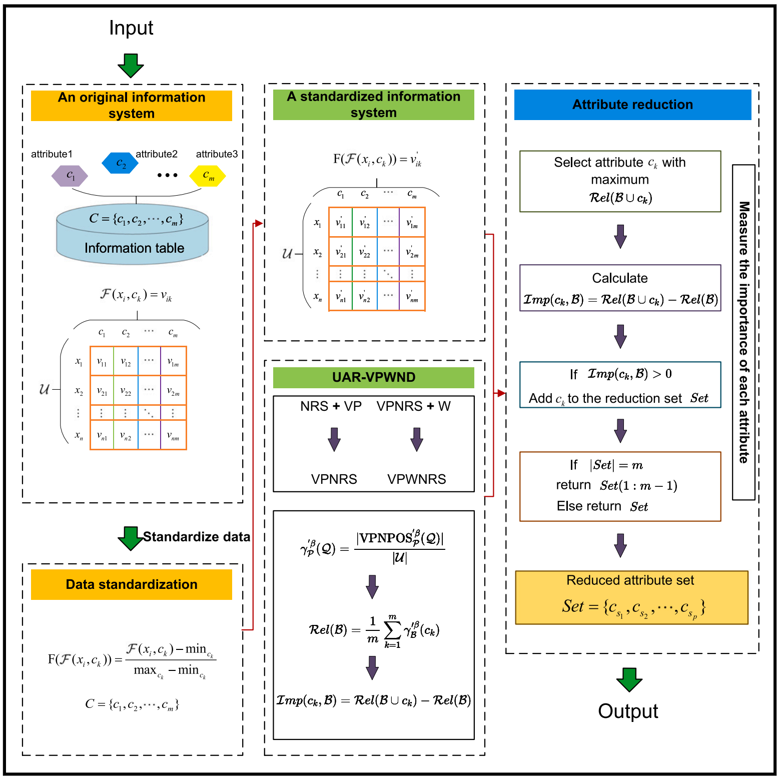
iScience

Article

Unsupervised attribute reduction based on variable precision weighted neighborhood dependency

Graphical abstract


Input
An original information system
attribute1
attribute2
attribute3
c_1
c_2
c_m
$C=\{c_1,c_2,\cdots,c_m\}$
Information table
$\mathcal{F}(x_i,c_k)=v_{ik}$
c_1 c_2 $\cdots$ c_m
$\mathcal{U}$
x_1 v_{11} v_{12} $\cdots$ v_{1m}
x_2 v_{21} v_{22} $\cdots$ v_{2m}
x_n v_{n1} v_{n2} $\cdots$ v_{nm}
Standardize data
Data standardization
$\mathrm{F}(\mathcal{F}(x_i,c_k))=\frac{\mathcal{F}(x_i,c_k)-\min_{c_k}}{\max_{c_k}-\min_{c_k}}$
$C=\{c_1,c_2,\cdots,c_m\}$
A standardized information system
$\mathrm{F}(\mathcal{F}(x_i,c_k))=v'_{ik}$
c_1 c_2 $\cdots$ c_m
$\mathcal{U}$
x_1 v'_{11} v'_{12} $\cdots$ v'_{1m}
x_2 v'_{21} v'_{22} $\cdots$ v'_{2m}
x_n v'_{n1} v'_{n2} $\cdots$ v'_{nm}
UAR-VPWND
NRS + VP
VPNRS + W
VPNRS
VPWNRS
$\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q})=\frac{|\mathrm{VPNPOS}'^{\beta}_{\mathcal{P}}(\mathcal{Q})|}{|\mathcal{U}|}$
$\mathcal{R}el(\mathcal{B})=\frac{1}{m}\sum_{k=1}^{m}\gamma'^{\beta}_{\mathcal{B}}(c_k)$
$\mathcal{I}mp(c_k,\mathcal{B})=\mathcal{R}el(\mathcal{B}\cup c_k)-\mathcal{R}el(\mathcal{B})$
Attribute reduction
Select attribute c_k with maximum $\mathcal{R}el(\mathcal{B}\cup c_k)$
Calculate $\mathcal{I}mp(c_k,\mathcal{B})=\mathcal{R}el(\mathcal{B}\cup c_k)-\mathcal{R}el(\mathcal{B})$
If $\mathcal{I}mp(c_k,\mathcal{B})>0$
Add c_k to the reduction set Set
If $|Set|=m$
return $Set(1:m-1)$
Else return Set
Reduced attribute set
$Set=\{c_{s_1},c_{s_2},\cdots,c_{s_p}\}$
Measure the importance of each attribute
Output

Highlights

- Variable precision neighborhood rough set
- Weighted neighborhood dependency
- Unsupervised attribute reduction

Authors

Yi Li, Benwen Zhang, Hongming Mo, Jiancheng Hu, Yuncheng Liu, Xingqiang Tan

Correspondence

txq@scun.edu.cn

In brief

Computer science; Computational mathematics; Artificial intelligence

Li et al., 2024, iScience *27*, 111270
December 20, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2024.111270

CellPress

iScience

CellPress OPEN ACCESS

Article

Unsupervised attribute reduction based on variable precision weighted neighborhood dependency

Yi Li,[1] Benwen Zhang,[1] Hongming Mo,[1] Jiancheng Hu,[2] Yuncheng Liu,[3] and Xingqiang Tan[1,4,*]

[1]Institute of Computer Application Research, Sichuan Minzu College, Kangding 626001, China
[2]College of Applied Mathematics, Chengdu University of Information Technology, Chengdu 610225, China
[3]School of Mathematics, Southwest Minzu University, Chengdu 610041, China
[4]Lead contact
*Correspondence: txq@scun.edu.cn

https://doi.org/10.1016/j.isci.2024.111270

SUMMARY

Neighborhood rough set (NRS) have been successfully applied to attribute reduction (AR). However, most current methods of AR based on NRS are supervised or semi-supervised. This limits their ability to process data without decision information. When granulating data samples, NRS considers only the number of samples within the neighborhood radius. It does not consider distribution information between samples, which can result in the loss of original data information. To address the aforementioned issue, we propose an unsupervised attribute reduction (UAR) strategy based on variable precision weighted neighborhood dependency (VPWND) (UAR_VPWND). We compare algorithm UAR_VPWND to existing classical UAR algorithms using public datasets. The experimental results show that algorithm UAR_VPWND can select fewer attributes to maintain or improve the performance of clustering learning algorithms.

INTRODUCTION

With the development of sensor technologies, people are able to comprehensively and multi-perspectively describe things. This results in a large amount of complex and high-dimensional data. These data can assist organizations in making science-based decisions, reducing costs, and creating value. However, these data are not only massive and high-dimensional but also contain significant noise. This noise significantly impacts knowledge discovery. Hence, performing AR on data is crucial. AR (also known as feature selection[1,2]) has the primary goal of eliminating unimportant or redundant attributes in data. It can reduce data dimensionality to maintain or improve the learning performance of the original data[3,4] AR has emerged as one of the most effective methods for handling high-dimensional data. The current AR methods can be divided into three main categories based on whether decision information is present in the data. They are supervised methods, semi-supervised methods, and unsupervised methods.[5] For supervised and semi-supervised methods, we must know the full or partial decision information of data in advance. In real life, obtaining the decision information of data is exceptionally difficult. Therefore, how to identify the optimal attribute subset from high-dimensional data in the absence of decision information is of great importance. And, it has attracted numerous researchers to continuously explore unsupervised attribute reduction (UAR).

Rough set theory (RST), proposed by Pawlak,[6] is a mathematical tool for information processing. It has been widely applied in a variety of fields, including AR,[7,8] knowledge discovery,[9] machine learning,[10] and image recognition. Classical RST is based on the concept of equivalence relations to perform granular partitioning of the samples. It defines upper and lower approximation sets and other related concepts through these granules to describe the uncertainty of data. Eventually, AR and generation of decision rules are achieved. However, classical RST can only handle nominal data. When dealing with numerical data, it is often necessary to discretize the data in advance.[11,12] It inevitably leads to the loss of original data information. To address this practical issue, researchers have extended classical RST to develop a more widely applicable version of RST. Neighborhood rough set (NRS) is one of the extended models,[13] which utilizes the neighborhood relation instead of the strict equivalence relation found in classical RST. It avoids information loss and expands the applicability of RST. NRS is currently a research hotspot in the field of RST.

NRS model has been successfully applied to AR,[14–17] multi-label AR,[18–20] data classification,[21,22] gene selection,[23–25] incremental learning,[26] and outlier detection,[27] etc. It holds promising research value and practical applications. For example: regarding the lower approximation in NRS, which only permits strictly containing samples. This strict partition condition reduces the tolerance of the NRS to noise. Hu et al. introduced the variable precision concept proposed by Ziarko into NRS and proposed the variable precision NRS.[13] Hu et al. proposed a method for heterogeneous attribute subset reduction based on NRS.[28] To our knowledge, most NRS AR algorithms assume that all samples within the neighborhood have the same importance when computing neighborhood dependency. The distribution information of samples within the neighborhood is ignored. In fact, the spatial distribution of each sample in the neighborhood is one of the basis for evaluating the importance of candidate attributes. For two different attribute subsets, even if their neighborhood samples are the same, if the spatial distribution of their neighborhood samples is different, they may convey different information. To address this issue, this paper presents the concept of weighted neighborhood, which takes into account the spatial distribution of neighborhood samples.

Based on the aforementioned discussion, we propose an UAR model based on VPWND. First, the average VPW dependency is calculated for all individual attribute subsets to define the relevance of the certain subset of attributes, thereby the importance is defined to indicate the importance of a candidate attribute. Next, an UAR algorithm based on VPWND is developed. Finally, the algorithm UAR_VPWND is compared and analyzed with existing classical UAR algorithms using public datasets. The experimental results show that the algorithm UAR_VPWND can select fewer attributes to maintain or improve the learning performance of the data in clustering learning algorithms. Besides, the hypothesis statistical analysis shows that the algorithm VPWND_UAR differs significantly from the majority of classical UAR algorithms. In summary, the main contributions of this study are as follows.

(1) To account for the distributional information between samples and reduce loss of information in the original data, we study weighted neighborhood and weighted neighborhood dependency.
(2) In order to deal with uncertainty and incomplete data without decision information. We use weighted neighborhood dependency as the measurement basis and integrate the advantages of variable precision NRS to design the algorithm UAR_VPWND.
(3) Experimental comparisons with other nine UAR algorithms on public datasets show that the proposed algorithm UAR_VPWND not only improves clustering performance but also reduces information loss by considering weighted attribute relationships. At the same time, statistical significance analysis verifies the effectiveness of the algorithm.

