Accelerated Feature Selection via Discernibility Hashing: A Rough Set Approach

Sheng Luo , Linxiang Shi * , Lin Chen , Xiaolin Cao

Posted Date: 3 November 2025

doi: 10.20944/preprints202510.2529.v1

Keywords: discernibility matrix; rough sets; feature selection; discernibility hashing; attribute reduction

 doi:10.20944/preprints202510.2529.v1

Article

Accelerated Feature Selection via Discernibility Hashing: A Rough Set Approach

Sheng Luo [1,2], Linxiang Shi[1,2,*], Lin Chen[1,2,] and Xiaolin Cao[1,2]

[1] School of Computer and Information, Shanghai Polytechnic University, Shanghai, 201209, China
[2] Artificial Intelligence Institute, Shanghai Polytechnic University, Shanghai, 201209, China
* Correspondence: lxshi@sspu.edu.cn

Abstract

As a foundational analytical tool, the discernibility matrix plays a pivotal role in the systematic reduction of knowledge in rough set-based systems. Recent advancements in rough set theory have witnessed the proliferation of discernibility matrix-based knowledge reduction algorithms, with notable applications in classical, neighborhood, covering, and fuzzy rough set models. However, the quadratic growth of the discernibility matrix's complexity (relative to domain size) imposes fundamental scalability limits, rendering it inefficient for real-world applications with massive datasets. To address this issue, we introduced a discernibility hashing strategy to limit the growth scale of the discernibility attribute sets, and propose a feature selection algorithm via discernibility hash based on rough set theory. First, on the premise of keeping the information of the original discernibility matrix unchanged, the method maps the discernibility attribute set of all objects to the storage unit through a hash function and records the number of collisions to construct a discernibility hash. By using this mapping, the two-dimensional matrix space can be reduced to a one- dimensional hash space, which greatly removes invalid and redundant elements. Secondly, based on the discernibility hash, an efficient knowledge reduction algorithm is proposed. The algorithm avoids invalid and redundant elements attribute sets to participate in the knowledge reduction process, and improves the efficiency of the algorithm. Finally, the experimental results show that the method is superior to the discernibility matrix method in terms of storage space and running time.

Keywords: discernibility matrix; rough sets; feature selection; discernibility hashing; attribute reduction

1. Introduction

The exponential growth of data volume exacerbates challenges for machine learning algorithms when processing high-dimensional data, including computational inefficiency, elevated storage costs and the curse of dimensionality[1]. Feature selection is an efficient preprocessing method for high-dimensional data, generating a compact low-dimensional version of the data while retaining information for downstream machine learning models[2][3][4][5]. The objective of feature selection is to identify and retain task-relevant features while discarding irrelevant ones. To address the critical issue of feature correlation assessment, Rough Set Theory[6] offers a quantitative method that evaluates the consistency between indiscernibility relationships induced by the selected feature set and the decision set, using upper and lower approximation spaces.

As a significant research area in Rough set theory, the concept of feature selection (also known as attribute reduction) has received considerable attention[7][8][9], and has been widely applied in various fields of artificial intelligence such as classification learning[10], multi-label learning[11], clustering analysis[12] and text analysis[13] etc. Similar to Principal Component Analysis (PCA) and Independent Component Analysis (ICA), attribute reduction aims to identify an optimal feature subset from the entire feature space of an information decision system while preserving equivalent

 doi:10.20944/preprints202510.2529.v1

discernibility to the original set[5]. Existing attribute reduction methods fall into two principal categories: information theoretic approaches that leverage heuristic measures[2][7][14][15][16], and discernibility matrix-based methods[3][6][9][8][17][18][19]. The first category employs entropy-based metrics and mutual information to guide feature selection, while the second utilizes discernibility matrices and equivalence class analysis to identify redundant features.

The discernibility matrix-based feature selection method operates by storing the discernibility attribute set between domain objects in a discernibility matrix, from which it subsequently derives the core and reduct w.r.t. the conditional attribute set[20]. By employing matrix-based representation, this approach offers intuitive simplicity and has been widely adopted in attribute reduction computations. Unfortunately, the discernibility matrix also has some drawbacks, such as the space complexity and time complexity of the algorithm for constructing the discernibility matrix are $O(|U|^2)$ and $O(|U|^2 \cdot |C|)$ respectively, as well as a large number of repetitive elements and invalid elements [8][21]. In response to the shortcomings of the discernibility matrix, some improvements have been made in the existing research. For instance, methods such as using the simplest discernibility matrix[8], binary tree structure[22], and tree structure[21][23] are employed to implement the calculation of kernel and reduction. The storage structures adopted in these methods have improved the algorithm efficiency compared to the discernibility matrix. However, there are still some aspects that need to be enhanced for the subsequent feature selection algorithms, such as the mutual information calculation between candidate attributes and decision attributes, the complexity brought by the new structure in selecting candidate attributes, and the difficulty in calculating the priority among attributes, etc. To simultaneously preserve the information entropy of the discernibility matrix while avoiding computational complexity in candidate attribute priority ranking, we propose a hash-based discernibility attribute set representation termed discernibility hash, enabling the construction of an accelerated feature selection algorithm. Through a combined hash mapping and conflict resolution mechanism, the discernibility hash intelligently filters repeated elements in the discernibility matrix, achieving both storage compression and computational efficiency. Due to the smaller size of stored elements brought by the transformation from 2D matrix representation to 1D hash representation of discernibility attribute set, the theoretical time complexity and space complexity of the subsequent feature selection algorithm are improved. Therefore, according to the structural characteristics of discernibility hash representation, we propose an efficient feature selection algorithm based on discernibility hash representation, denoted FSDH. The experimental results show that our method not only reduces the time complexity and space complexity, but also maintains the information entropy of the decision information system unchanged, and achieves that the selected feature subset is consistent with the result of the methods based on discernibility matrix.

