International Journal of Approximate Reasoning 187 (2025) 109544

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

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

Relative pre-reducts for computing the relative reducts of large data sets

Hajime Okawa [a], Yasuo Kudo [a,*], Tetsuya Murai [b]

[a] Graduate School of Engineering, Muroran Institute of Technology, 27-1 Mizumoto, Muroran, 050-8585, Hokkaido, Japan
[b] Department of Information Systems Engineering, Chitose Institute of Science and Technology, 758-65 Bibi, Chitose, 066-8655, Hokkaido, Japan

ARTICLE INFO

Keywords:
Attribute reduction
Relative reduct
Relative pre-reduct
Rough set theory

ABSTRACT

In this paper, we introduce the concept of relative pre-reducts to derive the relative reducts from a large dataset. The relative reduct is considered a consistency-based attribute reduction method that is commonly utilized to extract concise subsets of condition attributes. Nonetheless, calculating all relative reducts necessitates substantial time and memory to build a discernibility matrix. In this research, we demonstrate that all relative pre-reducts can be computed using a simplified matrix referred to as the partial discernibility matrix, which can be readily converted into relative reducts. We also suggest employing a data partitioning approach to generate the discernibility matrix. This method alleviates the issue of an increased number of results for each partition. The outcomes from this technique yield the relative pre-reducts proposed in this study. Since our enhancements to the computation of relative reducts are independent of other advancements, they can be implemented in conjunction with existing methods. Experimental findings indicate that utilizing relative pre-reducts for computing relative reducts is efficient for large datasets.

1. Introduction

Currently, attribute reduction (also known as feature selection [1]) is a critical aspect of data mining and machine learning. This process entails selecting a subset of attributes based on established criteria, subsequently eliminating any redundant attributes from this resulting set.

In the realm of machine learning, the presence of redundant attributes can result in decreased accuracy [2]. Furthermore, an increase in the number of attributes leads to a rise in model complexity. This complexity impacts not only model predictions but also the processes of model construction and learning, as well as the analysis of the models. Hence, attribute reduction has been extensively researched for its applicability in machine learning.

Various measures are employed for attribute reduction, including distance-based measures, probability-based measures, mutual information-based measures, consistency measures, and neighborhood graph-based measures. Within rough set theory, subsets of attributes are predominantly selected using the consistency measure [3]. This measure assesses the number of inconsistent samples and is not dependent on specific models. Inconsistent samples are those where attribute values are identical, yet the classes differ.

Relative reducts, based on rough sets, were introduced by Pawlak as a method for consistency-based attribute reduction [4]. Typically, there are two or more relative reducts; however, calculating all such reducts is NP-hard [5]. Numerous extended rough

* Corresponding author.
E-mail address: yask@muroran-it.ac.jp (Y. Kudo).

https://doi.org/10.1016/j.ijar.2025.109544
Received 24 March 2025; Received in revised form 11 June 2025; Accepted 1 August 2025

Available online 7 August 2025

Table 1
Example of a decision table.

Patient	Condition attributes				Decision attribute
	Headache	Sneeze	Temperature	Muscle pain	Flu
p_1	no	no	very high	yes	yes
p_2	no	yes	high	yes	yes
p_3	yes	no	high	no	no
p_4	yes	yes	high	no	yes
p_5	no	no	very high	yes	no
p_6	yes	no	normal	no	no

set models have emerged over the years, offering attribute reduction grounded in various measures. For instance, attribute reduction methods utilizing neighborhood rough set models (e.g., [6]) generally focus on distance-based measures. Conversely, the method [7] that is based on fuzzy rough set models leverages mutual information-based measures. The k-nearest neighbor rough set model provides attribute reduction based on neighborhood graph-based measures [8].

In response to these challenges, Kudo et al. [9,10] devised a method to identify some relative reducts using a probabilistic algorithm. This approach is beneficial for datasets containing a vast number of attributes. However, it may exceed available memory when handling datasets with a large number of samples due to the construction of discernibility matrices.

Several methods [11–14] have been proposed to tackle the computational complexity associated with the number of samples. The approach proposed by Chen et al. [11] mitigates excessive memory usage by incrementally simplifying the matrix. The other methods [12–14] segment a given table into groups of samples and compile the results obtained from each group. Since these two categories of improvements are independent, they can be implemented sequentially. However, the latter category of methods faces the challenge of potential overgrowth in the number of results within each group, thereby increasing computation time for the aggregation step. No experiments were reported in [13,14], while Piotr et al. [12] identified cases where calculations took longer than previous approaches.

In this study, we suggest employing a data partitioning approach to construct the discernibility matrix. This method successfully avoids the challenge of an increasing number of results for each segment. The output of this technique comprises the relative pre-reducts proposed in this study, which can be readily converted into relative reducts. Additionally, there are two versions of this approach; one can be integrated with the methods used by Kudo et al. [9,10]. Consequently, this method remains effective even when both sample and feature counts are substantial. Experimental results confirm that our approach yields some relative reducts more quickly than methods without partitioning when a considerable number of samples are present.

This research is an updated and expanded version of our contribution presented at the SCIS&ISIS 2022 conference [15]. The partial discernibility matrix method detailed in [15] is effectively comparable to the method introduced in this paper. However, the inclusion of relative pre-reducts in the current work allows us to demonstrate that the procedure accurately enumerates the relative reducts. Additionally, the development of a more efficient algorithm for generating relative pre-reducts is planned for future research.

The subsequent sections will cover the fundamental concepts associated with relative reducts and core attributes in rough set theory (Section 2), introduce relative pre-reducts along with algorithms for relative reducts based on these fundamental concepts (Section 3), outline the experimental findings (Section 4), discuss the methodology (Section 5), and offer concluding remarks (Section 6).

2. Preliminaries

Relative reducts are established using the lower approximation boundary found in decision tables. In this section, we will review the concepts of decision tables, lower approximations, and relative reducts. Additionally, we will discuss the computational methods employed for relative reducts and their intersection, referred to as the core. The content in this section primarily draws from [4,16].

2.1. Decision tables

Decision tables are used to represent the datasets that facilitate rough set data analysis. Formally, a *decision table* DT is defined as follows:

$$DT := (U, C \cup \{\mathsf{d}\}), \tag{1}$$

where U is a nonempty finite set of objects, C is a nonempty finite set of condition attributes, and $\mathsf{d} \notin C$ is the decision attribute. Each attribute $\mathsf{a} \in C \cup \{\mathsf{d}\}$ is a function from U to V_{a}, where V_{a} represents the complete set of values for attribute a.

Table 1 provides an example of a decision table, where

- $U = \{p_1, p_2, \ldots, p_6\}$,
- $C = \{\mathsf{Headache}, \mathsf{Sneeze}, \mathsf{Temperature}, \mathsf{Muscle\ pain}\}$, and
- $\mathsf{d} = \mathsf{Flu}$.

The value assigned to the attribute a for object $x \in U$ is found by identifying the intersection of the column for a and the row for x, e.g., $\mathsf{Headache}(p_1) =$ “no” as shown in Table 1.

2.2. Lower approximation and the positive region

In the context of rough set theory, the lower approximation consists solely of consistent objects, meaning all objects sharing identical attribute values (i.e., indiscernible objects) belong to the same decision class. Indiscernibility here indicates equivalence across all condition attributes. The indiscernibility relation for a subset $B \subseteq C$ is defined as follows:

$$IND(B) := \{(x, y) \in U \times U \mid \mathsf{b}(x) = \mathsf{b}(y), \forall \mathsf{b} \in B\}. \tag{2}$$

Given that the indiscernibility relation $IND(B)$ is an equivalence relation, for any object $x \in U$, we denote the equivalence class containing x as $[x]_B := \{y \in U \mid (x, y) \in IND(B)\}$. We denote the quotient set as $U/IND(B) := \{[x]_B \mid x \in U\}$. Specifically, the quotient set $U/IND(\{\mathsf{d}\})$ is referred to as $\mathcal{D}$, with its elements known as *decision classes*. For any two subsets $A, B \subseteq C$, the following properties are valid:

$$A \subseteq B \Longrightarrow [x]_B \subseteq [x]_A. \tag{3}$$

$$[x]_{A \cup B} = [x]_A \cap [x]_B. \tag{4}$$

For our example in Table 1, let $A = \{\mathsf{Headache}, \mathsf{Sneeze}\}$. Hence, we find:

$$\begin{aligned} IND(A) = \{&(p_1, p_1), (p_1, p_5), (p_2, p_2), (p_3, p_3), (p_3, p_6), \\ &(p_4, p_4), (p_5, p_1), (p_5, p_5), (p_6, p_3), (p_6, p_6)\}, \end{aligned} \tag{5}$$

which leads to

$$U/IND(A) = \{\{p_1, p_5\}, \{p_2\}, \{p_3, p_6\}, \{p_4\}\}. \tag{6}$$

The decision class set $\mathcal{D}$ is:

$$\mathcal{D} = \{\{p_1, p_2, p_4\}, \{p_3, p_5, p_6\}\}. \tag{7}$$

For any subset $A \subseteq C$ and any decision class $D \in \mathcal{D}$, the *lower approximation* $\underline{A}(D)$ is defined as follows:

$$\underline{A}(D) := \{x \in U \mid [x]_A \subseteq D\}. \tag{8}$$

As an illustration, let's calculate the lower approximations as outlined in Table 1. Let $D_1 = \{p_1, p_2, p_4\}$, $D_2 = \{p_3, p_5, p_6\}$, and $A = \{\mathsf{Headache}, \mathsf{Sneeze}\}$. Consequently, we have:

- $\underline{A}(D_1) = \{p_2, p_4\}$,
- $\underline{A}(D_2) = \{p_3, p_6\}$.

