = (U, C \cup D) be a decision table and B \subseteq C. U is a data set without duplicate samples. Then B is a relative reduction based on the following indistinguishable element-pair of S if B satisfies:

1. IEP_U(D|B) = IEP_U(D|C)
2. \forall a \in B, IEP_U(D|B - \{a\}) \neq IEP_U(D|B)

Definition 8 (see [30]). Let S = (U, A) be an information system. For any a \in A the value of object x about attribute a is f(a, x). Let Dis(a) = \{ (x_i, x_j) | f(a, x_i) \neq f(a, x_j) \} and Dis(A) = \cup_{a \in A} Dis(a), we called Dis(a) and Dis(A) the discernibility relations in terms of a and A, respectively.

\[ \text{Dis}(A) = \cup_{a \in A} \text{Dis}(a) \]

Definition 9 (see [30]). Suppose U = \{x_1, x_2, \ldots, x_n\}, let M_S = (c_{ij})_{n \times n} denote a n \times n matrix, where c_{ij} = \{a \in A: f(a, x_i) \neq f(a, x_j)\} for (x_i, x_j) \in \text{Dis}(A), otherwise c_{ij} = \emptyset, M_S is called the discernibility matrix of the information system S = (U, A).

3. The Algorithm Based on Indistinguishable Element-Pair

The indistinguishable element-pair algorithm obtains the importance of attributes by means of discernibility matrix information and does not create the discernibility matrix. We should compute the positive region and negative region at first, then achieve the simplified decision table. Reduce the calculation of duplicate data objects, saving a lot of time.

3.1. Compute the Positive and Negative Region (CPNR). In each detailed subdivision, if the decision value of the sample is different, then we put the first sample x into the negative region set and let U_{neg} = U_{neg} and \cup x the rest put the x into
the positive region set and let \( U = U_{pos} \cup U_{neg} \). In the process of calculating positive and negative regions, equivalence class division needs to be calculated continuously. Here is an ingenious method, which can greatly speed up the calculation speed of equivalence class partitioning. The details are described as follows: for \( i \) in range \( (n) \): list \([array[i]]\) Appendix Appendix for clarity. (i).

If the data object has an integer value on the attribute, the characteristics of an integer can be used. By collecting all the objects with the same value in the same subdivision, equivalence class division can be obtained quickly and accurately.

**Example 2.** There are six data objects, the values of attribute \( A \) are 1, 2, 1, 3, 2, and 1, respectively. Let \( Array = [1, 2, 1, 3, 2, 1] \), collecting the data objects with the same value into the same list. Array \([1] = Array \[3] = Array \[6] = 1\), we have list \([1] = \{1, 3, 6\}\). Array \([2] = Array \[5] = 2\), we have list \([2] = \{2, 5\}\). Array \([4] = 3\), we have list \([3] = \{4\}\).

Above mentioned all, data objects 1, 3, and 6 are divided into the same subdivision, and data objects 2 and 5 are divided into the same subdivision. (Algorithm 1)

**3.2. The Attribute Reduction Algorithm Based on Indistinguishable Element-Pair (IEP).** Suppose \( f(x, a) \) is the value of the data object \( x \) on the conditional attribute \( A \). There have two different data objects \( x_i \) and \( x_j \) if \( f(x, a) = 0 \), then a is recorded in the indiscernibility matrix. Another way to think about it is to take down indistinguishable data objects. After research, all samples divided by the same subdivision are indistinguishable. Algorithm IEP is described as follows: (Algorithm 2)

**Example 3.** Suppose \( U \) is a simplified decision table without repeated samples in Table 3. Based on the definition equivalence class, we have \( U/(a \cup D) = \{\{1, 4\}, \{2, 3\}, \{5, 6\}\}\). The bold data objects of the subdivision have the same value on the attribute \( a \cup D \). The asterisked (\* ) data objects belong to the negative region set. Based on the Definition 6, we have \( IEP_{U_r}(D[a]) = C_1^2 + 0 + C_2^2 = 2 \), \( U_r/b \cup D = \{\{1, 2, 3, 4\}, \{5, 6\}\}\). \( IEP_{U_r}(D[b]) = C_2^2 + C_2^2 + C_2^2 + 4 \), \( U_r/c \cup D = \{\{1, 2, 3, 6\}, \{5\}\}\). \( IEP_{U_r}(D[c]) = C_2^2 + C_2^2 + C_2^2 + 0 = 3 \), \( U_r/e \cup D = \{\{1, 3, 5, 6\}\}\). \( IEP_{U_r}(D[e]) = C_2^2 + C_2^2 + C_2^2 = 6 \).