The main contributions of this paper are summarized as follows:

- We proposed a method of discernibility hash mapping for the discernibility attribute set of a decision information system, which solves the problem of excessive storage space complexity of the discernibility matrix.
- A fast feature selection algorithm based on discernibility hashing was proposed, which avoids the participation of duplicate elements and invalid elements in the discernibility attribute sets, thereby reducing time complexity and improving the performance of the algorithm.
- We also conducted experiments on the proposed algorithm to verify its effectiveness and efficiency. The experimental results show that our algorithm can achieve the same feature selection results as discernibility matrix-based algorithm, while significantly reducing the time and space complexity.

The rest of the paper is organized as follows. In Section 2, we briefly introduced some essential basic concepts and notations of the rough set theory. Section 3 mainly describes our proposed algorithms. The main contents include some basic concepts and definitions, the construction of discernibility hash, the design and analysis of the feature selection algorithm based on discernibility hash etc. In Section 4, we introduce an example to demonstrate the results of the discernibility matrix and discernibility hash, analyze their differences, and discuss their structural connections. In Section

 doi:10.20944/preprints202510.2529.v1

5, we designed an algorithm comparison experiment based on the UCI dataset and analyzed the experimental results in detail. Section 6 concludes this work with a comprehensive summary.

2. Preliminary

In this section, we will describe related preliminary notions for the classic Pawlak rough set theory. And the research problem will be formulated with these concepts.

Definition 1 (Decision Information System). *A decision information system[6][5] is formally defined as a four-tuple structure, $S = (U, A, V, f)$, where:*

(1). $U = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ represents the object domain (universal set of all possible objects);
(2). A is the attribute set composed of the conditional attributes C and the decision attribute D (i.e., $A = C \cup D$);
(3). V represents the value domain, expressed as $V = \cup_{a \in A} V_a$, where V_a is the value set of attribute a;
(4). $f : U \times A \to V$ is a function which maps $U \times A$ to the value domain V.

The following table 1 shows an example of the decision information system. In the system, there are some calculate routines as follows, for example, $U = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_7\}$, $C = \{a, b, c, d, e, f\}$, $D = \{g\}$ and $A = C \cup D$, $V_a = \{1, 2\}$, $V_b = \{0, 1, 2\}$, $V_c = \{0, 1, 2\}$, $V_d = \{0, 1\}$, $V_e = \{0, 1, 2\}$. For the function f, the following are still some computational examples: $f(\boldsymbol{x}_1, a) = 1$, $f(\boldsymbol{x}_2, c) = 0$ and $f(\boldsymbol{x}_3, e) = 0$.

Table 1. A decision information system.

Obj.	a	b	c	d	e	f	g
$\boldsymbol{x}_1$	1	0	1	1	2	0	1
$\boldsymbol{x}_2$	1	0	0	0	2	1	1
$\boldsymbol{x}_3$	1	2	0	0	0	1	2
$\boldsymbol{x}_4$	1	2	2	1	0	0	0
$\boldsymbol{x}_5$	2	1	0	0	1	2	2
$\boldsymbol{x}_6$	2	1	1	0	1	0	2
$\boldsymbol{x}_7$	2	1	2	1	1	0	1

Definition 2 (Indiscernibility Relation). *Given a subset R of attributes $A(R \subseteq A)$, the indiscernibility relation[5] $IND(R) \subseteq U \times U$ is defined as,*

$$IND(R) = \{(\boldsymbol{x}, \boldsymbol{y}) | \forall \boldsymbol{x}, \boldsymbol{y} \in U, \forall a \in R, f(\boldsymbol{x}, a) = f(\boldsymbol{y}, a)\} \tag{1}$$

Let $R = \{a, b\}$, the indiscernibility relation $\mathrm{IND}(R)$ partitions the objects into equivalence classes $\mathrm{IND}(R) = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \{\boldsymbol{x}_3, \boldsymbol{x}_4\}, \{\boldsymbol{x}_6, \boldsymbol{x}_6, \boldsymbol{x}_7\}\}$.

Definition 3 (Reduct). *Given a decision information system S, the attribute set $R \subseteq A$ is defined as a reduct[6], if R satisfies the following conditions:*

(1). $IND(R) = IND(A)$;
(2). $IND(R - \{a\}) \neq IND(A), \forall a \in R$.

Definition 4 (Core). *The core of a decision information system S with respect to attribute set A is defined as the intersection of all its reducts, i.e.,*

$$Core(A) = \bigcap_{R \in Reduct(A)} R \tag{2}$$

where $Reduct(A)$ stands for the set of all possible reducts of attribute set A.

 doi:10.20944/preprints202510.2529.v1

Definition 5 (Discernibility Matrices). *Let S be a decision information system, then the discernibility matrix [20] $M = (m_{ij})$ of S is defined as a set, i.e.,*

$$m_{ij} = \begin{cases} \{a\}, & \forall a \in C, f(x_i, a) \neq f(x_j, a) \wedge f(x_i, D) \neq f(x_j, D); \\ \emptyset, & otherwise. \end{cases} \tag{3}$$

where m_{ij} denotes an element of the matrix M with the condition $1 \leq i, j \leq n$.