The lower approximation is characterized by the following properties:

Proposition 2.1. *Given a decision table $DT = (U, C \cup \{\mathsf{d}\})$ and two subsets A, B of C, the following properties are established for any decision classes $D, D_1, D_2 \in \mathcal{D}$ of DT.*

(i) $\underline{A}(D) \subseteq D$
(ii) $D = U \Longrightarrow \underline{A}(D) = D$
(iii) $A \subseteq B \Longrightarrow \underline{A}(D) \subseteq \underline{B}(D)$
(iv) $D_1 \neq D_2 \Longrightarrow \underline{A}(D_1) \cap \underline{A}(D_2) = \emptyset$

Attribute reduction based on rough set theory aims to identify the optimal subsets of condition attributes, known as relative reducts, while considering the consistency of objects throughout the decision table. To achieve this, we define the positive region of the decision table. Let the decision table DT be $(U, C \cup \{\mathsf{d}\})$, $A \subseteq C$, and let $\mathcal{D}$ represent the collection of all decision classes of DT. The *positive region* $Pos_A(\mathcal{D})$ of DT is then defined as:

$$Pos_A(\mathcal{D}) := \bigcup_{D \in \mathcal{D}} \underline{A}(D). \tag{9}$$

The positive region represents the disjoint union of the lower approximations associated with all decision classes. The following properties derive from Proposition 2.1:

Proposition 2.2. *Given a decision table $DT = (U, C \cup \{\mathsf{d}\})$ and subsets $A, B \subseteq C$, the following properties hold:*

(i) $Pos_A(\mathcal{D}) \subseteq U$,
(ii) $A \subseteq B \Longrightarrow Pos_A(\mathcal{D}) \subseteq Pos_B(\mathcal{D})$.

2.3. Relative reducts and core

Relative reducts are defined as the minimal subsets of condition attributes that maintain the consistency of objects. Given $DT = (U, C \cup \{\mathsf{d}\})$ and the set $\mathcal{D}$ of all decision classes of DT, a subset $R \subseteq C$ is termed a *relative reduct* of DT if:

(i) $Pos_R(\mathcal{D}) = Pos_C(\mathcal{D})$, and
(ii) R is minimal with respect to (i).

In general, a decision table can have multiple relative reducts. We denote the set of all relative reducts for a decision table DT as $Red(DT)$.

The intersection of all relative reducts is referred to as the *core* of DT, represented as $Core(DT)$:

$$Core(DT) := \bigcap_{R \in Red(DT)} R. \tag{10}$$

It is important to note that $Core(DT)$ consists of a set of condition attributes. The following properties are established.

Proposition 2.3. *Let $DT = (U, C \cup \{\mathsf{d}\})$ be a decision table, and let $\mathcal{D}$ denote the set of all decision classes of DT. For any subset $A \subseteq C$, the following statements hold:*

(i) $(\exists c_0 \in Core(DT), c_0 \notin A) \Longrightarrow Pos_A(\mathcal{D}) \subsetneq Pos_C(\mathcal{D})$,
(ii) $Pos_A(\mathcal{D}) = Pos_C(\mathcal{D}) \Longrightarrow Core(DT) \subseteq A$.

For instance, the collection of all the relative reducts for Table 1 is

$$\{\{\mathsf{Headache}, \mathsf{Sneeze}\}, \{\mathsf{Sneeze}, \mathsf{Temperature}\}, \{\mathsf{Sneeze}, \mathsf{Muscle\ pain}\}\}, \tag{11}$$

with the core being $\{\mathsf{Sneeze}\}$.

2.4. Enumeration of the relative reducts

This section reviews various computational methods [5,9–11] aimed at obtaining multiple relative reducts from a decision table. Techniques referenced in [9–11] are enhancements of the method formulated by Skowron and Rauszer [5] and can be utilized in unison.

2.4.1. Discernibility matrix

Herein, we introduce the discernibility matrix proposed by Skowron and Rauszer [5]. Given a decision table $DT = (U, C \cup \{\mathsf{d}\})$ where $U = \{x_1, x_2, \ldots, x_n\}$, the *discernibility matrix* DM of DT is an $n \times n$ symmetric matrix. The element located at the i-th row and j-th column of DM comprises the following set of condition attributes:

$$\delta_{ij} = \begin{cases} \left\{\mathsf{a} \in C \mid \mathsf{a}(x_i) \neq \mathsf{a}(x_j)\right\}, & \text{if } \mathsf{d}(x_i) \neq \mathsf{d}(x_j) \text{ and } \{x_i, x_j\} \cap Pos_C(\mathcal{D}) \neq \emptyset, \\ *, & \text{otherwise,} \end{cases} \tag{12}$$

where "$*$" represents a wildcard symbol. δ_{ij} ($\neq *$) implies that object x_i in one decision class can be distinguished from object x_j in another decision class by the value of at least one attribute $\mathsf{a} \in \delta_{ij}$. The notation $\delta_{ij} \in DM$ indicates that δ_{ij} is an element of the discernibility matrix DM. By definition, constructing the discernibility matrix involves a time and space complexity of $O(|U|^2 \times |C|)$, where $|X|$ denotes the cardinality of set X.

Theorem 2.4. [5] *Given $DT = (U, C \cup \{\mathsf{d}\})$ and its discernibility matrix $DM = (\delta_{ij})$, for any subset $Q \subseteq C$, the following two conditions are equivalent:*

1. $\forall \delta_{ij} \in DM, \delta_{ij} \neq * \Longrightarrow \delta_{ij} \cap Q \neq \emptyset$,
2. $Pos_Q(\mathcal{D}) = Pos_C(\mathcal{D})$.

Corollary 2.4.1. [5] *For the decision table $DT = (U, C \cup \{\mathsf{d}\})$ and its discernibility matrix $DM = (\delta_{ij})$, a subset $Q \subseteq C$ is minimal with respect to Condition 1 iff Q is minimal concerning Condition 2.*

1. $\forall \delta_{ij} \in DM, \delta_{ij} \neq * \Longrightarrow \delta_{ij} \cap Q \neq \emptyset$,
2. $Pos_Q(\mathcal{D}) = Pos_C(\mathcal{D})$.

Condition 2 of Corollary 2.4.1 aligns with Condition 1 of the relative reducts. To ascertain all relative reducts, we only need to compute all minimal sets of condition attributes that satisfy Condition 1 of Corollary 2.4.1.

Table 2
Discernibility matrix for Table 1.

	p_1	p_2	p_3	p_4	p_5	p_6
p_1	*					
p_2	*	*				
p_3	{H, T, M}	{H, S, M}	*			
p_4	*	*	{S}	*		
p_5	*	{S, T}	*	{H, S, T, M}	*	
p_6	{H, T, M}	{H, S, T, M}	*	{S, T}	*	*

To derive these sets, we extract the logical formula from the discernibility matrix and transform it into its minimal disjunctive normal form (DNF). The logical formula $L(DM)$ for DM is defined as follows:

$$L(DM) : \bigwedge_{1 \le j < i \le n,\ \delta_{ij} \neq *} L(\delta_{ij}), \tag{13}$$

where $L(\delta_{ij})$ represents the logical formula for the element δ_{ij} $(\neq *)$ in DM and is defined as follows:

$$L(\delta_{ij}) : \bigvee_{\mathsf{a} \in \delta_{ij}} \mathsf{a}. \tag{14}$$

$L(DM)'$ denotes the minimal DNF of $L(DM)$; $L(DM)'$ will then take the following form:

$$L(DM)' : \bigvee_{R \in Red(DT)} \bigwedge_{\mathsf{a} \in R} \mathsf{a}. \tag{15}$$

To convert $L(DM)$ into its minimal DNF, we must apply the idempotent law, absorption law, and distributive law, iteratively. Susmaga [17] has suggested parallel computation, and Bao et al. [18] have proposed a rapid algorithm for this transformation; however, the algorithm remains NP-hard.

Table 2 presents the discernibility matrix for Table 1, where the attribute names are abbreviated to their initial letters. Using Table 2, we construct $L(DM)$ and transform it into its minimal DNF:

$$\begin{aligned}
&(\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}) \wedge (\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}) \wedge (\mathsf{H} \vee \mathsf{S} \vee \mathsf{M}) \wedge (\mathsf{S} \vee \mathsf{T}) \\
&\wedge (\mathsf{H} \vee \mathsf{S} \vee \mathsf{T} \vee \mathsf{M}) \wedge \mathsf{S} \wedge (\mathsf{H} \vee \mathsf{S} \vee \mathsf{T} \vee \mathsf{M}) \wedge (\mathsf{S} \vee \mathsf{T}) \\
\equiv &(\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}) \wedge \mathsf{S} \\
\equiv &(\mathsf{H} \wedge \mathsf{S}) \vee (\mathsf{S} \wedge \mathsf{T}) \vee (\mathsf{S} \wedge \mathsf{M}).
\end{aligned}$$

Thus, we obtain all the relative reducts $\{\mathsf{Headache}, \mathsf{Sneeze}\}, \{\mathsf{Sneeze}, \mathsf{Temperature}\}, \{\mathsf{Sneeze}, \mathsf{Muscle\ pain}\}$ from Table 1.

2.4.2. Reduced decision tables

The reduced decision tables method, suggested by Kudo et al. [9,10], is designed to compute multiple relative reducts from a decision table that contains numerous attributes.

Let $DT = (U, C \cup \{\mathsf{d}\})$ be a decision table and let $DM = (\delta_{ij})$ represent its discernibility matrix. If a subset $C' \subseteq C$ fulfills the following condition:

$$\forall \delta_{ij} \in DM, \delta_{ij} \neq * \Longrightarrow \delta_{ij} \cap C' \neq \emptyset, \tag{16}$$

then the decision table $RDT = (U, C' \cup \{\mathsf{d}\})$ is referred to as a *reduced decision table* of DT.

Theorem 2.5. [9,10] *For a decision table $DT = (U, C \cup \{\mathsf{d}\})$, if $RDT = (U, C' \cup \{\mathsf{d}\})$ is its reduced decision table, and $R \subseteq C'$ is a subset of the condition attributes, then R is a relative reduct of RDT iff R is a relative reduct of DT.*

Algorithm 1 generates a reduced decision table from a given decision table. The algorithm incorporates randomization to yield different tables with each execution. By determining the relative reducts from the reduced decision tables, we can subsequently find the relative reducts for the original decision table.

2.5. Calculation of the core attributes

There exist several algorithms [5,19,20] that are capable of computing all the core attributes of a decision table. In terms of computational complexity, we examine the approach described by Ge et al. [20], which yields the following results.

Theorem 2.6. [20] *Let $DT = (U, C \cup \{\mathsf{d}\})$ represent a decision table. For any attribute $\mathsf{c} \in C$, the following two conditions are equivalent:*

1. $\mathsf{c} \in Core(DT)$,

Algorithm 1 Reduced decision table computation algorithm [10].