The rest of this paper is organized as follows. Related work presents UAR, and NRSAR. Preliminaries introduces the basic theory of NRS, and variable precision neighborhood rough set (VPNRS) UAR based on VPWND gives a VPNRS model considering conditional attribute sets, analyses the model, and proposes the algorithm UAR_VPWND. Results presents the experimental results and analyses. Finally, discussion introduces advantage and disadvantage of this paper.

Related work

In this section, we review related work, including methods for UAR, NRS AR methods, and methods of weighted.

UAR methods have gained significant attention in many research areas. This is primarily because they can identify and select relevant attributes without decision information. Similar to supervised and semi-supervised AR methods, UAR methods can be divided into three categories based on the AR strategy: filter approach, wrapper approach, and hybrid approach.[4,29] Filter approach selects the most relevant attributes based on the data itself. Their main characteristics are fast speed and strong scalability. There are two categories of UAR methods based on filter approach: univariate filter methods and multivariate filter methods.[30] In univariate filter methods, there are two prominent groups: methods based on information theory and methods based on spectral analysis.[31] Rao and Sastry proposed an unsupervised univariate filter method that ranks attributes using information theory.[31] DeLeon and Chough developed a new unsupervised spectral AR method for mixed data, called unsupervised spectral feature selection method (USFSM).[32] There are three main categories of multivariate filter methods are statistical/information methods, bio-inspired methods, and spectral/sparse learning methods. Mitra et al. proposed an AR method called feature similarity-based feature selection (FSFS) within the statistical/information-based methods category, which uses attribute similarity.[33] It is one of the most representative and most cited works. Tabakhi et al. proposed an UAR method based on ant colony optimization, called UFSACO.[34] It is one of the earliest methods based on bio-inspired concepts. Nie et al. proposed a structured optimal graph AR method based on sparse learning methods, called SOGFS.[35] This method performs AR and local structure learning simultaneously. Wrapper methods use the results of specific clustering algorithms to evaluate subsets of attribute. Its main drawback is the high computational cost. And, it can only be used with specific clustering algorithms. UAR methods based on wrapper approach are divided into three groups based on the attribute search strategy: sequential methods, bio-inspired methods, and iterative methods. In 2004, Dy and Brodey proposed an UAR method, which is considered one of the most outstanding sequential methods.[36] Dutta et al. proposed an UAR method based on bio-inspired.[36] Law et al. proposed a method that uses clustering algorithms to cluster data and perform AR simultaneously.[37] It is one of the most representative among iterative methods. To effectively utilize the advantages of both filter and wrapper methods, researchers have proposed hybrid methods. In filter stage, hybrid approach performs attribute ranking or selection based on measurements of intrinsic attributes of data. In wrapper stage, it evaluates the certain subsets of attribute using a specific clustering algorithm to identify the optimal attributes. Hybrid approach can be divided into two categories: ranking-based methods and non-ranking-based methods. Solorio-Fernandez et al. proposed a ranking-based hybrid UAR method.[38] Hruschka et al. proposed a non-ranking based hybrid UAR method.[39] The method combines k-Means clustering and Bayesian filtering.

The aforementioned UAR methods cannot handle uncertainty and incomplete data. Rough sets are a mathematical tool and method for dealing with uncertainty and incomplete data. It is widely used for high-dimensional data AR. The traditional rough set model is only applicable to nominal data. NRS model is an extension of the classical rough set model, due to the introduction of the concepts of neighborhood granularity and granularity space, which can be used to deal with numerical data. At the same time, do not discretize the process, reducing the loss of original data information. Researchers have done a great deal of work on NRS AR method. Wang et al. developed an NRS-based AR algorithm using self-information.[40] Wan et al. proposed a novel hybrid AR method considering attribute interaction in

NRS.[14] Chen et al. proposed an AR method for imbalanced data based on NRS.[41] Yang et al. proposed a method for AR based on NRS and distance metric learning.[42]

In the aforementioned studies on NRS AR, researchers did not account for the weighted of attributes, as each type of attributes has the different impact on the outcome. By considering the internal correlations between conditional attributes and decisions in advance, it becomes more feasible to select attributes with high correlation and dependency. Therefore, by using a weighted method to assign different weights to attributes, data can be processed more efficiently and accurately. Guo et al. used decision trees and proposed an adaptive weighted generalized multi-granulation interval-valued decision-theoretic rough sets model.[43] Tsang et al. used kernel-based fuzzy rough sets to define a weighted Parzen function, and proposed a new KNN classification algorithm based on the weighted Parzen function to improve classification accuracy.[44] Vluymans et al. proposed new weight selection strategies for ordered weighted average based fuzzy rough sets.[45] The weighting methods primarily include subjective weighting and objective weighting. Subjective weighted methods rely on human subjectivity for determining the weights of attributes, and the involvement of relevant experts is required when using these methods. Objective weighting methods can obtain weighted values by computing the original data, which are widely used. Recently, scholars have proposed some new and novel methods. Li et al. proposed an AR method based on improved Fuzzy C-Means with principle of refined justifiable granularity.[46] Liu et al. proposed a parameter-free multi-granularity AR method.[47] Li et al. proposed an interval dominance-based AR method for interval-valued ordered data.[48] Zhang et al. proposed an a possibilistic information fusion-based UAR method using information quality measures.[49] Li et al. proposed a double-quantitative AR method for multi-granularity ordered decision systems.[50] At the same time, some work has been done on attribute reduction based on machine learning. Sheikhpour at al. use generalized uncorrelated constraint for hessian-based semi-supervised AR.[51] Berahmand at al. use dual autoencoders for semi-supervised deep attribute clustering.[52]

The paper takes UAR, NRS AR, and etc. as the research basis to study UAR based on VPWND.

Preliminaries

In this section, we will review some basic definitions and notations regarding NRS, VPNRS, and VPWN. They will be used in subsequent sections of this paper.

NRS

Definition 1. Information System(IS) is a quadruple, i.e., *IS* = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$*. Where* $\mathcal{U} = \{x_1, x_2, \cdots, x_n\}$ *is a nonempty finite set of samples, called the universe;* $\mathcal{A}$ *is also a nonempty finite set, called the attribute set;* $\mathcal{V} = \cup \mathcal{V}_a (\forall a \in \mathcal{A})$*,* $\mathcal{V}_a$ *is the value universe of a;* $\mathcal{F} : \mathcal{U} \times \mathcal{A} \rightarrow \mathcal{V}$ *is an information function, which satisfies* $\forall x_i \in \mathcal{U}$ *and* $a \in \mathcal{A}$*, there exists* $\mathcal{F}(x_i, a) \in \mathcal{V}_a$*. For decision information system(DIS),* $\mathcal{A}$ *can be divided into two mutually disjoint subsets: the conditional attribute set* $\mathcal{C} = \{c_1, c_2, \cdots, c_m\}$ *and the decision attribute set* $\mathcal{D}$*,* $\mathcal{C} \cap \mathcal{D} = \varnothing, \mathcal{C} \cup \mathcal{D} = \mathcal{A}$*. DIS can be can be briefly written as DIS* = $(\mathcal{U}, \mathcal{C} \cup \mathcal{D}, \mathcal{V}, \mathcal{F})$.

Based on the *IS* = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$, we introduce the distance function Δ and neighborhood radius ε to measure the distinguishability of samples on the attribute set.

Definition 2. Given IS = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$*, for any nonempty subsets of conditional attributes* $\mathcal{B} \subseteq \mathcal{C}$*,* $\mathcal{B} = \{c_{k_1}, c_{k_2}, \cdots, c_{k_m}\}$ $(1 \le h \le m)$*,* $\forall x_i, x_j, x_k \in \mathcal{U}$ $(1 \le i,j,k \le m)$*, the distance function* $\Delta_{\mathcal{B}}(x_i, x_j)$ *needs to satisfy:*

(1) $\Delta_{\mathcal{B}}(x_i, x_j) \ge 0$ (*Non-negativity*);
(2) $\Delta_{\mathcal{B}}(x_i, x_j) = \Delta_{\mathcal{B}}(x_j, x_i)$ (*symmetry*);
(3) $\Delta_{\mathcal{B}}(x_i, x_k) \le \Delta_{\mathcal{B}}(x_i, x_j) + \Delta_{\mathcal{B}}(x_j, x_k)$ (*Triangular inequality*).

$\Delta_{\mathcal{B}}(x_i, x_j)$ *calculates the distance between* x_i *and* x_j *under* $\mathcal{B}$*. This paper adopts the Euclidean distance function from,*[13] *which is:*

$$\Delta_{\mathcal{B}}(x_i, x_j) = \left(\sum_{l=1}^{m} \left(\mathcal{F}(x_i, c_{k_l}) - \mathcal{F}(x_j, c_{k_l}) \right)^2 \right)^{\frac{1}{2}}. \quad \text{(Equation 1)}$$

Where $\mathcal{F}(x_i, c_{k_l})$ represent the attribute value of sample x_i under the attribute $c_{k_l} (1 \le l \le m)$; $\mathcal{F}(x_j, c_{k_l})$ represent the attribute value of sample x_j under the attribute c_{k_l}.

Based on the Euclidean distance function, we introduce the neighborhood radius ε ($\varepsilon \ge 0$) to granulate the universe forming neighborhood relations and neighborhood classes. Thus, the neighborhood information system (*NIS*) based on samples is constructed. Next, we present the definition of $\varepsilon -$ neighborhood.

Definition 3. Given IS = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$*, for* $\forall x_i \in \mathcal{U}$*,* $\mathcal{B} \subseteq \mathcal{C}$*, and* $\varepsilon \ge 0$*, the* $\varepsilon -$ *neighborhood of* x_i *on* $\mathcal{B}$ *is defined as*[13,53,54]

$$n^{\varepsilon}_{\mathcal{B}}(x_i) = \{x_j | \mathcal{D}_{\mathcal{B}}(x_i, x_j) \le \varepsilon, x_j \in \mathcal{U}\}. \quad \text{(Equation 2)}$$

$n^{\varepsilon}_{\mathcal{B}}(x_i)$ contains data samples in $\mathcal{U}$ whose distance from the x_i is no more than ε under $\mathcal{B}$.