The meaning of the entry M_{ij} is that the object pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be distinguished by any element in the attribute set M_{ij}. In another word, the objects $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are discerned precisely when the condition $m_{ij} \neq \emptyset$ is satisfied. It is obvious that discernibility matrix M is symmetric, i.e., $m_{ij} = m_{ji}$, and $m_{ii} = \emptyset, \forall i, j \in [1, n]$. Therefore, to minimize storage redundancy, we can retain only one triangular portion (either lower or upper) of the symmetric matrix, discarding the redundant counterpart. This approach preserves all essential information while reducing storage requirements by half.

Theorem 1. *Given a decision information system S and its discernibility matrix M, an attribute set R is a reduct if and only if [8],*

(1). $\forall (\boldsymbol{x}_i, \boldsymbol{x}_j) \in U \times U$ *and* $m_{ij} \neq \emptyset, [R \cap m_{ij} \neq \emptyset]$;
(2). $\forall a \in R, \exists (\boldsymbol{x}_i, \boldsymbol{x}_j) \in U \times U, [m_{ij} \neq \emptyset]$ *and* $[(R - \{a\}) \cap m_{ij} = \emptyset]$.

The proposition above enables the validation of attribute subsets as potential reducts by leveraging the discernibility matrix of a decision information system. However, the problem of how to construct a reduct based on the discernibility matrix still remains unsolved.

3. Proposed Method

As defined in the previous section's discernibility matrix formulation, the time complexity and the space complexity of the algorithm for constructing the discernibility matrix are $O(|U|^2)$ and $O(|U|^2 \times |C|)$ respectively. It is not difficult to find that there are redundant calculation elements [8][9] in the discernibility matrix. By eliminating these redundant elements, storage space can be significantly compressed, which in turn reduces the computational frequency of the attribute reduction algorithm and enhances overall efficiency. Moreover, as the object cardinality of the decision information system grows, the probability of repeated occurrences of the discernibility attribute set within the matrix proportionally increases. To mitigate redundant computations and storage overhead, we introduce a novel hash-based storage structure and a corresponding accelerated attribute reduction algorithm.

Definition 6 (Ordered Discernibility Attributes). *Given a decision information system S, the discernibility attribute of an object pair $(\boldsymbol{x}_i, \boldsymbol{x}_j), 1 \leq i, j \leq n$, is formally defined as:*

$$h_k = \begin{cases} \varphi(\{a\}), & \forall a \in C, f(\boldsymbol{x}_i, a) \neq f(\boldsymbol{x}_j, a) \wedge f(\boldsymbol{x}_i, D) \neq f(\boldsymbol{x}_j, D); \\ \emptyset, & otherwise. \end{cases} \tag{4}$$

where $\varphi(\cdot)$ is a sorting strategy that arranges the elements of the set $\{a\}$ in ascending (or descending) order, and k is a number that is much smaller than n^2 and it depends on the repetition rate of the discernibility attribute set.

For instance, suppose there are the following discernibility attribute sets,

$$\{a, b, d\}, \{c, a\}, \{b, d, a\}, \{a, b, c\}$$

 doi:10.20944/preprints202510.2529.v1

then, we can obtain the ordered discernibility attributes,

$$\begin{aligned}\varphi(\{a,b,d\}) &= \{a,b,d\}\\ \varphi(\{c,a\}) &= \{a,c\}\\ \varphi(\{b,d,a\}) &= \{b,d,a\}\\ \varphi(\{a,b,c\}) &= \{a,b,c\}\end{aligned}$$

It can be seen that the first line of the above equation is equivalent to the third line. This also reveals the implicit relationship between k and i,j, that is, h_k is equal to all the m_{ij} elements that have the same ordered discernibility attributes.

Definition 7 (Discernibility Hash)**.** *Given a decision information system S and its ordered discernibility attributes $h_k[h_k = \varphi(m_{ij}), \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in U]$, the discernibility hash function is defined as:*

$$\delta(h_k) = (h_k)_r \tag{5}$$

where r represents the encoding cardinality of the key. All the discernibility attributes of the decision information system S are calculated by the discernibility hash function to form a hash table called the discernibility hash.

Consider a decision information system S containing discernibility attributes, for example,

$$h_1 = \{a,c\}, h_3 = \{a,b,c\} \text{ and } h_4 = \{b,c\}.$$

The corresponding hash values are computed as,

$$\delta(h_1) = 101, \delta(h_3) = 111, \text{ and } \delta(h_4) = 011$$

respectively, by setting the parameter $r = 2$. To simplify the problem, it is also possible to simply adopt the identity function as the hash function, i.e., remove the digit '0' from the binary encoding, and then replace the digit '1' directly with the corresponding attribute name. Therefore, we can obtain the keys of the hash table as follow:

$$\delta(h_1) = ac, \delta(h_3) = abc, \text{ and } \delta(h_4) = bc$$

Unlike the discernibility matrix that stores all discernibility attributes explicitly, the hash-based approach maps key values directly to storage locations through deterministic hash functions. While both storage mechanisms exhibit comparable access time performance, the hash-based approach demonstrates superior space efficiency by effectively compressing redundant elements and empty slots inherent in the discernibility matrix. In hash-based storage systems, collisions inevitably occur when distinct keys share identical hash values. These collisions directly correspond to redundant elements in the discernibility matrix, where multiple object pairs may yield identical discernibility attribute sets. Therefore, the number of hash value collisions directly quantifies the redundancy in stored discernibility attributes, where each collision indicates a repeated entry in the discernibility matrix. For large-scale data sets, a limited set of attributes will inevitably produce a large number of repeated discernibility attribute subsets. Removing redundancy will optimize the calculation, which can greatly accelerate the calculation of attribute reduction.