When the conditional attribute with the lower indistinguishable degree has a stronger distinguishing ability, we select the attribute \( a \), and let \( red = \{a\} \). If the amount of data is not zero, we enter the next cycle. Based on the division of data objects \( \{b, c, e\} \), we obtain the following: \( U/(red \cup b \cup D) = \{\{1, 4\}, \{2, 3\}, \{5, 6\}\}\). \( IEP_{U_r}(D[red \cup b]) = C_4^2 + 0 + C_4^2 = 2 \), \( U/(red \cup c \cup D) = \{\{1, 2, 3\}, \{4\}, \{5, 6\}\}\). \( IEP_{U_r}(D[red \cup c]) = 0 \), \( U/(red \cup e \cup D) = \{\{1, 2, 3\}, \{4\}, \{5, 6\}\}\). \( IEP_{U_r}(D[red \cup e]) = 0 \).

Because \( IEP_{U_r}(D[red \cup c]) = 0 \) is the smallest, we select \( c \) to merge into the reduced set. Now, the amount of information is 0, and the algorithm terminates. Reduce result is \( red = \{a, c\} \).

**3.3. The Existing Static Reduction Algorithms.** The typical discernibility matrix algorithm and the related improved algorithms constantly revise the definition of discriminant matrix from the perspective of distinguishable data objects, leading to the inevitable consumption of a large amount of space resources to store the discernibility matrix. The phenomenon of memory overflow often occurs during the reduction of large data sets, which leads to the failure to complete the reduction task.

The IEP method does not need to store discernibility matrix and is suitable for reduction of large-scale data sets. In order to further verify the effectiveness of the IEP presented in this paper, let’s analyze the complexity of time and space and other similar algorithms based on discernibility matrix. In IEP algorithm, \( U \) is a decision table. Steps 2-3 focus on calculating the simplified decision table \( U_r \) and \( |U_r| \leq |U| \). The time complexity of computing \( U_r \) is \( O(|U_r|) \). The space complexity of steps 5 is \( O(|U_r|) \). The space complexity of data set is \( O(|U_r|) \). steps 5 want space \( O(|U|) \). Therefore, the total time complexity of algorithm IEP is \( O(|U_r|) + O(|U_r|) \). The space complexity of the algorithm MEDA is \( O(|U|) \). The space complexity of algorithm IEP is \( O(|U|) \) but the space complexity of storing the discernibility matrix of HU, DDMSE is \( O(|U|^2) \). Therefore, the space consumption of algorithm IEP is much less than that of algorithms HU, DDMSE, and MEDA.

**4. Incremental Attribute Reduction Algorithm Based on Indistinguishable Element-Pair**

The aforementioned algorithm IEP only adapts to the static data set. In reality, most data sets are dynamic. The traditional static methods are ineffective. Therefore, it is necessary to study some algorithms for dynamic data sets.

**4.1. An Incremental Method to Calculate Indistinguishable Element-Pair after Adding Some Objects (IEPAO).** There are two kinds of data objects as to updating in data set: increase and decrease. Let’s introduce the first one: to increase the data objects. When some data objects are added to data set, we only need to calculate the IEP of the updating part and obtain the amount of information with the help of the previous reduction result red. If the amount of information is zero, the updated reduction result is red. Otherwise, the added part objects will be merged with the basic data, we compute the amount of information based on detailed subdivision according to attribute set \( red \cup D \).

**Theorem 5.** Let \( S = (U, C \cup D) \) be a decision system, \( U/C \cup D = \{X_1, X_2, \ldots, X_m\} \). It is assumed that \( U_{\Delta x} \) is the new data objects,
Input: $U, R = \emptyset$, count = 0, C, D
Output: $U', U_{pos}', U_{neg}'$

/* $U'$ is a data set without duplicate samples, $U_{pos}'$ is a positive region, data set and $U_{neg}'$ is a negative region data set. */

Step 1: $U' = U_{pos}' = U_{neg}' = \emptyset$
Step 2: $U' = U$
Step 3: while (count $\leq |U| \& \& R \subseteq C$) do{
  Step 3.1: for any $C_i \subseteq C - R$, let $R = R \cup C_i$
  Step 3.2: compute $U' = U \cup U_{pos}'$.
  Step 3.3: statistics of the subdivisions regarded as $|U'| = 1$, let $U_{pos}' = U_{pos}' \cup U_{neg}'$, count add 1.
Step 4: scan the remaining subdivisions.

Algorithm 1: CPNR algorithm computes the positive and negative region method.