Input: decision table $DT = (U, C \cup \{\mathtt{d}\})$, the discernibility matrix DM of DT, the number b of the attributes selected
Output: reduced decision table $RDT = (U, C' \cup \{\mathtt{d}\})$

Select b attributes $\mathtt{c}_1, \ldots, \mathtt{c}_b$ from C at random by sampling without replacement
$C' \leftarrow \{\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathtt{c}_b\}$
for all $\delta_{ij} \in DM$ s.t. $i > j$ **do**
  if $\delta_{ij} \neq *$ **and** $\delta_{ij} \cap C' = \emptyset$ **then**
    Select $\mathtt{c} \in \delta_{ij}$ at random.
    $C' \leftarrow C' \cup \{\mathtt{c}\}$
  end if
end for
return $(U, C' \cup \{\mathtt{d}\})$

Algorithm 2 Computation of all core attributes [20].

Input: a decision table $DT = (U, C \cup \{\mathtt{d}\})$
Output: the set $Core$ of all core attributes of DT

$Core \leftarrow \emptyset$
for all $\mathtt{c} \in C$ **do**
  if $Pos_{C\setminus\{\mathtt{c}\}}(\mathcal{D}) \neq Pos_C(\mathcal{D})$ **then**
    $Core \leftarrow Core \cup \{\mathtt{c}\}$
  end if
end for
return $Core$

2. $Pos_{C\setminus\{\mathtt{c}\}}(\mathcal{D}) \neq Pos_C(\mathcal{D})$.

Therefore, we can determine all core attributes using Algorithm 2, which has a complexity of $O(|U| \times |C|^2)$. It is noteworthy that the complexity to compute the positive regions is $O(|U| \times |C|)$ [21].

3. Relative pre-reducts

Within the framework of relative pre-reducts, the redundancy of specific attributes is not assessed. However, when such attributes qualify as core attributes, evaluating their redundancy is redundant since, according to Proposition 2.3, they are inherently regarded as non-redundant. When there is only one attribute left to assess, it suffices to compute the positive region with respect to all other attributes to determine its redundancy.

In the subsequent sections, we will present the definition of relative pre-reducts for a decision table and demonstrate how relative pre-reducts can be easily converted into relative reducts.

3.1. Definition of the relative pre-reducts

Relative pre-reducts are defined for every subset of condition attributes $B \subseteq C$; therefore, we refer to the outcomes as relative B-reducts.

Definition 3.1. Let $DT = (U, C \cup \{\mathtt{d}\})$ signify a decision table, and let $B \subseteq C$ denote a subset of condition attributes. A subset $R \subseteq C$ is termed a relative B-reduct of DT if it satisfies the following conditions:

(i) $Pos_{R\cup B}(\mathcal{D}) = Pos_C(\mathcal{D})$, and
(ii) R is minimal with respect to (i).

In general, there may be multiple relative B-reducts for a decision table DT; hence, the set of relative B-reducts is denoted as $Red^{\cup B}(DT)$.

Proposition 3.2. *Let $DT = (U, C \cup \{\mathtt{d}\})$ be a decision table, and let $B \subseteq C$ be a subset of condition attributes. For any relative B-reducts $R \in Red^{\cup B}(DT)$, the following two conditions hold:*

1. $\forall \mathtt{c} \in R, \mathtt{c} \notin B$,
2. $Pos_{R\setminus\{\mathtt{c}\}\cup B}(\mathcal{D}) \neq Pos_C(\mathcal{D}), \forall \mathtt{c} \in R$.

Proof. (1) Suppose that there exists $\mathtt{c} \in R$ such that $\mathtt{c} \in B$. Then $R \cup B = (R \setminus \{\mathtt{c}\}) \cup B$, hence

$$Pos_{R\setminus\{\mathtt{c}\}\cup B}(\mathcal{D}) = Pos_{R\cup B}(\mathcal{D}) = Pos_C(\mathcal{D}). \tag{17}$$

This contradicts the minimality of R.

(2) Because R is minimal with respect to the condition $Pos_{R\cup B}(\mathcal{D}) = Pos_C(\mathcal{D})$, then $Pos_{(R\setminus\{\mathsf{c}\})\cup B}(\mathcal{D}) \neq Pos_C(\mathcal{D})$. (Note that because $\mathsf{c} \notin B$, $(R\setminus\{\mathsf{c}\})\cup B \subsetneq R\cup B$.) □

3.2. Calculation method for the relative pre-reducts

We propose a method to compute the relative pre-reducts using a partial discernibility matrix [15].

Definition 3.3. [15] Let $DT = (U, C\cup\{\mathsf{d}\})$ denote a decision table, and let DM represent its discernibility matrix. The element at the i-th row and the j-th column in DM is denoted by δ_{ij}. If $B \subsetneq C$ is a nonempty subset of the condition attributes and n is the number of objects in DT (i.e., $n = |U|$), then the partial discernibility matrix $PDM(B)$ of DT with respect to B is defined as a symmetric $n\times n$ matrix, where the element at the i-th row and j-th column is defined as follows:

$$\delta_{ij}^B = \begin{cases} \delta_{ij}, & \text{if } (x_i, x_j) \in IND(B), \\ *, & \text{otherwise,} \end{cases} \tag{18}$$

where "$*$" is a wildcard symbol.

The notation $\delta_{ij}^B \in PDM(B)$ indicates that δ_{ij}^B is an element of the partial discernibility matrix $PDM(B)$. By definition, the time and space complexity for constructing this matrix is $O\left(|U|\times|C| + \sum_{E\in U/IND(B)} |E|^2\times|C|\right)$. It is important to note that if $|U/IND(B)| \geq 2$, then $|U|^2 > \sum_{E\in U/IND(B)} |E|^2$.

The following theorem and its corollary illustrate that relative B-reducts can be calculated using the partial discernibility matrix with respect to B.

Theorem 3.4. *Given $DT = (U, C\cup\{\mathsf{d}\})$, $B \subseteq C$ and its partial discernibility matrix $PDM(B) = \left(\delta_{ij}^B\right)$, for any subset $Q \subseteq C$, the following two conditions are equivalent:*

1. $\forall \delta_{ij}^B \in PDM(B), \delta_{ij}^B \neq * \Longrightarrow \delta_{ij}^B \cap Q \neq \emptyset$,
2. $Pos_{Q\cup B}(\mathcal{D}) = Pos_C(\mathcal{D})$.

Proof. ($\Rightarrow$) Because, by Proposition 2.2, $Pos_A(\mathcal{D}) \subseteq Pos_C(\mathcal{D})$ for any $A \subseteq C$, it is sufficient to show that $Pos_C(\mathcal{D}) \subseteq Pos_{Q\cup B}(\mathcal{D})$. Suppose that $x_i \in Pos_C(\mathcal{D})$ and $x_i \notin Pos_{Q\cup B}(\mathcal{D})$. Then for any $D \in \mathcal{D}$, $[x_i]_{Q\cup B} \nsubseteq D$, i.e., there exists some $x_j \in [x_i]_{Q\cup B}$ such that $\mathsf{d}(x_i) \neq \mathsf{d}(x_j)$. Then $\{x_i, x_j\} \cap Pos_C(\mathcal{D}) \neq \emptyset$, and $(x_i, x_j) \in IND(Q\cup B)$ implies $(x_i, x_j) \in IND(B)$. By definition, $\delta_{ij}^B \neq *$, then $\delta_{ij}^B \cap Q \neq \emptyset$. Thus, there exists some $\mathsf{q} \in Q$ such that $\mathsf{q}(x_i) \neq \mathsf{q}(x_j)$. This contradicts $x_j \in [x_i]_{Q\cup B}$. Hence $x_i \in Pos_C(\mathcal{D})$ implies $x_i \in Pos_{Q\cup B}(\mathcal{D})$, and thus we have $Pos_C(\mathcal{D}) \subseteq Pos_{Q\cup B}(\mathcal{D})$.

($\Leftarrow$) Suppose that $\delta_{ij}^B \neq *$; then $\{x_i, x_j\} \cap Pos_C(\mathcal{D}) \neq \emptyset$. Assume, without loss of generality, that $x_i \in Pos_C(\mathcal{D})$. Then, on the assumption that $Pos_{Q\cup B}(\mathcal{D}) = Pos_C(\mathcal{D})$, we obtain $x_i \in Pos_{Q\cup B}(\mathcal{D})$. There then exists some $D \in \mathcal{D}$ such that $[x_i]_{Q\cup B} \subseteq D$. Because $\delta_{ij} \neq *$, we obtain $(x_i, x_j) \in IND(B)$ and $\mathsf{d}(x_i) \neq \mathsf{d}(x_j)$, and therefore (x_i, x_j) must not be an element of $IND(Q\cup B)$. This implies that there exists some $\mathsf{q} \in Q$ such that $\mathsf{q}(x_i) \neq \mathsf{q}(x_j)$. Hence, $\mathsf{q} \in \delta_{ij}^B \cap Q$, and hence $\delta_{ij}^B \cap Q \neq \emptyset$. □

Corollary 3.4.1. *Consider a decision table $DT = (U, C\cup\{\mathsf{d}\})$ and a subset $B \subsetneq C$ along with its partial discernibility matrix $PDM(B) = (\delta_{ij}^B)$. A subset $Q \subseteq C$ is minimal concerning Condition 1 iff Q is also minimal concerning Condition 2.*

1. $\forall \delta_{ij}^B \in PDM(B), \delta_{ij}^B \neq * \Longrightarrow \delta_{ij}^B \cap Q \neq \emptyset$,
2. $Pos_{Q\cup B}(\mathcal{D}) = Pos_C(\mathcal{D})$.