Theorem 2. *Given a decision information system S and its discernibility hash H, an attribute set R is a reduct if and only if*

(1). $\forall h \in H, \quad R \cap h \neq \emptyset$;

(2). $\forall a \in R, \exists h \in H, \quad (R - \{a\}) \cap h = \emptyset.$

 doi:10.20944/preprints202510.2529.v1

Proof. Let $\mathrm{IND}(C) = \{C_1, C_2, ..., C_l\}$ and $\mathrm{IND}(R) = \{R_1, R_2, ..., R_l\}$, where l denotes the cardinality of the equivalence classes induced by the conditional attribute set C and the decision attribute D. For any non-empty set $h \in H$, it indicates that

$$\{a\} = \delta^{-1}(\varphi^{-1}(h)) \tag{6}$$

Then, there exists $\boldsymbol{x}_i, \boldsymbol{x}_j$ such that $f(\boldsymbol{x}_i, a) \neq f(\boldsymbol{x}_j, a) \wedge f(\boldsymbol{x}_i, D) \neq f(\boldsymbol{x}_j, D)$. That is, the object pair $\boldsymbol{x}_i, \boldsymbol{x}_j$ belongs to different partitions, i.e.

$$\text{if } \boldsymbol{x}_i \in C_i \wedge \boldsymbol{x}_j \in C_j. \quad \text{then } C_i \neq C_j. \tag{7}$$

If $R \cap h = \emptyset$, then we can claim that, the object pair $\boldsymbol{x}_i, \boldsymbol{x}_j$ belongs to the identical partition that induced by the attribute set R, which is contradictory to Equation 7., i.e.,

$$\mathrm{IND}(C) \neq \mathrm{IND}(R)$$

As a result, Condition 1 is satisfied. Suppose that after removing element a from R, there exists no $h \in H$ such that $(R \setminus \{a\}) \cap h = \emptyset)$. This implies $(R \setminus \{a\})$ remains a reduct, which contradicts the definition of reducts. Therefore, Condition 2 must hold. □

Condition 1 establishes that R is sufficient for distinguishing all discernible object pairs. It is easy to see that the union of all the elements of discernibility hash H must satisfy Condition 1. Condition 2 shows whether each element in R can be removed.

Theorem 3. *The discernibility hash and the discernibility matrix derived from the same information system are informationally equivalent, preserving identical discernibility attribute sets despite their distinct storage representations.*

Proof. The probability that the element $m_{ij}(1 \leq i, j \leq n)$ of the arbitrary discernibility matrix M is:

$$p_{\{attrs\}} = \frac{2 \times \#(\{attrs\})}{n(n-1)} \tag{8}$$

where $\#(\cdot)$ is a function to calculate the occurrence frequency of attribute set $\{attrs\} \triangleq m_{ij}$ within the discernibility matrix M, and n, $\frac{n(n-1)}{2}$ are the total number of objects and entries in the decision information system S, respectively. Then, the information entropy of the matrix M is:

$$\mathrm{entropy}(M) = \sum_{\{attrs\}} -p_{\{attrs\}} \cdot \log(p_{\{attrs\}}) \tag{9}$$

For any ordered discernibility attribute $h_k \in H$, the probability of it appearing in the discernibility hash is,

$$p_{\mathrm{key}} = \frac{\mathrm{freq}(\delta(\mathrm{key})) \times 2}{n(n-1)} \tag{10}$$

where $\mathrm{key} \triangleq h_k(1 \leq k \leq n)$, $\mathrm{freq}(\cdot)$ computes the frequency of the key appearing in discernibility hash H. For each discernibility attribute pair $(m_{ij}, h_k) \in M \times H$, the following equation necessarily holds:

$$\#(m_{ij}) = \mathrm{freq}(\delta(h_k)), \quad \text{if } \forall 1 \leq i, j, k \leq n, \mathrm{sorted}(m_{ij}) = h_k. \tag{11}$$

In other words, any discernibility attribute set appears the same number of times in the discernibility matrix and the discernibility hash, regardless of the type of storage structure. Therefore, the following equation holds true:

$$\mathrm{entropy}(M) = \mathrm{entropy}(H). \tag{12}$$

 doi:10.20944/preprints202510.2529.v1

□

As a consequence, after using discernibility hash representation, the amount of information entropy in the system is equal to that of the discernibility matrix. The two approaches are logically equivalent, differing solely in their storage architecture (e.g., adjacency list vs. hash table), with no information loss during representation. Obviously, by taking advantage of hash collisions, we can significantly reduce the size of the discernibility attributes set, thereby lowering the complexity of the subsequent processing algorithm.

3.1. Efficient Knowledge Reduction Algorithm

The knowledge reduction algorithm is able to divided into two stages. The initial stage systematically computes all discernibility attributes for object pairs in the decision information system, subsequently storing them in a target hash table. The final stage employs the discernibility hash strategy to construct the reduct of the decision information system, thereby optimizing both computational efficiency and storage utilization. The construction algorithm of the discernibility hash table and the corresponding fast knowledge reduction algorithm are presented below.

Algorithm 1 Discernibility hash construction algorithm.

Input: S: the decision information system; $\delta(\cdot)$: the discernibility hash function.

Output: H: the discernibility hash table of S.

for $i = 1, 2, ..., N$ **do**
  for $j = 1, 2, ..., i$ **do**
    $m_{ij} \leftarrow$ Eq.(3);
    $key \leftarrow$ Eq.(4);
    $loc_{key} \leftarrow$ Eq.(5): calculate the mapping addresses of the discernibility attributes;
    insert loc_{hk} into the target hash table H and track the number of collision;
  end for
end for
return the discernibility hash H.