Input: $S = (U, C \cup D)$
Output: red
Step 1: $\text{red} \leftarrow \emptyset$, $B \leftarrow \emptyset$
Step 2: Calculate the positive region $U_{pos}$ and negative region $U_{neg}$ with CPNR
Step 3: get the simplified decision table $U'$ through Step 2
Step 4: $\text{IEP} \leftarrow \text{Sys}.\text{maxsize}$, $\text{IEP}_{list} \leftarrow \emptyset$
Step 5: while $(\text{IEP} > 0)$ do {
  Step 5.1: $\text{IEP}(b, B, D) = \min(\text{IEP}(a, B, D))$, $a \in C - B$
  Step 5.2: $\text{IEP}_{list} \leftarrow \text{IEP}_{list}(b, B, D)$
  Step 5.3: delete all the subdivisions and the card is 1 of $\text{IEP}_{list}$
  Step 5.4: $B = B \cup \{b\}$
Step 6: $\text{red} \leftarrow \text{B} \text{ return red}$

Algorithm 2: The indistinguishable element-pair algorithm (IEP).

Table 3: Example of decision table.

<table>
<thead>
<tr>
<th>No</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>e</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 4: A comparison of time and space complexity of IEP, HU, DDMSE, and MEDA.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEP</td>
<td>$O(</td>
<td>U</td>
</tr>
<tr>
<td>HU</td>
<td>$O(</td>
<td>U</td>
</tr>
<tr>
<td>DDMSE</td>
<td>$O(</td>
<td>U</td>
</tr>
<tr>
<td>MEDA</td>
<td>$O(</td>
<td>U</td>
</tr>
</tbody>
</table>

$U_{ax} / C \cup D = \{Y_1, Y_2, \ldots, Y_m\}$, $X_i \cup Y_i = X_i$, $1 \leq i \leq k$.
According to the division of $U_{ax} / C \cup D$, $U_{ax} / C \cup D = \{X_1, X_2, \ldots, X_k, X_{k+1}, X_{k+2}, \ldots, Y_m\}$. Then $\text{IEP}_{ax}(D(C)) = \text{IEP}_{ax}(D(C)) + \text{IEP}_{ax}(D(C)) + |U||U_{ax}| - \sum_{i=1}^{k} |X_i||Y_i|$

Proof. See Appendix 1 for the proof process.

According to Theorem 5, the value of $\text{IEP}_{ax}(D(C))$ is related to $\text{IEP}_{ax}(D(C))$, where add data objects. We propose the algorithm IEPAO based on this characteristic. (Algorithm 3)

In Table 5, red is the reduction result before adding data, red’ is the final reduction result IEP can only reduce static data set the time complexity is $O(|U||C|) + O((|C|^2 - |\text{red}|) \cdot |U|)$. If data objects $U_{ax}$ is added, the time complexity becomes $O((|U| + |U_{ax}|) \cdot |C|) + O((|C|^2 - |\text{red}|^2)(|U| + |U_{ax}|))$. The IEPAO algorithm uses the previous reduction results, the time complexity is $O(|U_{ax}| \cdot |C|) + O((|C|^2 - |\text{red}|^2 - (|\text{red}'| - |\text{red}|^2) \cdot \min(|U_{ax}|, |U|))$. It clearly shows that the calculation time of IEPAO is less than IEP.

4.2 Incremental Updating Attribute Reduction Algorithm When Delete Some Objects (IEPDO). In reality, some data will be discarded after a long time. IEPDO algorithm can reduce the deleted data objects dynamically.

Theorem 6. Let $S = \{U, C \cup D\}$ be a decision system and $U / C \cup D = \{X_1, X_2, \ldots, X_m\}$. We assume that the deleted data object set is $U_{ax}$ and $U_{ax} / C \cup D = \{Y_1, Y_2, \ldots, Y_k\}$. From the definition of equivalence class divided, we have $(U - U_{ax}) / C \cup D = \{X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_m\}$, where $X_i = X_i - Y_i$, $i = 1, 2, \ldots, k$. If delete the data objects $U_{ax}$ from $U$, the indistinguishable amount of information is
Input: $U, U/\text{red} \cup D$, red and incremental object sets $U_{\Delta k}$, where $U$ is the simplified decision table before update.

Output: Updated reduction set red'

Step 1: Mark the negative region data objects, list $= U/\text{red} \cup D$, red$'\leftarrow$red

Step 2: Compute $\text{IEP}_{U_{\Delta k}}(D)$, red$'$

Step 3: We may assume that $(U/\text{red} \cup D) \cap (U_{\Delta k}/\text{red}')$ has $k$ subdivisions and $X_i \in U, Y_i \in U_{\Delta k}$, then compute $\sum_{i=1}^{k} |X_i||Y_i|$.

