Information Sciences 689 (2025) 121472

Contents lists available at ScienceDirect

Information Sciences

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

Parallel attribute reduction algorithm for unlabeled data based on fuzzy discernibility matrix and soft deletion behavior

Haotong Wen [a], Yi Xu [a,*], Meishe Liang [b]

[a] School of Computer Science and Technology, Anhui University, Hefei 230601, China
[b] Department of Mathematics and Statistics, Shijiazhuang Tiedao University, Shijiazhuang, China

ARTICLE INFO

Dataset link: http://archive.ics.uci.edu/ml

Keywords:
Fuzzy rough set
Unsupervised attribute reduction
Discernibility matrices
Algorithm acceleration

ABSTRACT

In attribute reduction algorithms, discernibility matrix-based methods and heuristic-based methods are two highly effective approaches. While the prevailing view is that heuristic algorithms are faster than discernibility matrix-based methods, the rise of GPUs and other matrix-based parallel computing devices has enabled discernibility matrix reduction methods to achieve faster computation speeds by leveraging their matrix characteristics. However, few discernibility matrix-based methods can directly adapt to GPU devices, and for unlabeled data, existing discernibility matrix-based methods fail to fully utilize the fuzzy information, resulting in unsatisfactory outcomes. In this paper, we propose a parallel attribute reduction algorithm based on fuzzy discernibility matrices and soft deletion behavior. To achieve parallel computing, we transform the traditional 3-dimensional discernibility matrix into a 2-dimensional matrix. To maximize the use of fuzzy discernibility information, we introduce a fuzzy deletion function, which can effectively update the discernibility matrix by incorporating fuzzy discernibility information. Finally, we propose a stopping mechanism for the algorithm, enabling it to select fewer attributes under appropriate conditions. Experiments demonstrate that our algorithm significantly increases computation speed compared to traditional heuristic algorithms and reduces the number of attributes while maintaining and enhancing the effectiveness of downstream tasks.

1. Introduction

In the realm of data analysis and knowledge discovery, attribute reduction algorithms are pivotal mechanisms that enhance the efficiency and interpretability of machine learning models [1–3]. These algorithms aim to identify and retain the most informative features within a dataset while discarding redundant or irrelevant ones. Since Pawlak introduced the concept of rough sets, attribute reduction algorithms based on rough set theory have achieved considerable success. However, classical rough set theory cannot be applied to numerical data, and although discretization can address this issue, the fuzzy information in the data is lost in the process. Dubois and Prade [4,5] subsequently proposed the theory of fuzzy rough sets, which can effectively utilize the fuzzy information in data. Researchers have developed various attribute reduction algorithms based on this foundation [4–15].

Attribute reduction algorithms based on fuzzy rough set theory can be broadly categorized into two types: heuristic algorithms [16] and those based on discernibility matrices. Heuristic algorithms select attributes by constructing attribute evaluation functions [17–20], while discernibility matrix-based algorithms [21] take a different approach. Although heuristic algorithms are

* Corresponding author.
E-mail addresses: 1202132011@student.stdu.edu.cn (H. Wen), xuyi1023@126.com (Y. Xu), liangmeishe@stdu.edu.cn (M. Liang).

https://doi.org/10.1016/j.ins.2024.121472
Received 17 May 2024; Received in revised form 26 August 2024; Accepted 9 September 2024

Available online 12 September 2024

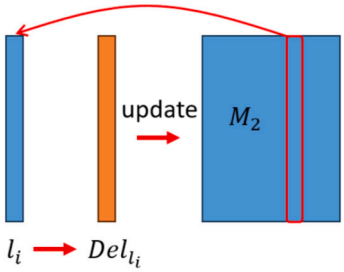


Fig. 1. Reduction Framework.

generally considered faster, the development of GPUs has significantly accelerated large-scale matrix computations and parallel computing techniques. Given that many AI technologies require GPU training [22–25], developing an attribute reduction algorithm optimized for GPUs not only aligns the reduction algorithm with intelligent computing tasks but also substantially accelerates the reduction process. Heuristic algorithms, however, pose challenges when adapting to matrix-based parallel structures. For example, in positive region reduction, calculating the lower approximation of each object's fuzzy set for all target sets requires at least one layer of iteration [26]. Additionally, heuristic algorithms often necessitate calculating the importance of multiple attributes in each iteration, which complicates parallelization [3,27,28]. Although discernibility matrix-based methods can be easily adapted for parallelization, most existing methods have not been modified accordingly. Most acceleration for discernibility matrix-based algorithms focus on bypassing ineffective computations to achieve faster performance. For instance, Cheng et al. [29] proposed a graph-based attribute reduction algorithm that utilizes information from fuzzy lower approximations when constructing the discernibility matrix. Liu et al. [30] introduced a discernibility matrix-based incremental feature selection on fused decision tables. Dai et al. [31] proposed an attribute reduction algorithm based on fuzzy discernibility pairs, proving the attribute importance function constructed by this algorithm to be equivalent to SAT in [32]. However, these methods are generally applicable only in specific scenarios within datasets.

The aforementioned algorithms also face the issue of fuzzy information loss during the reduction. Although fuzzy information is employed in constructing fuzzy discernibility matrices, the loss of discernibility information persists because the resulting matrices remain binary. Furthermore, in algorithms that construct reductions based on satisfaction (SAT), the issue of information loss persists; the operation of selecting maximum discernibility leads to the underutilization of other discernibility information. Sowkuntla et al. [33] introduced a MapReduce-based acceleration algorithm for the fast reduction of large datasets. However, this method also fails to effectively leverage the fuzzy similarity information inherent in the data. Despite this, algorithms like this still achieve satisfactory performance since the labels in dataset provide sufficient information. These algorithms that require labels are generally referred to as supervised algorithms [34–36], in contrast to unsupervised attribute reduction algorithms.

Despite the effectiveness of supervised algorithms, the widespread use of smart devices across various industries has led to an explosion of redundant data without labels. This large-scale data is costly to label manually and raises the expense of extracting key information. Scenarios such as duplicate photos from burst mode, multiple copies of the same document across devices, repeated logs from monitoring systems, and redundant sensor data in smart homes or industrial environments can generate massive amounts of redundant, unlabeled data. Existing unsupervised attribute reduction algorithms can achieve satisfactory performance in these situations, such as Yuan et al. [17–19] proposed an unsupervised algorithm based on fuzzy rough sets, which updates attribute significance using the average dependency of all conditional attributes, and they also developed an unsupervised attribute reduction algorithm using fuzzy complementary entropy, Xu et al. [37] introduced a graph-based attribute reduction algorithm that considers the relationship between each attribute in each iteration. However, challenges such as the inability to parallelize still remain. Moreover, adapting existing discernibility matrix methods to unsupervised algorithms results in suboptimal performance due to the lack of label information. Therefore, there is a need for a new type of reduction method that not only facilitates accelerated computations on GPUs but also effectively utilizes fuzzy information within unlabeled data.

Based on the above discussions, we propose an unsupervised attribute reduction algorithm based on fuzzy discernibility matrices (UDMAR). This algorithm can effectively utilize the fuzzy information and accelerate computations on GPUs. Initially, we transform the traditional three-dimensional fuzzy discernibility matrix into a two-dimensional discernibility matrix M for GPU-accelerated computation. Secondly (as shown in Fig. 1), to fully utilize fuzzy discernibility information, we introduce the concept of fuzzy deletion and accordingly propose the definition of a fuzzy deletion function. The core idea is to select a column from the 2D discernibility matrix that meets certain criteria and use the discernibility information in this column to remove the corresponding information from the other columns, thereby updating the matrix. For example, in Fig. 1, we extract the column l_i with the highest sum from the 2D discernibility matrix and turn it into a fuzzy deletion operator Del_{l_i}. This operator is then applied to each column of M_2, gradually erasing the discernibility information to varying degrees. This cycle repeats until the information in M_2 drops below a certain threshold. To determine this threshold, we introduce an automatic stopping mechanism, which sets the stopping point by comparing the discernibility information in the reduced attribute set with that of the remaining attributes. The effectiveness of UDMAR is verified by five aspects: downstream task experiment, runtime experiment, ablation study, parameter sensitivity experiment, and hypothesis testing. The experimental results demonstrate that our algorithm maintains or even improves the accuracy of reduction results in downstream tasks, while significantly accelerating the process through matrix operations. The ablation study confirms the effectiveness of the fuzzy deletion function, and hypothesis testing further supports these conclusions.

Innovation and Contributions:

1. We present an attribute reduction framework designed for GPU compatibility, enabling efficient processing of large-scale datasets in machine learning and data analysis.
2. We introduce the concept of a fuzzy deletion function, enabling the algorithm to fully leverage the fuzzy information.
3. We propose a stopping mechanism that allows the algorithm to select the appropriate number of attributes, preventing it from stopping too early and missing important attributes, or too late and introducing redundancy.

In this paper, Section 2 provides an introduction to the fundamentals of fuzzy rough sets and fuzzy discernibility matrix. Section 3 introduces the concept of a fuzzy deletion function and the definition of ordered reduction. Sections 4 and 5 detail a specific fuzzy deletion function and the corresponding reduction algorithm. In Section 6, we present an example to demonstrate the computational process of the algorithm. Finally, in Section 7, we validate our method through experiments conducted across five aspects. Section 8 concludes the paper, discussing the conclusions derived from this study and outlining potential directions for future work.

2. Basic knowledge

2.1. Fuzzy rough set

A quadruple $FIS = (U, A, V, f)$ is a fuzzy decision information system, U is a universe which is non-empty and finite; $A = C \cup D$, where C is condition attribute set and D is decision attribute set; $FIS = (U, C, V, f)$ is a fuzzy information system if $D = \emptyset$. V is an union of attribute domain V_a, where V_a is the attribute domain of the attribute a; f as a mapping: $U \times A \to V$, satisfies $f(x, a) \in V_a$, where $\forall a \in A$ and $x \in U$.

Let $U = \{x_1, x_2, \ldots, x_n\}$. If $\mathcal{A} : U \to [0, 1]$, then $\mathcal{A}$ is a fuzzy set on U. $\forall x \in U$, $\mathcal{A}(x)$ is called the membership function of x. The fuzzy set $\mathcal{A}$ can be expressed as $\mathcal{A} = (\mathcal{A}(x_1), \mathcal{A}(x_2), \ldots, \mathcal{A}(x_n))$ or $\mathcal{A} = \sum_{i=1}^{n} \mathcal{A}(x_i)/x_i$. The set of all fuzzy sets on U is denoted as $\mathcal{F}(U)$.

Definition 1. [4,38,39] A fuzzy relation R on U is defined as
$R : U \times U \to [0, 1]$. $\forall (x, y) \in U \times U$, the membership degree $R(x, y)$ indicates the degree to which x and y have a relationship R. The set of all fuzzy relations on U is denoted as $\mathcal{F}(U \times U)$.

Suppose $R \in \mathcal{F}(U \times U)$, $\forall (x, y) \in U$, if it satisfies the following conditions

- Reflexivity $R(x, x) = 1$;
- Symmetry $R(x, y) = R(y, x)$;
- Transitivity $R(x, z) \geq \sup_{y \in U} \min\{R(x, y), R(y, z)\}$,

then R is called a fuzzy equivalence relation on U. Besides, if R only satisfies reflexivity and symmetry, then R is called a fuzzy similarity relation on U. $\forall R_1, R_2 \in \mathcal{F}(U \times U)$, we have

- $R_1(x, y) \leq R_2(x, y) \Rightarrow R_1 \subseteq R_2$;
- $(R_1 \cap R_2)(x, y) = \min\{R_1(x, y), R_2(x, y)\}$;
- $(R_1 \cup R_2)(x, y) = \max\{R_1(x, y), R_2(x, y)\}$.

Definition 2. [4,38,39] Let $FIS = (U, A, V, f)$ be a fuzzy information system, and R be a fuzzy similarity relation on U. For any $X \in \mathcal{F}(U)$, the lower approximation $\underline{R}X$ and upper approximation $\overline{R}X$ of X are a pair of fuzzy sets on U. The membership functions respectively are

$$\underline{R}X(x) = \inf_{y \in U} \max\{1 - R(x, y), X(y)\}, \tag{1}$$

$$\overline{R}X(x) = \sup_{y \in U} \min\{R(x, y), X(y)\}. \tag{2}$$

Let $U = \{x_1, \ldots, x_n\}$ and $C = \{c_1, \ldots, c_m\}$ be a conditional attribute set. $\forall B \subseteq C$, B can induce a fuzzy similarity relation $\mathcal{R}_B$ on U. It can be denoted by fuzzy relation matrix $S(\mathcal{R}_B) = (r_{ij}^B)_{n \times n}$, where $r_{ij}^B = \mathcal{R}_B(x_i, x_j)$, each row $(r_{i1}^B, r_{i2}^B, \ldots, r_{in}^B)$ denotes a fuzzy set. The fuzzy set induced by $\mathcal{R}_B$ is defined as

$$[x_i]_{\mathcal{R}_B} = \frac{r_{i1}^B}{x_1} + \frac{r_{i2}^B}{x_2} + \ldots + \frac{r_{in}^B}{x_n} = (r_{i1}^B, r_{i2}^B, \ldots, r_{in}^B). \tag{3}$$

The common methods for determining $\mathcal{R}_B(x_i, x_j)$ are outlined in [40]. In the fuzzy rough set-based method, the conjunction approach is typically used, calculated as $\mathcal{R}_B(x_i, x_j) = \wedge_{b \in B} \mathcal{R}_b(x_i, x_j)$.

2.2. Discernibility matrix reduction model based on rough sets

In this subsection, two existing discernibility matrix-based reduction methods are introduced.

Definition 3. [30] Given a fuzzy information system $FIS = (U, C \cup D, V, f)$, $\forall x, y \in U$, the discernbility matrix can be described as follows:

$$M(x,y) = \begin{cases} \{R \subseteq \mathbb{R} : 1 - R(x,y) > \underline{R}[x]_D(x)\}, & \text{if } y \notin [x]_D \\ \emptyset, & \text{otherwise} \end{cases}$$

Based on the discernibility matrix, the reduction can be obtained through the discernibility function f as below.

$$f(c_1, c_2, \ldots, c_m) = \wedge\{\vee M(x,y) | \forall x, y \in U, M(x,y) \neq \emptyset\}.$$

B is a reduction of C if B satisfies the following conditions:

- $\forall x, y \in U,\ M(x,y) \neq \varnothing \Rightarrow B \cap M(x,y) \neq \varnothing$
- $\forall B' \subset B,\ \exists x, y \in U, M(x,y) \neq \varnothing \Rightarrow B' \cap M(x,y) = \varnothing$

The above method can only generate a discernibility matrix with binary elements. In [32], a reduction method based on a discernibility matrix with fuzzy elements was proposed.