Proof. The proof is evident from Theorem 3.4. □

Condition 2 in Corollary 3.4.1 is equivalent to Condition 1 of the relative B-reducts. Consequently, relative B-reducts can be derived using the same method applied to relative reducts. The logical formula $L(PDM(B))$ for $PDM(B)$ is defined as follows:

$$L(PDM(B)) : \bigwedge_{1\leq j<i\leq n,\ \delta_{ij}^B\neq *} L(\delta_{ij}^B), \tag{19}$$

where $L(\delta_{ij}^B)$ is the logical formula for the element δ_{ij}^B ($\neq *$) of DM and is defined as follows:

$$L(\delta_{ij}^B) : \bigvee_{\mathsf{a}\in\delta_{ij}^B} \mathsf{a}. \tag{20}$$

Let $L(PDM(B))'$ represent the minimal DNF of $L(PDM(B))$, so it can be stated as:

Table 3
Partial discernibility matrix $PDM(\{\mathsf{S}\})$ for Table 1.

	p_1	p_3	p_5	p_6
p_1	*			
p_3	{H, T, M}	*		
p_5	*	*	*	
p_6	{H, T, M}	*	*	*

	p_2	p_4
p_2	*	
p_4	*	*

Table 4
Partial discernibility matrix $PDM(\{\mathsf{T}\})$ for Table 1.

	p_1	p_5
p_1	*	
p_5	*	*

	p_2	p_3	p_4
p_2	*		
p_3	{H, S, M}	*	
p_4	*	{S}	*

	p_6
p_6	*

$$L(PDM(B))' : \bigvee_{R \in Red^{\cup B}(DT)} \bigwedge_{\mathsf{a} \in R} \mathsf{a}. \tag{21}$$

To convert $L(PDM(B))$ into its minimal DNF $L(PDM(B))'$, it is sufficient to apply the Reduct Computation Algorithm (RCA) [18] to $L(PDM(B))$.

Example 3.5. Let's consider that DT is the decision table illustrated in Table 1 and that $C_0 = Core(DT) = \{\mathsf{S}\}$. The partial discernibility matrix $PDM(C_0)$ is shown in Table 3. Since the elements δ_{ij}, such that $(p_i, p_j) \notin IND(C_0)$, must be treated as wildcards, we omit these elements.

The logical formula for $PDM(C_0)$ is as follows:

$$(\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}) \wedge (\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}); \tag{22}$$

its minimal DNF $L(PDM(C_0))'$ is:

$$\mathsf{H} \vee \mathsf{T} \vee \mathsf{M}. \tag{23}$$

Consequently, the set $Red^{\cup C_0}(DT)$ of the relative C_0-reducts is $\{\{\mathsf{Headache}\}, \{\mathsf{Temperature}\}, \{\mathsf{Muscle\ pain}\}\}$.

Example 3.6. Suppose DT is the decision table displayed in Table 1. The partial discernibility matrix $PDM(\{\mathsf{T}\})$ is illustrated in Table 4. Here too, the elements δ_{ij} where $(p_i, p_j) \notin IND(\{\mathsf{T}\})$ must be treated as wildcards, so we omit them.

The logical formula for $PDM(\{\mathsf{T}\})$ is represented as follows:

$$(\mathsf{H} \vee \mathsf{S} \vee \mathsf{M}) \wedge \mathsf{S}; \tag{24}$$

its minimal DNF $L(PDM(\{\mathsf{T}\}))'$ is:

$$\mathsf{S}. \tag{25}$$

Thus, there exists only one relative $\{\mathsf{T}\}$-reduct, $\{\mathsf{Sneeze}\}$, of DT.

3.3. Transformation to relative reducts

Relative B-reducts can be converted into relative reducts under two conditions: when $B = Core(DT)$ or when $B = \{\mathsf{b}\}$ (a singleton set).

Theorem 3.7. *Consider $DT = (U, C \cup \{\mathsf{d}\})$ as a decision table, and let $C_0 = Core(DT)$. Then,*

$$Red(DT) = \left\{ R \cup C_0 \;\middle|\; R \in Red^{\cup C_0}(DT) \right\}. \tag{26}$$

Proof. According to Proposition 2.3, for any relative reduct $R \in Red(DT)$, it holds that $C_0 \subseteq R$. Let $P = R \setminus C_0$; consequently,

$$Pos_{P \cup C_0}(\mathcal{D}) = Pos_C(\mathcal{D}). \tag{27}$$

Because $R = P \cup C_0$ is a relative reduct and $\forall Q \subsetneq P, Q \cup C_0 \subsetneq R$; then

$$Pos_{Q\cup C_0}(\mathcal{D}) \neq Pos_C(\mathcal{D}). \tag{28}$$

Therefore, P is a relative C_0-reduct of DT.

Conversely, we take any relative C_0-reduct P. Let $R = P \cup C_0$; Therefore, we can conclude that:

$$Pos_R(\mathcal{D}) = Pos_C(\mathcal{D}). \tag{29}$$

However, from Propositions 2.3 and 3.2,

$$Pos_{R\setminus\{\mathsf{c}\}}(\mathcal{D}) \neq Pos_C(\mathcal{D}), \forall \mathsf{c} \in R. \tag{30}$$

Therefore, $R = P \cup C_0$ is a relative reduct of DT.

Hence, we have

$$Red(DT) = \left\{ R \cup C_0 \;\middle|\; R \in Red^{\cup C_0}(DT) \right\}. \quad \square \tag{31}$$

This means that all relative reducts of DT can be obtained by calculating the unions of C_0 with each relative C_0-reduct.

Theorem 3.8. *Let $DT = (U, C \cup \{\mathsf{d}\})$ be a decision table, let $\mathsf{b} \in C$ be any condition attribute, and let R be a relative $\{\mathsf{b}\}$-reduct. If $Pos_R(\mathcal{D}) = Pos_C(\mathcal{D})$, then R is a relative reduct of DT; otherwise, then $R \cup \{\mathsf{b}\}$ is a relative reduct of DT.*

Proof. Assume $Pos_R(\mathcal{D}) = Pos_C(\mathcal{D})$. From Proposition 3.2, we have:

$$Pos_{R\setminus\{\mathsf{c}\}\cup\{\mathsf{b}\}}(\mathcal{D}) \neq Pos_C(\mathcal{D}), \forall \mathsf{c} \in R. \tag{32}$$

Because, by Proposition 3.2, $R \setminus \{\mathsf{c}\} \subsetneq R \setminus \{\mathsf{c}\} \cup \{\mathsf{b}\}$, then from Proposition 2.2 we have

$$Pos_{R\setminus\{\mathsf{c}\}}(\mathcal{D}) \subseteq Pos_{R\setminus\{\mathsf{c}\}\cup\{\mathsf{b}\}}(\mathcal{D}). \tag{33}$$

Therefore, from (32) and (33),

$$Pos_{R\setminus\{\mathsf{c}\}}(\mathcal{D}) \neq Pos_C(\mathcal{D}). \tag{34}$$

Therefore, R is recognized as a relative reduct of DT.

Conversely, if we assume $Pos_R(\mathcal{D}) \neq Pos_C(\mathcal{D})$, then according to Definition 3.1,

$$Pos_{R\cup\{\mathsf{b}\}}(\mathcal{D}) = Pos_C(\mathcal{D}). \tag{35}$$

On the assumption that R is a relative $\{\mathsf{b}\}$-reduct,

$$Pos_Q(\mathcal{D}) \neq Pos_C(\mathcal{D}), \forall Q \subsetneq R \cup \{\mathsf{b}\}. \tag{36}$$

Hence, $R \cup \{\mathsf{b}\}$ is a relative reduct of DT. $\square$

In general, various relative reducts are produced per Theorem 3.8. Note that this method may not extract all possible relative reducts.

Example 3.9. Consider the decision table referred to as DT, which is illustrated in Table 1. From Example 3.5, we find that $Red^{\cup C_0}(DT) = \{\{\mathsf{H}\}, \{\mathsf{T}\}, \{\mathsf{M}\}\}$ and $C_0 = Core(DT) = \{\mathsf{S}\}$. According to Theorem 3.7, we can determine all the relative reducts $Red(DT)$ of DT as follows:

$$Red(DT) = \left\{ R \cup C_0 \;\middle|\; R \in Red^{\cup C_0}(DT) \right\} \tag{37}$$

$$= \{\{\mathsf{H}\} \cup \{\mathsf{S}\}, \{\mathsf{T}\} \cup \{\mathsf{S}\}, \{\mathsf{M}\} \cup \{\mathsf{S}\}\} \tag{38}$$

$$= \{\{\mathsf{H}, \mathsf{S}\}, \{\mathsf{T}, \mathsf{S}\}, \{\mathsf{M}, \mathsf{S}\}\}. \tag{39}$$

Example 3.10. In this case, DT is again the decision table shown in Table 1. From Example 3.6, we find that $Red^{\cup\{\mathsf{T}\}}(DT) = \{\{\mathsf{S}\}\}$. By applying Theorem 3.8, we can derive a relative reduct $\{\mathsf{T}, \mathsf{S}\}$ of DT as follows:

$$\left\{ R \cup \{\mathsf{T}\} \;\middle|\; R \in Red^{\cup\{\mathsf{T}\}}(DT) \right\} \tag{40}$$

$$= \{\{\mathsf{S}\} \cup \{\mathsf{T}\}\} \tag{41}$$

$$= \{\{\mathsf{T}, \mathsf{S}\}\}. \tag{42}$$

Algorithm 3 Relative reduct computation algorithm using the relative $Core(DT)$-reducts [15].

Input: a decision table DT such that $Core(DT) \neq \emptyset$
Output: the set of all relative reducts Red of DT
// Calculate $Core(DT)$.
1: $C_0 \leftarrow Core(DT)$
2: Construct the partial discernibility matrix $PDM(C_0)$
3: $L \leftarrow$ the minimal DNF of $L(PDM(C_0))$
// Calculate $Red^{\cup C_0}(\mathcal{D})$ with $PDM(C_0)$.
4: $pRed \leftarrow \left\{ A \subseteq C \;\middle|\; \bigwedge_{\mathsf{a} \in A} \mathsf{a} \text{ coincides with a term of } L \right\}$
5: $Red \leftarrow \{ A \cup C_0 \mid A \in pRed \}$
6: **return** Red

Algorithm 4 Relative reduct computation algorithm using relative pre-reducts with respect to a single attribute [15].