Step 4: Compute $\text{Info} = \text{IEP}_U(D)$, $\text{IEP}_{U_{\Delta k}}(D)$, $|U||U_{\Delta k}| - \sum_{i=1}^{k} |X_i||Y_i|$

Step 5: If $\text{Info} = 0$, algorithm is terminated else

While $\text{Info} \neq 0$ do {
  for a in $C - \text{red}'$ {
    Compute $\text{IEP}_{U_{\Delta k} \cup \{a\}}(D)\text{red}'' \cup a)$, $\text{Info} = \text{IEP}_U(D)$, $\text{IEP}_{U_{\Delta k}}(D)$, $|U||U_{\Delta k}|$
    Compute $\text{Info} = \text{IEP}_{U_{\Delta k} \cup \{a\}}(D)\text{red}''\cup a)$
    list $\leftarrow (U \cup U_{\Delta k})/\text{red}''$
  }

Step 6: return red$'$

Algorithm 3: Incremental updating algorithm based on IEP when adding some objects (IEPAO).

Table 5: The time complexity of each step of algorithm IEPAO.

<table>
<thead>
<tr>
<th>Step no</th>
<th>Time complexity</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>$O(</td>
<td>U_{\Delta k}</td>
</tr>
<tr>
<td>Step 2</td>
<td>$O(</td>
<td>U/\text{red} \cup D</td>
</tr>
<tr>
<td>Step 3</td>
<td>$O(</td>
<td>U\text{red} -</td>
</tr>
<tr>
<td>Step 4</td>
<td>$O(1)$</td>
<td>Get info</td>
</tr>
<tr>
<td>Step 5</td>
<td>$O(</td>
<td>U_{\Delta k}</td>
</tr>
</tbody>
</table>

$\text{IEP}_{U\cup U_{\Delta k}}(D|C)$, then $\text{IEP}_{U\cup U_{\Delta k}}(D|C) = \text{IEP}_{U}(D|C) + \text{IEP}_{U_{\Delta k}}(D|C) - |U||U_{\Delta k}| + \sum_{i=1}^{k} |X_i||Y_i|.$

Proof.

$\text{IEP}_{U\cup U_{\Delta k}}(D|C) = C^2_{|U \cup U_{\Delta k}|} - \sum_{i=1}^{k} C^2_{|X_i - Y|} - \sum_{i=k+1}^{m} C^2_{|X_i|}$

$= C^2_{|U|} + C^2_{|U_{\Delta k}|} - |U||U_{\Delta k}| - \sum_{i=1}^{k} \left(C^2_{|X_i|} + C^2_{|Y|} + \sum_{i=1}^{m} |X_i||Y_i| - \sum_{i=k+1}^{m} C^2_{|X_i|}ight) = \text{IEP}_{U}(D|C) + \text{IEP}_{U_{\Delta k}}(D|C) - |U||U_{\Delta k}| + \sum_{i=1}^{k} |X_i||Y_i|.$

$$ (14) $$

5. Experiment Analysis

In this section, lots of experiments are conducted on both static and dynamic data sets to verify the efficiency of the proposed attribute reduction algorithms. In the experiments, fifteen data sets are downloaded from UCI. Table 7 displays the basic information of each data set, where $|U|$ represents the number of samples, $|C|$ represents the number of conditional attributes, $|D|$ represents the number of decision classes, and Type represents the decision system is consistency (Y in short) or inconsistency (N in short), respectively. For the convenience of the following description,
5.1. Performance Comparison between Algorithm IEP and Other Discernibility Matrix Algorithms Based on Static Data Sets. In experiment, we consider the fifteen data sets from UCI listed in Table 7. These selected data sets are reasonably distributed, including large data sets for Letters and small data sets for Hepatitis, Audiology, consistent data sets (Gene and Mushroom, etc.), and inconsistent data sets (Mass and Spect heart). In order to show the time effect of each algorithm, we refer to the SpeedupRatio = $T_{\text{baseline}}/T$ method proposed by literature [31], is the executing time of a typical algorithm. $T_{\text{baseline}}$ reaches its maximum when the typical algorithm cannot perform the reduction task. Then SpeedupRatio ∈ [0, ∞).

For the different data sets, the SpeedupRatio of IEP and the other three algorithms (Hu, DDMSE, and MEDA) is also different. Table 8 shows the SpeedupRatio of IEP, Hu, DDMSE, and MEDA on Letters and Connect-4 data sets because these two data sets

Algorithm 4: Increment updating algorithm based on IEP when delete some objects (IEPDO).