Definition 4. [32] Given a fuzzy information system $FIS = (U, C \cup D, V, f)$, $\forall x, y \in U$, the fuzzy discernbility matrix M' is presented as follows:

$$M'(x,y) = \{a_{N(\mu_a(x,y))} | a \in C\},$$

where μ_a describes the fuzzy indiscernibility between x and y with respect to attribute a.

Let $\mu_a = R_a(x,y)$, and $N(\mu_a(x,y)) = 1 - \mu_a(x,y)$. Then, $N(\mu_a(x,y))$ represents the degree of distinction between x and y under attribute a. For example, $M'(x,y) = \{a_{0.45}, b_{0.32}, c_{0.1}\}$. The fuzzy discernibility function is as follows:

$$f'_D(c_1, c_2, \ldots, c_m) = \wedge\{\vee M'(x,y) \leftarrow N(\mu_D(x,y)) \mid \forall x, y \in U\},$$

where "$\leftarrow$" represents the fuzzy implication, and D denotes the decision attributes.

The degree of satisfaction of an attribute subset B with respect to the decision attribute D is proposed as follows:

$$SAT_{B,D}(M'(x,y)) = \max_{a \in B}\{N(\mu_a(x,y))\} \leftarrow N(\mu_D(x,y)).$$

Thus, the satisfiability of B is defined as:

$$SAT(B) = \frac{\sum_{x,y \in U, x \neq y} SAT_{B,D}(M'(x,y))}{\sum_{x,y \in U, x \neq y} SAT_{C,D}(M'(x,y))}.$$

A conditional attribute subset B can then be regarded as a reduction if it satisfies the following conditions:

- $SAT(B) = SAT(C) = 1$
- $\forall B' \subset B, SAT(B') < SAT(B)$.

3. Unsupervised attribute reduction algorithm based on fuzzy discernibility matrix

In this section, a new reduction framework is proposed, which primarily achieves reduction through the continuous updating of the discernibility matrix using a fuzzy deletion function. Based on Definition 4, we present the following definition.

Definition 5. Let $FIS = (U, C, V, f)$ be a fuzzy information system and M is a corresponding fuzzy discernilibity matrix, $\forall x, y \in U$ and $\forall c \in C$, the unsupervised discernibility matrix-based object distinction sets (UMBO) of attribute c are defined as follows:

$$UMBO(c) = \{N(\mu_c(x,y)) | x, y \in U\}.$$

Based on Definition 5, we introduce the following definition to measure how effectively an attribute distinguishes objects or the extent to which it covers object pairs in the discernibility matrix.

Definition 6. Let $FIS = (U, C, V, f)$ be a fuzzy information system, and let M be the corresponding fuzzy discernibility matrix. For $\forall c \in C$, the evaluation function for an attribute's covering ability is defined as follows:"

$$Ucov(c,k) = \sum_{N(\mu_c) \in UMBO(c)} N^k(\mu_c),$$

where $k > 0$ is a hyperparameter. $Ucov(c, 1)$ represents the discernibility degree of attribute c, which can be denoted as $Ucov(c)$. More details about k can be found in Section 5.

In the traditional discernibility matrix updating process, the algorithm deletes attributes that provide the same discernibility ability as those already selected. We apply a fuzzy approach to this deletion process, enabling more fuzzy information to be fully utilized. For example, given $a \in UMBO(c_i)$ and $b \in UMBO(c_j)$, if b is used to delete a, the deletion is based on the proportion of information b contributes to a. Based on this, the definition of the fuzzy deletion function is as follows:

Definition 7. Given discernibility information $a \geq 0$ and $b \geq 0$, a function Del is called a deletion factor if it satisfies the following conditions:

- $\forall b \in (0, 1)$, $Del_b(a) < a$ and $Del_b(0) = 0$.
- $Del_0(a) = a$ or $Del_1(a) = 0$.
- $\forall b \in [0, 1]$, the derivative of $\mathrm{Del}_b(a)$ with respect to a exists, and satisfies $\frac{\mathrm{dDel}_b(a)}{\mathrm{d}a} \geq 0$.

Definition 7 outlines the basic criteria that a deletion operation should fulfill. The initial condition requires that when any non-zero information b is deleted from a, the value of a should decrease. If $a = 0$, deleting any information still results in 0. The second condition states that no information is deleted from a when $b = 0$, and all information is deleted from a when $b = 1$, i.e., $Del_b(a) = 0$. The last condition indicates that the deletion function is monotonically increasing with respect to a, meaning that for a given $b \in [0, 1]$, the more information a contains, the greater $Del_b(a)$ will be.

When a function satisfies the above conditions, we refer to it as a discernibility information deletion function. Its output represents the remaining amount of information in a after the information b has been deleted. For clarity, we provide the following stipulations:

1. $\forall b, c \in [0, 1]$, given x, let $B_{(\leq)} = \{b, c\}$ be an ordered set, then we can define:

$$Del_{B_{(\leq)}}(x) = Del_c(Del_b(x)).$$

2. Given the ordered sets $A_{(\leq)}$ and $B_{(\leq)}$, then the ordered set $\{A, B\}$ can be represented as $A_{(\leq)} \oplus B_{(\leq)}$.

Next, we present a theorem on the monotonicity of the Del function with respect to the discernibility information x, based on its properties and definition.

Theorem 1. *Given any ordered set $B_{(\leq)} = \{b_1, \ldots, b_k\}$, for any $x \in [0, 1]$, $Del_{B_{(\leq)}}(x) \leq x$.*

Proof. Let $x_i = Del_{\{b_1, b_2, \ldots, b_i\}}(x)$ with $x_0 = Del_0(x) = x$. Then, we can derive:

$$Del_{\{b_1, b_2, \ldots, b_i, b_{i+1}\}} = Del_{b_{i+1}}(Del_{\{b_1, b_2, \ldots, b_i\}}(x)) = Del_{b_{i+1}}(x_i).$$

Since $Del_{b_i}(x) < x$, it follows that $x_i < x_{i-1}$, which leads to:

$$x_k < x_{k-1} < \ldots < x_0 = x.$$

If $\exists b_i = 0$ for $i = 1, 2, \ldots, k$, then $x_i = x_{i-1}$, finally we obtain $Del_{\{b_1, b_2, \ldots, b_k\}}(x) \leq x$. □

Through Theorem 1, it can be deduced that the monotonicity of Del with respect to x still holds when the information deletion variable is an ordered set. Building on this, Theorem 2 presents the properties of the Del function with respect to ordered sets.

Theorem 2. *Given any ordered set $A_{(\leq)} = \{a_1, \ldots, a_k\}$, $B_{(\leq)} = A_{(\leq)} \oplus C_{(\leq)}$ where $C_{(\leq)} \in R^{1 \times n}$, then $Del_{B_{(\leq)}}(x) \leq Del_{A_{(\leq)}}(x)$.*

Proof. Since $B_{(\leq)} = A_{(\leq)} \oplus C_{(\leq)}$, we have $Del_{B_{(\leq)}}(x) = Del_{C_{(\leq)}}(Del_{A_{(\leq)}}(x))$. Based on Theorem 1, it follows that:

$$Del_{B_{(\leq)}}(x) = Del_{C_{(\leq)}}(Del_{A_{(\leq)}}(x)) \leq Del_{A_{(\leq)}}(x).$$

Finally, we conclude that $Del_{B_{(\leq)}}(x) \leq Del_{A_{(\leq)}}(x)$. □

The discernibility matrix provides essential information for reduction. However, the traditional discernibility matrix, often viewed as a three-dimensional tensor, is computationally inefficient due to its nested matrix and set structures. We reformulate the matrix into a representation more suitable for matrix and parallel computing.

Definition 8. [32] Given an information system $FIS = (U, C, V, f)$, if its fuzzy discernibility matrix M_{dis} is denoted as

$$M_{dis}(x, y) = \{a_{N(\mu_a(x,y))} | a \in C\},$$

then its two-dimensional form M is as follows:

$$M((x_i, y_j), a) = N(\mu_a(x_i, y_j)),$$

which can be specifically represented as follows,

$$\begin{array}{c} \\ (x_1,y_1) \\ (x_1,y_2) \\ \vdots \\ (x_n,y_n) \end{array} \begin{array}{c} \begin{array}{cccc} a_1 & & \cdots & a_m \end{array} \\ \begin{pmatrix} N(\mu_{a_1}(x_1,y_1)) & \cdots & N(\mu_{a_m}(x_1,y_1)) \\ N(\mu_{a_1}(x_1,y_2)) & \cdots & N(\mu_{a_m}(x_1,y_2)) \\ \vdots & \ddots & \vdots \\ N(\mu_{a_1}(x_n,y_n)) & \cdots & N(\mu_{a_m}(x_n,y_n)) \end{pmatrix} \end{array}.$$

Based on the above form, the two-dimensional discernibility matrix can be derived from either an upper triangular or lower triangular discernibility matrix to obtain a matrix with a smaller size. Typically, $M \in \mathbb{R}^{\frac{1}{2}(\sum_{x,y\in U} 1), |A|}$. For example, consider the fuzzy discernibility matrix M_{dis} given as:

$$M_{dis} = \begin{pmatrix} \{0,0,0,0\} & \{0,1,1,0\} & \{1,0.1,0.3,0.2\} & \{0.1,0,0,0\} \\ \{0,1,1,0\} & \{0,0,0,0\} & \{0,0,0,0\} & \{0,1,1,1\} \\ \{1,0.1,0.3,0.2\} & \{0,0,0,0\} & \{0,0,0,0\} & \{1,0,0,1\} \\ \{0.1,0,0,0\} & \{0,1,1,1\} & \{1,0,0,1\} & \{0,0,0,0\} \end{pmatrix},$$

its two-dimensional discernibility matrix M is:

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0.1 & 0.3 & 0.2 \\ 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Based on the above definition, a two-dimensional discernibility matrix M can be represented as a group of vectors $M = [\vec{a_1}, \vec{a_2}, \ldots, \vec{a_m}]$. Then, for a vector $x = [x_1, x_2, \ldots, x_m]$, we can define the action of a deletion factor on the vector as follows.

Definition 9. Given a vector $\vec{x} = [x_1, x_2, \ldots, x_m]$, for any $a \in [0, 1]$,

$$Del_{\vec{x}}(a) = [Del_{x_1}(a), Del_{x_2}(a), Del_{x_3}(a), \ldots, Del_{x_m}(a)].$$

If $\vec{a}$ is a vector, then

$$Del_{\vec{x}}(\vec{a}) = [Del_{x_1}(a_1), Del_{x_2}(a_2), Del_{x_3}(a_3), \ldots, Del_{x_m}(a_m)].$$

Let $M = [\vec{a_1}, \ldots \vec{a_m}]$, we have

$$Del_{\vec{x}}(M) = [Del_{\vec{x}}(\vec{a_1}), \ldots, Del_{\vec{x}}(\vec{a_m})].$$

When the variables change from scalar a and scalar b to vector $\vec{a}$ and vector $\vec{b}$, we accordingly present some properties of the fuzzy deletion function regarding vectors $\vec{a}$ and $\vec{b}$ below, as an extension of Definition 7 and Theorem 1.

Property 1. *Given a vector $\vec{b} = [b_1, \ldots, b_m]$ as the information deletion variable. If the Del function satisfies the Definition 7, then the following properties hold.*

- *Given a vector $\vec{a} = [a_1, \ldots, a_m]$, we have $|Del_{\vec{b}}(\vec{a})| < |\vec{a}|$.*
- *Given two vectors $\vec{a_1}$ and $\vec{a_2}$, if $a_{1i} \le a_{2i}$ for $\forall i \le m$, then $|Del_{\vec{b}}(\vec{a_1})| \le |Del_{\vec{b}}(\vec{a_2})|$.*

It should be noted that for a vector $\vec{x}$, $|\vec{x}| = \sum_{x_i \in \vec{x}} x_i$.

Proof. Since $Del_{\vec{b}}(\vec{a}) = [Del_{b_1}(a_1), Del_{b_2}(a_2), \ldots, Del_{b_m}(a_m)]$, for every $Del_{b_i}(a_i)$ we have $Del_{b_i}(a_i) < a_i$, thus we can finally get $|Del_{\vec{b}}(\vec{a_1})| \le |Del_{\vec{b}}(\vec{a_2})|$.

Since $\frac{\mathrm{d}Del_b(x)}{\mathrm{dx}} \ge 0$, the *Del* function is a non-increasing function. Thus for $a_{1i} \le a_{2i}$, we obtain $Del_{b_i}(a_{1i}) \le Del_{b_i}(a_{2i})$. Then, we have $|Del_{\vec{b}}(\vec{a_1})| \le |Del_{\vec{b}}(\vec{a_2})|$. □

Theorem 2 describes the monotonicity of the fuzzy deletion function with respect to the scalar b. Based on Theorem 2, the monotonicity theorem of the fuzzy deletion function with respect to the vector $\vec{b}$ is given as follows.

Theorem 3. *Let $A_{\preceq} = \{\vec{a}_1, \vec{a}_1, \ldots, \vec{a}_t\}$ and $C_{\preceq} = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_k\}$ be ordered sets, $t, k \in R$, $B_{\preceq} = A_{\preceq} \oplus C_{\preceq}$. For $\vec{x} \in \mathbb{R}^{1\times m}$, let $a = Del_{A_{\preceq}}(\vec{x})$, $b = Del_{B_{\preceq}}(\vec{x})$, then $Ucov(b,k) \le Ucov(a,k)$.*

Proof. Let a_{ij} represent the j-th element of the i-th vector. Since $Del_{\vec{a}_i}(\vec{x}) = [Del_{a_{i1}}(x_1), Del_{a_{i2}}(x_2), \ldots, Del_{a_{im}}(x_m)]$ and $Del_{\vec{c}_i}(\vec{x}) = [Del_{c_{i1}}(x_1), Del_{c_{i2}}(x_2), \ldots, Del_{c_{im}}(x_m)]$, based on Theorem 1, it follows that the ordered sets $A^i_{\preceq}$ and $B^i_{\preceq}$, formed by each component of the vector, satisfy $Del_{A^i_{\preceq}}(x_i) \le Del_{B^i_{\preceq}}(x_i)$. Therefore $|Del_{A_{\preceq}}(\vec{x})| \le |Del_{B_{\preceq}}(\vec{x})|$. For any $k \in R$, we have $|Del_{A_{\preceq}}(\vec{x})^k| \le |Del_{B_{\preceq}}(\vec{x})^k|$. Thus, $Ucov(b) \le Ucov(a)$. □

Based on Property 1, we can obtain the operational properties of the Del function with respect to the matrix $M = [a_1, a_2, \ldots, a_m]$ as follows:

- Given a vector $\vec{b}$, we have $|Del_{\vec{b}}(M)| < |M|$.
- Given a vector $\vec{b}$, if $|M_1| \le |M_2|$, it follows that $|Del_{\vec{b}}(M_1)| \le |Del_{\vec{b}}(M_2)|$.