Algorithm 2 Efficient Feature Selection algorithm based on Discernibility Hash, FSDH.

Input: H: the discernibility hash H.

Output: $CORE(A)$: the core of the attribute set A; $RED(A)$: a reduct of the attribute set A.

$attr\_freq \leftarrow$ the frequency of all attributes of the difference hash;
$CORE(A) \leftarrow \emptyset$;
for key in H **do**
  if $len(key) == 1$ and key not in CORE(A) **then**
    add key to the set CORE(A);
  end if
end for
$RED(A) \leftarrow CORE(A)$;
repeat
  $a \leftarrow max\_item(attr\_freq)$; //select the highest frequency item from $attr\_freq$
  add a to the set RED(A);
  remove all keys containing a from the discernibility hash H;
until {RED(A) meets the reduction criteria (Def. 3)}
return $CORE(A), RED(A)$;

An important criterion for a hash function, Eq.(5), is that it must ensure that different keys are mapped to different addresses, while the same key must be mapped to the same address. The purpose of the discernibility hash is to utilize the hash conflicts to compress the repeated occurrence of the same key in the discernibility. If different sets of discernibility attributes are mapped to the same storage location, it will cause calculation difficulties for subsequent knowledge reduction. This is the most distinctive feature that sets Eq.(5) apart from others. Obviously, the time complexity of Algorithm **??** is the same as that of discernibility matrix based algorithm, both being $O(n^2)$,

Based on the output of Algorithm 1, we propose an efficient feature selection algorithm based on discernibility hash strategy, termed FSDH, in Algorithm 2

The algorithm JohnsonsReduct[21] is an attribute reduction algorithm that utilizes the Johnson heuristic strategy based on the discernibility matrix. The difference between JohnsonReduct and FSDH lies solely in their storage representation of the identical discernibility attribute set, that is, JohnsonReduct employs a discernibility matrix whereas FSDH utilizes a hash table.

 doi:10.20944/preprints202510.2529.v1

Theorem 4. *FSDH and JohnsonReduct have the same output result, and the time upper bound of FSDH is that of JohnsonReduct, while the lower bound is determined by the storage compression rate ρ (Equation 17).*

Proof. Suppose the set of the discernibility attribute set is As, the discernibility matrix is $M = (m_{ij})$, and the discernibility hash is $H = [h : v_h]$. According to Theorem 3, for any discernibility attribute set $s \in As$, it can be concluded that,

$$\begin{aligned} \forall i, j \in [1, n], \quad & \exists m_{ij} \wedge \text{freq}(m_{ij}) = \text{freq}(s) \\ \forall k \in [1, |H|], \quad & \exists h_k = \delta(\varphi(m_{ij})) \wedge v_{h_k} = \text{freq}(s) \end{aligned} \tag{13}$$

where $\text{freq}(s)$ is a counting function, and v_{h_k} stands for the value of the key h_k. Consequently, since the discernibility attribute sets stored in matrix and hash formats are statistically equivalent, the outputs of the two algorithms are consistent.

Algorithm **??** demonstrates that the discernibility matrix and hash differ exclusively in their first and 12th rows. Given identical storage structures, these rows exhibit equivalent time and space complexities. If the discernibility matrix M is used, the time complexity of the first and 12th rows is,

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=1}^{|C|} 1 \\ &= |C| \sum_{i=1}^{n} i = |C| \frac{n(n+1)}{2} \end{aligned} \tag{14}$$

Therefore, the time complexity is $O(n^2)$ when $|C| < n$, otherwise it is $O(n^3)$. For the discernibility hash H, since the discernibility attribute set has been encoded, it can be directly mapped to the physical address. Hence, the operation of determining whether a candidate attribute belongs to a discernibility attribute set can be directly performed using the encoding value through subtraction, with a time complexity of $O(1)$. Then, the time complexity of the discernibility hash H is,

$$F(n) = \sum_{i=1}^{|H|} 1 = O(|H|) \tag{15}$$

where $|H|$ is the length of hash table H, and $|H| = \frac{n^2}{2}\rho$ (ρ represents the compression rate, with a value ranging from [0, 1]). in conclusion, the following inequalities hold,

$$O\left(n^2 \cdot \frac{\rho}{2}\right) \leq F(n) \leq T(n) \tag{16}$$

That is, the upper bound of the time complexity of FSDH is JohnsonReduct, while the lower bound is determined by the compression rate ρ. □

The computational bottleneck of the feature selection algorithm primarily arises from traversing the discernibility matrix, where repeated attribute comparisons dominate the runtime complexity. The proposed algorithm achieves significant space complexity reduction, transitioning from $O(n^2 \times |C|)$ in traditional knowledge reduction algorithm to $O(|H|)$ in our hash-based approach, where $|H| \approx \frac{n^2}{2} \cdot \rho$. This improvement is particularly notable as $\frac{n^2}{2} \cdot \rho \ll n^2 \times |C|$ for large-scale datasets. In the subsequent experimental section, the algorithm's results on the UCI dataset confirmed the correctness of this statement.

4. Case Study

This section uses an example to analyze the differences between the discernibility hash and the discernibility matrix. We adopt the benchmark dataset [21], referred to Toy1, as our target decision

 doi:10.20944/preprints202510.2529.v1

information system for evaluation. The dataset Toy1, as presented in Table 2, comprises four conditional attributes (columns $a - d$), one decision attribute (column e) and seven objects.