Definition 4. Given IS = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$*, for* $\forall \mathcal{B} \subseteq \mathcal{C}$*,* $\varepsilon \ge 0$*, and* $\forall x_i, x_j \in \mathcal{U}$*. Then, the neighborhood relation in U determined by B can be defined as*[13,53,54]

$$nr_{\mathcal{B}} = \{(x_i, x_j) | x_j \in n^{\varepsilon}_{\mathcal{B}}(x_i)\}. \quad \text{(Equation 3)}$$

From the definition 4, which is clear that the neighborhood relation among the samples in the $\mathcal{U}$ necessarily satisfies reflexivity and symmetry, and it is a class of similarity relations. In the NIS, when $\varepsilon = 0$, it can be said that $nr_{\mathcal{B}}$ is the finest granularity on $\mathcal{U}$. It is actually a degenerate equivalence relation, which is applicable for nominal data. When $\varepsilon > 0$, $nr_{\mathcal{B}}$ is a coarse equivalence relation on $\mathcal{U}$, which is applicable for numerical data. It is clear that the neighborhood relation actually describes the similarity or indistinguishability between samples in the $\mathcal{U}$.

Next, we will give the property of neighborhood.

Property 1. Given IS = $(\mathcal{U},\mathcal{A},\mathcal{V},\mathcal{F})$*, for* $\forall x_i, x_j \in \mathcal{U}$*, and* $\forall \mathcal{P}, \mathcal{Q} \subseteq \mathcal{C}$*. From* definition 3 *and* definition 4*, the* $\varepsilon -$ *neighborhood and neighborhood relationship have the following properties.*[13,53,54]

(1) $x_i \in n^{\varepsilon}_{\mathcal{B}}(x_i) \Rightarrow n^{\varepsilon}_{\mathcal{B}}(x_i) \ne \varnothing$;
(2) $\forall \mathcal{P} \subseteq \mathcal{Q} \Rightarrow n^{\varepsilon}_{\mathcal{P}}(x_i) \supseteq n^{\varepsilon}_{\mathcal{Q}}(x_i)$;
(3) $\forall 0 \le \varepsilon_1 \le \varepsilon_2 \Rightarrow n^{\varepsilon_1}_{\mathcal{P}}(x_i) \subseteq n^{\varepsilon_2}_{\mathcal{P}}$.

(4) $1 \leq \left|n_{\mathcal{B}}^{\varepsilon}(x_i)\right| \leq \mathcal{U}$;
(5) $\cup_{i=1}^{n} n_{\mathcal{B}}^{\varepsilon}(x_i) = \mathcal{U}$;
(6) $\forall \mathcal{P} \subseteq \mathcal{Q} \Rightarrow nr_{\mathcal{P}} \supseteq nr_{\mathcal{Q}}$.

Definition 5. Given IS $= (\mathcal{U}, \mathcal{A}, \mathcal{V}, \mathcal{F})$, *for* $\forall \mathcal{B} \subseteq \mathcal{C}$, $\varepsilon \geq 0$, $nr_{\mathcal{B}}$ *is the* $\mathcal{B} - \varepsilon$ *neighborhood relationship induced on* $\mathcal{U}$ *by* $\mathcal{B}$. $\mathcal{NR}_{\mathcal{C}} = \{nr_{\mathcal{B}} : \mathcal{B} \subseteq \mathcal{C}\}$ *represents all neighborhood relations on the* $\mathcal{U}$. *The quadruple NIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F})$ *is a NIS. Similarly, the neighborhooddec decision information system(NDIS) can be represented as NDIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{B}} \cup \mathcal{D}, \mathcal{V}, \mathcal{F})$.[13,53,54]

VPNRS

Definition 6. Given NIS $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F})$, *for* $\forall \mathcal{X} \subseteq \mathcal{U}$, $nr_{\mathcal{B}} \in \mathcal{NR}_{\mathcal{C}}$, *and introduced variable precision parameter* $\beta(0.5 \leq \beta \leq 1)$. *Then, the variable precision neighborhood information (VPNIS) can be represented as VPNIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$. *Similarly, the variable precision neighborhood decision information system(VPNDIS) can be represented as VPNDIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}} \cup \mathcal{D}, \mathcal{V}, \mathcal{F}, \beta)$.[13]

In the VPNIS, variable precision upper and lower approximation sets can be defined.

Definition 7. Given VPNIS $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, *for* $\forall \mathcal{X} \subseteq \mathcal{U}$, $nr_{\mathcal{B}} \in \mathcal{NR}_{\mathcal{C}}$. *The variable precision upper approximation sets* $\overline{nr_{\mathcal{B}}}(\mathcal{X})$ *and variable precision lower approximation sets* $\underline{nr_{\mathcal{B}}}(\mathcal{X})$ *of* $\mathcal{X}$ *based on the neighborhood relation* $nr_{\mathcal{B}}$ *are defined as*[13]

$$\overline{nr_{\mathcal{B}}^{\beta}}(\mathcal{X}) = \left\{x \left| \frac{\left|n_{\mathcal{B}}^{\varepsilon}(x) \cap \mathcal{X}\right|}{\left|n_{\mathcal{B}}^{\varepsilon}(x)\right|} \geq 1 - \beta, x \in \mathcal{U} \right.\right\}, \quad \text{(Equation 4)}$$

$$\underline{nr_{\mathcal{B}}^{\beta}}(\mathcal{X}) = \left\{x \left| \frac{\left|n_{\mathcal{B}}^{\varepsilon}(x) \cap \mathcal{X}\right|}{\left|n_{\mathcal{B}}^{\varepsilon}(x)\right|} \geq \beta, x \in \mathcal{U} \right.\right\}. \quad \text{(Equation 5)}$$

The NRS relaxes the strict definition of the positive region by introducing variable precision parameter β. The value of β is inversely related to the size of the positive region, i.e., the smaller the β, the larger the positive region. Therefore, VPNRS have the certain tolerance to data inconsistency and exhibit good noise resistance.

VPWN

Definition 8. Given VPNIS $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$. *Then, the weighted neighborhood of* $x_i \in \mathcal{U}$ *on* $\mathcal{B} \subseteq \mathcal{C}$ *can be defined as*

$$n_{\mathcal{B}}^{\prime\varepsilon} = \left\{w \times x_j \middle| \Delta(x_i, x_j) \leq \varepsilon, w = 1 - \Delta_{\mathcal{B}}(x_i, x_j), x_j \in \mathcal{U}\right\}. \quad \text{(Equation 6)}$$

In order to make the expression for calculating the weighted neighborhood in the subsequent steps clearer and more intuitive, the weighted neighborhood object $w \times x_j$ is simplified to wx_j.

From Equation 6, it can be observed that when the x_i is relatively far from the x_j within its neighborhood, the weight of the x_j is smaller, and its contribution to the classification is also smaller. Conversely, it is larger.

Based on definition 4 and definition 8, the weighted neighborhood relation can be defined.

Definition 9. Given VPNIS $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, *for* $\forall \mathcal{B} \subseteq \mathcal{C}, \varepsilon \geq 0$ *and* $\forall x_i, x_j \in \mathcal{U}$. *Then, the weighted relation on* $\mathcal{U}$ *determined by* $\mathcal{B}$ *can be defined as*

$$nr'_{\mathcal{B}}(x_i) = \left\{(x_i, wx_j) \middle| wx_j \in n_{\mathcal{B}}^{\prime\varepsilon}\right\}. \quad \text{(Equation 7)}$$

Based on definition 7 and definition 9, we can define the variable precision neighborhood upper and lower approximation sets.

Definition 10. Given VPNIS $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, *for* $\forall \mathcal{X} \subseteq \mathcal{U}$, $\forall \mathcal{B} \subseteq \mathcal{C}$, $n_{\mathcal{B}}^{\prime\varepsilon}$, *and* $nr'_{\mathcal{B}}$. *Then, the variable precision neighborhood upper approximation sets* $\overline{nr'_{\mathcal{B}}}(\mathcal{X})$ *and the variable precision neighborhood lower approximation sets* $\underline{nr'_{\mathcal{B}}}(\mathcal{X})$ *based on the weighted neighborhood relation* $nr'_{\mathcal{B}}$ *are defined as*

$$\overline{nr'_{\mathcal{B}}}(\mathcal{X}) = \left\{x_i \left| \frac{\left|n_{\mathcal{B}}^{\prime\varepsilon}(x_i) \cap \mathcal{X}\right|}{\left|n_{\mathcal{B}}^{\prime\varepsilon}(x)\right|} \geq 1 - \beta, x_i \in \mathcal{U} \right.\right\}, \quad \text{(Equation 8)}$$

$$\underline{nr'_{\mathcal{B}}}(\mathcal{X}) = \left\{x_i \left| \frac{\left|n'_{\mathcal{B}}(x_i) \cap \mathcal{X}\right|}{\left|n'_{\mathcal{B}}(x)\right|} \geq \beta, x_i \in \mathcal{U} \right.\right\}. \quad \text{(Equation 9)}$$

Then, the $\overline{nr'_{\mathcal{B}}}(X)$ and $\underline{nr'_{\mathcal{B}}}(X)$ based on $nr'_{\mathcal{B}}$ also can be further defined as

$$\overline{nr'_{\mathcal{B}}}(\mathcal{X}) = \left\{x_i \left| \frac{\left|n_{\mathcal{B}}^{\varepsilon}(x_i) \cap \mathcal{X}\right|}{\left|n_{\mathcal{B}}^{\varepsilon}(x_i)\right|} \times \frac{\sum_{\mathbf{m} \in n_{\mathcal{B}}^{\varepsilon}(x_i) \cap \mathcal{X}} (1 - \Delta_{\mathcal{B}}(x_i, \mathbf{m}))}{\sum_{\mathbf{n} \in n_{\mathcal{B}}^{\varepsilon}(x_i)} (1 - \Delta_{\mathcal{B}}(x_i, \mathbf{n}))} \geq 1 - \beta, x_i \in \mathcal{U} \right.\right\}, \quad \text{(Equation 10)}$$

$$\underline{nr'_{\mathcal{B}}}(\mathcal{X}) = \left\{x_i \left| \frac{\left|n_{\mathcal{B}}^{\varepsilon}(x_i) \cap \mathcal{X}\right|}{\left|n_{\mathcal{B}}^{\varepsilon}(x_i)\right|} \times \frac{\sum_{\mathbf{m} \in n_{\mathcal{B}}^{\varepsilon}(x_i) \cap \mathcal{X}} (1 - \Delta_{\mathcal{B}}(x_i, \mathbf{m}))}{\sum_{\mathbf{n} \in n_{\mathcal{B}}^{\varepsilon}(x_i)} (1 - \Delta_{\mathcal{B}}(x_i, \mathbf{n}))} \geq \beta, x_i \in \mathcal{U} \right.\right\}. \quad \text{(Equation 11)}$$

UAR based on VPWND

In this section, a new VPNRS selection model that considers conditional attribute sets is first developed. Next, an UAR model based on VPWND is proposed. Finally, we design the algorithm VPWND_UAR and provide a specific example to illustrate it.