Input: a decision table DT
Output: the set of all or some relative reducts Red of DT
// Select an attribute based on EstSize$(\cdot)$
1: $\mathsf{b} \leftarrow \arg\min_{\mathsf{a} \in C} \mathrm{EstSize}(\mathsf{a})$
// Calculate relative pre-reducts using $PDM(\{\mathsf{b}\})$
2: Construct $PDM(\{\mathsf{b}\})$ of DT
3: $L \leftarrow$ the minimal DNF of $L(PDM(\{\mathsf{b}\}))$
// Calculate $Red^{\cup\{\mathsf{b}\}}(\mathcal{D})$ with $PDM(\{\mathsf{b}\})$.
4: $pRed \leftarrow \left\{ A \subseteq C \;\middle|\; \bigwedge_{\mathsf{a} \in A} \mathsf{a} \text{ coincides with a term of } L \right\}$
5: $Red \leftarrow \emptyset$
// For each $R \in pRed$, check whether b can be eliminated from R.
6: **for all** $R \in pRed$ **do**
7: **if** $Pos_{R \setminus \{\mathsf{b}\}}(\mathcal{D}) = Pos_C(\mathcal{D})$ **then**
8: $Red \leftarrow Red \cup \{R\}$
9: **else**
10: $Red \leftarrow Red \cup \{R \cup \{\mathsf{b}\}\}$
11: **end if**
12: **end for**
13: **return** Red

3.4. Relative reduct computation algorithms that use relative pre-reducts

Utilizing Theorems 3.7 and 3.8, we perform the computation of relative reducts for a decision table based on relative pre-reducts. Algorithm 3 outlines the method that employs relative pre-reducts relative to core attributes.

In Step 1, we identify the set $Core(DT)$ containing all core attributes from the given decision table DT, assigning it to the variable C_0. Next, in Steps 2 and 3, we construct the partial discernibility matrix $PDM(C_0)$ and compute the minimal DNF for the formula $L(PDM(C_0))$, storing it in the variable L. Step 4 involves obtaining the sets of attributes corresponding to the terms in L. In Step 5, we compute the union of each identified set with C_0 and append these results to the set Red. Finally, in Step 6, we return the resulting family Red of sets, which contains all the relative reducts of DT.

Algorithm 4 presents a method that utilizes relative pre-reducts concerning a single attribute. To optimize for computational complexity, we choose an attribute whose partial discernibility matrix is relatively small. To estimate the approximate size of a partial discernibility matrix, we define the function $\mathrm{EstSize} : C \to \mathbb{N}$, where $\mathbb{N}$ represents the set of natural numbers, as follows:

$$\mathrm{EstSize}(\mathsf{a}) := \sum_{X \in U/IND(\{\mathsf{a}\})} |X|^2. \tag{43}$$

$\delta_{ij}^{\{\mathsf{b}\}} \neq *$ only if pair (x_i, x_j) of objects is indiscernible with respect to $\mathsf{b} \in C$.

In Step 1, we select an attribute b such that EstSize(b) yields the minimum value. Steps 2–4 follow a sequence akin to Steps 2–4 in Algorithm 3. In Steps 5–13, we assess whether b is redundant within each $R \in pRed$.

Because algorithms 3 and 4 necessitate the calculation of the minimal DNF, the overall computational complexity is NP-hard. The complexities associated with each step in Algorithm 3 are as follows: Step 1 involves calculating all core attributes, resulting in a complexity of $O(|U| \times |C|^2)$. Thus, it becomes challenging to apply this algorithm to datasets that contain a large number of attributes. In Step 2, we construct the $PDM(Core(DT))$, which incurs a complexity of $O(\sum_{E \in U/IND(B)} |E|^2 \times |C|)$.

For Steps 2–4 in Algorithm 4, the computational complexity mirrors that of Algorithm 3, with $Core(DT)$ being substituted by $\{\mathsf{b}\}$. In Step 1, $U/IND(\{\mathsf{a}\})$ is calculated for every condition attribute $\mathsf{a} \in C$. The computation of a partition can be conducted using counting sort, leading to a complexity of $O(n)$, making the complexity of Step 1 is $O(|U| \times |C|)$. In Steps 6–13, the positive region $Pos_{R \setminus \{\mathsf{b}\}}(\mathcal{D})$ is computed for each relative pre-reduct $R \in pRed$, resulting in a complexity of $O\left(\left|Red^{\cup\{\mathsf{b}\}}(DT)\right| \times |U| \times |C|\right)$.

3.5. Reduced decision table for relative pre-reducts

Relative pre-reducts can be determined for a dataset containing a large number of attributes by utilizing a reduced decision table. We now define the concept of a reduced decision table for relative pre-reducts.

Algorithm 5 Reduced decision table computation algorithm for the relative pre-reducts.

Input: decision table $DT = (U, C \cup \{\mathtt{d}\})$, the attribute set $B \subseteq C$, the partial discernibility matrix $PDM(B)$ of DT, the number b of attributes selected
Output: B-reduced decision table $RDT(B) = (U, C' \cup \{\mathtt{d}\})$

```
Select b attributes c_1, ..., c_b from C \ B at random by sampling without replacement
C' ← {c_1, c_2, ..., c_b} ∪ B
for all δ^B_ij ∈ PDM(B) s.t. i > j do
    if δ^B_ij ≠ * and δ^B_ij ∩ C' = ∅ then
        Select c ∈ δ^B_ij at random.
        C' ← C' ∪ {c}
    end if
end for
return (U, C' ∪ {d})
```

Definition 3.11. Let $DT = (U, C \cup \{\mathtt{d}\})$ represent a decision table, and let $PDM(B) = (\delta^B_{ij})$ be its partial discernibility matrix with respect to $B \subseteq C$. Let $A \subseteq C$ be a set of condition attributes such that $A \cap B = \emptyset$, and let $C' = A \cup B$. The decision table $RDT(B) = (U, C' \cup \{\mathtt{d}\})$ is considered a B-reduced decision table iff a subset $A \subseteq C$ meets the following condition:

$$\forall \delta^B_{ij} \in PDM(B), \delta^B_{ij} \neq * \Longrightarrow \delta^B_{ij} \cap A \neq \emptyset. \tag{44}$$

Theorem 3.12. *Let $DT = (U, C \cup \{\mathtt{d}\})$ be a decision table, let $PDM(B) = (\delta^B_{ij})$ represent its partial discernibility matrix with respect to $B \subseteq C$, and let $RDT(B) = (U, C' \cup \{\mathtt{d}\})$ denote a B-reduced decision table. A subset $R \subseteq C'$ is a relative B-reduct of $RDT(B)$ iff R is a relative B-reduct of DT.*

Proof. Because A satisfies the condition (44), it follows from Theorem 3.4 that $Pos_{A \cup B}(\mathcal{D}) = Pos_{C'}(\mathcal{D}) = Pos_C(\mathcal{D})$. Consequently, for any subset $Q \subseteq C'$, it holds that

$$Pos_{Q \cup B}(\mathcal{D}) = Pos_{C'}(\mathcal{D}) \Leftrightarrow Pos_{Q \cup B}(\mathcal{D}) = Pos_C(\mathcal{D}). \tag{45}$$

Thus, condition (i) for relative B-reducts in $RDT(B)$ is equivalent to condition (i) of relative B-reducts in DT. Therefore, for any $R \subseteq C'$, R is a relative B-reduct of $RDT(B)$ iff R is a relative B-reduct of DT. □

The partial discernibility matrix for a reduced decision table pertaining to relative pre-reducts can be directly computed from the original decision table's partial discernibility matrix.

Proposition 3.13. *Given two decision table $DT = (U, C \cup \{\mathtt{d}\})$ and $DT' = (U, C' \cup \{\mathtt{d}\})$ such that $C' \subseteq C$, and the partial discernibility matrix $PDM(B) = \left(\delta^B_{ij}\right)$ of DT where B is a set of condition attributes such that $B \subseteq C'$, then the element δ'^B_{ij} of the partial discernibility matrix $PDM'(B)$ of DT' satisfies the following two conditions:*

(i) $\delta^B_{ij} \neq *$ *iff* $\delta'^B_{ij} \neq *$,
(ii) *if* $\delta^B_{ij} \neq *$, *then* $\delta'^B_{ij} = \delta^B_{ij} \cap C'$.

Proof. (i) From the proof for Theorem 3.12, $Pos_{C'}(\mathcal{D}) = Pos_C(\mathcal{D})$. Hence, we obtain

$$\begin{aligned} \delta^B_{ij} \neq * &\Leftrightarrow (x_i, x_j) \in IND(B), \mathtt{d}(x_i) \neq \mathtt{d}(x_j), \text{ and } \{x_i, x_j\} \cap Pos_C(\mathcal{D}) \neq \emptyset \\ &\Leftrightarrow (x_i, x_j) \in IND(B), \mathtt{d}(x_i) \neq \mathtt{d}(x_j), \text{ and } \{x_i, x_j\} \cap Pos_{C'}(\mathcal{D}) \neq \emptyset \\ &\Leftrightarrow \delta'^B_{ij} \neq * . \end{aligned} \tag{46}$$

(ii) From (i) and the definition of the partial discernibility matrix, both δ^B_{ij} and δ'^B_{ij} are sets of condition attributes. Hence,

$$\delta'^B_{ij} = \{\mathtt{c} \in C' \mid \mathtt{c}(x_i) \neq \mathtt{c}(x_j)\} = \{\mathtt{c} \in C \mid \mathtt{c}(x_i) \neq \mathtt{c}(x_j)\} \cap C' = \delta^B_{ij} \cap C'. \quad \square \tag{47}$$

Similar to a standard reduced decision table, a reduced decision table for relative pre-reducts can be computed using Algorithm 5. Algorithm 6 outlines the computation of B-relative reducts based on B-reduced decision tables.

The complexities of each step in Algorithm 5 are as follows. The complexities of Steps 1 and 2 are $O(|C|)$. Step 3–8 process each element of the partial discernibility matrix; therefore, their complexity is $O(\sum_{E \in U/IND(B)} |E|^2 \times |C|)$. Overall, the complexity of Algorithm 5 is therefore $O(\sum_{E \in U/IND(B)} |E|^2 \times |C|)$.

Since Algorithm 6 involves computing the minimal DNF, its total complexity is NP-hard. The complexities for each step are as follows: Step 2 constructs the partial discernibility matrix, with its complexity dependent on whether $B = Core(DT)$ or $B = \{\mathtt{b}\}$. Step 4 generates a reduced decision table using Algorithm 5, which has a complexity of $O(\sum_{E \in U/IND(B)} |E|^2 \times |C|)$.