Based on Theorem 3, we can derive the monotonicity theorem of the Del function when the variable M is related to the deletion of information variables.

Theorem 4. *Let $A_{\preceq} = \{\vec{a}_1, \vec{a}_1, \ldots, \vec{a}_t\}$ and $C_{\preceq} = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_k\}$ be ordered sets, $t, k \in R$, $B_{\preceq} = A_{\preceq} \oplus C_{\preceq}$. For $M = [\vec{x}_1, \ldots, \vec{x}_m]$, we have $|Del_{B_{\preceq}}(M)| \le |Del_{A_{\preceq}}(M)|$.*

Proof. It can be proofed by Theorem 3. □

Based on Theorem 4, we propose the following definition of reduction for the two-dimensional fuzzy identification matrix M.

Definition 10. For the information system $FIS = (U, C, V, f)$, its two-dimensional fuzzy discernibility matrix is denoted by M. Given sets B and R such that $B \cup R = C$, for any ordered set $R_{\preceq}$, let $C_{\preceq} = B_{\preceq} \oplus R_{\preceq}$. The ordered set $B_{\preceq}$ is a reduction of C if it satisfies the following conditions:

- $|Del_{B_{\preceq}}(M)| = |Del_{C_{\preceq}}(M)|$.
- $\forall B'$ s.t. $B_{\preceq} - B'_{\preceq} \neq \emptyset$, $|Del_{B_{\preceq}}(M)| < |Del_{B'_{\preceq}}(M)|$.

4. Specific model

Based on the above methods, we propose a Del function model to serve as the deletion function for attribute reduction.

Theorem 5. *Given a deletion variable b, for $\forall x \in [0,1]$, a Del function can be defined as follows:*

$$Del_b(x) = (1-b)\cdot x. \tag{4}$$

Proof. Based on $Del_b(x) = (1-b)\cdot x$, for $\forall a > 0$, we have $Del_b(a) < a$ and $Del_b(0) = 0$ with $b \in (0,1)$. It follows that $Del_0(x) = x$ and $Del_1(x) = 0$.

For any $b \in [0,1]$ and $x_0 \in [0,1]$, the derivative of $Del_b(x_0)$ can be expressed as follows:

$$\frac{dDel_b(x_0)}{dx_0} = \lim_{\delta x \to 0} \frac{Del_b(x_0 + \Delta x) - Del_b(x_0)}{\Delta x} = 1 - b.$$

Since $\lim_{\delta x\to 0} \frac{Del_b(x_0+\Delta x) - Del_b(x_0)}{\Delta x}$ exist and $1 - b \ge 0$, we can deduce that Del_b satisfies Definition 7. □

In the case where the discernibility matrix is a binary matrix, we can observe that the update results based on the two-dimensional discernibility matrix are equivalent to the update results of the traditional discernibility matrix. For instance, given an information system $(U, \{b_1, b_2, b_3, b_4\}, V, f)$, the discernibility matrix generated by Definition 3 is denoted as

$$M_{dis} = \begin{pmatrix} \{0,0,0,0\} & \{0,1,1,0\} & \{1,0,1,0\} & \{0,0,1,0\} \\ \{0,1,1,0\} & \{0,0,0,0\} & \{0,0,0,0\} & \{0,1,1,1\} \\ \{1,0,1,0\} & \{0,0,0,0\} & \{0,0,0,0\} & \{1,0,0,1\} \\ \{0,0,1,0\} & \{0,1,1,1\} & \{1,0,0,1\} & \{0,0,0,0\} \end{pmatrix}.$$

$$M' = \begin{pmatrix} 0 & 0.2 & 0 & 0.1 \\ 0.1 & 0.3 & 0 & 0.1 \\ 0.1 & 0.3 & 0.9 & 0 \\ 0 & 0.3 & 0.7 & 0 \\ 0.2 & 0.3 & 0 & 0.1 \\ 0.1 & 0.2 & 0 & 0.1 \end{pmatrix}$$

Fig. 2. Discernibility Matrix.

We then transform it into a two-dimensional discernibility matrix

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

For any column of M, constructing a fuzzy operator and applying it to M yields results equivalent to updating M_{dis} by selecting the corresponding attribute.

Therefore, based on the above discussion, the behavior of the fuzzy deletion function proposed in this paper is entirely equivalent to the behavior of updating the traditional discernibility matrix.

5. Reasoning cut

In attribute reduction algorithms, the importance of attributes typically decreases as the algorithm progresses. Generally, the algorithm stops when the attribute significance reaches zero. If an earlier stop is desired, a threshold needs to be set, halting the algorithm when the importance falls below this threshold. A suitable stopping threshold can remove more redundant attributes while retaining high-quality attributes. In this section, we propose a method to determine the stopping threshold based on the updated state of the discernibility matrix. The relevant definitions are as follows:

Definition 11. Given a 2D discernibility matrix $M = (b_i^j)$ and a deletion function Del, let the reduction pool be denoted by B. The stopping condition for this iteration is then defined as $E(Del_{B_{\leq}}(B))$, where E is a stopping condition function (cut function).

Based on the above definition, the max cut function is given as follows:

Let $B_{\leq} = \{\vec{b}^1, \ldots, \vec{b}^m\}$. Then we have:

$$E(Del_{B_{\leq}}(B)) = \max_{b^j \in B} |Del^k_{B_{\leq}}(\vec{b}^j)|,$$

where k is a manually set hyperparameter.

According to the above definition, the algorithm stops when the attribute significance $Ucov(c,k)$ is less than the threshold $E(Del_{B_{\leq}}(B))$. The parameter k, as a hyperparameter, plays a role in adjusting the value of $E(Del)$. In practical situations, the algorithm may encounter the following scenario when it stops.

For example, given the information system (U, C, V, f), $C = \{c_1, c_2, c_3, c_4\}$, c_1 and c_2 have already been selected into the reduction set, the corresponding updated discernibility matrix M' is shown as Fig. 2,

where the two columns corresponding to the attributes c_1 and c_2 are surrounded by red frames.

We can determine the threshold value using the max cut, which is the larger value of $Ucov(M'(:,1),3)$ and $Ucov(M'(:,2),3)$. Since

$$Ucov(M'(:,1),3) = 0.0110, Ucov(M'(:,2),3) = 0.1240,$$

we obtain

$$E(Del_{B_{\leq}}(B)) = \max\{Ucov(M'(:,1),3), \quad Ucov(M'(:,2),3)\} = 0.1240.$$

Since

$$Ucov(M'(:,3),3) > Ucov(M'(:,4),3)$$

and

$$Ucov(M'(:,3),3) = 0^3 + 0^3 + 0.9^3 + 0.7^3 + 0^3 + 0^3 = 1.0720,$$

we find that

$$Ucov(M'(:,3),3) > E(Del_{B_{\leq}}(B)).$$

Thus, the stop condition is not met. However, if $k = 1$, we have

$$Ucov(M'(:,2),1) = 1.6, Ucov(M'(:,3),1) = 1.6,$$

the stop condition is satisfied. However, c_3 still has good discrimination ability, which leads to the fact that c_3 cannot be selected into the reduction set when $k = 1$. In fact, k can be considered an evaluation parameter for discrimination ability. By adjusting k, the originally small discernibility values become even smaller when raised to the power of k. For attributes with different levels of discrimination quality, a higher value of k results in a larger gap between their $Ucov$ values.

Based on the above discussion, we ultimately derive Algorithm 1.

Algorithm 1: UDMAR.

Input: $FIS = (U, C, V, f)$, λ, $Del_b(x) = (1-b)x$, k, $|C| = m$, $|U| = n$
Output: Reduced attribute set R

$Red \leftarrow \emptyset$, $Label$;
for $h \leftarrow 1$ **to** m **do**
  Calculate the similarity matrix Sim_{c_h} using a suitable method;
end
Convert $Sim_{c_1}, Sim_{c_2}, \cdots, Sim_{c_m}$ into a tensor $M_{sim}^{m\times n\times n}$, followed by $M_{dis} = \vec{1} - M_{sim}$;
Convert M_{dis} to a 2-dimensional matrix M using Definition 8;
while $Label$ **do**
  $\{Ucov(c_1), \ldots, Ucov(c_m)\} = \vec{1}_{1\times n} \cdot M$;
  $c_i' = \arg\max_{c_i \in C}\{Ucov(c_i)\}$;
  $M = Del_{UMBO(c_i')}(M)$;
  if $E(Del_{Red}^k(Red)) \geq Ucov(c_i')$ **then**
    $Label = 0$;
  else
    $Red = Red \oplus c_i'$;
  end
end
Return Red

6. Example

In this section, the previously discussed method is applied to compute the reduction of Table 1.

We employ the min-max method for data normalization [41]. Additionally, we use the following formula to fuzzify the data [19].

$$r_{ij}^{c_k} = \begin{cases} 1, f(x_i,c_k) = f(x_j,c_k) \text{ and } c_k \text{ is nominal}; \\ 0, f(x_i,c_k) \neq f(x_j,c_k) \text{ and } c_k \text{ is nominal}; \\ 0, |f(x_i,c_k) - f(x_j,c_k)| > \varepsilon_k \text{ and } c_k \text{ is numerical}; \\ 1 - |f(x_i,c_k) - f(x_j,c_k)|, |f(x_i,c_k) - f(x_j,c_k)| \leq \varepsilon_k \text{ and } c_k \text{ is numerical}. \end{cases} \tag{5}$$

From Eq. (5), a fuzzy similarity relation matrix $M_{R_{c_k}}$ can be derived. r_{ij} represents the fuzzy similarity relationship between attributes, and ε_{c_k} is a conditionally fuzzy similarity radius, with the specific calculation formula as follows:

$$\varepsilon_{c_k} = \frac{std(c_k)}{\lambda}, \tag{6}$$

where $std(c_k)$ is the standard deviation of attribute c_k, and the preset parameter λ is used to adjust the fuzzy similarity radius.

First of all, given a parameter $\lambda = 1$, by the core function introduced by [18], we can obtain the similarity matrices Sim_{c_i} for each c_i. Based on step 5, a three-dimensional tensor M_{sim} is expressed as follows:

$$M_{sim}(1) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, M_{sim}(2) = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$M_{sim}(3)=\begin{pmatrix}1.0000 & 0.9677 & 0.9032 & 0 & 1.0000\\ 0.9677 & 1.0000 & 0.9355 & 0 & 0.9677\\ 0.9032 & 0.9355 & 1.0000 & 0 & 0.9032\\ 0 & 0 & 0 & 1.0000 & 0\\ 1.0000 & 0.9677 & 0.9032 & 0 & 1.0000\end{pmatrix},$$

$$M_{sim}(4)=\begin{pmatrix}1.0000 & 0.9474 & 0.8947 & 0 & 1.0000\\ 0.9474 & 1.0000 & 0.9474 & 0 & 0.9474\\ 0.8947 & 0.9474 & 1.0000 & 0 & 0.8947\\ 0 & 0 & 0 & 1.0000 & 0\\ 1.0000 & 0.9474 & 0.8947 & 0 & 1.0000\end{pmatrix},$$

$$M_{sim}(5)=\begin{pmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}.$$

Then, by step 6, $M_{dis}=\vec{1}-M_{sim}$ is transformed into a 2D matrix $M^{(k\times m)}$ as shown below, where $k=\frac{n(1+n)}{2}$.

$$M^{(k\times m)}=\begin{pmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0323 & 0.0526 & 1.0000\\ 0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0968 & 0.1053 & 1.0000\\ 0 & 0 & 0.0645 & 0.0526 & 0\\ 0 & 0 & 0 & 0 & 0\\ 1.0000 & 0 & 1.0000 & 1.0000 & 1.0000\\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0\\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0\\ 0 & 0 & 0 & 0 & 0\\ 1.0000 & 0 & 0 & 0 & 1.0000\\ 1.0000 & 1.0000 & 0.0323 & 0.0526 & 1.0000\\ 1.0000 & 1.0000 & 0.0968 & 0.1053 & 1.0000\\ 1.0000 & 0 & 1.0000 & 1.0000 & 1.0000\\ 0 & 0 & 0 & 0 & 0\end{pmatrix}.$$

By step 8, we can calculate

$$\{Ucov(c_1),Ucov(c_2),\ldots,Ucov(c_m)\}=\vec{1}^{(1\times k)}\cdot M$$

$$=\begin{bmatrix}7.0000 & 6.0000 & 4.3226 & 4.3684 & 7.0000\end{bmatrix}.$$

According to step 9 in Algorithm 1, the c_1 have maximum value $\max\limits_{c_i\in C-Red}(Ucov(c_i))=7$.

Then, update M by step 10. Based on Definition 9 and Eq. (4), we apply $Del_{c_1}(\vec{x})=(\vec{1}-M(:,c_1))\cdot\vec{x}$ to each column of M, where $\vec{x}$ is the column vector of M. i.e., we multiply $1-M(:,c_1)$ with each column of M as follows (The symbol $.*$ indicates element-wise multiplication of the column vector with each column of M.):

$$M=UMBO(c_1).*M=\begin{pmatrix}1\\1\\1\\1\\1\\1\\0\\0\\0\\1\\0\\0\\0\\0\\1\end{pmatrix}.*\begin{pmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0323 & 0.0526 & 1.0000\\ 0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0968 & 0.1053 & 1.0000\\ 0 & 0 & 0.0645 & 0.0526 & 0\\ 0 & 0 & 0 & 0 & 0\\ 1.0000 & 0 & 1.0000 & 1.0000 & 1.0000\\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0\\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0\\ 0 & 0 & 0 & 0 & 0\\ 1.0000 & 0 & 0 & 0 & 1.0000\\ 1.0000 & 1.0000 & 0.0323 & 0.0526 & 1.0000\\ 1.0000 & 1.0000 & 0.0968 & 0.1053 & 1.0000\\ 1.0000 & 0 & 1.0000 & 1.0000 & 1.0000\\ 0 & 0 & 0 & 0 & 0\end{pmatrix}=\begin{pmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0323 & 0.0526 & 1.0000\\ 0 & 0 & 0 & 0 & 0\\ 0 & 1.0000 & 0.0968 & 0.1053 & 1.0000\\ 0 & 0 & 0.0645 & 0.0526 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ & & \cdots & & \\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{pmatrix}.$$

Table 1
Mixed data table.