Table 6: The time complexity of algorithm IEPDO and IEP.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEP</td>
<td>$O((</td>
</tr>
<tr>
<td>IEPDO</td>
<td>$O((</td>
</tr>
</tbody>
</table>

Table 7: A description of data sets.

| Dataset       | |U|  |C|  |D|  |Consistency? |
|---------------|---|---|---|---|---|--------------|
| Mushroom      | 8124 | 22 | 2 | Y |
| Audiology     | 226  | 70 | 24| Y |
| Blance-scale  | 625  | 4  | 3 | Y |
| Breast        | 286  | 9  | 2 | Y |
| Car           | 1728 | 6  | 4 | Y |
| Letters       | 20000| 17 | 26| Y |
| Ticdata2000   | 5822 | 85 | 2 | Y |
| Gene          | 3190 | 60 | 3 | Y |
| Nursery       | 12960| 8  | 5 | Y |
| Handwritten   | 5620 | 64 | 10| Y |
| Mass          | 830  | 6  | 2 | N |
| Hepatitis     | 155  | 19 | 2 | N |
| Connect-4     | 67557| 42 | 3 | Y |
| Chess kr-kp   | 3196 | 36 | 2 | Y |
| Spect heart   | 267  | 22 | 2 | N |

the data set Letters recognition is abbreviated as Letters, Mammographic Mass as Mass. All the character or string features are normalized into an integer. All of the experiments have been implemented on a PC with Windows 10, Core™ i7-10710U CPU 1.10 GHz 1.61 Hz and 8 G memory. All of the algorithms are coded in python, and the used software is PyCharm Community Edition 2020.2.3 × 64 and Weak3.2.

Input: $U, U_{\text{red}} \cup D$, red and delete object sets $U_{\Delta x}$, where $U$ is the simplified decision table before update.

Output: red'

Step 1: Mark the deleted data objects is $U_{\Delta x}$, list←$U_{\text{red}} \cup D$, red'←red

Step 2: Compute $\text{IEP}_{U_{\Delta x}}(D)[\text{red}']$;

Step 3: We may assume that $(U/\text{red}') \cap (U_{\Delta x}/\text{red}')$ has k subdivisions and $X_i \in U, Y_j \in U_{\Delta x}$, then compute $\sum_{i=1}^{k} |X_i||Y_j|$

Step 4: Compute $\text{Info} = \text{IEP}_{U_{\Delta x}}(D)[\text{red}'] + \text{IEP}_{U_{\Delta x}}(D)[\text{red}'] - |U||U_{\Delta x}| + \sum_{i=1}^{k} |X_i||Y_j|$

Step 5: Let list←$(U - U_{\Delta x})/\text{red}'$
   
   for a in red' {
   
   Compute Info = $\text{IEP}_{U-U_{\Delta x}}(D)[\text{red}' - a]$, if Info = 0 then red'←red' - a * and list←$(U - U_{\Delta x})/\text{red}'$
   else
   
   break;

Step 6: return red'
are too large. The Speedup Ratios of IEP are 1.6109 and 2.4286, respectively, on data sets Audiology and Hepatitis. Since the values are greater than 1, the speed of IEP is faster than Hu on Audiology and Hepatitis. From Table 8, IEP is the fastest but the difference is not obvious among the Hu, DDMSE and MEDA based on small data sets. Overall, the Speedup Ratio is related to $|U|^2 \cdot |C|$. The smaller the $|U|^2 \cdot |C|$, the faster the speed.

Table 9 shows the reduction and performance time of the comparisons of four algorithms. In Table 9, time is measured in seconds, red is reduction and the data in bold represents the minimum reduction time of many algorithms. The IEP takes only 1.953 seconds to reduce the Mushroom data set, while DDMSE takes 230.071 seconds. The main reason is that DDMSE modifies the definition of discernibility information to improve the distinguishing ability, leading to the increasing number of compared element pairs. Then, it makes the space for storing discernibility matrix larger and larger. On data sets Letters and Connect-4, IEP can quickly and effectively obtain the reduction results, while Hu, DDMSE, and MEDA cannot complete the reduction task due to insufficient memory. On the small data set Breast and Balance-scale, IEP needs to waste 0.035 seconds, while the other three algorithms take 0.191, 0.152 and 0.044 seconds respectively. For the reduction on a small data set, the time effect of IEP and other algorithms is not obvious.

Compared with other discernibility matrix algorithms, IEP has less time consumption, while getting the same reduction results based on the same data sets. Especially, the reduction effect is more obvious on large-scale data sets.

5.2. Time Comparison of IEPAO and IEP When Adding Data Objects. In the following experiments, we select nine data sets for dynamic update experiments from Table 7. For each data set, 50% of the data objects are randomly selected as the original object set, and the remaining data are randomly generated at the proportions of 10%, 20%, 30%, 40%, and 50% as incremental object sets, respectively. The incremental part is divided into 5 groups of experiments, each group is executed 10 times and computes the average time. Experimental results are outlined in Figure 1. In the experiment of IEP, time statistics do not include the calculating time of original objects.