VPNRS model considering conditional attributes

Given *VPNIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, where $\mathcal{U} = \{x_1, x_2, \cdots, x_n\}$, $\mathcal{C} = \{c_1, c_2, \cdots, c_m\}$. Let $\mathcal{B} = \{c_{k_1}, c_{k_2}, \cdots, c_{k_h}\}(1 \leq h \leq m) \subseteq \mathcal{C}$, For $\forall x_i, x_j \in \mathcal{U}$, the weighted relation $nr'_{\mathcal{B}}$ under $\mathcal{B}$ can be expressed as

$$r'^{\mathcal{B}}_{ij} = \left\{ \begin{array}{c} w = 1 - \Delta_{\mathcal{B}}(x_i, x_j), \Delta_{\mathcal{B}}(x_i, x_j) \leq \varepsilon; \\ 0, \Delta_{\mathcal{B}}(x_i, x_j) > \varepsilon. \end{array} \right\} \quad \text{(Equation 12)}$$

From Equation 12, it can be seen that when the Euclidean distance between x_i and x_j is less than or equal to the ε, we consider that x_i and x_j are indistinguishable under $\mathcal{B}$ and assign a weight value of w between them. In traditional NRS, when x_i and x_j are indistinguishable, we use 1 to denote the neighborhood relationship between them.[14,15] It does not take into account the distributional information between samples, which may lead to loss of information in the original data. To address this, we use w to denote the neighborhood relationship between x_i and x_j. $\Delta_{\mathcal{B}}(x_i, x_j)$ is smaller, w is larger; $\Delta_{\mathcal{B}}(x_i, x_j)$ is larger, w is smaller. To a certain extent, it can prevent the loss of information in the original data. When the Euclidean distance between x_i and x_j is greater than the ε, x_i and x_j are considered distinguishable under $\mathcal{B}$. This approach reflects people's tolerance for numerical data noise. $nr'_{\mathcal{B}}$ is the weighted relation on $\mathcal{U}$ w.r.t. $\mathcal{B}$. Then, the weighted relation matrix can be expressed as

$$\mathcal{M}'\left(nr'_{\mathcal{B}}\right) = \begin{pmatrix} r'^{\mathcal{B}}_{11} & r'^{\mathcal{B}}_{12} & \cdots & r'^{\mathcal{B}}_{1n} \\ r'^{\mathcal{B}}_{21} & r'^{\mathcal{B}}_{22} & \cdots & r'^{\mathcal{B}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r'^{\mathcal{B}}_{n1} & r'^{\mathcal{B}}_{n2} & \cdots & r'^{\mathcal{B}}_{nn} \end{pmatrix}. \qquad \text{(Equation 13)}$$

The weighted neighborhood $n'^{\varepsilon}_{\mathcal{B}}$ of x_i can also be represented as a vector, i.e., $n'^{\varepsilon}_{\mathcal{B}}(x_i) = (r'^{\mathcal{B}}_{i1}, r'^{\mathcal{B}}_{i2}, \cdots, r'^{\mathcal{B}}_{in})(0 \le r'^{\mathcal{B}}_{i2} \le 1)$.

Next, we define the neighborhood radius parameter ε.

Definition 11. *Given VPNIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$,*for* $\forall \mathcal{B} \subseteq \mathcal{C}$, *let* $\mathcal{B} = \{c_{k_1}, c_{k_2}, \cdots, c_{k_h}\}(0 \le h \le n)$. *Then, the equation for calculating the neighborhood radius is defined as*

$$\varepsilon = \frac{\sum_{k=1}^{h} std\left(c_{k_h}\right)}{h}. \qquad \text{(Equation 14)}$$

Where $std(c_{k_h})$ is the standard deviation of the attribute values of c_{k_h}, and h is the number of individual attribute subset in B.

Definition 12. *The family of neighborhood granules generated by* $nr_{\mathcal{B}}$ *is defined as*

$$\mathcal{G}(nr_{\mathcal{B}}) = \left\{n^{\varepsilon}_{\mathcal{B}}(x_1), n^{\varepsilon}_{\mathcal{B}}(x_2), \cdots, n^{\varepsilon}_{\mathcal{B}}(x_n)\right\}. \qquad \text{(Equation 15)}$$

Where $n^{\varepsilon}_{\mathcal{B}}(x_i)$ is the set of neighborhood granules containing x_i induced by $nr_{\mathcal{B}}$. Without causing confusion, $\mathcal{B}$ can also be used to replace $nr_{\mathcal{B}}$, i.e., $\mathcal{G}(nr_{\mathcal{B}})$ can be simply written as $\mathcal{G}(\mathcal{B})$.

Based on definition 9 and definition 12. We will next provide the definition of the family of the weighted neighborhood granules.

Definition 13. *The family of the weighted neighborhood granules generated by* $nr'_{\mathcal{B}}$ *is defined as*

$$\mathcal{G}'\left(nr'_{\mathcal{B}}\right) = \left\{n'^{\varepsilon}_{\mathcal{B}}(x_1), n'^{\varepsilon}_{\mathcal{B}}(x_2), \cdots, n'^{\varepsilon}_{\mathcal{B}}(x_n)\right\}. \qquad \text{(Equation 16)}$$

Without causing confusion, $\mathcal{B}'$ can also be used to replace $nr'_{\mathcal{B}}$, i.e., $\mathcal{G}'(nr'_{\mathcal{B}})$ can be simply written as $\mathcal{G}'(\mathcal{B}')$.

In the VPNIS, the upper and lower approximation sets of the variable precision neighborhood information granules are proposed for the classification of the decision attribute. Based on this, this paper will introduce the concept of VPW neighborhood upper and lower approximation sets for the classification of conditional attributes.

Definition 14. *Given VPNIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, *for* $\forall \mathcal{P}, \mathcal{Q} \subseteq \mathcal{C}$, $\forall x_i, x_j \in \mathcal{U}$. *Let* $\mathcal{G}(\mathcal{P}) = \{n^{\varepsilon}_{\mathcal{P}}(x_1), n^{\varepsilon}_{\mathcal{P}}(x_2), \cdots, n^{\varepsilon}_{\mathcal{P}}(x_n)\}$, $\mathcal{G}(\mathcal{Q}) = \{n^{\varepsilon}_{\mathcal{Q}}(x_1), n^{\varepsilon}_{\mathcal{Q}}(x_2), \cdots, n^{\varepsilon}_{\mathcal{Q}}(x_n)\}$. $\forall n^{\varepsilon}_{\mathcal{P}}(x_i) \in \mathcal{G}(\mathcal{P})$, $\forall n^{\varepsilon}_{\mathcal{Q}}(x_j) \in \mathcal{G}(\mathcal{Q})$. *Then, the variable precision neighborhood upper and lower approximation sets of* $n^{\varepsilon}_{\mathcal{Q}}(x_j)$ *with respect to* $nr_{\mathcal{P}}$ *are defined as*

$$\overline{nr^{\beta}_{\mathcal{P}}}\left(n^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \left\{x_i \middle| \frac{\left|n^{\varepsilon}_{\mathcal{P}}(x_i) \bigcap n^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n^{\varepsilon}_{\mathcal{P}}(x_i)\right|} \ge 1 - \beta, x_i \in \mathcal{U}\right\}, \qquad \text{(Equation 17)}$$

$$\underline{nr^{\beta}_{\mathcal{P}}}\left(n^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \left\{x_i \middle| \frac{\left|n^{\varepsilon}_{\mathcal{P}}(x_i) \bigcap n^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n^{\varepsilon}_{\mathcal{P}}(x_i)\right|} \ge \beta, x_i \in \mathcal{U}\right\}. \qquad \text{(Equation 18)}$$

The upper and lower approximations sets of the VPNRS are based on the β tolerance partition, which allows the certain degree of misclassification. The size of β determines the sample coverage of the upper and lower approximation sets.

Based on definition 13 and definition 14, the VPW neighborhood upper and lower approximation sets can be defined considering conditional attributes.

Definition 15. *Given VPNIS* $= (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, *for* $\forall \mathcal{P}, \mathcal{Q} \subseteq \mathcal{C}$, $\forall x_i, x_j \in \mathcal{U}$. *Let* $\mathcal{G}'(\mathcal{P}') = \{n'^{\varepsilon}_{\mathcal{P}}(x_1), n'^{\varepsilon}_{\mathcal{P}}(x_2), \cdots, n'^{\varepsilon}_{\mathcal{P}}(x_n)\}$, $\mathcal{G}'(\mathcal{Q}') = \{n'^{\varepsilon}_{\mathcal{Q}}(x_1), n'^{\varepsilon}_{\mathcal{Q}}(x_2), \cdots, n'^{\varepsilon}_{\mathcal{Q}}(x_n)\}$. $\forall n'^{\varepsilon}_{\mathcal{P}}(x_i) \in \mathcal{G}'(\mathcal{P}')$, *and* $\forall n'^{\varepsilon}_{\mathcal{Q}}(x_j) \in \mathcal{G}'(\mathcal{Q}')$. *Then, the VPW neighborhood upper and lower approximation sets of* $n'_{\mathcal{Q}}(x_j)$ *w.r.t.* $nr'_{\mathcal{P}}$ *are defined as*

$$\overline{nr'^{\beta}_{\mathcal{P}}}\left(n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \left\{x_i \middle| \frac{\left|n'^{\varepsilon}_{\mathcal{P}}(x_i) \bigcap n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n'^{\varepsilon}_{\mathcal{P}}(x_i)\right|} \ge 1 - \beta, x_i \in \mathcal{U}\right\}, \qquad \text{(Equation 19)}$$

$$\underline{nr'^{\beta}_{\mathcal{P}}}\left(n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \left\{x_i \middle| \frac{\left|n'^{\varepsilon}_{\mathcal{P}}(x_i) \bigcap n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n'^{\varepsilon}_{\mathcal{P}}(x_i)\right|} \ge \beta, x_i \in \mathcal{U}\right\}. \qquad \text{(Equation 20)}$$