U	c_1	c_2	c_3	c_4	c_5
x_1	A	2	36.1	96.9	2
x_2	A	1	36.2	97.2	1
x_3	A	1	36.4	97.5	1
x_4	B	2	39.2	102.6	1
x_5	C	2	36.1	96.9	5

Based on step 11, Max Cut is used to calculate $E(Del_{Red}(Red)) = \max_{c_i \in Red} Ucov(c_i)$. Since $Red = \varnothing$, we set $E(Del_{Red}(M(:, Red))) = 0$. Thus, we have $E(Del_{Red}(M(:, Red))) < 7$. By step 14, we update Red to $Red \oplus c_1 = \{c_1\}$.

Back to step 7, since M has been updated, we have

$$\{Ucov(c_1), \ldots, Ucov(c_m)\} = \{0, 2.0000, 0.1935, 0.2105, 2.0000\}.$$

Thus, c_2 has the maximum value, with $\max_{c_i \in C-Red}(Ucov(c_i)) = 2$. After updating M, we have:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0604 & 0.0492 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ & & \cdots & & \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Since $E(Del_{Red}(Red)) = \max_{c_i \in Red} Ucov(c_i)$ remains $0 < 2$, we update Red to $Red \oplus c_2 = \{c_1, c_2\}$.

Then, going to the next iteration, we can get $Red = \{c_1, c_2, c_3\}$.

Finally, in the last iteration, since $E(Del_{Red}(Red)) = \max_{c_i \in Red} Ucov(c_i) = 0.0604$, $\max_{c_i \in C-Red}(Ucov(c_i)) = Ucov(c_4) = 0.0492$. Therefore, $E(Del_{Red}(Red)) > Ucov(c_4)$, the algorithm stops at this point.

The reduction of Table 1 is $Red = \{c_1, c_2, c_3\}$.

7. Experiment

In this section, we conduct a comprehensive evaluation across five aspects: downstream tasks (classification and clustering), parameter sensitivity, runtime analysis, and hypothesis testing. Before presenting the results, we provide an overview of the datasets and benchmark algorithms used in our experiments. The code can be found at: https://github.com/Doctorwngd/UDMAR/.

7.1. Experimental preparation

We use 26 datasets collected from the UCI machine learning repository, encompassing nominal, numerical, and mixed data [42]. Detailed descriptions are provided in Table 2. For handling missing entries, we employed the maximal probability value method, substituting missing values with the most frequently occurring value for each attribute.

To compare our method with existing techniques, we utilized six SOTA methods as benchmark algorithms.

- Baseline: The results obtained from all original attribute sets are utilized as the baseline.
- USFSM [43]: This algorithm determines feature correlation by integrating kernels with advanced spectral-based feature evaluation measures, resulting in a feature ranking ordered by their significance.
- UFRFS [44]: This method uses the fuzzy positive domain to designate the significance of attributes, ultimately generating a feature subset.
- USQR [45]: The USQR method is a quick reduct based on the conventional rough positive domain, typically resulting in a reduced subset.
- UEBR [46]: This method employs conditional entropy to determine attribute significance, ultimately producing a feature subset.
- FRUAR [19]: Similar to UFRFS, this algorithm uses the positive domain to define attribute significance and applies a similarity kernel function capable of processing mixed data.
- HKCMI [18]: The HKCMI approach utilizes a novel kernel function to define inter-element similarity and uses complementary entropy to establish attribute significance.
- UDMAR: UDMAR is the unsupervised attribute reduction algorithm based on fuzzy discernibility matrix introduced in this paper.

Table 2
Basic descriptive statistics of the 26 adopted data sets.

ID	Data set	Object	Conditional attribute	Decision class	Type
1	abalone	4177	8	29	Mixed
2	autos	205	25	6	Mixed
3	bands	531	39	2	Mixed
4	cleveland	303	13	5	Mixed
5	Credit approval	690	15	2	Mixed
6	diabetes	768	8	2	Numeric
7	ecoli	336	7	8	Numeric
8	flag	194	28	8	Mixed
9	german	1000	20	2	Mixed
10	glass	214	10	7	Numeric
11	heart disease	270	13	2	Mixed
12	hepatitis	155	19	2	Mixed
13	hypothyroid	3163	25	2	Mixed
14	labor	57	16	2	Mixed
15	lymphography	148	18	4	Nominal
16	monks	432	6	2	Nominal
17	nursery	12960	8	5	Nominal
18	parkinsons	768	22	2	Numeric
19	sick	3772	29	2	Mixed
20	wbc	699	9	2	Numeric
21	wdbc	569	31	2	Numeric
22	wpbc	198	33	2	Numeric
23	sonar	208	61	2	Mixed
24	SCADI	70	206	2	Mixed
25	Musk	476	167	2	Numeric
26	Tiger	1220	231	2	Numeric

7.2. Experiment settings

For the classification experiments, we use three algorithms: Classification and Regression Trees (CART), Naive Bayes (NB), and k-Nearest Neighbor (kNN). These algorithms are employed to rigorously evaluate UMDAR, executed in MATLAB 2021a. We use ten-fold cross-validation for all algorithms to ensure accuracy. The "DistributionNames" parameter for the NB algorithm is set to "kernel". We repeat each experiment 10 times, and the average and standard deviation of the classification accuracy are calculated as the final results.

For the clustering experiments, k-Means is employed to evaluate the performance of these attribute reduction methods. The number of clusters (k) is set according to the true classes present in the data. The clustering algorithm produces the predicted classifications for the dataset, and clustering accuracy (ACC) is used as the metric to assess clustering performance [47]. A higher ACC score indicates better clustering performance. Each clustering experiment is repeated ten times, with the final results presented as the average and standard deviation from these ten runs.

In the ablation experiment, we compare our method with traditional discernibility matrix methods. Since constructing a discernibility matrix without labels cannot directly apply a method like Definition 3, we generate different binary discernibility matrices by thresholding the fuzzy discernibility matrix at various levels. We compare then UDMAR with all generated binary discernibility matrices, as shown in Fig. 3. The threshold range used is [0:0.1:1].

Table 3 presents the average number of reductions across four downstream tasks, derived from the optimal feature sets of each attribute reduction algorithm, demonstrating their ability to eliminate redundancy. In Table 3, since the USFSM algorithm is an attribute ranking algorithm, it is excluded from the comparison. An underscore indicates the minimum number of attributes in the reduction set. UDMAR achieves the smallest number of reductions on most datasets and has the lowest average number of reductions overall.

In the runtime analysis experiments, we examined two aspects: the comparison of execution times and the variation in time as the data size increases for each attribute reduction algorithm. We set the parameter λ to 1 for FRUAR, HKCMI, and UDMAR as defined in Eq. (5).

For methods requiring discretization, FCM was adopted due to its effectiveness, categorizing numeric data into four classes [47,48]. The fuzzification function is given by Eq. (5), and the range of λ in Eq. (6) is [0.1:0.1:3]. For the SCADI, Musk, and Tiger datasets, k is set to 5, and for the others, k is set to 1. We then compared the results with original reduction subsets from other algorithms, highlighting our algorithm's superior ability to eliminate redundancy and select the most informative features.

7.3. Classification and clustering experiments

In this section, classification downstream tasks CART, NB, and KNN, along with the clustering downstream task k-Means, are employed to evaluate the performance of the above attribute reduction methods. The experimental results are presented in Tables 4, 5, 6 and 7 respectively.

Table 3
Average number of selected attributes across four downstream tasks.

Date set	Baseline	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR
abalone	8	8	7	8	7	7	5.5
autos	25	17	16	13	15	22.5	14.75
bands	39	19	7	5	8	30.5	6.75
cleveland	13	13	11	9	10	12	9.75
creditA	15	13	14	13	12.5	14	9.5
diabetes	8	8	7	8	7	7	7
ecoli	7	7	5	6	6	6	4.75
flag	28	20	15	13	16.5	22	14.25
german	20	19	16	11	13.5	19	12.25
glass	10	8	9	8	9	9	3
heart	13	13	12	10	12	10	10
hepatitis	19	18	12	13	16.25	18	9.5
hypothyroid	25	23	23	21	22.75	24	6.25
sonar	16	8	9	9	43.25	59	13.5
labor	16	13	9	9	10.5	15	10.25
lymphography	18	11	10	10	10	10	9
monks	6	6	5	6	5	5	6
nursery	8	8	8	7	7	7	8
parkinsons	22	8	11	10	11.75	21	7
SCADI	206	17	13	13	15	118	11.5
sick	29	27	18	18	23	27	8
wbc	9	9	8	9	8	8	6.25
wdbc	31	9	13	15	28	30	8.75
wpbc	33	8	9	7	14.75	33	9.5
Musk	167	19	19	13	20.5	165	19.5
Tiger	231	51	10	14	41	108	26.75
Average	39.31	14.62	11.38	10.69	15.13	31.04	9.89

Table 4
The classification results of the reduced data on CART (%).

Date set	Baseline	USFSM	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR
abalone	20.99±0.51	20.85±0.42	20.43±0.68	20.82±0.72	20.96±0.42	21.15±0.71	21.46±0.48	**21.86±0.51**
autos	73.38±3.36	**75.98±2.56**	68.11±3.26	75.35±2.24	73.49±1.20	74.22±3.14	75.24±1.80	75.72±1.45
bands	74.32±2.34	71.53±1.83	72.41±1.12	69.55±1.64	70.49±1.56	73.63±1.46	**75.34±1.51**	74.07±1.97
cleveland	51.42±1.78	51.25±1.95	51.00±1.18	49.51±1.58	50.04±1.49	52.15±1.59	51.47±2.37	**53.25±1.93**
creditA	81.20±1.51	**84.52±0.87**	81.71±0.90	81.36±0.87	81.41±1.08	81.87±0.86	81.97±1.21	83.09±0.54
diabetes	70.51±1.07	70.85±0.93	70.10±1.27	71.16±1.82	69.97±0.70	71.28±1.22	70.92±1.15	**71.36±0.93**
ecoli	81.39±0.88	80.18±1.30	80.97±1.21	80.87±1.45	81.38±0.74	81.57±1.60	81.76±1.23	**82.07±1.07**
flag	47.32±3.88	51.71±3.07	45.35±2.61	48.66±1.65	49.96±2.20	**51.73±2.31**	48.37±2.90	50.32±2.84
german	69.38±1.03	70.30±1.01	67.05±1.43	69.46±1.11	69.68±0.90	70.36±0.96	69.78±0.74	**70.84±1.01**
glass	98.69±0.29	98.60±0.30	98.65±0.41	98.56±0.46	97.94±0.57	98.78±0.24	98.88±0.33	**98.89±0.32**
heart	77.04±1.62	**78.93±1.53**	75.93±1.86	74.81±1.63	75.63±2.31	77.70±1.64	77.26±1.67	78.37±2.07
hepatitis	56.07±2.17	60.68±2.65	59.07±2.56	58.48±2.54	59.60±3.16	59.84±3.50	**59.37±3.14**	58.50±4.71
hypothyroid	98.89±0.09	**98.96±0.08**	98.89±0.09	98.85±0.09	98.92±0.07	98.92±0.06	**98.96±0.08**	97.34±0.14
sonar	71.89±1.90	69.28±1.69	52.48±2.90	68.52±3.12	67.13±2.82	74.26±2.42	72.95±2.39	**75.85±1.57**
labor	82.67±3.09	88.33±2.39	84.63±1.64	80.87±2.92	78.93±3.49	**91.13±1.87**	84.40±3.34	83.83±3.15
lymphography	76.80±2.46	78.84±1.69	75.49±2.90	**78.35±1.66**	76.23±3.31	78.14±1.39	76.39±3.22	76.79±2.00
monks	100.00±0.00	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
nursery	98.75±0.04	95.17±0.05	98.78±0.09	42.10±0.35	98.77±0.07	42.04±0.25	93.04±0.06	**98.81±0.06**
parkinsons	86.62±1.67	86.09±2.28	86.00±1.50	84.42±2.28	87.84±1.70	88.82±1.79	87.54±1.55	**88.24±1.55**
SCADI	77.14±1.78	71.14±2.68	70.57±1.81	75.43±1.76	77.71±1.20	80.71±1.81	78.86±1.31	**84.57±1.31**
sick	98.78±0.11	98.83±0.11	98.77±0.11	98.87±0.08	98.89±0.13	98.84±0.11	**98.97±0.13**	98.36±0.16
wbc	94.08±0.44	94.21±0.51	93.95±0.40	93.96±0.64	93.98±0.36	94.38±0.52	94.21±0.53	**95.12±0.37**
wdbc	92.55±0.99	93.16±0.93	93.06±0.58	92.46±0.75	93.16±0.82	93.46±0.79	92.90±0.92	**93.57±0.57**
wpbc	68.94±1.11	69.84±2.69	66.12±1.45	70.66±2.03	71.95±2.35	73.80±2.64	69.03±2.25	**75.20±2.48**
Musk	77.74±1.78	75.63±1.65	75.09±1.40	79.94±1.38	79.30±1.96	**81.51±1.36**	78.77±1.06	79.84±1.29
Tiger	74.63±1.09	55.47±0.00	73.20±0.74	71.09±0.83	69.17±1.56	73.43±1.10	74.57±1.04	**75.68±0.57**
Average	76.9689	76.5512	75.3004	74.3888	76.6358	76.2970	77.4004	**78.5208**

- From Tables 4 to 7, it can be observed that UDMAR can maintain or improve accuracy across all datasets for both classification and clustering tasks. UDMAR achieves the best average accuracy among all the compared algorithms.
- In Table 4, it can be observed that UDMAR maintains the best accuracy in 15 datasets. However, for USFSM, UFRFS, USQR, UEBR, FRUAR, HKCMI, it maintains the best accuracy only in 5, 1, 2, 1, 4, 5 datasets respectively.

Table 5
The classification results of the reduced data on NB (%).