In each subfigures of Figure 1, the $x$-coordinate represents the increment ratio, and the unit is the proportion of the increased part of the data in the total data. The value of the $y$-coordinate $y$ is the time of computing reduction in different incremental, which are measured by seconds. In Figure 1, the curve with a five-pointed star mark shows the change in the running time of IEPAO, while the curve with a circle mark indicates the variation of IEP.

It can be seen in Figure 1 that as the size of data set expands, the time of calculating reduction will increase. The calculation time of IEPAO is much less than IEP. The main reason for this phenomenon is that when we add the new objects into the data set, IEPAO only needs to calculate the added part of the data, and then combine the previous reduction results in obtaining the changed result quickly. But, IEP can only process static data sets. When new data is added, it takes longer time to recalculate the original data and the added part. On the whole, the performance of IEPAO is relatively stable in Figure 1. With the increase of updated data, the calculation time is also increasing. But, IEPAO has an anomaly in that the calculation time decreased as the data increasing. The subfigure (f) of Figure 1 displays that IEPAO takes 4.543 seconds to reduce the Letters data set with a 20% increment, and 4.253 seconds to reduce the data with 30% increment. The main reason is that when the data increases, it accelerates the division of data on conditional attributes. Since the amount of information is zero based on subdivision of cardinality 1, these objects are constantly deleted during the reduction process to accelerate the convergence speed.

5.3. Time Comparison of IEPDO and IEP When Deleting Data Objects. The same as Section 5.2, nine data sets are selected in Table 7. Take the original data of each data sets as the basic data, and randomly select 10%, 20%, 30%, 40%, and 50% objects in the remaining data to delete respectively. Use IEP and IEPDO to reduce the updated data. Each group of data was selected to repeat the experiment for 10 times and averaged the time of 10 times. The experimental results show in Figure 2. In each subfigure of Figure 2, $x$-axis represents the proportion of decrement data objects, while $y$-axis represents the computational time. The time is measured by seconds. The curve with five-pointed star mark in Figure 2 shows the change in the running time of the IEPDO, while as shown in Figure 2, when we delete some data objects from data set, the computational time of IEP and IEPDO will decrease accordingly. In the same computing environment, IEPDO takes less time than IEP. The IEPDO method only needs to calculate the updated objects when deleting data objects. Some updated objects when deleting data objects. Some conditional attributes with zero indistinguishable element-pair are removed from the reduction set. However, IEP takes longer time because of calculating the reduced data set from all conditional attributes. For the IEP selected, it takes 0.0338 seconds on the decreased 30% data objects of Hepatitis, while taking 0.0352 seconds to compute the deleted 40% data. It costs 1.233 seconds to calculate the decrease of 30% data of Nursery but takes 1.2451 seconds on the decrease of 40% data. Why the calculation time increases with the decreasing data is that the speed of dividing equivalence class is slowed down as it randomly selects some weak distinguish power on conditional attributes. The performance of IEPDO is relatively stable. By deleting data, the calculation time decreases. From nine sub-figure of Figure 2, when we reduce the large-scale data, the time-consuming effect of IEPDO is more obvious with the decreased data objects, while the effect is not significant on small data sets, such as data sets of Chess kr-kp and Hepatitis.

5.4. Classification Accuracy Analysis of IEP, IEPAO, and IEPDO. In this section, the precision of classification is calculated on the selection of reducts obtained by the algorithms IEP, IEPAO, and IEPDO. Firstly, we take 50% of objects of each data set in nine data sets from Table 7 as the basic data set, the rest 50% data as the incremental objects
Table 8: A speedup ratio comparison of IEP, MEDA, HU, and DDMSE.

<table>
<thead>
<tr>
<th>Data set</th>
<th>IEP</th>
<th>MEDA</th>
<th>HU</th>
<th>DDMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom</td>
<td>81.916</td>
<td>4.3723</td>
<td>1</td>
<td>0.6954</td>
</tr>
<tr>
<td>Audiology</td>
<td>1.6109</td>
<td>1.5839</td>
<td>1</td>
<td>0.7454</td>
</tr>
<tr>
<td>Blance-scale</td>
<td>17.857</td>
<td>9.1912</td>
<td>1</td>
<td>1.1220</td>
</tr>
<tr>
<td>Breast</td>
<td>5.4571</td>
<td>4.3409</td>
<td>1</td>
<td>1.2566</td>
</tr>
<tr>
<td>Car</td>
<td>41.639</td>
<td>10.5291</td>
<td>1</td>
<td>1.2320</td>
</tr>
<tr>
<td>Letters</td>
<td>∞</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Ticdata2000</td>
<td>19.3021</td>
<td>18.1108</td>
<td>1</td>
<td>1.5578</td>
</tr>
<tr>
<td>Gene</td>
<td>26.4713</td>
<td>4.5756</td>
<td>1</td>
<td>0.5976</td>
</tr>
<tr>
<td>Nursery</td>
<td>1397.65</td>
<td>34.368</td>
<td>1</td>
<td>3.5097</td>
</tr>
<tr>
<td>Handwritten</td>
<td>54.5751</td>
<td>1.0171</td>
<td>1</td>
<td>0.4773</td>
</tr>
<tr>
<td>Mass</td>
<td>22.3890</td>
<td>13.4333</td>
<td>1</td>
<td>1.0824</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>2.42862</td>
<td>1.8478</td>
<td>1</td>
<td>0.9884</td>
</tr>
<tr>
<td>Connect-4</td>
<td>∞</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Chess kr-kr</td>
<td>26.7601</td>
<td>1.797</td>
<td>1</td>
<td>1.3320</td>
</tr>
<tr>
<td>Spect heart</td>
<td>10.0600</td>
<td>2.086</td>
<td>1</td>
<td>0.9860</td>
</tr>
</tbody>
</table>