Algorithm 6 Relative pre-reduct computation based on reduced decision tables.

Input: decision table $DT = (U, C \cup \{\mathsf{d}\})$, the number b of attributes extracted, the number I of iterations
Output: The set $pRed$ of B-relative reducts of DT

```
pRed ← ∅
Construct PDM(B) of DT
for i = 1 to I do
    Generate RDT(B) using Algorithm 5
    Construct PDM′(B) of RDT(B) according to Proposition 3.13
    L ← the minimal DNF of L(PDM′(B))
    S ← { A ⊆ C | ⋀_{a∈A} a coincides with a term of L }
    pRed ← pRed ∪ S
end for
return pRed
```

Table 5
$\{\mathsf{T}\}$-reduced decision table $(U, \{\mathsf{H}, \mathsf{S}, \mathsf{T}\} \cup \{\mathsf{Flu}\})$.

Patient	Condition Attributes			Decision Attribute
	Headache	Sneeze	Temperature	Flu
p_1	no	no	very high	yes
p_5	no	no	very high	no
p_2	no	yes	high	yes
p_3	yes	no	high	no
p_4	yes	yes	high	yes
p_6	yes	no	normal	no

Table 6
The partial discernibility matrix $PDM(\{\mathsf{T}\})$ of the $\{\mathsf{T}\}$-reduced decision table $(U, \{\mathsf{H}, \mathsf{S}, \mathsf{T}\} \cup \{\mathsf{Flu}\})$.

	p_1	p_5
p_1	*	
p_5	*	*

	p_2	p_3	p_4
p_2	*		
p_3	{H, S}	*	
p_4	*	{S}	*

	p_6
p_6	*

Example 3.14. Let's define $A = \{\mathsf{H}, \mathsf{S}\}$ and $C' = A \cup \{\mathsf{T}\}$. Based on Table 4, $(U, C' \cup \{\mathsf{d}\})$ (see Table 5) is identified as a $\{\mathsf{T}\}$-reduced decision table derived from Table 1. The partial discernibility matrix of the $\{\mathsf{T}\}$-reduced decision table is presented in Table 6. From Table 6, we can derive the logical formula and subsequently convert it into its minimal DNF.

$$(\mathsf{H} \vee \mathsf{S}) \wedge \mathsf{S} = \mathsf{S}. \tag{48}$$

Therefore, we obtain only one relative $\{\mathsf{T}\}$-reduct $\{\mathsf{S}\}$. Because

$$Pos_C(\mathcal{D}) = \{p_2, p_3, p_4, p_6\} \text{ and } Pos_{\{\mathsf{S}\}}(\mathcal{D}) = \{p_2, p_4\}, \tag{49}$$

from Theorem 3.8, we then obtain a relative reduct $\{\mathsf{T}\} \cup \{\mathsf{S}\} = \{\mathsf{S}, \mathsf{T}\}$.

4. Experiments

To illustrate the effectiveness of our proposed method, we conducted a comparison with the traditional discernibility matrix method [5] and Hońko's horizontal decomposition method [12]. The horizontal decomposition method processes portions of the discernibility matrix, yet ultimately computes all elements. In our experiments, the hyperparameter "number of middle subtables" in Hońko's approach was set to 4. When a decision class had an insufficient number of objects, decomposition was not applied. These methods require the computation of the positive region and the minimal DNF. We utilized the algorithm proposed by Yin [21] for calculating the positive region. While various fast algorithms for calculating the minimal DNF have been proposed (e.g., Susmaga [17] and Bao et al. [18]), our experiments employed the most recent approach [18], which is the Reduct Computation Algorithm. Chen et al. [11] proposed constructing a discernibility matrix by removing elements using the idempotent law and absorption law techniques. Although this approach could be integrated into these methods, we did not implement it in the current comparison. All methods employed for relative reduct computation in the experiments were executed within the environment described in Table 7.

In our experiments, we utilized 14 datasets sourced from the UCI Machine Learning Repository [22], which is an online database widely used in machine learning research. The specific details of the datasets used are provided in Table 8. For the Internet Advertisements, Nomao, and Gisette_train datasets, which contain a relatively large number of attributes, we employed a combined strategy that involved reduced decision tables. None of the earlier methods were able to enumerate the relative reducts for these datasets. Although TAO-all2 does not qualify as a decision table, we refer to its attribute "s. s. temp." as a decision attribute. Missing values were operationalized as a singular value.

Table 7
The implementation and execution environment for the relative reduct computation.

Language	C++
CPU	Intel® Xeon® CPU E5-2650 v4 @ 2.20 GHz
RAM	132 GB
OS	Ubuntu 22.04.5 LTS
Compiler	g++ 11.4.0
Optimization options	O3

Table 8
Description of the data sets used in this research.

Data sets	$\|U\|$	$\|C\|$
Zoo	100	16
Car	1,728	6
Kr-vs-kp	3,196	36
Nursery	12,960	8
Letter-Recognition	20,000	16
Adult	32,561	14
Shuttle	58,000	9
TAO-all2	178,080	10
Diabetes 130-US hospitals for years 1999–2008	101,766	47
Cover-type	581,012	54
Connect-4	67,557	42
Internet Advertisements	3,279	1,559
Nomao	34,465	119
Gisette_train	6,000	5,001

Table 9
Average computation times (in seconds) for both the conventional and core-based reduct enumeration algorithms.

Data set	DM	Hońko [12]	Algorithm 3 (Core-based)	
			Step 1 (Calculate core)	All steps
Zoo	**0.004*** ± 0.000	0.006 ± 0.000	0.010 ± 0.001	0.012 ± 0.000
Car	0.348 ± 0.018	0.400 ± 0.014	0.037 ± 0.002	**0.052*** ± 0.002
Kr-vs-kp	2.822 ± 0.057	2.726 ± 0.016	1.882 ± 0.008	**2.003*** ± 0.008
Nursery	28.418 ± 0.813	33.667 ± 0.317	0.869 ± 0.006	**1.225*** ± 0.004
Letter-Recog.	140.214 ± 2.531	524.662 ± 18.345	8.799 ± 0.019	**10.165*** ± 0.019
Adult	149.886 ± 1.323	152.552 ± 0.388	19.567 ± 0.018	**23.382*** ± 0.017
Shuttle	431.225 ± 24.694	692.349 ± 18.344	34.690 ± 0.050	**188.575*** ± 0.873
TAO-all2	Memory	Time out	418.019 ± 0.491	**496.037** ± 0.475
Diabetes	Memory	Time out	-	Memory
Cover-type	Memory	Memory	Time out	-
Connect-4	Memory	**2,566.991** ± 82.628	No core	-

*: $p < 0.05$.

Throughout the experiments, we measured both execution time and the number of solutions produced. This process was repeated 10 times, and we calculated the averages of the measurements. To facilitate discussion, the measurement of execution time was divided into two phases. For the core-based method, we segmented it into two phases: one for computing the core attributes, and another for subsequent processes. Likewise, the single-attribute-based method was divided into two phases: the first leading up to the computation of the {b}-relative reducts, followed by a second phase. Additionally, we performed a paired t-test ($p < 0.05$) to analyze total computational time. No test was conducted for the method utilizing the reduced decision table because no comparison was available.

Table 9 presents the average computation times along with their standard deviations for the previous methods alongside the core-based techniques for each dataset. The results that show significantly shorter computation times are highlighted in bold. Any measurement that demonstrated a significant difference by Welch's t-test ($p < 0.05$) is labeled with a "$*$". For instance, the measurement **2.003*** ± 0.008 in column 4 for the kr-vs-kp dataset indicates an average execution time of 2.003 (s) with a standard deviation of 0.008 (s), observed when Algorithm 3 was executed 10 times; Algorithm 3 was notably faster than both the discernibility matrix method and Hońko's horizontal decomposition method. The term "Memory" signifies that the program exhausted available memory, while "No core" indicates that the dataset lacked any core attributes and thus could not be processed. "Time out" denotes that the program exceed the allotted time limit, which was set to 4 hours.

Table 10 displays the average computation times measured, along with their standard deviations, for the two conventional methods and the single-attribute-based approach. The other details in this table are consistent with those presented in Table 9.

Table 11 indicates the count of relative reducts generated for each employed method. The core-based method, referred to as Algorithm 3, successfully enumerated all relative reducts, whereas the single-attribute-based method, Algorithm 4, may not have identified all relative reducts.

Table 10
Average computation times (in seconds) for both the conventional and single-attribute-based reduct enumeration algorithms.

Data set	DM	Hońko [12]	Algorithm 4 (Single-attribute-based)	
			Steps 1–4	All steps
Zoo	**0.004*** ± 0.000	0.006 ± 0.000	0.002 ± 0.000	0.018 ± 0.001
Car	0.348 ± 0.018	0.400 ± 0.014	0.101 ± 0.003	**0.106*** ± 0.003
Kr-vs-kp	2.822 ± 0.057	2.726 ± 0.016	1.531 ± 0.009	**1.815*** ± 0.027
Nursery	28.418 ± 0.813	33.667 ± 0.317	5.979 ± 0.014	**6.055*** ± 0.016
Letter-Recog.	140.214 ± 2.531	524.662 ± 18.345	13.891 ± 0.019	**36.493*** ± 0.035
Adult	149.886 ± 1.323	152.552 ± 0.388	4.718 ± 0.031	**6.734*** ± 0.030
Shuttle	431.225 ± 24.694	692.349 ± 18.344	11.506 ± 0.018	**14.654*** ± 0.029
TAO-all2	Memory	Time out	75.136 ± 0.168	**76.266** ± 0.165
Diabetes	Memory	Time out	202.687 ± 0.351	Time out
Cover-type	Memory	Memory	792.477 ± 1.019	Time out
Connect-4	Time out	2,566.991 ± 82.628	1,120.350 ± 10.106	**2,183.120*** ± 9.493

*: $p < 0.05$.

Table 11
Numbers of relative reducts obtained by using the conventional, core-based, and single-attribute-based methods.