Data set	Baseline	USFSM	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR
abalone	25.00 ± 0.14	25.29 ± 0.18	25.04 ± 0.14	25.13 ± 0.16	24.96 ± 0.15	25.29 ± 0.16	25.51 ± 0.27	**26.40 ± 0.12**
autos	60.47 ± 1.35	56.99 ± 1.60	54.11 ± 1.20	59.43 ± 1.31	64.34 ± 1.34	65.00 ± 1.38	61.28 ± 1.22	**66.09 ± 1.77**
bands	70.49 ± 0.30	70.66 ± 0.73	65.67 ± 0.27	67.69 ± 0.40	68.90 ± 0.96	70.86 ± 0.47	**71.45 ± 0.38**	71.09 ± 0.45
cleveland	55.99 ± 1.32	56.60 ± 0.62	56.23 ± 0.97	55.11 ± 0.83	55.33 ± 0.80	**58.36 ± 1.12**	57.16 ± 0.73	57.67 ± 0.56
creditA	66.58 ± 0.47	68.33 ± 0.44	66.64 ± 0.73	67.46 ± 0.76	82.55 ± 0.26	82.55 ± 0.41	67.71 ± 0.49	**85.67 ± 0.24**
diabetes	73.49 ± 0.50	72.93 ± 0.49	73.54 ± 0.45	75.63 ± 0.47	73.40 ± 0.48	75.87 ± 0.30	75.73 ± 0.49	**77.17 ± 0.33**
ecoli	84.57 ± 0.88	84.83 ± 0.60	84.35 ± 0.80	84.44 ± 0.65	84.32 ± 0.53	85.11 ± 0.62	84.80 ± 0.79	**85.11 ± 0.49**
flag	41.41 ± 2.13	**46.13 ± 2.18**	39.04 ± 1.91	35.88 ± 1.36	36.24 ± 2.25	43.27 ± 1.23	37.19 ± 2.79	39.72 ± 2.28
german	70.97 ± 0.32	**72.07 ± 0.34**	70.76 ± 0.32	71.96 ± 0.36	71.62 ± 0.42	71.98 ± 0.48	71.09 ± 0.39	71.98 ± 0.38
glass	90.27 ± 0.45	91.54 ± 1.06	87.99 ± 1.12	90.26 ± 1.11	93.66 ± 0.41	92.01 ± 0.85	91.87 ± 0.64	**96.58 ± 0.80**
heart	79.93 ± 0.42	79.44 ± 1.08	79.48 ± 0.56	79.70 ± 0.80	77.00 ± 0.93	**81.44 ± 0.51**	80.22 ± 0.53	81.15 ± 0.44
hepatitis	68.83 ± 1.84	71.00 ± 2.38	67.46 ± 1.10	68.68 ± 1.84	68.28 ± 2.31	**71.32 ± 1.31**	69.77 ± 1.28	68.86 ± 1.66
hypothyroid	96.83 ± 0.05	**98.96 ± 0.08**	97.01 ± 0.06	97.03 ± 0.06	97.38 ± 0.06	97.34 ± 0.10	97.08 ± 0.06	97.13 ± 0.05
sonar	76.59 ± 0.79	74.61 ± 0.92	57.38 ± 1.00	70.06 ± 0.99	67.13 ± 2.82	74.26 ± 2.42	72.95 ± 2.39	**77.83 ± 1.04**
labor	82.97 ± 4.07	81.63 ± 3.05	87.50 ± 1.95	82.93 ± 2.60	82.47 ± 2.77	89.30 ± 2.82	85.83 ± 2.22	**90.87 ± 1.82**
lymphography	83.18 ± 1.87	79.30 ± 1.94	75.41 ± 1.83	79.66 ± 2.17	80.35 ± 1.84	80.85 ± 1.35	80.46 ± 1.32	**81.37 ± 1.84**
monks	96.25 ± 0.91	**96.92 ± 0.31**	96.20 ± 0.69	96.80 ± 0.34	96.20 ± 1.10	96.76 ± 0.53	96.57 ± 0.63	96.58 ± 0.70
nursery	89.26 ± 0.15	89.12 ± 0.17	89.30 ± 0.13	50.66 ± 1.13	89.27 ± 0.15	50.62 ± 1.11	89.12 ± 0.22	**89.39 ± 0.17**
parkinsons	74.61 ± 0.66	74.54 ± 1.17	77.24 ± 0.65	79.89 ± 1.03	77.21 ± 0.79	82.18 ± 0.63	75.90 ± 0.77	**84.66 ± 0.54**
SCADI	77.29 ± 0.81	55.29 ± 1.79	55.29 ± 1.18	69.43 ± 1.81	74.43 ± 0.81	72.43 ± 2.52	**77.43 ± 0.60**	71.43 ± 0.95
sick	93.74 ± 0.02	93.88 ± 0.00	93.75 ± 0.02	93.74 ± 0.03	93.68 ± 0.04	93.75 ± 0.01	93.75 ± 0.03	**96.52 ± 0.07**
wbc	96.47 ± 0.10	96.48 ± 0.07	96.47 ± 0.07	96.54 ± 0.11	96.47 ± 0.07	96.42 ± 0.10	**96.60 ± 0.09**	96.52 ± 0.10
wdbc	93.71 ± 0.40	94.50 ± 0.22	93.11 ± 0.40	92.95 ± 0.25	93.16 ± 0.15	93.74 ± 0.49	93.78 ± 0.38	**95.55 ± 0.12**
wpbc	65.08 ± 1.41	73.82 ± 1.37	61.61 ± 2.46	73.11 ± 1.25	**78.06 ± 1.41**	77.97 ± 1.41	66.51 ± 1.70	76.42 ± 1.52
Musk	82.77 ± 0.69	74.55 ± 1.00	79.79 ± 0.51	75.14 ± 0.63	73.74 ± 0.74	79.29 ± 0.46	82.98 ± 0.76	**85.88 ± 0.48**
Tiger	45.07 ± 0.79	45.01 ± 0.00	46.93 ± 0.64	62.57 ± 0.35	71.01 ± 0.50	45.73 ± 0.87	45.22 ± 0.76	**71.70 ± 0.19**
Average	74.6854	74.0162	72.2038	73.1492	75.8138	75.1423	74.9215	**78.4362**

Table 6
The classification results of the reduced data on KNN (%).

Date set	Baseline	USFSM	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR
abalone	19.77 ± 0.25	20.75 ± 0.32	19.99 ± 0.38	19.28 ± 0.37	19.93 ± 0.40	20.72 ± 0.28	**20.88 ± 0.29**	20.85 ± 0.35
autos	72.85 ± 0.81	77.79 ± 0.98	71.47 ± 1.16	74.33 ± 1.31	72.67 ± 1.20	**77.44 ± 1.33**	73.67 ± 1.62	77.40 ± 1.94
bands	78.64 ± 0.93	78.42 ± 0.53	77.65 ± 0.95	77.53 ± 0.76	76.82 ± 1.28	78.26 ± 0.81	78.44 ± 0.73	**78.44 ± 0.64**
cleveland	54.35 ± 0.93	54.63 ± 0.88	54.48 ± 0.76	54.17 ± 0.74	52.04 ± 0.75	54.35 ± 1.71	**55.14 ± 0.87**	54.39 ± 1.20
creditA	80.83 ± 0.54	81.19 ± 0.38	80.32 ± 0.69	81.09 ± 0.51	81.22 ± 0.56	**82.10 ± 0.31**	81.17 ± 0.41	81.90 ± 0.56
diabetes	70.38 ± 0.52	70.10 ± 0.60	70.52 ± 0.78	69.13 ± 0.63	70.83 ± 0.77	**71.12 ± 0.58**	69.19 ± 0.78	70.67 ± 0.52
ecoli	81.08 ± 0.54	80.02 ± 0.56	81.01 ± 1.07	80.86 ± 0.37	80.75 ± 0.89	81.07 ± 1.09	**81.35 ± 0.66**	80.08 ± 0.35
flag	45.37 ± 1.23	48.94 ± 1.56	42.01 ± 1.32	46.76 ± 1.56	47.32 ± 1.34	48.37 ± 1.12	46.24 ± 1.08	**46.61 ± 1.05**
german	68.66 ± 0.58	67.73 ± 0.77	67.95 ± 0.74	69.88 ± 0.80	68.51 ± 0.74	69.91 ± 0.60	69.44 ± 0.44	**69.73 ± 0.57**
glass	90.71 ± 0.61	93.42 ± 0.49	91.07 ± 0.62	91.17 ± 0.52	90.78 ± 0.96	93.52 ± 0.48	91.42 ± 0.66	**95.71 ± 0.51**
heart	75.04 ± 0.93	76.15 ± 0.89	74.70 ± 1.28	74.26 ± 0.84	73.00 ± 0.98	78.70 ± 1.21	74.44 ± 1.18	**78.48 ± 0.95**
hepatitis	60.35 ± 1.97	63.65 ± 1.62	58.52 ± 2.07	53.50 ± 1.63	58.43 ± 1.23	60.42 ± 1.78	59.81 ± 1.11	**58.78 ± 2.09**
hypothyroid	97.15 ± 0.13	97.27 ± 0.05	97.13 ± 0.11	97.19 ± 0.14	97.17 ± 0.09	97.21 ± 0.09	97.21 ± 0.06	**97.61 ± 0.09**
sonar	87.02 ± 0.87	**79.71 ± 0.63**	54.47 ± 0.95	78.73 ± 1.10	76.29 ± 1.12	88.51 ± 0.66	87.36 ± 0.61	86.16 ± 1.39
labor	87.33 ± 1.35	**94.30 ± 1.20**	87.93 ± 0.72	85.40 ± 1.35	86.10 ± 1.53	89.10 ± 0.72	88.87 ± 1.36	88.93 ± 1.19
lymphography	78.19 ± 1.39	79.21 ± 1.01	73.23 ± 1.66	78.34 ± 1.47	77.71 ± 1.01	78.54 ± 1.08	75.69 ± 1.48	**76.74 ± 1.61**
monks	91.95 ± 0.65	99.35 ± 0.50	90.90 ± 0.51	99.12 ± 0.56	91.38 ± 0.64	98.94 ± 0.54	99.17 ± 0.47	**91.53 ± 1.05**
nursery	84.73 ± 0.08	**94.25 ± 0.08**	84.71 ± 0.05	43.54 ± 0.37	84.73 ± 0.09	43.58 ± 0.13	91.96 ± 0.13	84.75 ± 0.10
parkinsons	96.26 ± 0.61	93.99 ± 0.63	94.66 ± 0.48	93.85 ± 1.03	96.43 ± 0.60	96.67 ± 0.68	96.37 ± 0.27	**97.33 ± 0.47**
SCADI	79.86 ± 0.45	64.14 ± 1.57	64.43 ± 1.84	71.86 ± 1.36	79.00 ± 0.69	73.14 ± 0.90	80.29 ± 0.60	78.00 ± 0.74
sick	96.25 ± 0.10	96.82 ± 0.13	96.16 ± 0.11	96.21 ± 0.10	96.33 ± 0.03	96.30 ± 0.14	96.28 ± 0.11	**96.89 ± 0.09**
wbc	95.61 ± 0.29	95.79 ± 0.30	95.65 ± 0.34	95.95 ± 0.22	95.91 ± 0.30	**96.08 ± 0.30**	95.94 ± 0.33	95.92 ± 0.15
wdbc	95.40 ± 0.25	95.08 ± 0.24	94.15 ± 0.27	94.49 ± 0.50	94.67 ± 0.26	**95.89 ± 0.41**	95.62 ± 0.49	94.98 ± 0.30
wpbc	69.82 ± 1.80	74.95 ± 1.20	66.12 ± 1.24	72.57 ± 1.13	70.23 ± 1.21	**76.07 ± 1.39**	70.84 ± 1.01	76.42 ± 1.21
Musk	85.75 ± 0.84	77.37 ± 0.86	80.81 ± 0.92	82.20 ± 0.47	82.50 ± 0.60	**86.82 ± 0.61**	86.82 ± 0.61	86.45 ± 0.69
Tiger	77.52 ± 0.52	45.07 ± 0.00	74.07 ± 0.64	70.34 ± 0.66	68.41 ± 0.60	76.17 ± 0.45	77.25 ± 0.70	**78.17 ± 0.39**
Average	77.7258	76.9265	74.7735	75.0673	76.5062	77.2692	78.4946	**78.5738**

- In Table 5, UDMAR maintains the best accuracy in 15 datasets. However, for USFSM, UFRFS, USQR, UEBR, FRUAR, HKCMI, it maintains the best accuracy only in 4, 0, 0, 1, 3, 3 datasets respectively.
- In Table 6, UDMAR maintains the best accuracy in 12 datasets. However, for USFSM, UFRFS, USQR, UEBR, FRUAR, HKCMI, it maintains the best accuracy only in 3, 0, 0, 0, 7, 3 datasets respectively.

Table 7
The clustering results of the reduced data on k-Means (%).