The bolded values are the fastest speed ratio.

Table 9: A time and reduced comparison IEP, MEDA, HU, and DDMSE.

| Data sets   | | IEP | | MEDA | | HU | | DDMSE |
|-------------| | Time (s) | | | | Time (s) | | | Time (s) |
| Mushroom    | | 4 | | 1.953 | | 5 | | 36.59 | | 4 | | 159.981 | | 4 | | 230.071 |
| Audiology   | | 14 | | 0.293 | | 14 | | 0.298 | | 13 | | 0.472 | | 14 | | 0.633 |
| Blance-scale| | 4 | | 0.035 | | 4 | | 0.068 | | 4 | | 0.625 | | 4 | | 0.557 |
| Breast      | | 9 | | 0.035 | | 9 | | 0.044 | | 9 | | 0.191 | | 9 | | 0.152 |
| Car         | | 6 | | 0.133 | | 6 | | 0.526 | | 6 | | 5.338 | | 6 | | 4.495 |
| Letters     | | 12 | | 5.892 | | — | | — | | ∞ | | — | | — | | ∞ |
| Ticdata2000 | | 23 | | 7.724 | | 23 | | 8.232 | | 23 | | 149.085 | | 23 | | 95.701 |
| Gene        | | 10 | | 2.593 | | 10 | | 15.001 | | 10 | | 68.639 | | 10 | | 114.863 |
| Nursery     | | 8 | | 1.32 | | 8 | | 53.679 | | 8 | | 1844.866 | | 8 | | 525.649 |
| Handwritten | | 7 | | 4.688 | | 8 | | 251.558 | | 7 | | 255.848 | | 7 | | 536.041 |
| Mass        | | 5 | | 0.054 | | 5 | | 0.09 | | 5 | | 1.209 | | 5 | | 1.117 |
| Hepatitis   | | 8 | | 0.039 | | 9 | | 0.046 | | 8 | | 0.085 | | 8 | | 0.086 |
| Connect-4   | | 34 | | 64.358 | | — | | — | | ∞ | | — | | — | | ∞ |
| Chess kp-kr | | 29 | | 2.22 | | 29 | | 33.059 | | 29 | | 59.407 | | 29 | | 44.568 |
| Spect heart | | 18 | | 0.035 | | 18 | | 0.1685 | | 18 | | 0.3521 | | 18 | | 0.357 |

The time shown in bold is the least time.

Figure 1: Continued.
Figure 1: The time comparison of IEP and IEPAO when adding data objects. (a) Chess kr-kp. (b) Connect-4. (c) Gene. (d) Handwritten. (e) Hepatitis. (f) Letters. (g) Mushroom. (h) Nursery. (i) Ticdata2000.

Figure 2: Continued.
and select the algorithms IEP and IEPAO to reduce. Secondly, we delete 50% of objects randomly from each dataset, using the algorithms IEP and IEPAO to process. Then, the classification accuracies are acquired by using J48, Naive-Bayes (NB), RandomForest (RF), SMO classifier, and 10-fold cross-validation. The experimental results are shown in Tables 10 and 11.