From definition 10, *we can further derive the VPW neighborhood upper and lower approximation sets, as follows*

$$\begin{aligned} \overline{nr'^{\beta}_{\mathcal{P}}}\left(n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \Bigg\{x_i \Bigg| & \frac{\left|n^{\varepsilon}_{\mathcal{P}}(x_i) \cap n^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n^{\varepsilon}_{\mathcal{P}}(x)\right|} \\ & \times \frac{\sum_{\boldsymbol{m}' \in n^{\varepsilon}_{\mathcal{P}}(x) \cap n^{\varepsilon}_{\mathcal{Q}}(x_j)} (1 - \Delta_{\mathcal{P}}(x_i, \boldsymbol{m}'))}{\sum_{\boldsymbol{n}' \in n^{\varepsilon}_{\mathcal{P}}(x)} (1 - \Delta_{\mathcal{P}}(x_i, \boldsymbol{n}'))} \\ & \ge 1 - \beta, x_i \in \mathcal{U}\Bigg\}, \end{aligned} \qquad \text{(Equation 21)}$$

$$\begin{aligned} \underline{nr'^{\beta}_{\mathcal{P}}}\left(n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right) = \Bigg\{x_i \Bigg| & \frac{\left|n^{\varepsilon}_{\mathcal{P}}(x_i) \cap n^{\varepsilon}_{\mathcal{Q}}(x_j)\right|}{\left|n^{\varepsilon}_{\mathcal{P}}(x)\right|} \\ & \times \frac{\sum_{\boldsymbol{m}' \in n^{\varepsilon}_{\mathcal{P}}(x) \cap n^{\varepsilon}_{\mathcal{Q}}(x_j)} (1 - \Delta_{\mathcal{P}}(x_i, \boldsymbol{m}'))}{\sum_{\boldsymbol{n}' \in n^{\varepsilon}_{\mathcal{P}}(x)} (1 - \Delta_{\mathcal{P}}(x_i, \boldsymbol{n}'))} \\ & \ge \beta, x_i \in \mathcal{U}\Bigg\}. \end{aligned} \qquad \text{(Equation 22)}$$

Next, we will define the VPWND. Based on this, the concept of the VPW neighborhood positive region is first introduced.

Definition 16. *The positive region of the VPW neighborhood of* $\mathcal{Q}$ *w.r.t.* $\mathcal{P}$ *can be defined as*

$$VPNPOS'^{\beta}_{\mathcal{P}}(\mathcal{Q}) = \bigcup_{n'^{\varepsilon}_{\mathcal{Q}}(x_j) \in \mathcal{G}'(\mathcal{Q}')} \underline{nr'^{\beta}_{\mathcal{P}}}\left(n'^{\varepsilon}_{\mathcal{Q}}(x_j)\right), \qquad \text{(Equation 23)}$$

The VPW neighborhood positive region defines the union of the VPW neighborhood lower approximation sets of $\mathcal{Q}$ with respect to $\mathcal{P}$. The size of it can reflect the relevance of the conditional attributes. Next, we will define the VPWND.

Definition 17. *The VPWND of $\mathcal{Q}$ w.r.t. $\mathcal{P}$ can be defined as*

$$\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q}) = \frac{\left|VPNPOS'^{\beta}_{\mathcal{P}}(\mathcal{Q})\right|}{|\mathcal{U}|}. \qquad \text{(Equation 24)}$$

Clearly, $0 \le \gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q}) \le 1$. This shows the relevance between conditional attributes. When $\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q}) = 1$, it is said that $\mathcal{Q}$ is fully dependent on $\mathcal{P}$; When $\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q}) = 0$, it is said that $\mathcal{Q}$ is completely independent of $\mathcal{P}$. The goal of this paper is to find the subset of attributes with the maximum dependency. under the circumstances, the attributes within the subset of attributes have higher relevance to each other.

Given $VPNIS = (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, for $\mathcal{P}$, $\mathcal{Q} \subseteq \mathcal{C}$, $\forall n'^{\varepsilon}_{\mathcal{Q}}(x_j) \in \mathcal{G}'(\mathcal{Q}')$. From definition 15, it is known that $\underline{n\gamma'^{\beta}_{\mathcal{P}}}$ is non-monotonic. Then, $\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q})$ is also non-monotonic.

$\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q})$ mainly focuses on the relevance between the $\mathcal{P}$ and $\mathcal{Q}$. In order to compute $\gamma'^{\beta}_{\mathcal{P}}(\mathcal{Q})$, We should not attempt to examine all subsets within $\mathcal{C}$. Since there are $2^{|\mathcal{C}|}$ subsets of attributes in $\mathcal{C}$, we will obtain $2^{2|\mathcal{C}|}$ VPWND. This will result in its time complexity being exponential. Therefore, this paper only considers the VPWND w.r.t. all individual attribute subset.

Next, we will define the relevance of the conditional attribute subset $\mathcal{B}$.

Definition 18. *The relevance of the $\mathcal{B}$ can be defined as*

$$\mathcal{R}el(\mathcal{B}) = \frac{1}{m}\sum_{k=1}^{m} \gamma'^{\beta}_{\mathcal{P}}(c_k). \qquad \text{(Equation 25)}$$

Where $0 \le \mathcal{R}el(\mathcal{B}) \le 1$, the relevance reflects the $\mathcal{B}$ and all individual condition attribute subsets. It can be used to define the importan of an attribute.

Definition 19. *If $\mathcal{R}el(\mathcal{B}) = \mathcal{R}el(\mathcal{B} - b)$, then b is said to be redundant in $\mathcal{B}$; otherwise, b is said to be necessary in $\mathcal{B}$. If every attribute in $\mathcal{B}$ is necessary, then $\mathcal{B}$ is said to be independent.*

Definition 20. *Let $\mathcal{B} \subseteq \mathcal{C}$, if $\mathcal{B}$ satisfies*

(1) $\mathcal{R}el(\mathcal{B}) = \mathcal{R}el(\mathcal{C})$;
(2) $\forall b \in \mathcal{B}, \mathcal{R}el(\mathcal{B} - b) < \mathcal{R}el(\mathcal{B})$.

Then, $\mathcal{B}$ is called a reduction of $\mathcal{C}$.

In the definition 20, the condition (1) requires that the reduction cannot decrease the distinguishing ability of the system, and the condition attributes after the reduction have the same distinguishing ability as all the condition attributes; the condition (2) requires that there are no redundant attributes after the reduction, and all attributes should be indispensable. It is formally completely consistent with the definition of supervised AR in classical rough sets. The model uses the VPWND to measure the correlation between conditional attributes. However, AR of supervised uses the decision attribute to define the importance of the conditional attributes. Therefore, they are applicable to completely different scenarios.

Relevance defines the correlation between conditional attributes Therefore, it can serve as an indicator for evaluating the importance of an attribute.

Definition 21. *$\forall b \in \mathcal{C} - \mathcal{B}$, we define the importance of b relative to $\mathcal{B}$ as*

$$\mathcal{I}mp(b, \mathcal{B}) = \mathcal{R}el(\mathcal{B} \cup b) - \mathcal{R}el(\mathcal{B}). \qquad \text{(Equation 26)}$$

Since $0 \le \mathcal{R}el(B) \le 1$ and $\mathcal{R}el(\mathcal{B}) \le \mathcal{R}el(\mathcal{B} \cup b)$, we have $0 \le \mathcal{I}mp(b,\mathcal{B}) \le 1$. If $\mathcal{I}mp(b,\mathcal{B}) = 0$, the b is considered redundant in B; otherwise, it is considered necessary.

VPWND based on AR algorithm

So far, the paper has established an unsupervised AR model based on VPWND, which called VPWND based on UAR.

The description of data samples often differ in magnitude and dimension. In order to obtain accurate data processing results and avoid the influence between different data, we need to standardize the original data.[55] The paper adopts min-max normalization. Its computational formula is as

$$F(\mathcal{F}(x_i, c_k)) = \frac{\mathcal{F}(x_i, c_k) - \min_{c_k}}{\max_{c_k} - \min_{c_k}}. \qquad \text{(Equation 27)}$$

Where $\mathcal{F}(x_i, c_k)$ represents the value of x_i under the c_k. max_{c_k} and min_{c_k} are maximum and minimum values of c_k, respectively. After normalization, the range of values for these attributes is $[0,1]$. Next, we will design the algorithm UAR_VPWND and analyze its time complexity.

Algorithm 1. the algorithm UAR_VPWND

```
Input: VPNIS = $(\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, where $|\mathcal{C}| = m$, variable precision threshold parameter β
Output: $\mathcal{S}et$
$\mathcal{S}et \leftarrow \varnothing$, $Label \leftarrow 1$, $\mathcal{B} \leftarrow \mathcal{C} - \mathcal{S}et$;
for $k \leftarrow 1$ to $m$ do
    Calculate $\mathcal{M}'(nr'(c_k))$;
end.
while Label do
    Let $\mathcal{B} = \{c_{k_1}, c_{k_2}, \cdots, c_{k_h}\}$;
    for $l \leftarrow 1$ to $h$ do
        for $s \leftarrow 1$ to $m$ do
            Calculate $\gamma'^{\beta}_{\mathcal{S}et \cup c_{k_l}}(\{c_s\})$;
        end.
        Calculate $\mathcal{R}el(\mathcal{S}et \cup \{c_{k_l}\})$;
    end.
    Select attribute $c_{k_{l'}}$, s.t. $\mathcal{R}el(\mathcal{S}et \cup \{c_{k'_l}\})$ have a maximum value;
    Calculate $\mathcal{I}mp(c_{k_{l'}}, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_{k_{l'}}\}) - \mathcal{R}el(\mathcal{S}et)$;
    if $\mathcal{I}mp(c_{k_{l'}}, \mathcal{S}et) > 0$ then
        $\mathcal{S}et \leftarrow \mathcal{S}et \cup \{c_{k_{l'}}\}$, $\mathcal{B} \leftarrow \mathcal{B} - \{c_{k_{l'}}\}$;
        $\mathcal{B} = \{c_{k_1}, c_{k_2}, \cdots, c_{k_p}\}$;
        for $j \leftarrow 1$ to $p$ do
            Calculate $\mathcal{M}'(nr'(c_j))$
        end.
        else.
            $Label \leftarrow 0$;
        end.
    end.
end.
if $|\mathcal{S}et| = = m$ then
    return $\mathcal{S}et \leftarrow \mathcal{S}et(1 : m - 1)$;
    else.
        return $\mathcal{S}et$.
    end.
end.
```

Table 1. Initial and standardized the IS

$\mathcal{U}$	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
x_1	1.4	7.5	3.9	4.6	0.4444	0.5000	1.0000	0.6250
x_2	1.5	7.6	3.2	4.5	0.5556	0.6250	0.2222	0.5000
x_3	1.0	7.9	3.3	4.2	0.0000	1.0000	0.3333	0.1250
x_4	1.6	7.6	3.6	4.2	0.6667	0.6250	0.6667	0.1250
x_5	1.2	7.1	3.0	4.1	0.2222	0.0000	0.0000	0.0000
x_6	1.9	7.3	3.8	4.9	1.0000	0.2500	0.8889	1.0000

Table 2. The VPWND between two individual attributes

$\gamma'^{\beta}_{\{c_i\}}(\{c_j\})$	c_1	c_2	c_3	c_4
c_1	1.0000	0.3333	0.3333	0.1667
c_2	0.6667	1.0000	0.1667	0.5000
c_3	0.6667	0.3333	1.0000	0.5000
c_4	0.3333	0.3333	0.6667	1.0000

The algorithm UAR_VPWND starts with the empty set. The importance of all remaining attributes is calculated each time. The algorithm UAR_VPWND selects the attribute with the highest importance and add it to the reduction set until the importance of all remaining attributes becomes 0. That is, adding any new attribute, the system's importance no longer changes. In the algorithm UAR_VPWND, the number of loops in steps $2-4$ runs m times, the number of loops in step 3 runs for $n\times n$ times, the number of loops in steps $7-12$ runs $h\times m$ times, the number of loops in steps $8-10$ runs m times, the number of loops in steps $18-20$ runs p times, and the number of loops in step 19, runs $n\times n$ times. The total number of loops in the algorithm UAR_VPWND is $m\times n\times n+h\times m+p\times n\times n$. Therefore, the time complexity of the algorithm UAR_VPWND is $O(n^2)$.