Table 2. Example dataset Toy1.

Obj.	a	b	c	d	e
$\boldsymbol{x}_1$	1	0	1	1	1
$\boldsymbol{x}_2$	1	0	0	0	1
$\boldsymbol{x}_3$	1	2	0	0	2
$\boldsymbol{x}_4$	1	2	2	1	0
$\boldsymbol{x}_5$	2	1	0	0	2
$\boldsymbol{x}_6$	2	1	1	0	2
$\boldsymbol{x}_7$	2	1	2	1	1

Applying Equation 3 to the Toy1 dataset yields the discernibility matrix presented in Table 3, where each entry m_{ij} quantifies the attribute-level dissimilarity between objects $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. As demonstrated in Table 3, the discernibility matrix consists of 21 elements, including 6 empty sets and repeated sets of discernibility attributes. Among these, the empty sets can be completely ignored, while the repeated attribute sets can be compressed. Obviously, the discernibility matrix exhibits significant redundancy in stored attribute sets. By employing hash-based compression techniques, these repetitive elements can be systematically eliminated, achieving dual objectives: (1) reducing space complexity form $O(n^2)$ to $O(|H|)$, and (2) accelerating subsequent knowledge reduction operations through streamlined data access.

Table 3. The discernibility matrix of the dataset Toy1

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	$\boldsymbol{x}_6$
$\boldsymbol{x}_2$	$\emptyset$					
$\boldsymbol{x}_3$	{bcd}	{b}				
$\boldsymbol{x}_4$	{bc}	{bcd}	{cd}			
$\boldsymbol{x}_5$	{abcd}	{ab}	$\emptyset$	{abcd}		
$\boldsymbol{x}_6$	{abd}	{abc}	$\emptyset$	{abcd}	$\emptyset$	
$\boldsymbol{x}_7$	$\emptyset$	$\emptyset$	{abcd}	{ab}	{cd}	{cd}

The discernibility hash table for the dataset Toy1, constructed using Algorithm **??**, is presented in Table 4. This table systematically maps discernibility attribute sets to their corresponding hash values, demonstrating the algorithm's efficiency in reducing storage redundancy. The symbol ρ in the table stands for the storage compression rate which is calculated by the following equation,

$$\rho = \frac{\text{the total number of unique discernibility attribute sets}}{\text{the total number of discernibility attribute sets}} \tag{17}$$

The compression rate ρ reflects the degree of conflict in discernibility hash table. The higher the degree of conflict, the more sets of discernibility attributes are repeated, and thus the smaller the compressible space will be.

By comparing Table 3 and Table 4, it can be seen that, compared with the discernibility matrix, the size of the discernibility hash has decreased from 21 to 8, and the compression storage rate is $8/21 \approx 38.1\%$. In other words, the discernibility hash method achieves a 38.1% reduction in storage space compared to the discernibility matrix, as it eliminates the need to store and compute 61.8% of duplicate entries in subsequent steps. Therefore, the advantage of discernibility hash H becomes evident, that is, while keeping the system's information entropy unchanged, the storage space has been effectively compressed.

 doi:10.20944/preprints202510.2529.v1

Table 4. The discernibility hash table of the dataset Toy1

key	item frequency
{ab}	2
{abc}	1
{abcd}	4
{abd}	1
{b}	1
{bc}	1
{bcd}	2
{cd}	3
compression rate ρ:	38.1%

5. Experiments

The empirical study of the FSDH is given in this section. We first setup the experiments by introducing the datasets. Then we conduct a comprehensive performance evaluation by analyzing both time complexity and space usages through rigorous empirical testing.

5.1. Experimental setup

The experimental platform utilized in this study comprised a Core i7-8550U CPU (1.80GHz base clock) with 16GB DDR4 RAM, ensuring sufficient computational capacity for all test. To test the performance of our method, we employed seven benchmark datasets for evaluation: six from the UCI Machine Learning Repository (Weather, Contact-lenses, Zoo, Breast-cancer, Vote and Tictactoe) and one synthetic dataset (Toy1). Table 5 presents comprehensive descriptive statistics for the seven benchmark datasets, including sample sizes, attribute sizes and class sizes.

Table 5. Selected data sets.

Name	Abbreviation	Instances	Attributes	Classes
toy1	D1	7	4	3
weather	D2	14	4	2
contact-lenses	D3	24	4	3
zoo	D4	101	17	7
breast-cancer	D5	286	9	2
vote	D6	435	16	2
tictactoe	D7	958	9	2

In order to verify the performance differences between discernibility matrics and discernibility hash tables, we chose the algorithm JohnsonsReduct [21], which uses the same heuristic strategy as FSDH, for comparison.

5.2. Results

Table 6 presents a comparative analysis of temporal and spatial complexity across seven benchmark datasets, with execution time (in seconds) and storage consumption (in unit amounts) quantitatively evaluated under uniform experimental conditions. In the table, the column "Count of units" represents the number of discernibility attribute sets generated and stored by the algorithm JohnsonReduct and FSDH, respectively, and the column "Time(sec.)" stands for the time consumed by the algorithm, measured in seconds. Furthermore, the meaning of "Time rate", denoted as τ in the table, is as shown in the following equation:

$$\tau = \frac{\text{Executation time of ERDH}}{\text{Executation time of JohnsonReduct}} \tag{18}$$

 doi:10.20944/preprints202510.2529.v1

The ρ in the table is consistent with Equation 17, both representing the compression degree of all storage units.

Table 6. Experimental results of the UCI datasets.