Date set	Baseline	USFSM	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR
abalone	15.57 ± 0.67	15.16 ± 0.71	15.36 ± 0.48	15.56 ± 0.79	15.71 ± 0.72	15.93 ± 0.35	15.71 ± 0.55	**15.82 ± 0.44**
autos	30.63 ± 2.59	**39.52 ± 1.75**	29.66 ± 2.42	28.54 ± 1.22	28.34 ± 1.09	31.37 ± 1.74	32.05 ± 1.30	30.88 ± 2.07
bands	58.76 ± 0.20	58.83 ± 0.18	58.87 ± 0.16	58.76 ± 0.20	**59.27 ± 0.09**	58.87 ± 0.16	58.87 ± 0.16	58.87 ± 0.16
cleveland	35.21 ± 3.82	39.70 ± 3.60	36.80 ± 2.64	41.49 ± 6.98	39.21 ± 1.79	42.57 ± 8.70	41.62 ± 7.67	**41.98 ± 8.69**
creditA	56.74 ± 3.21	58.41 ± 2.39	57.78 ± 3.60	56.00 ± 3.09	57.07 ± 3.48	58.58 ± 2.79	58.58 ± 2.79	**59.28 ± 2.34**
diabetes	66.35 ± 0.93	68.85 ± 4.09	66.80 ± 0.00	66.35 ± 0.67	66.80 ± 0.00	70.72 ± 0.70	66.68 ± 0.11	**70.55 ± 2.47**
ecoli	62.02 ± 9.18	52.56 ± 6.64	61.82 ± 6.87	58.90 ± 6.44	55.39 ± 2.43	60.74 ± 7.49	63.87 ± 9.30	**57.65 ± 4.38**
flag	26.60 ± 2.18	**40.93 ± 4.27**	27.32 ± 1.67	26.13 ± 1.93	27.68 ± 1.56	27.63 ± 2.29	27.84 ± 1.75	28.04 ± 1.98
german	52.70 ± 0.00	**60.93 ± 3.40**	52.70 ± 0.00	52.70 ± 0.00	52.70 ± 0.00	53.62 ± 2.91	52.80 ± 0.32	54.38 ± 2.71
glass	62.20 ± 11.51	58.74 ± 5.73	62.15 ± 8.52	61.07 ± 6.75	62.29 ± 6.58	65.47 ± 4.41	65.79 ± 6.29	**61.31 ± 0.30**
heart	74.59 ± 5.39	**76.30 ± 0.00**	74.59 ± 5.39	72.85 ± 7.26	61.63 ± 4.78	**76.30 ± 0.00**	**76.30 ± 0.00**	**76.30 ± 0.00**
hepatitis	61.61 ± 0.34	59.23 ± 2.91	61.03 ± 0.33	60.06 ± 1.98	59.48 ± 0.67	61.29 ± 0.00	62.19 ± 0.92	**62.52 ± 4.05**
hypothyroid	77.26 ± 11.72	85.06 ± 5.60	79.48 ± 8.95	77.72 ± 7.99	74.39 ± 7.98	83.95 ± 6.43	84.11 ± 7.15	**75.31 ± 4.77**
sonar	54.66 ± 1.60	**53.85 ± 0.00**	56.30 ± 0.15	58.27 ± 0.30	52.16 ± 0.41	65.53 ± 0.32	55.38 ± 0.78	63.70 ± 0.52
labor	73.33 ± 7.71	69.65 ± 11.07	72.11 ± 10.70	71.05 ± 11.26	70.70 ± 12.01	78.07 ± 4.24	**79.30 ± 4.74**	80.70 ± 5.16
lymphography	50.34 ± 7.42	48.24 ± 7.13	45.95 ± 7.26	45.81 ± 6.76	50.41 ± 5.32	46.96 ± 6.04	46.49 ± 7.54	**51.82 ± 5.64**
monks	62.69 ± 3.81	63.98 ± 0.29	63.19 ± 1.56	65.74 ± 3.70	64.31 ± 3.71	63.89 ± 0.00	63.24 ± 4.66	**65.56 ± 5.27**
nursery	31.52 ± 1.85	**32.67 ± 2.22**	30.12 ± 0.89	30.71 ± 0.92	29.69 ± 1.02	30.51 ± 0.82	30.59 ± 1.12	31.22 ± 1.69
parkinsons	62.87 ± 0.26	69.95 ± 3.03	66.46 ± 3.65	68.62 ± 0.63	65.49 ± 4.17	74.05 ± 0.36	65.13 ± 1.99	**76.10 ± 1.11**
SCADI	61.29 ± 7.48	39.00 ± 3.87	38.86 ± 2.92	47.71 ± 10.57	52.71 ± 3.40	47.57 ± 6.74	55.43 ± 3.61	55.14 ± 4.37
sick	77.79 ± 7.05	83.00 ± 7.40	76.13 ± 7.06	78.80 ± 11.48	73.59 ± 7.73	75.67 ± 8.91	81.88 ± 6.67	**74.54 ± 8.37**
wbc	95.85 ± 0.00	95.85 ± 0.00	95.85 ± 0.00	95.99 ± 0.00	95.85 ± 0.00	95.75 ± 0.10	95.99 ± 0.00	**95.85 ± 0.00**
wdbc	92.79 ± 0.00	92.79 ± 0.00	88.56 ± 9.13	91.49 ± 0.09	90.72 ± 0.34	92.72 ± 0.15	93.29 ± 0.07	**94.09 ± 0.15**
wpbc	59.85 ± 0.27	61.67 ± 4.95	62.17 ± 9.48	56.36 ± 2.52	57.78 ± 2.73	63.69 ± 5.34	61.77 ± 4.96	**66.01 ± 3.88**
Musk	53.97 ± 0.60	54.10 ± 3.81	53.93 ± 0.10	55.88 ± 0.00	53.07 ± 0.20	54.10 ± 0.70	54.58 ± 0.95	**56.62 ± 3.31**
Tiger	55.21 ± 0.58	54.92 ± 0.00	55.52 ± 0.33	57.98 ± 3.29	57.20 ± 3.39	61.89 ± 0.10	55.25 ± 0.54	**62.69 ± 0.03**
Average	58.1692	59.2031	57.4050	57.8027	56.7916	59.9462	59.4216	**60.2639**

- In Table 7, UDMAR maintains the best accuracy in 18 datasets. However, for USFSM, UFRFS, USQR, UEBR, FRUAR, HKCMI, it maintains the best accuracy only in 6, 0, 0, 1, 1, 2 datasets respectively.

From the analysis above, it is evident that UDMAR consistently maintains the highest accuracy across most datasets while achieving a relatively smaller reduction subset. This is due to UDMAR's ability to fully utilize the fuzzy information within the fuzzy discernibility matrix and its adaptive stopping mechanism. Therefore, UDMAR produces superior and higher-quality reduction results compared to other algorithms.

7.4. Ablation study

To evaluate UDMAR's capability in applying fuzzy information, we conduct an ablation study in this section. The specific experimental results are shown in Fig. 3. As illustrated, CART and NB are used to demonstrate the information utilization ability of the fuzzy deletion function. Compared to the thresholding method, our approach retains more information from the data, particularly in the NB task, leading to improved performance in the downstream tasks.

However, while our algorithm demonstrates clear advantages on most datasets, traditional methods still achieve higher accuracy on some datasets and under certain parameters. For instance, on the sick dataset, the accuracy of the traditional method is slightly higher than that of UDMAR. Similarly, on the labor and autos datasets, there are instances where the precision for specific parameter segments exceeds the precision of UDMAR. This could be due to our algorithm's stopping method halting too early, leading to information loss.

In summary, the results from the majority of datasets demonstrate UDMAR's ability to retain information, proving that the fuzzy deletion function outperforms traditional binary methods by fully utilizing the information within the datasets. While the stopping mechanism works effectively on most datasets, there may still be situations of information loss on some datasets.

7.5. Runtime analysis

Runtime experiments are effective in evaluating the efficacy of algorithms. Detailed information on the performance of the seven attribute reduction algorithms can be found in Table 8 and Fig. 4.

In Table 8, UDMAR's runtime is the lowest among all datasets. In some datasets, its runtime is tens of times faster than the second-ranked algorithm. For example, in bands dataset, UDMAR's execution time is 77.5372 times lower compared to HMCMI, and in SCADI, it is 1241.3 times lower. UDMAR is highly efficient on large-scale datasets. In Musk, UDMAR's runtime is just over 5 seconds, while in Tiger, UDMAR's runtime is also less than a minute. In terms of average runtime, UDMAR still maintains the lowest. "—" indicates that the runtime for the Nursery dataset was not obtained due to memory overflow.

Table 8
Runtimes of the seven algorithms (in seconds).

Date set	USFSM	UFRFS	USQR	UEBR	FRUAR	HKCMI	UDMAR	UDAMR (GPU)
abalone	3607.71±284.77	4096.58±113.37	574.05±14.09	381.73±6.29	11035.82±1899	220.67±7.77	**123.45±0.43**	117.92±0.11
autos	9.28±0.88	1.65±0.10	22.78±0.93	3.35±0.16	74.08±10.78	3.99±0.41	**0.06±0.00**	0.06±0.01
bands	572.54±65.71	34.34±5.32	515.01±129.37	19.77±2.20	2240.41±175.40	93.82±1.42	**0.81±0.01**	0.63±0.03
cleveland	46.36±6.68	2.09±0.09	7.67±0.28	1.45±0.02	22.98±4.10	1.27±0.10	**0.06±0.00**	0.07±0.00
creditA	342.47±27.69	29.89±5.57	78.77±8.46	13.78±0.76	421.06±143.41	9.85±0.31	**0.74±0.00**	0.48±0.00
diabetes	217.73±34.65	41.87±3.66	23.73±1.53	4.83±0.06	166.12±6.92	3.26±0.15	**0.55±0.02**	0.33±0.00
ecoli	41.39±6.59	1.49±0.06	2.13±0.14	0.42±0.01	6.88±3.28	0.27±0.01	**0.04±0.00**	0.04±0.00
flag	9.15±1.46	1.65±0.10	20.50±1.20	3.51±0.18	46.01±21.60	4.31±0.29	**0.06±0.00**	0.06±0.00
german	771.56±94.79	113.98±16.20	360.24±15.97	52.84±4.49	1588.93±112.95	48.49±2.95	**2.78±0.02**	1.71±0.06
glass	3.96±0.60	0.85±0.32	2.43±0.20	0.48±0.02	19.59±6.81	0.29±0.01	**0.03±0.00**	0.03±0.00
heart	26.17±3.16	1.64±0.18	6.34±0.26	1.32±0.04	21.72±3.30	0.96±0.06	**0.05±0.00**	0.05±0.00
hepatitis	4.87±0.47	0.62±0.02	4.70±0.08	1.29±0.02	15.28±2.77	1.03±0.10	**0.03±0.00**	0.03±0.00
hypothyroid	5257±151	6479±710	5785±247	2360±11	16750±2216	1618±27.15	**155.49±1.74**	152.68±0.98
labor	0.83±0.20	0.06±0.01	0.52±0.04	0.15±0.02	0.76±0.11	0.15±0.00	**0.01±0.00**	0.01±0.00
lymphography	3.50±0.99	0.51±0.02	3.60±0.44	0.80±0.00	3.84±0.67	0.65±0.05	**0.02±0.00**	0.02±0.00
monks	50.68±20.08	2.81±0.24	1.38±0.22	0.43±0.01	1.65±0.21	0.30±0.02	**0.06±0.00**	0.07±0.00
nursery	35119±1546	121738±26488	10581±383	8210±388	10695±987	5161±113	**3927±11.38**	—
parkinsons	8.42±3.00	1.24±0.05	12.67±2.27	1.97±0.01	151.20±29.66	2.55±0.31	**0.05±0.01**	0.05±0.00
sick	7941±1524	11154±958	10478±526	3575±32	27957±2925	3108±75	**324±0.85**	358.44±24.93
wbc	178.26±107.12	20.86±4.21	20.03±2.39	3.50±0.05	28.59±4.49	2.70±0.11	**0.47±0.02**	0.36±0.10
wdbc	421.31±237.72	30.27±2.48	329.79±27.81	36.13±1.31	16260±7605	56.44±1.64	**1.02±0.14**	0.87±0.27
wpbc	13.79±5.23	2.04±0.08	25.23±4.48	3.97±0.33	615.19±19.49	8.05±0.14	**0.08±0.02**	0.09±0.02
SCADI	10.26±2.11	2.28±0.26	157.30±5.52	39.28±3.00	214.28±33.77	111.72±57.20	**0.06±0.01**	0.10±0.03
sonar	25.20±7.56	4.52±0.74	99.87±11.25	14.47±1.14	5048.42±928.80	50.20±0.61	**0.18±0.06**	0.16±0.01
Musk	20.13±0.33	95.3±0.85	5016±730	774±3.18	27505±14563	6092±104.4	**5.45±1.13**	2.63±0.12
Tiger	305.31±8.78	2075±50.78	35448±616	9817±61.28	457018±42332	88758±2038	**48.99±4.90**	45.81±8.16
Average	2116	5613	2676	974	22227	4052	**177**	—

We present the results of the algorithms' runtime as data volume increases in Fig. 4. The horizontal axis represents the data volume at different proportions, while the vertical axis represents the algorithms' runtime. It is evident that UDMAR achieves the lowest runtime across various proportions in the presented datasets.

From the experimental results above, it is clear that the UDMAR algorithm holds a significant advantage in runtime compared to other unsupervised attribute reduction algorithms. Therefore, UDMAR can significantly reduce computational time.

7.6. Parameter sensitive analysis

The threshold λ plays a crucial role within the UDMAR algorithm, acting as a parameter to control the granularity of fuzzy rough data analysis. This allows for the generation of distinct attribute subsets at various levels of granularity, leading to subsets that are suited for various downstream tasks. The impact of the λ parameter under different downstream tasks is depicted in Fig. 5.

As the λ parameter fluctuates, most datasets exhibit stable performance across most downstream tasks, as exemplified in the autos, cleveland, creditA, flag, glass, heart lymphography, parkinsons, sick and SCADI datasets. In summary, our algorithm maintains stable precision across most datasets as λ varies and is capable of selecting the optimal precision.

Building upon these analyses, we infer that UDMAR is particularly well-suited for experiments involving downstream tasks such as classification and clustering.

7.7. Hypothesis test

In this subsection, a hypothesis testing experiment is employed to evaluate the statistical significance of the algorithm, implementing both the Friedman [49] and Nemenyi tests. [50]

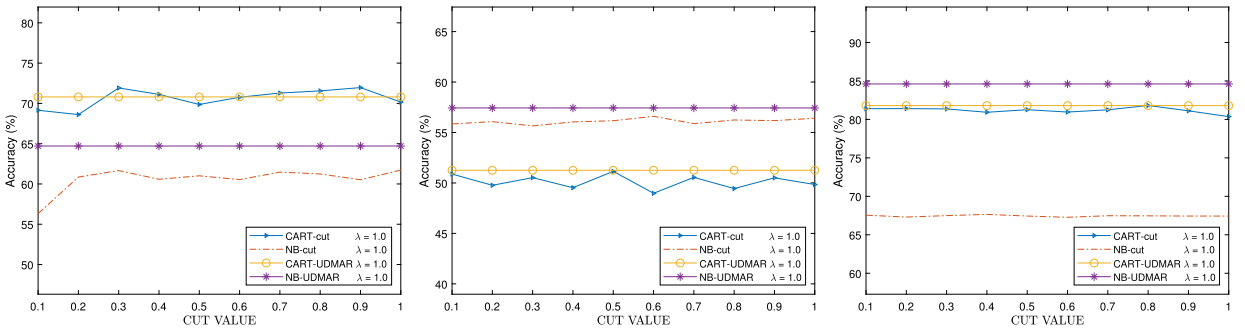
In the Friedman test, the first step is to rank the accuracies of each algorithm across diverse datasets. These accuracies are arranged in an ascending order, with each assigned a distinct natural number (1, 2, …), referred to as 'ranks'. If two accuracies are identical, their ranks are averaged. Suppose we are evaluating M algorithms over N datasets, the average rank of each specific algorithm is denoted as r_i. Following this, the Friedman test statistic is shown as follows:

$$\tau_{\chi^2} = \frac{12N}{M(M+1)}\left(\sum_{i=1}^{M} r_i^2 - \frac{M(M+1)^2}{4}\right).$$