A special example

Given $VPNIS = (\mathcal{U}, \mathcal{NR}_{\mathcal{C}}, \mathcal{V}, \mathcal{F}, \beta)$, the values on the left side of Table 1 are the numerical data. Where $\mathcal{U} = \{x_1, x_2, x_3, \cdots, x_6\}$, $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$. First, we need to perform min-max normalization on the original data using Equation 27. The normalized results are shown on the right side of Table 1. Let $\mathcal{S}et = \varnothing$, $\mathcal{B} = \mathcal{C}$, $\beta = 0.75$.

$\forall c_k \in \mathcal{C}$, the weighted relation matrixs are as follows.

$$\mathcal{M}\left(nr'_{c'_1}\right) = \begin{pmatrix} 1.0000 & 0.8889 & 0.0000 & 0.7778 & 0.7778 & 0.0000 \\ 0.8889 & 1.0000 & 0.0000 & 0.8889 & 0.6667 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.7778 & 0.0000 \\ 0.7778 & 0.8889 & 0.0000 & 1.0000 & 0.0000 & 0.6667 \\ 0.7778 & 0.6667 & 0.7778 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.6667 & 0.0000 & 1.0000 \end{pmatrix},$$

$$\mathcal{M}\left(nr'_{c'_2}\right) = \begin{pmatrix} 1.0000 & 0.8750 & 0.0000 & 0.8750 & 0.0000 & 0.7500 \\ 0.8750 & 1.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.8750 & 1.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.7500 \\ 0.7500 & 0.0000 & 0.0000 & 0.0000 & 0.7500 & 1.0000 \end{pmatrix},$$

$$\mathcal{M}\left(nr'_{c'_3}\right) = \begin{pmatrix} 1.0000 & 0.0000 & 0.0000 & 0.6667 & 0.0000 & 0.8889 \\ 0.0000 & 1.0000 & 0.8889 & 0.0000 & 0.7778 & 0.0000 \\ 0.0000 & 0.8889 & 1.0000 & 0.6667 & 0.6667 & 0.0000 \\ 0.6667 & 0.0000 & 0.6667 & 1.0000 & 0.0000 & 0.7778 \\ 0.0000 & 0.7778 & 0.6667 & 0.0000 & 1.0000 & 0.0000 \\ 0.8889 & 0.0000 & 0.0000 & 0.7778 & 0.0000 & 1.0000 \end{pmatrix},$$

$$\mathcal{M}\left(nr'_{c'_4}\right) = \begin{pmatrix} 1.0000 & 0.8750 & 0.0000 & 0.0000 & 0.0000 & 0.6250 \\ 0.8750 & 1.0000 & 0.6250 & 0.6250 & 0.0000 & 0.0000 \\ 0.0000 & 0.6250 & 1.0000 & 1.0000 & 0.8750 & 0.0000 \\ 0.0000 & 0.6250 & 1.0000 & 1.0000 & 0.8750 & 0.0000 \\ 0.0000 & 0.0000 & 0.8750 & 0.8750 & 1.0000 & 0.0000 \\ 0.6250 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}.$$

The VPWND between two individual attributes are calculated from Equation 24, Which is shown in Table 2.

By definition 18, the relevance for each attribute can be calculated as follows. $\mathcal{R}el(\{c_1\}) = \frac{1}{m}\sum_{k=1}^{m}\gamma_{c_1}^{'\beta}(c_k) = \frac{1}{4}(1.0000 + 0.3333 + 0.3333 + 0.1667) \approx 0.4583$; $\mathcal{R}el(\{c_2\}) = 0.5833$; $\mathcal{R}el(\{c_3\}) = 0.6250$; $\mathcal{R}el(\{c_4\}) = 0.5833$.

Similarly, the importance of each attribute is as follows. (Since $\mathcal{S}et = \varnothing$, then $\mathcal{R}el(\mathcal{S}et) = 0$.) $\mathcal{I}mp(c_1, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_1\}) - \mathcal{R}el(\mathcal{S}et) = 0.4583 - 0.0000 = 0.4583$; $\mathcal{I}mp(c_2, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_2\}) - \mathcal{R}el(\mathcal{S}et) = 0.5833 - 0.0000 = 0.5833$; $\mathcal{I}mp(c_3, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_3\}) - \mathcal{R}el(\mathcal{S}et) = 0.6250 - 0.0000 = 0.6250$; $\mathcal{I}mp(c_4, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_4\}) - \mathcal{R}el(\mathcal{S}et) = 0.5833 - 0.0000 = 0.5833$.

Next, the attribute with the highest importance is added to $\mathcal{S}et$. That is the attribute c_3 is added to $\mathcal{S}et$, we can get $\mathcal{S}et = \{c_3\}$, $B = C - \{c_3\} = \{c_1, c_2, c_4\}$.

Similarly, we can calculate $\mathcal{I}mp(c_1, \mathcal{S}et) = \mathcal{R}el\ (\mathcal{S}et \cup \{c_2\}) - \mathcal{R}el(\mathcal{S}et) = 1.0000 - 0.6250 = 0.3750$; $\mathcal{I}mp(c_2, \mathcal{S}et) = \mathcal{R}el\ (\mathcal{S}et \cup \{c_3\}) - \mathcal{R}el(\mathcal{S}et) = 0.9583 - 0.6250 = 0.3333$; $\mathcal{I}mp(c_4, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_4\}) - \mathcal{R}el(\mathcal{S}et) = 0.0000 - 0.6250 = -0.6250$.

At this point, we can get $\mathcal{S}et = \{c_3, c_1\}, B = \{c_2, c_4\}$.

Further, we can calculate $\mathcal{I}mp(c_2, \mathcal{S}et) = \mathcal{R}el\ (\mathcal{S}et \cup \{c_2\}) - \mathcal{R}el(\mathcal{S}et) = 1.0000 - 1.0000 = 0.0000$; $\mathcal{I}mp\ (c_3, \mathcal{S}et) = \mathcal{R}el(\mathcal{S}et \cup \{c_4\}) - \mathcal{R}el(\mathcal{S}et) = 1.0000 - 1.0000 = 0.0000$.

Therefore, $\mathcal{S}et = \{c_3, c_1\}$.

RESULTS

This section evaluates the algorithm UAR_VPWND through the performance clustering tasks. It mainly includes the experimental datasets, and experimental schemes, experimental results, parameter analyses, and hypothesis testing.

Experimental datasets

From publicly available databases, 16 numerical datasets are selected for experiments. For the missing values in some datasets, the maximum method is used to fill the missing values. The datasets used for clustering are described as shown in Table 3.

Table 3. Datasets basic information

ID	Datasets	Abbreviation	Sample	Conditional attributes	Decision class
1	Dermatology	Derm	366	34	6
2	Diabetes	Diab	768	8	2
3	Ecoli	Ecoli	336	7	8
4	Glass identification	Glass	214	10	6
5	Ionosphere	Iono	351	33	2
6	Movement_libras	Move	360	90	15
7	Page_blocks	Page	5473	10	5
8	Parkinsons	Park	195	22	2
9	Sonar	Sonar	208	60	2
10	Wisconsin Breast Cancer	WBC	699	9	2
11	Wisconsin Diagnostic Breast Cancer	WDBC	569	31	2
12	Wilt	Wilt	4839	5	2
13	Wine	Wine1	178	13	3
14	winequality_white	Wine2	4898	11	7
15	winequality_red	Wine3	1599	11	6
16	Wisconsic Prognostic Breast Cancer	WPBC	198	34	2

Table 4. The average optimal number of attributes based on different UAR algorithms

Dateset	Original Attribute	FRUAR	UFRFS	HKCMI	LS	FSFS	USFSM	SPEC	FMIUFS	UAR_mSMU	UAR_VPWND
Derm	34.0	16.0	19.0	7.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
Diab	8.0	7.0	**8.0**	7.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Ecoli	7.0	6.0	**7.0**	6.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Glass	10.0	9.0	8.0	9.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
Iono	33.0	32.0	12.0	12.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Move	90.0	49.0	5.0	8.0	11.5	11.5	11.5	11.5	11.5	11.5	11.5
Page	10.0	9.0	8.0	5.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Park	22.0	18.0	8.0	8.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
Sonar	60.0	30.0	8.0	8.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
WBC	9.0	8.0	**9.0**	8.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
WDBC	31.0	14.0	9.0	12.0	3.5	3.5	3.5	3.5	3.5	3.5	3.5
Wilt	5.0	4.0	**5.0**	4.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Wine1	13.0	12.0	8.0	9.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Wine2	11.0	10.0	10.0	10.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Wine3	11.0	10.0	10.0	10.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
WPBC	34.0	24.0	8.0	9.0	14.0	14.0	14.0	14.0	14.0	14.0	14.0
Average	24.3	16.1	8.9	8.3	5.8	5.8	5.8	5.8	5.8	5.8	5.8

Experimental schemes

On the datasets listed in Table 3, we compared the algorithm UAR_VPWND with the fuzzy rough set-based UAR (FRUAR),[4] unsupervised fuzzy rough set-based feature selection (UFRFS),[56] hybrid kernel-based fuzzy complementary mutual information (HKCMI),[57] Laplacian score-based (LS),[58] feature similarity-based feature selection (FSFS),[33] unsupervised spectral feature selection method (USFSM),[32] spectral analysis-based (SPEC),[59] a novel unsupervised approach to heterogeneous feature selection based on fuzzy mutual information (FMIUFS),[60] and a possibilistic information fusion-based unsupervised feature selection method using information quality measures (UAR_mSMU).[49] Where, algorithms LS and FSFS are typically applicable to numerical data; the algorithm USFSM is applicable to mixed data. Among them, algorithms FRUAR, UFRFS, HKCMI, FMUFS, and UAR_mSMU are based on fuzzy rough set theory. Where algorithms UFRFS and UAR_mSMU are applicable to numerical data; the algorithms FRUAF, HKCMI, and FMIUFS are applicable to mixed data. For the algorithm FRUAR, which adjust the parameter λ in the range of [0.1, 3.0] with a step size of 0.1 to get optimal result. For the algorithm HKCMI, which adjust the parameter ρ in the range of [0.1 1.0] with a step size of 0.1 and another parameter $\varepsilon \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ to get the optimal result. For the algorithm FMUFS, which adjusts the parameter λ in the range of [0.1, 2.0] with a step size of 0.1 to get the optimal result. For each dataset, they can select an optimal subset of attributes.