	JohnsonReduct		FSDH		Time rate(τ)	Compression rate(ρ)
	Count of units	Time(sec.)	Count of units	Time(sec.)		
Toy1	21	4.96E-04	8	7.73E-05	15.59%	38.10%
Weather	91	8.38E-04	13	9.04E-05	10.79%	14.29%
Contact-lenses	276	3.53E-04	15	1.05E-04	29.76%	5.43%
Zoo	5050	4.91E-02	956	2.95E-03	6.00%	18.93%
Breast	40755	2.93E-01	488	7.30E-03	2.50%	1.20%
Vote	94395	7.85E+00	10548	1.06E+00	13.53%	11.17%
Tictactoe	458403	2.27E+01	501	7.15E-03	0.03%	0.11%

As can be seen form Table 6, the algorithm FSDH that based on discernibiltiy hash shows significant improvement in efficiency compared to the algorithm JohnsonReduct which based on discernibility matrix, both in terms of time and space cost. In these data sets, the algorithm demonstrates its most significant improvement in data set Tictactoe, where the entry size of discernibility matrix is reduced from 458403 to 501 (achieving a compression rate of 0.11%). As a result, the execution time of the subsequent feature selection algorithm decreases to merely 0.03% of the original algorithm's time. For data set Contact-lenses, despite having the lowest compression rate among the data sets, the storage units still achieve a compression rate of 29.76%, which constitutes a substantial improvement for the discernibility matrix.

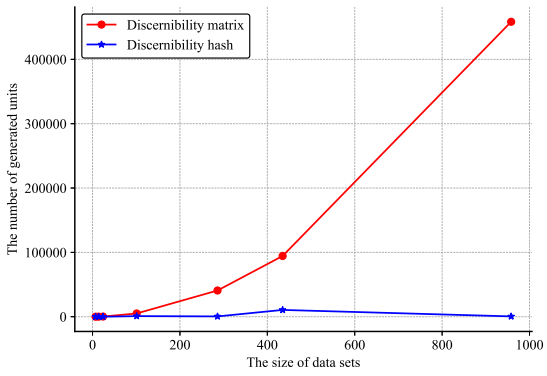

Figure 1. The comparison of storage unit amount

Figure 1 shows the comparison of storage units produced by the two structures on datasets of different sizes. As the size of the dataset increases from 7 to 958, the number of discernibility matrix cells also grows accordingly at a rate of $O(n^2)$, while the growth rate of hash cells is approximately linear, at $O(n)$. It indicates that as the sample size increases, the repetition rate of the discernibility attributes also shows an increasing trend. Therefore, a large number of repetitive units in the discernibility matrices are completely unnecessary to retain. After deletion, it can save computing resources and avoid redundant calculations. Avoiding repetition is precisely what discernibility hashing is good at. By using the hashing mechanism, conflicts can be utilized to filter out duplicate elements, without any loss of the system's information entropy, and achieving the goal of simultaneously saving computing time and storage space.

 doi:10.20944/preprints202510.2529.v1

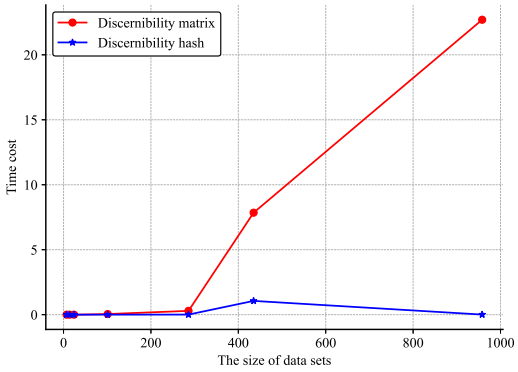

Figure 2. Execution time comparison

The removal of repetitive elements via filtering directly contributes to a marked increase in the computational efficiency of feature selection algorithms. Figure 2 demonstrates the performance of discernibility matrix and discernibility hash-based approaches across datasets of varying scales, revealing their distinct impacts on feature selection efficiency. From lines 7 and 12 of Algorithm **??**, it can be seen that the time complexity of this algorithm is closely related to the size of all unique discernibility attribute sets. Therefore, the size of the unique discernibility attribute sets is the key factor determining the efficiency of the feature selection algorithms. The redundant storage can prove to be completely unnecessary and increase the complexity of the algorithm. This can be seen form Figure 2, especially when the sample data reaches a certain level, for example 400 in the figure, this characteristic becomes very significant.

In summary, the feature selection algorithm based on discernibility hashing not only reduces redundant storage units without losing the system information entropy, but also significantly improves the execution time compared to the discernibility matrix based feature selection algorithms.

6. Conclusions

Due to the redundant storage units, it is a challenging task that how to effectively and robustly discover principal component of a decision information system, especially when the size of training set reaches a certain level. In this work, we proposed a novel representation for discernibility attributes, i.e., discernibility hash, and developed an efficient feature selection algorithm to choose the most important feature of the decision information system for the subsequent tasks. Experiments show that our method is superior to the discernibility matrix based methods.

Several aspects of the new method are worth investigating in further depth, including how to compress the discernibility hash, feature selection strategies and how to handle the numerical feature etc. In the future, our work will be focused on the compression of discernibility hash, because a good data representation will be able to reduce algorithm complexity and also improve algorithm efficiency.