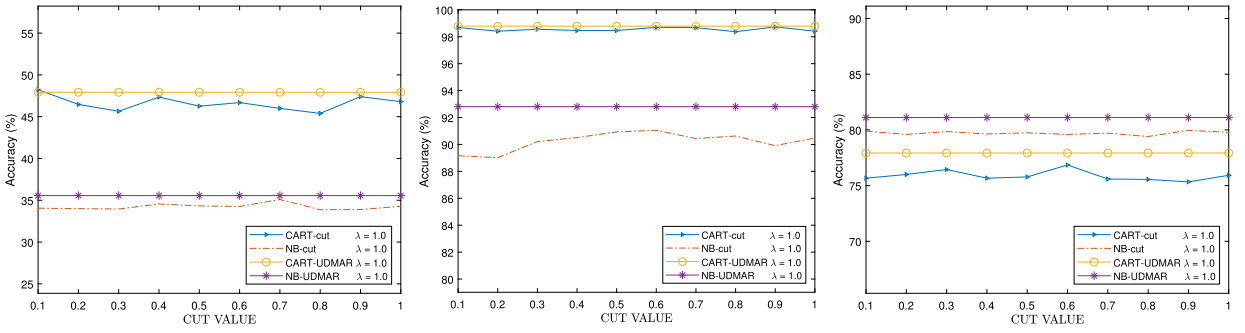
However, given the conservative nature of the original Friedman test statistic, we introduce an alternate statistic, denoted as τ_F:

$$\tau_F = \frac{(N-1)\tau_{\chi^2}}{N(M-1)-\tau_{\chi^2}},$$

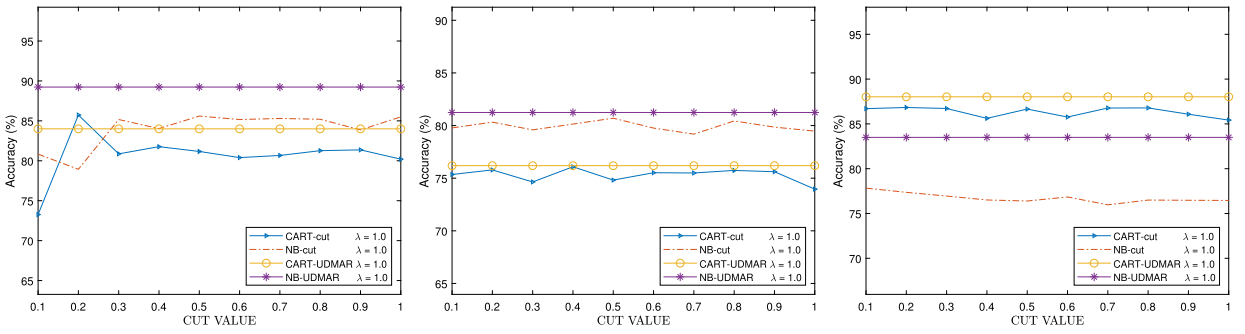
where τ_F follows an F-distribution with degrees of freedom $(M-1)$ and $(M-1)(N-1)$. The null hypothesis of the Friedman test suggests that there are no significant differences in the performance of the algorithms across datasets. A rejection of this null hypothesis indicates significant discrepancies in the algorithms' performance. To further investigate these differences, we perform post-hoc tests, utilizing the Nemenyi test in this study. This test is primarily used to compare the differences between algorithms to



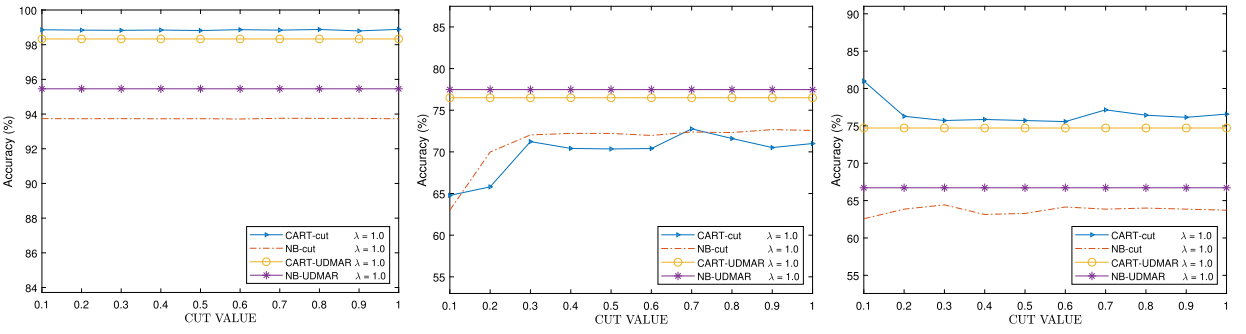
(a) autos (b) cleveland (c) creditA



(d) flag (e) glass (f) heart

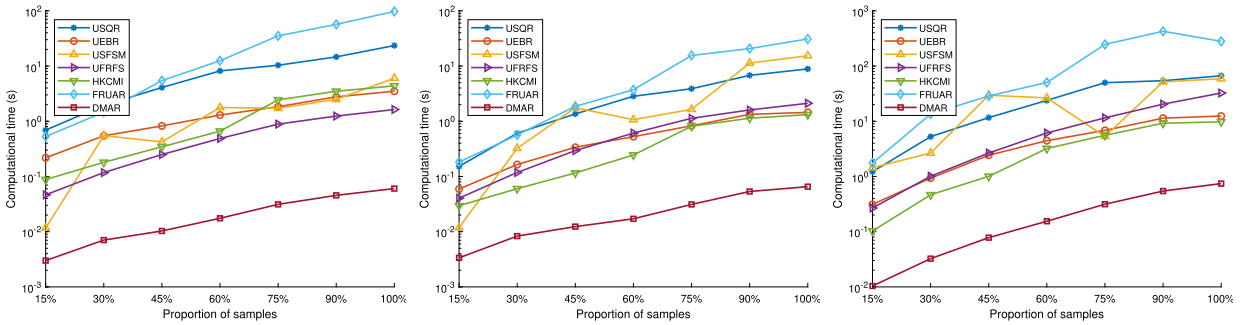


(g) labor (h) lymphography (i) parkinsons

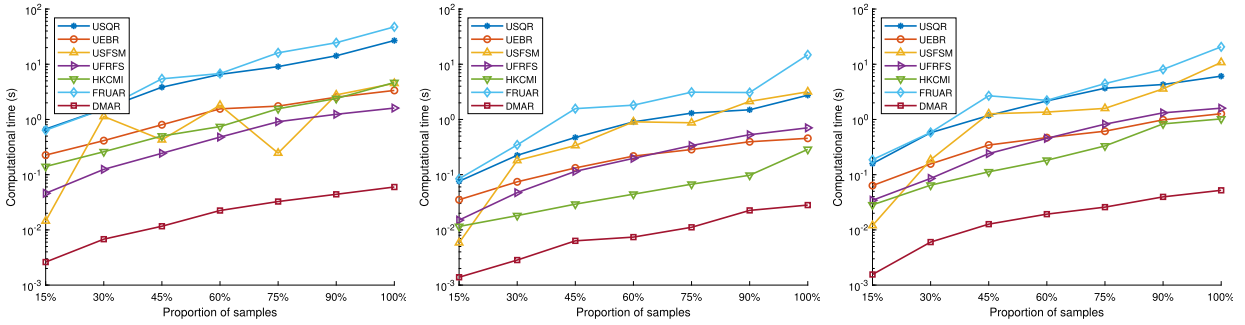


(j) sick (k) sonar (l) SCADI

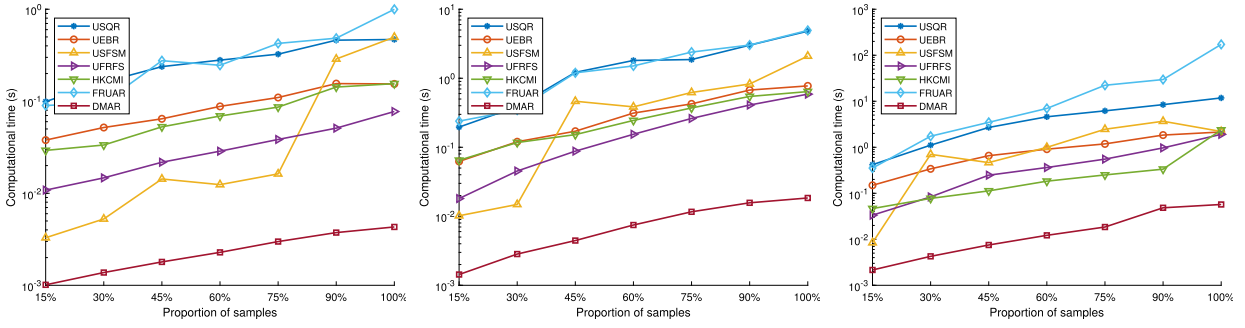
Fig. 3. Ablation study experiment.



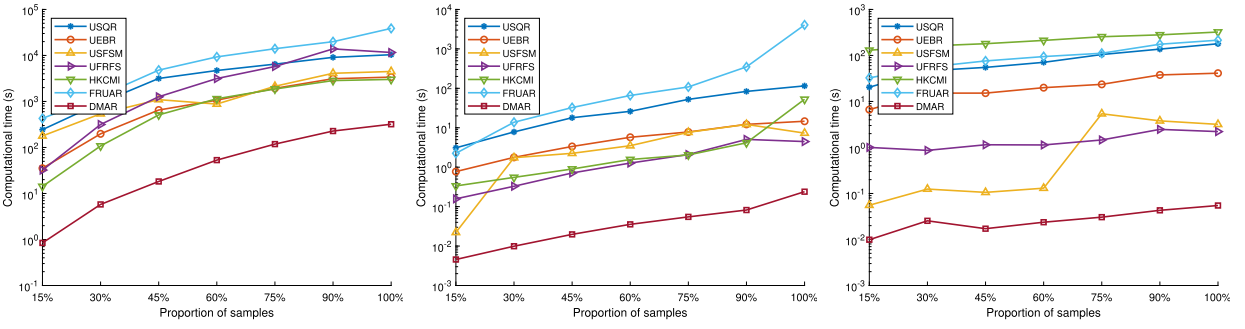
(a) autos (b) cleveland (c) creditA



(d) flag (e) glass (f) heart

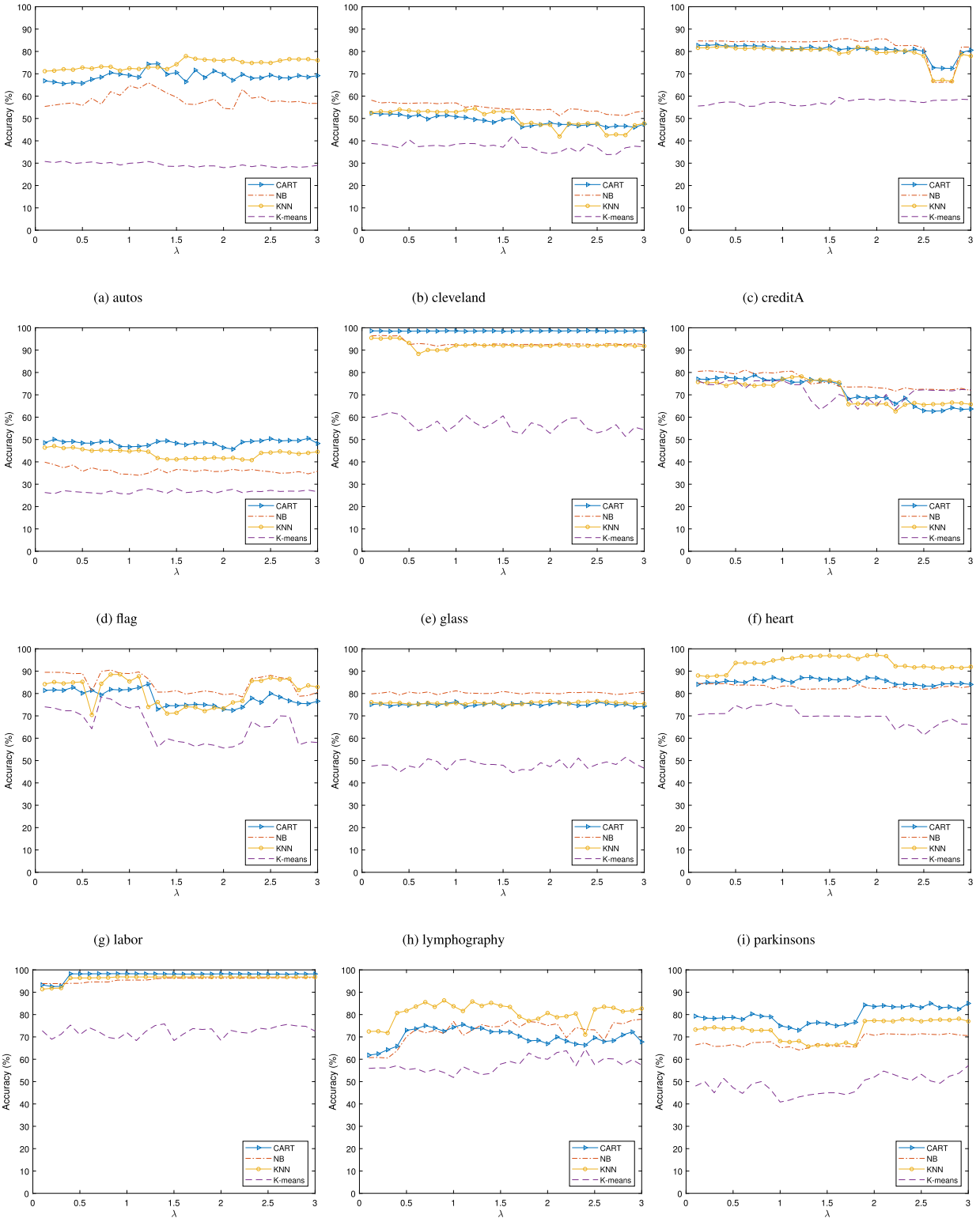


(g) labor (h) lymphography (i) parkinsons



(j) sick (k) sonar (l) SCADI

Fig. 4. Runtime analysis experiment.



(a) autos (b) cleveland (c) creditA

(d) flag (e) glass (f) heart

(g) labor (h) lymphography (i) parkinsons

(j) sick (k) sonar (l) SCADI

Fig. 5. Parameter sensitive experiment.