From Table 10, it is clear that when some objects are added into the information systems, the average classification accuracy of the reduction found by incremental algorithm IEPAO is better than those of algorithm IEP in data sets Chess, Connect-4, Gene, Handwritten, Hepatitis, Letters and Ticdata2000 are coincide with those of algorithm IEP in datasets, e.g., Chess kr-kp, Mushroom, and Nursery. The experimental results show that the incremental algorithm IEPAO can find a feasible attribute reduction when incremental algorithm IEPAO replaces algorithm IEP. Moreover, the algorithm IEPAO can obtain high-quality attribute reduction with less time consumption. Similarly, when some objects are deleted from the original object set, the average classification accuracy of the reduction obtained by the algorithm IEPDO is better than

<table>
<thead>
<tr>
<th>Data set</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
<th>J48 (%)</th>
<th>IEPAO (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chess kr-kp</td>
<td>99.4368</td>
<td>99.4368</td>
<td>99.0926</td>
<td>99.0926</td>
<td>88.3292</td>
<td>88.3292</td>
<td>95.4318</td>
<td>95.4318</td>
<td>95.4318</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connect-4</td>
<td>80.9006</td>
<td>83.6573</td>
<td>82.1099</td>
<td>85.0756</td>
<td>72.2027</td>
<td>74.7908</td>
<td>80.2002</td>
<td>82.4056</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene</td>
<td>65.7053</td>
<td>68.0923</td>
<td>70.3762</td>
<td>72.9031</td>
<td>67.2424</td>
<td>69.6412</td>
<td>66.6771</td>
<td>69.4532</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handwritten</td>
<td>65.3203</td>
<td>67.5629</td>
<td>71.9757</td>
<td>72.9084</td>
<td>62.6868</td>
<td>64.0985</td>
<td>67.4021</td>
<td>70.2341</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hepatitis</td>
<td>79.3548</td>
<td>82.5094</td>
<td>82.5806</td>
<td>84.5620</td>
<td>80.0001</td>
<td>83.8007</td>
<td>80.0001</td>
<td>83.0924</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Letters</td>
<td>87.5150</td>
<td>89.3318</td>
<td>95.2050</td>
<td>95.2050</td>
<td>60.2500</td>
<td>62.6780</td>
<td>76.6250</td>
<td>79.9138</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mushroom</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>98.6091</td>
<td>98.6091</td>
<td>100.00</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nursery</td>
<td>97.0525</td>
<td>97.0525</td>
<td>99.0664</td>
<td>99.0664</td>
<td>90.3241</td>
<td>90.3241</td>
<td>93.0787</td>
<td>93.0787</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticdata2000</td>
<td>82.0225</td>
<td>84.4521</td>
<td>80.1498</td>
<td>83.4097</td>
<td>77.9026</td>
<td>79.8094</td>
<td>82.7157</td>
<td>86.7732</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The bold values are the classification accuracy with the best classification performance.

Accordingly, we can conclude that the incremental algorithm IEPDO can find a feasible attribute reduction.

Hence, the experimental results verified that the proposed incremental methods IEPAO and IEPDO can obtain an efficient attribute reduction and provide a quick data preprocessing method for dynamic data sets.

6. Conclusions and Further Study

Attribute reduction can effectively eliminate redundant information. Though the discernibility matrix method is one of the intuitive and effective reduction methods, it cannot deal with the reduction of large-scale data sets effectively because of memory overflow. The attribute reduction mechanism based on IEP can effectively solve the problem of space consumption analyzed in this paper. During the reduction process, IEP effectively prunes the subdivisions with cardinality 1, which speeds up the calculation of equivalence class division. IEP has better time and space effects in reduction, but it only adapts to the environment of static data sets. Considering the constant updating of data in reality, IEPAO and IEPDO are proposed on the basis of IEP to deal with the reduction of adding data objects and deleting data objects respectively. As to IEP, the entire data set has to be reduced again and consumed a lot of time with the data changes. IEPAO and IEPDO only compute the changed part data and combine the previous reduction results, which can obtain the data set with fewer redundancies and better outcomes.

Of course, the algorithm proposed has some shortcomings in this paper. For example, (1) The IEP method can only reduce integer or character data, but cannot adapt to process other types of data. (2) The incremental update algorithm proposed in this paper does not consider the changes in attributes and values.

In the future, we will conduct the research from the following aspects: design an increment algorithm adapting to different types of data; develop a reduction method regarding the change values of data objects; propose an incremental mechanism with adding and deleting some attributes. Additionally, those approaches should adapt to an incomplete decision system.

Appendix

The proof of Theorem 5

Proof. From Definition 8, we have

\[ \text{IEP}_{\text{new}}(DC) = \sum_{i=1}^{n} \text{IEP}_{\text{old}}(DC) + \frac{1}{2} \sum_{i=1}^{n} |X_i| \]

(A.1)
Data Availability
All the data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This work was partially supported by the Natural Science Foundation of China (61836016), the Quality Improvement Project of Chaohu University on Discipline Construction (kj21gczx03), Special Support Plan for Innovation and Entrepreneurship Leaders in Anhui Province, the Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province (KJ2021A1030), and the Key Subject Subprojects of Chaohu University ZDXK-201815.

References