Author Contributions: Conceptualization, Luo S.; methodology, Luo S.; software, Luo S.; validation, Shi L., Cao X.; formal analysis, Shi L.; investigation, Luo S.; resources, Luo S.; data curation, Chen L.; writing—original draft preparation, Luo S.; writing—review and editing, Shi L.; visualization, Cao X.; supervision, Chen L.; project administration, Shi L.; funding acquisition, Shi L. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ayesha, S.; Hanif, M.K.; Talib, R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion* **2020**, *59*, 44–58. https://doi.org/10.1016/j.inffus.2020.01.005.

 doi:10.20944/preprints202510.2529.v1

2. Yuan, K.; Miao, D.; Zhang, H.; Pedrycz, W. An Efficient and Robust Feature Selection Approach Based on Zentropy Measure and Neighborhood-Aware Model. *IEEE Transactions on Neural Networks and Learning Systems* **2025**, *36*, 16351–16365. https://doi.org/10.1109/TNNLS.2025.3565320.
3. Qian, W.; Wan, L.; Shu, W. Semi-supervised feature selection based on discernibility matrix and mutual information. *Applied Intelligence* **2024**, *54*, 7278–7295.
4. Liu, C.; Lin, B.; Miao, D. A novel adaptive neighborhood rough sets based on sparrow search algorithm and feature selection. *Information Sciences* **2024**, *679*, 121099. https://doi.org/10.1016/j.ins.2024.121099.
5. Sheng, L.; Duoqian, M.; Zhifei, Z. A neighborhood rough set model with nominal metric embedding. *Information Sciences* **2020**, *520*, 373–388.
6. Pawlak, Z.; Skowron, A. Rudiments of rough sets. *Information Sciences* **2007**, *177*, 3–27.
7. Wang, C.; Huang, Y.; Ding, W.; Cao, Z. Attribute reduction with fuzzy rough self-information measures. *Information Sciences* **2021**, *549*, 68–86. https://doi.org/10.1016/j.ins.2020.11.021.
8. Yao, Y.; Zhao, Y. Discernibility matrix simplification for constructing attribute reducts. *Information Sciences* **2009**, *179*, 867–882.
9. Lang, G.; Li, Q.; Guo, L. Discernibility matrix simplification with new attribute dependency functions for incomplete information systems. *Knowledge and Information Systems* **2013**, *37*, 611–638.
10. Trabelsi, A.; Elouedi, Z.; Lefevre, E. An ensemble classifier through rough set reducts for handling data with evidential attributes. *Information Sciences* **2023**, *635*, 414–429. https://doi.org/10.1016/j.ins.2023.01.091.
11. Wang, Z.; Chen, D.; Che, X. Learning Operator-Valued Kernels From Multilabel Datesets With Fuzzy Rough Sets. *IEEE Transactions on Fuzzy Systems* **2025**, *33*, 1311–1321. https://doi.org/10.1109/TFUZZ.2024.3522466.
12. Vidhya, K.A.; Geetha, T.V. Rough set theory for document clustering: A review. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2165–2185.
13. Singh, G.K.; Mandal, S. Cluster Analysis using Rough Set Theory. *Journal of Informatics and Mathematical Sciences* **2017**, *9*, 509–520. https://doi.org/10.26713/jims.v9i3.754.
14. Ji, X.; Li, J.; Yao, S.; Zhao, P. Attribute reduction based on fusion information entropy. *International Journal of Approximate Reasoning* **2023**, *160*, 108949. https://doi.org/10.1016/j.ijar.2023.108949.
15. Gao, C.; Zhou, J.; Miao, D.; Yue, X.; Wan, J. Granular-conditional-entropy-based attribute reduction for partially labeled data with proxy labels. *Information Sciences* **2021**, *580*, 111–128. https://doi.org/10.1016/j.ins.2021.08.067.
16. Wang, P.; Qu, L.; Zhang, Q. Information entropy based attribute reduction for incomplete heterogeneous data. *J. Intell. Fuzzy Syst.* **2022**, *43*, 219–236. https://doi.org/10.3233/JIFS-212037.
17. Wei, W.; Wu, X.; Liang, J.; Cui, J.; Sun, Y. Discernibility matrix based incremental attribute reduction for dynamic data. *Knowledge-Based Systems* **2018**, *140*, 142–157. https://doi.org/https://doi.org/10.1016/j.knosys.2017.10.033.
18. Ma, F.; Ding, M.; Zhang, T.; Cao, J. Compressed binary discernibility matrix based incremental attribute reduction algorithm for group dynamic data. *Neurocomputing* **2019**, *344*, 20–27. NEURAL LEARNING IN LIFE SYSTEM AND ENERGY SYSTEM, https://doi.org/10.1016/j.neucom.2018.01.094.
19. Liu, Y.; Zheng, L.; Xiu, Y.; Yin, H.; Zhao, S.; Wang, X.; Chen, H.; Li, C. Discernibility matrix based incremental feature selection on fused decision tables. *International Journal of Approximate Reasoning* **2020**, *118*, 1–26. https://doi.org/10.1016/j.ijar.2019.11.010.
20. Skowron, A.; Rauszer, C. *The Discernibility Matrices and Functions in Information Systems*; Kluwer, 1992.
21. Yang, M.; Yang, P. A novel condensing tree structure for rough set feature selection. *Neurocomputing* **2008**, *71*, 1092–1100.
22. Lu, Z.; Qin, Z.; Jin, Q.; Li, S. Constructing Rough Set Based Unbalanced Binary Tree for Feature Selection. *Chinese Journal of Electronics* **2014**, *23*, 474–479. https://doi.org/10.23919/CJE.2014.10851196.
23. Jiang, Y.; Yu, Y. Minimal attribute reduction with rough set based on compactness discernibility information tree. *Soft Computing* **2016**, *20*, 2233–2243.