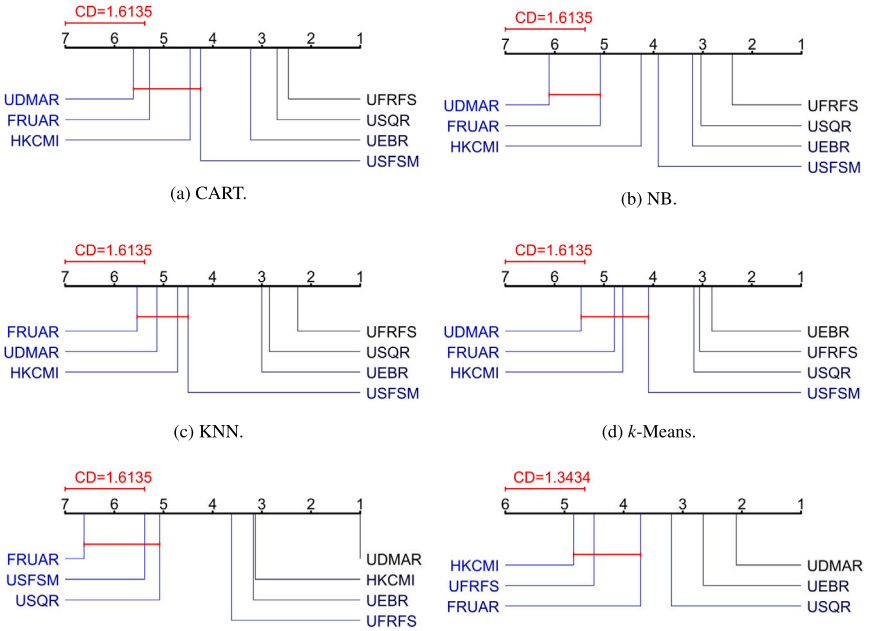


Fig. 6. Nemenyi's test.

Table 9
τ_F on CART, NB, KNN, k-Means, runtime analysis and the number of selected attributes.

Methods	τ_F	Critical value ($\alpha = 0.1$)
CART	12.2613	1.8138
NB	13.4622	
KNN	13.3298	
k-Means	7.0206	
Time	68.3057	
Red	11.8835	1.8939

a critical difference. Computing this critical difference reveals significant variations at the defined confidence level. In the Nemenyi test, the calculation of the critical difference (CD) is performed by:

$$CD_\alpha = q_\alpha \sqrt{\frac{M(M+1)}{6N}},$$

where q_α represents the critical value from Table 9, based on the significance level α as explained in [49]. If the difference between the algorithms exceeds the mean rank difference, it indicates a significant difference between them.

The Nemenyi test table provides a clear visual representation of the statistical differences between the reduction algorithms. In this chart, the average ranks of all reduction algorithms are shown on a numerical scale. A red line connecting algorithms indicates no substantial differences between them. For the four downstream task algorithms and the runtime experiment, with $M = 7$ and $N = 26$, the τ_F distribution has 6 and 150 degrees of freedom. In the reduction comparison experiment, with M = 6 and N = 26, the τ_F distribution has 5 and 125 degrees of freedom. Based on the Friedman test, we present the corresponding τ_F values and critical values (with $\alpha = 0.1$) for different learning algorithms in Table 9. The τ_F values for all algorithms in the downstream tasks and runtime analysis experiment exceed the critical value of 1.8138. For the reduction number comparison, the critical value is 1.8939. Therefore, we reject the null hypothesis of no substantial differences across the reduction algorithms, confirming significant differences among them. Post-hoc examinations are then conducted. At a significance level of $\alpha = 0.1$, we calculate the critical differences $CD_{0.1}$ to be 1.6135 and 1.3434. We then generate the Nemenyi test chart, displaying the comparative performances of different reduction algorithms across four downstream tasks and two comparison tables, as shown in Fig. 6. As shown in Fig. 6, UDMAR shows significant differences compared to most evaluated algorithms, particularly under subgraphs (a) CART, (b) NB, (d) k-Means, (e) Time, and (f) Reduction. UDMAR shows no significant differences with only 1 or 2 algorithms in (a) CART, (b) NB, and (d) k-Means. In (e) Time, UDMAR achieves the best performance. This indicates UDMAR significantly improves runtime while maintaining effectiveness in downstream tasks. However, in (c) KNN, FRUAR's rank is superior to UDMAR's, though there is no significant difference between

them. Overall, UDMAR greatly accelerates the algorithm process while maintaining and improving accuracy and reducing the number of reductions.

Upon analyzing the data, it is evident that UDMAR shows significant differences across most downstream tasks due to its superior performance.

8. Conclusion

Overall, the attribute reduction algorithm proposed in this paper effectively utilizes fuzzy information within datasets and accelerates the algorithm through parallel computation. This is attributed to the efficient handling of fuzzy information by the fuzzy deletion function. We have demonstrated the effectiveness of the proposed algorithm through experiments.

However, despite the advantages in speed and accuracy, this method still has limitations in certain situations. The fuzzy deletion function only considers the relationships and interactions among fine-grained information, without taking into account other coarse-grained information in the dataset, such as the information carried in the positive, negative, and boundary regions of fuzzy sets. On nominal datasets, which typically do not provide sufficient fuzzy information, the algorithm may degrade into a traditional discernibility matrix reduction method, resulting in poorer performance in removing redundant attributes. To incorporate coarse-grained information, the following directions have the potential to enhance the performance of this method: 1. Consider integrating information such as the positive region constructed by fuzzy sets with the variables in the fuzzy deletion function. 2. Calculate the similarity relationships between attributes and integrate this similarity information with the discernibility matrix. 3. Apply delayed decision-making methods such as three-way decisions to construct algorithm stopping conditions, thereby improving the ability to remove redundancy more comprehensively.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

CRediT authorship contribution statement

Haotong Wen: Writing – review & editing, Software, Methodology. **Yi Xu:** Writing – review & editing. **Meishe Liang:** Writing – review & editing.

Declaration of competing interest

The 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.

Data availability

The datasets generated during and/or analyzed during the current study are available in the UCI repository (http://archive.ics.uci.edu/ml).

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions, which have greatly improved this paper. This work was supported by the National Natural Science Foundation of China (No. 62076002, 61402005, 61972001), the Natural Science Foundation of Anhui Province, China (No. 2008085MF194, 1308085QF114, 1908085MF188), and the Higher Education Natural Science Foundation of Anhui Province, China (No. KJ2013A015).

References

[1] B.P. Doppala, D. Bhattacharyya, M. Chakkravarthy, T.-h. Kim, A hybrid machine learning approach to identify coronary diseases using feature selection mechanism on heart disease dataset, Distrib. Parallel Databases (2023) 1–20.
[2] B. Yuzhang, M. Jusheng, Adaptive intuitionistic fuzzy neighborhood classifier, Int. J. Mach. Learn. Cybern. 15 (2024) 1855–1871.
[3] M. Akram, H.S. Nawaz, M. Deveci, Attribute reduction and information granulation in Pythagorean fuzzy formal contexts, Expert Syst. Appl. 222 (2023) 119794.
[4] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (1990) 191–209.
[5] D. Dubois, H. Prade, Putting rough sets and fuzzy sets together, intelligent decision support, in: Handbook of applications and advances of the rough sets theory, 1992, pp. 203–232.
[6] G. Lang, M. Cai, H. Fujita, Q. Xiao, Related families-based attribute reduction of dynamic covering decision information systems, Knowl.-Based Syst. 162 (2018) 161–173.
[7] C. Lee, D.A. Landgrebe, Feature extraction based on decision boundaries, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 388–400.
[8] K. Liu, X. Yang, H. Yu, H. Fujita, X. Chen, D. Liu, Supervised information granulation strategy for attribute reduction, Int. J. Mach. Learn. Cybern. 11 (2020) 2149–2163.
[9] Y. Qian, J. Liang, W. Pedrycz, C. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, Pattern Recognit. 44 (2011) 1658–1670.

[10] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 26, 2012, pp. 1026–1032.
[11] M. Qian, C. Zhai, Robust unsupervised feature selection, in: Twenty-Third International Joint Conference on Artificial Intelligence, Citeseer, 2013.
[12] L. Dong, D. Chen, N. Wang, Z. Lu, Key energy-consumption feature selection of thermal power systems based on robust attribute reduction with rough sets, Inf. Sci. 532 (2020) 61–71.
[13] P. Zhang, T. Li, G. Wang, C. Luo, H. Chen, J. Zhang, D. Wang, Z. Yu, Multi-source information fusion based on rough set theory: a review, Inf. Fusion 68 (2021) 85–117.
[14] X. Zhang, H. Yao, Z. Lv, D. Miao, Class-specific information measures and attribute reducts for hierarchy and systematicness, Inf. Sci. 563 (2021) 196–225.
[15] X. Zhang, C. Mei, D. Chen, J. Li, Feature selection in mixed data: a method using a novel fuzzy rough set-based information entropy, Pattern Recognit. 56 (2016) 1–15.
[16] Y. Wang, X. Chen, K. Dong, Attribute reduction via local conditional entropy, Int. J. Mach. Learn. Cybern. 10 (2019) 3619–3634.
[17] Z. Yuan, H. Chen, T. Li, Exploring interactive attribute reduction via fuzzy complementary entropy for unlabeled mixed data, Pattern Recognit. 127 (2022) 108651.
[18] Z. Yuan, H. Chen, X. Yang, T. Li, K. Liu, Fuzzy complementary entropy using hybrid-kernel function and its unsupervised attribute reduction, Knowl.-Based Syst. 231 (2021) 107398.
[19] Z. Yuan, H. Chen, T. Li, Z. Yu, B. Sang, C. Luo, Unsupervised attribute reduction for mixed data based on fuzzy rough sets, Inf. Sci. 572 (2021) 67–87.
[20] H. Wen, S. Zhao, M. Liang, Unsupervised attribute reduction algorithm for mixed data based on fuzzy optimal approximation set, Mathematics 11 (2023) 3452.
[21] R. Jensen, Rough set-based feature selection: a review, in: Rough Computing: Theories, Technologies and Applications, 2008, pp. 70–107.
[22] B. Huang, R. Cheng, Z. Li, Y. Jin, K.C. Tan, Evox: a distributed gpu-accelerated framework for scalable evolutionary computation, IEEE Trans. Evol. Comput. (2024).
[23] Z. Guo, T.-W. Huang, Y. Lin, Accelerating static timing analysis using cpu–gpu heterogeneous parallelism, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 42 (2023) 4973–4984.
[24] C. Park, Y. Lu, S. Saha, T. Xue, J. Guo, S. Mojumder, D.W. Apley, G.J. Wagner, W.K. Liu, Convolution hierarchical deep-learning neural network (c-hidenn) with graphics processing unit (gpu) acceleration, Comput. Mech. 72 (2023) 383–409.
[25] T. Um, B. Oh, B. Seo, M. Kweun, G. Kim, W.-Y. Lee, Fastflow: accelerating deep learning model training with smart offloading of input data pipeline, Proc. VLDB Endow. 16 (2023) 1086–1099.
[26] P. Wang, J. He, Z. Li, Attribute reduction for hybrid data based on fuzzy rough iterative computation model, Inf. Sci. 632 (2023) 555–575.
[27] X. Ji, J. Li, S. Yao, P. Zhao, Attribute reduction based on fusion information entropy, Int. J. Approx. Reason. 160 (2023) 108949.
[28] J. Dai, Q. Liu, W. Chen, C. Zhang, Multi-label feature selection based on fuzzy mutual information and orthogonal regression, IEEE Trans. Fuzzy Syst. (2024).
[29] J. Chen, J. Mi, Y. Lin, A graph approach for fuzzy-rough feature selection, Fuzzy Sets Syst. 391 (2020) 96–116.
[30] Y. Liu, L. Zheng, Y. Xiu, H. Yin, S. Zhao, X. Wang, H. Chen, C. Li, Discernibility matrix based incremental feature selection on fused decision tables, Int. J. Approx. Reason. 118 (2020) 1–26.
[31] J. Dai, H. Hu, W.-Z. Wu, Y. Qian, D. Huang, Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets, IEEE Trans. Fuzzy Syst. 26 (2017) 2174–2187.
[32] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Trans. Fuzzy Syst. 17 (2008) 824–838.
[33] P. Sowkuntla, P.S. Prasad, Mapreduce based parallel fuzzy-rough attribute reduction using discernibility matrix, Appl. Intell. 52 (2022) 154–173.
[34] C. Wang, Y. Huang, W. Ding, Z. Cao, Attribute reduction with fuzzy rough self-information measures, Inf. Sci. 549 (2021) 68–86.
[35] M. Hu, Y. Guo, D. Chen, E.C. Tsang, Q. Zhang, Attribute reduction based on neighborhood constrained fuzzy rough sets, Knowl.-Based Syst. 274 (2023) 110632.
[36] R. Jensen, Q. Shen, Fuzzy–rough attribute reduction with application to web categorization, Fuzzy Sets Syst. 141 (2004) 469–485.
[37] W. Xu, M. Huang, Z. Jiang, Y. Qian, Graph-based unsupervised feature selection for interval-valued information system, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[38] D.S. Yeung, D. Chen, E.C. Tsang, J.W. Lee, W. Xizhao, On the generalization of fuzzy rough sets, IEEE Trans. Fuzzy Syst. 13 (2005) 343–361.
[39] Q. Hu, D. Yu, W. Pedrycz, D. Chen, Kernelized fuzzy rough sets and their applications, IEEE Trans. Knowl. Data Eng. 23 (2010) 1649–1667.
[40] Z. Yuan, H. Chen, P. Xie, P. Zhang, J. Liu, T. Li, Attribute reduction methods in fuzzy rough set theory: an overview, comparative experiments, and new directions, Appl. Soft Comput. 107 (2021) 107353.
[41] Z. Yuan, X. Zhang, S. Feng, Hybrid data-driven outlier detection based on neighborhood information entropy and its developmental measures, Expert Syst. Appl. 112 (2018) 243–257.
[42] D. Dheeru, E.K. Taniskidou, 2017, Uci machine learning repository.
[43] S. Solorio-Fernández, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, A new unsupervised spectral feature selection method for mixed data: a filter approach, Pattern Recognit. 72 (2017) 314–326.
[44] N. Mac Parthaláin, R. Jensen, Unsupervised fuzzy-rough set-based dimensionality reduction, Inf. Sci. 229 (2013) 106–121.
[45] C. Velayutham, K. Thangavel, Unsupervised quick reduct algorithm using rough set theory, J. Electron. Sci. Technol. 9 (2011) 193–201.
[46] C. Velayutham, K. Thangavel, A novel entropy based unsupervised feature selection algorithm using rough set theory, in: IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM-2012), IEEE, 2012, pp. 156–161.
[47] P. Zhu, W. Zhu, Q. Hu, C. Zhang, W. Zuo, Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (2017) 364–374.
[48] Q. Hu, D. Yu, Z. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, Pattern Recognit. Lett. 27 (2006) 414–423.
[49] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[50] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.