For clustering experiments, the clustering effects of the aforementioned comparison algorithms are also evaluated by calling algorithms k-Means, and fuzzy C-Means (FCM) in MATLAB R2021a. The parameters of all clustering experiments are kept as initial values. The number of clusters is set to the number of real decision classes. The clustering algorithms return the predicted decisions for the dataset. This paper uses clustering accuracy (ACC),[49,61] and adjusted rand index (ARI)[60] to evaluate the performance of the clustering algorithms. ACC is defined as

$$ACC = \frac{\sum_{n=1}^{n} \delta(\mathcal{F}(x_i), \mathcal{F}'(x_i))}{n}. \qquad \text{(Equation 28)}$$

Where $\mathcal{F}(x_i)$ represents the true decision value of x_i, and $\mathcal{F}'(x_i)$ represents the predicted decision value of x_i. If $\mathcal{F}(x_i) = \mathcal{F}'(x_i)$, then $\delta(\mathcal{F}(x_i), \mathcal{F}'(x_i)) = 1$. Otherwise, $\delta(\mathcal{F}(x_i), \mathcal{F}'(x_i)) = 0$. The higher the ACC or AIR value, the better the performance of the clustering algorithm. Since clustering algorithms k-Means, and FCM have random, therefore, we need repeat the experiment 10 times. Finally, this paper uses the average and standard deviation of the clustering ACC as the final results.[4,49]

In the experiment, all the attribute values of datasets are normalized to the [0.0, 1.0] using the min-max normalization Equation 28. We follow the experimental setting in,[62] which sets the neighborhood size to be fixed at 5 when the algorithm LS calculates the neighbor graph matrix. For the algorithm FSFS, the "style" is set as -1. Algorithms LS, FSFS, USFSM, SPEC, and FMIUFS output a sequence of attributes arranged in descending order of attribute importance. In order to compare with the algorithm UAR_VPWND proposed in this paper, we select attribute subsequences with the same number of attributes selected by the algorithm UAR_VPWND for the clustering experiments. The algorithm UAR_VPWND has a precision threshold parameter $\beta(0.5 \leq \beta \leq 1.0)$, which is used to adjust the partitioning of the positive region and has a significant impact on the algorithm. Generally, different values of β will lead to different results. Therefore, we need to adjust the parameter β from 0.5 to 1.0, with a step size of 0.05. Through extensive experimental results show that the algorithm UAR_VPWND is usually difficult to achieve the stop condition that the importance

Table 5. The clustering ACC(%) of the k-Means algorithm on reduced dataset

Dataset	Original attribute	FRUAR	UFRFS	HKCMI	LS	FSFS	USFSM	SPEC	FMIUFS	UAR_mSMU	UAR_VPWND	β
Derm	68.42 ± 6.91	70.38 ± 6.10	63.61 ± 11.60	56.48 ± 3.74	65.49 ± 6.16	68.52 ± 6.67	70.93 ± 6.49	70.14 ± 8.33	71.75 ± 4.84	**72.92** ± **6.60**	69.15 ± 2.93	0.95
Diab	66.81 ± 0.04	65.78 ± 0.91	66.80 ± 0.00	66.60 ± 0.07	64.80 ± 4.45	65.65 ± 0.67	66.02 ± 0.62	65.70 ± 0.68	65.31 ± 1.91	66.02 ± 0.42	**67.19** ± **0.00**	0.80
Ecoli	57.95 ± 5.38	53.66 ± 5.81	**58.84** ± **3.67**	56.90 ± 6.75	40.65 ± 2.61	40.95 ± 3.45	39.40 ± 2.42	38.90 ± 1.69	40.71 ± 3.41	40.60 ± 3.15	52.59 ± 4.28	0.90
Glass	56.36 ± 7.25	59.58 ± 8.20	59.07 ± 5.39	61.31 ± 7.14	68.50 ± 6.25	67.34 ± 6.51	66.87 ± 6.62	**70.93** ± **6.74**	67.34 ± 7.64	68.55 ± 5.31	65.93 ± 6.59	0.90
Iono	71.00 ± 0.12	71.17 ± 0.12	71.08 ± 4.55	71.34 ± 0.06	77.66 ± 4.41	77.32 ± 5.05	**83.87** ± **3.31**	82.88 ± 4.35	80.57 ± 4.92	75.90 ± 3.06	82.05 ± 0.00	1.00
Move	43.47 ± 3.15	43.22 ± 2.05	31.69 ± 1.03	40.89 ± 1.68	31.69 ± 1.15	31.75 ± 1.39	32.83 ± 1.30	31.75 ± 0.94	31.94 ± 1.43	32.25 ± 1.50	**43.83** ± **3.36**	0.80
Page	46.94 ± 6.64	50.35 ± 5.46	45.71 ± 6.54	50.71 ± 5.02	81.68 ± 16.54	31.12 ± 0.62	75.59 ± 20.03	**91.14** ± **0.00**	89.05 ± 1.37	85.07 ± 12.24	91.04 ± 0.31	0.60
Park	62.77 ± 0.26	64.87 ± 1.35	74.82 ± 1.12	75.90 ± 1.05	74.77 ± 4.50	75.49 ± 3.01	72.10 ± 3.82	75.79 ± 2.20	74.87 ± 3.48	76.05 ± 1.33	**79.28** ± **2.50**	0.70
Sonar	54.86 ± 1.48	55.05 ± 2.47	56.30 ± 0.15	61.20 ± 2.89	53.37 ± 0.68	53.89 ± 0.70	53.70 ± 0.79	54.04 ± 0.61	53.46 ± 0.74	53.89 ± 0.70	**63.89** ± **0.15**	0.50
WBC	95.85 ± 0.00	95.57 ± 0.00	95.85 ± 0.00	95.99 ± 0.00	93.71 ± 0.00	93.71 ± 0.00	93.71 ± 0.00	93.71 ± 0.00	93.71 ± 0.00	93.71 ± 0.00	**95.99** ± **0.00**	0.90
WDBC	92.79 ± 0.00	91.14 ± 0.15	91.48 ± 0.09	90.46 ± 0.08	88.12 ± 0.39	85.61 ± 8.11	78.00 ± 13.28	88.08 ± 0.36	80.33 ± 12.27	77.86 ± 13.16	**93.25** ± **0.56**	0.50
Wilt	50.98 ± 0.05	51.14 ± 0.13	51.10 ± 0.17	51.06 ± 0.01	55.07 ± 0.08	54.97 ± 0.13	55.20 ± 12.90	54.95 ± 0.13	**58.99** ± **12.48**	54.79 ± 0.13	55.02 ± 0.12	0.85
Wine1	94.66 ± 0.55	**95.79** ± **0.89**	93.54 ± 0.30	94.89 ± 1.31	78.20 ± 0.24	78.20 ± 0.24	78.26 ± 0.27	78.20 ± 0.24	78.26 ± 0.27	78.26 ± 0.27	93.54 ± 1.00	0.65
Wine2	26.96 ± 1.09	**27.74** ± **1.55**	27.38 ± 2.11	27.54 ± 2.67	26.78 ± 0.79	26.65 ± 0.55	26.46 ± 0.13	26.67 ± 0.38	26.45 ± 0.14	26.20 ± 0.77	26.72 ± 1.53	0.95
Wine3	30.99 ± 1.83	31.15 ± 1.93	29.79 ± 2.01	30.01 ± 2.42	29.15 ± 2.60	29.17 ± 2.83	28.44 ± 1.69	30.97 ± 2.27	28.32 ± 2.98	28.57 ± 2.37	**32.35** ± **4.08**	1.00
WPBC	59.75 ± 0.24	57.12 ± 6.27	60.96 ± 9.21	56.36 ± 2.84	60.56 ± 2.01	61.41 ± 3.60	62.63 ± 3.23	62.47 ± 3.20	61.16 ± 2.48	63.43 ± 3.53	**66.57** ± **1.19**	0.80
Average	61.29 ± 2.19	61.48 ± 2.71	61.13 ± 3.00	61.73 ± 2.36	61.89 ± 3.30	58.86 ± 2.72	61.50 ± 4.81	63.52 ± 2.01	62.64 ± 3.77	62.14 ± 3.53	**67.40** ± **1.79**	–

of all remaining attributes is 0. Therefore, the algorithm UAR_VPWND sets the importance stop condition to 0.0001. Finally, the algorithm UAR_VPWND selects an optimal subset of attributes for each dataset. All decision attributes in all datasets will be removed before the reduction.

Experimental results

Next, we will analyze the results of the clustering experiments. Its comparison results are shown in Tables 4, 5, and 6. Table 4 shows the average optimal number of attributes for different attribute reduction algorithms. Where, bold indicates that corresponding reduction algorithm obtains the original set of conditional attributes as the result. From Table 4, we can observe that for all datasets, algorithms FRUAR, HKCMI, and UAR_VPWND are able to remove some redundant and irrelevant attributes. However, the algorithm UFRFS cannot remove any conditional attributes on certain datasets, such as datasets Diab, Ecoli, WBC, and Wilt. On all datasets, the algorithm UAR_VPWND select the number of attributes is smaller than the number of condition attributes in the original dataset. Such as, for datasets Iono, Move, Page, Park, Sonar, WDBC, Wine1, and WPBC, the algorithm UAR_VPWND select the average number of attributes of 4.0, 11.5, 1.0, 5.0, 6.0, 3.5, 3.0, and 14.0, respectively. This is significantly less than the original number of conditional attributes in datasets. Besides, for the average number of attributes, the algorithm UAR_VPWND is significantly lower than the algorithms FRUAR, UFRFS, and HKCMI. This indicates that the proposed algorithm UAR_VPWND can more efficiently remove redundant and irrelevant attributes from the dataset.