Data set	DM	Hońko [12]	Algorithm 3 (Core-based)	Algorithm 4 (Single-attribute-based)
Zoo	33	33	33	33 ± 0
Car	1	1	1	1 ± 0
Kr-vs-kp	4	4	4	4 ± 0
Nursery	1	1	1	1 ± 0
Letter-Recog.	61	61	61	45 ± 0
Adult	2	2	2	2 ± 0
Shuttle	19	19	19	8 ± 0
TAO-all2	-	-	4	1 ± 0
Diabetes	-	-	-	1,008.9 ± 7.4
Cover-type	-	-	-	35.5 ± 0.5
Connect-4	-	171	-	170 ± 0

Table 12
Measured average computation times (in seconds) for the combined single-attribute-based and reduced decision table [9] methods ($b = 25$, $I = 10$).

Data set	Algorithm 4 and Algorithm 6 (Single-attribute and reduced decision table)	
	Steps 1–4 (Algorithm 6)	All steps
Internet Advertisements	65.819 ± 0.212	125.359 ± 24.205
Nomao	80.660 ± 1.732	926.566 ± 238.809
Gissete_train	8,039.525 ± 1,687.061	8,672.210 ± 1,740.792

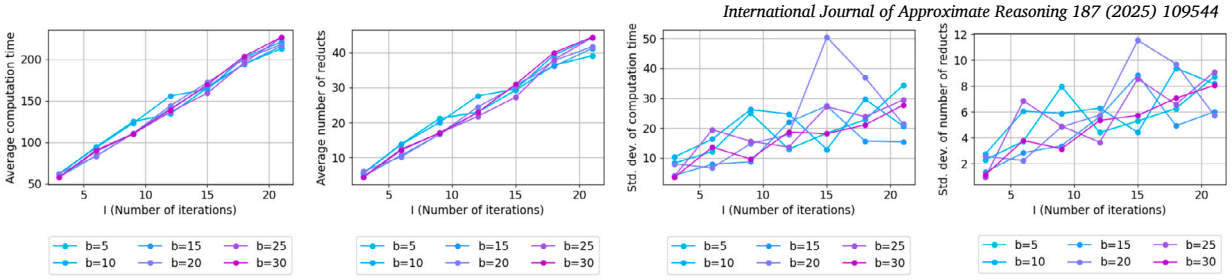
Table 13
Number of relative reducts obtained from the single-attribute-based method combined with the reduced decision table method.

Data set	Number of relative reducts
Internet Advertisements	19.5 ± 8.45
Nomao	569.5 ± 145.89
Gissete_train	515.6 ± 138.17

Table 12 presents the average computation times measured for the combined methods of single-attribute-based and reduced decision tables [9]. For Algorithm 6, the number b of initially selected attributes was set to 25, and the number I of iterations was set to 10. In this approach, Steps 1–4 were replaced by Algorithm 6 due to the utilization of a reduced decision table. Apart from this modification, the specifics of this table align with those found in Table 9.

Table 13 shows the quantity of relative reducts obtained through the application of the single-attribute-based method in conjunction with the reduced decision table method. It should be noted that this method does not guarantee the enumeration of all possible relative reducts.

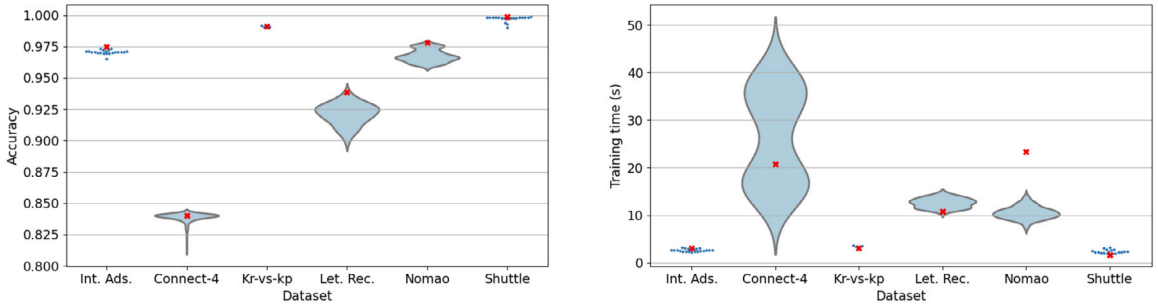
Additionally, we explored how the parameters impacted the method utilizing the reduced decision table. Fig. 1 depicts the trends in average computation time over 10 runs, along with the corresponding standard deviation (SD) for the Internet Advertisements dataset. Similar trends were observed for the other datasets; therefore, their corresponding graphs were omitted to prevent redundancy. In this particular experiment, the iteration number I was varied from 3 to 21 in increments of 3, while the number of randomly selected attributes b was adjusted from 5 to 30 in increments of 5. The upper limit for b was capped at 30, as larger values would substantially raise the computational cost of transforming logical formula due to the NP-hard nature of the problem.



(a1) Computation time (s) (a2) Number of relative reducts (b1) Computation time (s) (b2) Number of relative reducts

(a) Average (b) Standard deviation

Fig. 1. Impact of parameters b (number of randomly selected attributes) and I (number of iterations) on the reduced decision table method for the Internet Advertisements dataset.



(a) Accuracy (b) Training time (s)

Fig. 2. Average accuracy and training time over 5-fold cross-validation using the CatBoost classifier [23], comparing relative reducts to the complete feature set. Results for the complete feature set are indicated in red. A violin plot is used when the number of reducts is 30 or more; otherwise, a swarm plot is presented.

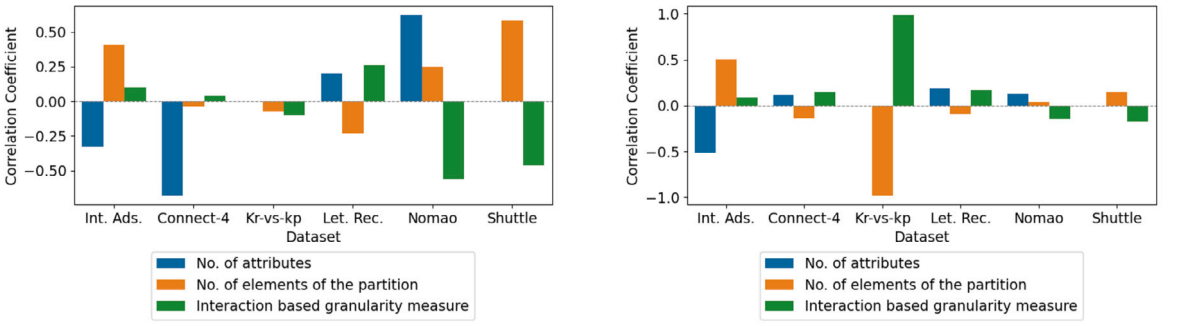
Table 14
Implementation and execution environment for the classification problems.

Python	3.11.11
CatBoost	1.2.8
CPU	Intel® Core™ i9-14900KF 6 GHz
GPU	NVIDIA GeForce RTX™ 4070 Ti
RAM	32 GB
OS	Windows 11 Home 10.0.26100

Furthermore, we evaluated the performance of relative reducts against the complete feature set using the CatBoost classifier [23], which is recognized for achieving state-of-the-art results in classification tasks involving categorical features. The average accuracy (depicted in Fig. 2a) and training time (shown in Fig. 2b) were computed over 5-fold cross-validation. Red points represent the results corresponding to the complete feature set. For relative reducts, violin plots are employed when the number of reducts equals 30 or more; otherwise, swarm plots are used. All features were treated as categorical. Detailed implementation specifics and the execution environment are presented in Table 14.

In our experiments, we utilized the CatBoost classifier with specific hyperparameters: 1,000 iterations, a learning rate of 0.1, a tree depth of 6, and early stopping after 50 rounds without improvement. The evaluation metric focused on accuracy, with training conducted using GPU acceleration. A fixed random seed was implemented to ensure reproducibility. The selection of hyperparameters in our research was informed by a mix of standard practices, empirical observations, and guidelines from the CatBoost documentation. Notably, a learning rate of 0.1 and a tree depth of 6 are established values known to perform consistently across different datasets. The iteration count of 1,000, combined with early stopping after 50 rounds without improvement, allows for efficient model convergence while mitigating overfitting risk.

Fig. 3 illustrates the correlation coefficients between model performance (in terms of accuracy and training time) and three standard evaluation metrics for relative reducts: the number of attributes, the number of partition elements, and the interaction-based granularity measure [24].



(a) Correlation with accuracy

(b) Correlation with training time

Fig. 3. Correlation coefficients between model performance and the three standard evaluation metrics for the relative reducts in the CatBoost training procedure.

5. Discussion

Based on Tables 9 and 10, both proposed methods exhibited the shortest computation times across all datasets, except for the Zoo dataset. This suggests that constructing the partial discernibility matrix is less computationally intensive than constructing the discernibility matrix for larger datasets. Computing the partial discernibility matrix involves only a subset of the operations required for the full discernibility matrix. Conversely, Hońko's horizontal decomposition method ultimately computes all elements of the full discernibility matrix, leading to a higher computational cost.

The core-based method (Algorithm 3) necessitated the computation of all core attributes in the decision table to construct a partial discernibility matrix (Step 1). In our experiments, we employed the core computation method proposed by Ge et al. [20], which has a computational complexity of $O(|U| \times |C|^2)$. This complexity escalates gradually as the number of objects increases. However, the computational complexity of Ge et al.'s approach is proportional to the square of the number of attributes. As shown in Table 9, the core-based method encounters a timeout during the core computation phase (Step 1) on the Cover-type dataset, which has the highest number of attributes.

For the single-attribute-based method (Algorithm 4), the attribute for which the evaluation function $\mathrm{EstSize}(\cdot)$ yields its minimum value is utilized to construct the partial discernibility matrix. The computational complexity of this process is $O(|U| \times |C|)$, which remains feasible even with datasets containing a large number of objects and attributes. However, Algorithm 4 also requires the computation of the positive region for all $\{\mathrm{b}\}$-relative reducts. Consequently, the overall computational complexity becomes $O(|U| \times |C| \times |pRed|)$, causing the computational cost of Algorithm 4 to rise as $|pRed|$ increases. Table 10 demonstrates that Algorithm 4 times out during Steps 6–11 on both the Diabetes and Cover-type datasets, with $|pRed| = 17,479$ for the Diabetes dataset and $|pRed| = 424$ for the Cover-type data set. As this method sequentially checks each element of $pRed$ to confirm if it represents a solution, it can provide the solution at that moment.

Table 12 indicates that the standard deviation (SD) of Algorithm 4 across all steps is proportionally higher compared to other methods. This variance is due to the differing number of elements in $pRed$, resulting from the probabilistic selection used to create the reduced decision table. Specifically, the SD of the number of results in Table 13 is greater than the equivalent value in Table 11. Moreover, the computation time for Steps 1–4 (Algorithm 4 combined with Algorithm 6) on the Gisette_train dataset is approximately ten times greater than that on the other datasets. This significant time difference is attributed to the time required to apply the idempotent law and the absorption law to L when determining the minimal DNF, indicating that the partial discernibility matrix contains a substantial number of elements.

With Algorithm 4, it is possible that not all relative reducts were enumerated. As shown in Table 11, Algorithm 4 failed to identify some relative reducts in the Letter-Recognition, Shuttle, TAO-all2, and Connect-4 datasets. This is evidenced by the fact that Algorithm 3 and Hońko's method consistently enumerate all the relative reducts.

Fig. 1 depicts how the parameters b and I affect the reduced decision table method for the Internet Advertisements dataset. The overall trends in both the average computation time (Fig. 1a1) and the average number of reducts (Fig. 1a2) are similar, indicating that the number of reducts influences the computation time directly. Both metrics increase monotonically with respect to the parameter I, as the reduced decision table is generated and the relative pre-reducts are calculated I times.

Conversely, the parameter b has a negligible effect on both computation time and the number of reducts. This suggests that the size of the reduced decision tables remains nearly constant, regardless of the initially selected number of attributes.

Regarding the SD, both the computation time (Fig. 1b1) and the number of reducts (Fig. 1b2) exhibited an upward trend as I increases, while no consistent pattern was observed for b. The increase in variation with respect to I may be due to a higher likelihood of generating reduced decision tables that either terminate significantly quicker or slower. The lack of a clear pattern for b aligns with the average outcomes and likely arises from the minimal impact of b on the size of the reduced decision tables.

It is noteworthy that the significant SD observed at $b = 20$ and $I = 15$ could be attributable to randomness in the selection process, which resulted in the creation of several reduced decision tables characterized by unusually quick or slow computation times.

Fig. 2 provides a comparison between the performance of relative reducts and the complete feature set when utilizing the CatBoost classifier. As illustrated in Fig. 2a, many relative reducts yield slightly lower classification accuracy than the complete feature set. However, certain reducts demonstrate marginal improvements in accuracy. This indicates that the CatBoost classifier can partially identify and reduce the impact of redundant features.

In contrast, Fig. 2b shows that specific relative reducts significantly decrease the training time. For instance, on the Connect-4 dataset, some reducts require merely one-tenth of the training time, whereas for the Nomao dataset, the training time is cut by nearly half. Given that these two datasets have a relatively high number of features (see Table 8) and relative reducts (see Tables 11 and 13), it follows that the observed improvements in training time can be linked to the removal of redundant features, which would otherwise increase computational overhead.

It is important to note that for the Connect-4 and Letter-Recognition datasets, there are instances where the application of relative reducts not only fails to reduce training time but may increase it. This can be attributed to the fact that these relative reducts are derived independently of the classifier, potentially resulting in a misalignment with CatBoost's internal structure or optimization strategy.

As depicted in Fig. 3, the three standard metrics for the relative reducts do not show consistent correlations with model performance. This inconsistency may stem from the relative reducts being constructed without considering the specific classifier used. These findings imply that the practical effectiveness of relative reducts in classification tasks can vary. Therefore, it may be advantageous to explore the extraction of relative reducts as thoroughly as possible.

6. Conclusion

In this research, we introduced the concept of relative pre-reducts for computing the relative reducts of large datasets. Relative pre-reducts can be generated using the partial discernibility matrix. Depending on the focus of the attribute set when constructing the partial discernibility matrix, two methods for relative reduct enumeration are available: core-based and single-attribute-based methods. While the core-based method cannot be applied to datasets lacking core attributes, it allows for the enumeration of all relative reducts. Conversely, the single-attribute-based method may not be able to enumerate all the relative reducts, but can be utilized with decision tables devoid of core attributes. Nevertheless, computational complexity poses a significant challenge, as it increases in line with the number of relative pre-reducts.

The experimental results indicate that the proposed method outperforms previous methodologies on large datasets. Notably, the single-attribute-based method, in conjunction with reduced decision tables, can be effectively applied to datasets with a substantial number of objects and attributes. The experiments revealed that the computation time of the core-based method escalated as the number of attributes increased because the computational complexity of the core attributes is $O(|U| \times |C|^2)$. Conversely, the single-attribute-based method successfully computed relative reducts for datasets that caused the core-based method to time out. It was also observed that computation time increased for datasets with a large number of results.

The main limitation of these methods lies in the duration required for calculating the core attributes and the positive regions of the various $\{b\}$-relative reducts. Enhancing the efficiency of these computations will be a focal challenge in future research.

In this study, we introduced relative pre-reducts and discernibility matrices that are computed only partially; however, the discernibility matrix can apply to rough sets beyond classical rough sets. For instance, there are reduct computation methods [25,26] applicable to fuzzy rough sets [27] and covering-based rough sets [28]. In our future work, we aim to investigate the possibility of extending the concept of partial discernibility matrices to these alternative rough sets.

CRediT authorship contribution statement

Hajime Okawa: Writing – original draft, Visualization, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Yasuo Kudo:** Writing – review & editing, Supervision. **Tetsuya Murai:** Supervision.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used GPT-4o to improve the readability and language of the paper. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hajime Okawa reports financial support was provided by Japan Science and Technology Agency. The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by JST SPRING, Grant Number JPMJSP2153.

Data availability

Data will be made available on request.

References

[1] M. Rong, D. Gong, X. Gao, Feature selection and its use in big data: challenges, methods, and trends, IEEE Access 7 (2019) 19709–19725.
[2] P. Dhal, C. Azad, A comprehensive survey on feature selection in the various fields of machine learning, Appl. Intell. (2021) 1–39.
[3] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1) (2003) 155–176.
[4] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Theory and Decision Library, Kluwer Academic Publishers, 1991.
[5] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Słowiński (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Springer, Netherlands, Dordrecht, 1992, pp. 331–362.
[6] K. Liu, T. Li, X. Yang, X. Yang, D. Liu, Neighborhood rough set based ensemble feature selection with cross-class sample granulation, Appl. Soft Comput. 131 (2022) 109747.
[7] B. Sang, L. Yang, W. Xu, H. Chen, T. Li, W. Li, Vcos: multi-scale information fusion to feature selection using fuzzy rough combination entropy, Inf. Fusion 117 (2025) 102901.
[8] C. Wang, Y. Shi, X. Fan, M. Shao, Attribute reduction based on k-nearest neighborhood rough sets, Int. J. Approx. Reason. 106 (2019) 18–31.
[9] Y. Kudo, T. Murai, A parallel computation method for heuristic attribute reduction using reduced decision tables, J. Adv. Comput. Intell. Intell. Inform. 17 (3) (2013) 371–376.
[10] S. Akama, Y. Kudo, T. Murai, A parallel computation method for heuristic attribute reduction using reduced decision tables, in: Topics in Rough Set Theory: Current Applications to Granular Computing, Springer International Publishing, Cham, 2020, pp. 101–111.
[11] D. Chen, S. Zhao, L. Zhang, Y. Yang, X. Zhang, Sample pair selection for attribute reduction with rough set, IEEE Trans. Knowl. Data Eng. 24 (11) (2012) 2080–2093.
[12] P. Hońko, Attribute reduction: a horizontal data decomposition approach, Soft Comput. 20 (3) (2016) 951–966.
[13] M. Ye, C. Wu, Decision table decomposition using core attributes partition for attribute reduction, in: 2010 5th International Conference on Computer Science & Education, 2010, pp. 23–26.
[14] P. Zhang, H. Liang, The discussion of certain increment operator in rough set data analysis, Int. J. Inf. Technol. 12 (5) (2006).
[15] H. Okawa, Y. Kudo, T. Murai, Partial discernibility matrices for enumerating relative reducts of large datasets, in: 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), 2022.
[16] L. Polkowski, Rough Sets: Mathematical Foundations, Advances in Soft Computing, Physica-Verlag, 2002.
[17] R. Susmaga, Parallel computation of reducts, in: L. Polkowski, A. Skowron (Eds.), Rough Sets and Current Trends in Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 450–458.
[18] Y. Bao, X. Du, M. Deng, N. Ishii, An efficient method for computing all reducts, Trans. Jpn. Soc. Artif. Intell. 19 (3) (2004) 166–173.
[19] W. Cai, Z. Xu, W. Song, B. Yang, A quick algorithm for computing core based on the positive region, in: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), vol. 3, 2007, pp. 676–681.
[20] H. Ge, L. Li, C. Yang, Quick algorithm for computing core attribute, Control Decis. 24 (5) (2009) 738–742 (in Chinese).
[21] L. Yin, Z. Jiang, A fast attribute reduction algorithm based on a positive region sort ascending decision table, Symmetry 12 (7) (2020).
[22] D. Dua, C. Graff, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017.
[23] A.V. Dorogush, V. Ershov, A. Gulin, Catboost: gradient boosting with categorical features support, arXiv preprint, arXiv:1810.11363, 2018.
[24] Y. Yao, L. Zhao, A measurement theory view on the granularity of partitions, Inf. Sci. 213 (2012) 1–13.
[25] E.C.C. Tsang, D. Chen, D.S. Yeung, X. Wang, J.W.T. Lee, Attributes reduction using fuzzy rough sets, IEEE Trans. Fuzzy Syst. 16 (5) (2008) 1130–1141.
[26] C. Wang, Q. He, D. Chen, Q. Hu, A novel method for attribute reduction of covering decision systems, Inf. Sci. 254 (2014) 181–196.
[27] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (2–3) (1990) 191–209.
[28] Y. Yao, B. Yao, Covering based rough set approximations, Inf. Sci. 200 (2012) 91–107.