Tables 5, 6, 7, and 8 present the optimal clustering ACC and ARI results based on the k-Means and FCM algorithm, respectively. The bold values in the tables indicate the optimal values achieved by the different AR algorithms. From the results in Tables 5, 6, 7, and 8, we can get some corresponding analyses as follows.

(1) From the results in Tables 5 and 6, the algorithm UAR_VPWND achieves the best clustering ACC in 15 cases compared to the other 9 UAR algorithms. However, algorithms FRUAR, UFRFS, HKCMI, LS, FSFS, USFSM, SPEC, FMIUFS, and UAR_mSMU achieve the best clustering ACC in only 5, 2, 1, 1, 1, 2, 4, 2, and 3 cases, respectively. Besides, for the clustering ACC of the algorithm UAR_VPWND, there are 24, 25, 22, 26, 28, 26, 24, 26, and 27 cases higher than algorithms FRUAR, UFRFS, HKCMI, LS, FSFS, USFSM, SPEC, FMIUFS, and UAR_mSMU respectively.

(2) From the results in Tables 7 and 8, the algorithm UAR_VPWND achieves the best clustering ARI in 12 cases compared to the other 9 UAR algorithms. However, algorithms FRUAR, UFRFS, HKCMI, LS, FSFS, USFSM, SPEC, FMIUFS, and UAR_mSMU achieve the best clustering AIR in only 5, 1, 3, 1, 0, 0, 3, 3, and 4 cases, respectively. Besides, for the clustering AIR of the algorithm UAR_VPWND, there are 24, 26, 22, 27, 29, 30, 27, 26, and 26 cases higher than algorithms FRUAR, UFRFS, HKCMI, LS, FSFS, USFSM, SPEC, FMIUFS, and UAR_mSMU respectively.

Table 6. The clustering ACC(%) of the FCM algorithm on reduced dataset

Dataset	Original attribute	FRUAR	UFRFS	HKCMI	LS	FSFS	USFSM	SPEC	FMIUFS	UAR_mSMU	UAR_VPWND	β
Derm	**69.73** ± **6.07**	48.31 ± 0.12	62.32 ± 0.09	51.53 ± 0.19	50.22 ± 0.22	50.33 ± 0.22	50.41 ± 0.19	51.15 ± 2.68	50.36 ± 0.22	50.66 ± 0.67	60.41 ± 2.14	0.95
Diab	66.67 ± 0.00	67.04 ± 0.04	66.67 ± 0.00	66.93 ± 0.00	64.66 ± 0.07	64.64 ± 0.07	64.65 ± 0.07	64.67 ± 0.06	64.64 ± 0.07	64.64 ± 0.07	**67.06** ± **0.00**	0.80
Ecoli	50.36 ± 1.95	**51.01** ± **2.11**	50.60 ± 2.13	50.18 ± 2.26	33.15 ± 0.21	33.39 ± 0.39	33.42 ± 0.37	33.45 ± 0.40	33.36 ± 0.30	33.63 ± 0.40	47.71 ± 1.84	0.90
Glass	54.86 ± 2.36	55.42 ± 2.71	57.99 ± 0.15	55.42 ± 2.71	64.58 ± 1.69	64.25 ± 1.72	63.18 ± 2.23	63.27 ± 1.38	63.93 ± 1.69	64.58 ± 1.69	**66.03** ± **1.03**	0.90
Iono	70.94 ± 0.00	71.23 ± 0.00	67.52 ± 0.00	69.52 ± 0.00	**87.18** ± **0.00**	**87.18** ± **0.00**	**87.18** ± **0.00**	**87.18** ± **0.00**	**87.18** ± **0.00**	**87.18** ± **0.00**	77.78 ± 0.00	1.00
Move	15.94 ± 1.14	15.11 ± 0.42	30.94 ± 1.10	25.64 ± 0.86	30.83 ± 0.65	30.89 ± 0.60	30.97 ± 0.66	31.17 ± 0.75	31.31 ± 0.75	31.06 ± 0.96	**37.61** ± **1.74**	0.75
Page	35.74 ± 0.00	35.70 ± 0.01	35.72 ± 0.00	35.68 ± 0.01	86.16 ± 0.69	31.08 ± 0.00	85.40 ± 1.61	**90.61** ± **0.01**	85.47 ± 1.35	85.63 ± 1.12	**90.61** ± **0.03**	0.60
Park	67.18 ± 0.00	67.69 ± 0.00	**75.38** ± **0.00**	74.87 ± 0.00	74.36 ± 0.00	74.36 ± 0.00	74.36 ± 0.00	74.36 ± 0.00	74.36 ± 0.00	74.36 ± 0.00	73.85 ± 0.00	0.80
Sonar	55.48 ± 0.61	56.92 ± 1.29	56.73 ± 0.00	62.07 ± 0.15	53.85 ± 0.00	53.85 ± 0.00	53.85 ± 0.00	53.85 ± 0.00	53.85 ± 0.00	53.85 ± 0.00	**63.94** ± **0.00**	0.50
WBC	95.28 ± 0.00	95.57 ± 0.00	95.28 ± 0.00	95.28 ± 0.00	93.85 ± 0.00	93.85 ± 0.00	93.85 ± 0.00	93.85 ± 0.00	93.85 ± 0.00	93.85 ± 0.00	**95.99** ± **0.00**	0.75
WDBC	92.79 ± 0.00	91.74 ± 0.00	91.74 ± 0.00	91.04 ± 0.00	88.58 ± 0.00	88.58 ± 0.00	88.58 ± 0.00	88.58 ± 0.00	88.58 ± 0.00	88.58 ± 0.00	**93.50** ± **0.00**	0.65
Wilt	51.44 ± 0.00	51.48 ± 0.00	51.44 ± 0.00	51.60 ± 0.00	53.81 ± 0.01	53.81 ± 0.01	51.44 ± 0.00	53.81 ± 0.00	53.81 ± 0.00	**53.82** ± **0.01**	53.81 ± 0.01	0.85
Wine1	**94.94** ± **0.00**	**94.94** ± **0.00**	92.70 ± 0.00	94.38 ± 0.00	78.09 ± 0.00	78.09 ± 0.00	78.09 ± 0.00	78.09 ± 0.00	78.09 ± 0.00	78.09 ± 0.00	93.82 ± 0.00	0.60
Wine2	30.15 ± 1.06	**31.78** ± **1.03**	30.91 ± 0.90	30.45 ± 1.12	21.50 ± 0.39	21.54 ± 0.29	21.56 ± 0.52	21.70 ± 0.45	21.85 ± 0.39	21.57 ± 0.53	23.61 ± 0.15	0.65
Wine3	**30.45** ± **0.88**	29.40 ± 1.06	28.66 ± 0.51	29.97 ± 0.59	22.67 ± 0.66	23.30 ± 0.18	22.38 ± 0.30	22.35 ± 0.06	22.45 ± 0.32	22.39 ± 0.29	28.95 ± 0.05	0.95
WPBC	58.08 ± 0.00	58.28 ± 2.98	60.40 ± 2.26	**63.13** ± **0.00**	59.09 ± 0.00	59.09 ± 0.00	59.09 ± 0.00	59.09 ± 0.00	59.09 ± 0.00	59.09 ± 0.00	62.12 ± 0.00	0.80
Average	58.75 ± 0.88	57.60 ± 0.74	59.69 ± 0.54	59.23 ± 0.49	60.16 ± 0.29	56.76 ± 0.22	59.90 ± 0.37	60.45 ± 0.36	60.14 ± 0.32	60.19 ± 0.36	**64.80** ± **0.44**	–

(3) From the perspective of the average clustering ACC and ARI, the algorithm UAR_VPWND achieves relatively higher average values among two clustering algorithms.
(4) In terms of standard deviation, the algorithm UAR_VPWND achieves relatively lower values among two clustering algorithms. This indicates that its results are relatively more stable.

In summary, the algorithm UAR_VPWND not only can select a relatively small subset of attributes but also can maintain or improve the clustering ACC and ARI. Therefore, the algorithm UAR_VPWND is suitable for UAR in numerical datasets for clustering tasks.

The analyses aforementioned have examined the results of the clustering experiments. Next, we will conduct comprehensive and in-depth analysis of the experimental results, and further findings are as follows.

(1) In general, the greater the number of conditional attributes in the original dataset, the more efficient the reduction will be. The greater the number of decision classes, the lower the ACC. Typically, sample size does not affect performance.
(2) The algorithm UAR_VPWND does not necessitate replacement or discretization, thereby enabling it to retain a greater amount of genuine information within the dataset. It uses the advantages of NRS to effectively handle data with uncertainty and incompleteness. Therefore, the algorithm UAR_VPWND has good clustering performance.

Hypothesis testing analyses

This paper adopts the method described in[4,57,61] to evaluate the statistical importance of algorithms, i.e., the Friedman's test and the Nemenyi's post hoc test. First, we use the Friedman's test to determine if these algorithms have the same performance. Assume we compare $\mathcal{M}$ algorithms on $\mathcal{N}$ datasets, where r_i represents the average ordinal value of the i algorithm, then the Friedman's test is calculated as

$$\tau_{\chi^2} = \frac{12\mathcal{N}}{\mathcal{M}(\mathcal{M}+1)}\left(\sum_{i=1}^{\mathcal{M}} r_i^2 - \frac{\mathcal{M}(\mathcal{M}+1)^2}{4}\right), \tau_{\mathcal{F}} = \frac{(\mathcal{N}-1)\tau_{\chi^2}}{\mathcal{N}(\mathcal{M}-1)-\tau_{\chi^2}}. \quad \text{(Equation 29)}$$

Where $\tau_{\mathcal{F}}$ obeys the F distribution with $(\mathcal{M}-1)$ and $(\mathcal{M}-1)(\mathcal{N}-1)$ degrees of freedom. If the null hypothesis that "all algorithms have the same performance" is rejected, then, it means that there are significant differences in the performance of the algorithms. Next, we typically use the Nemenyi's post hoc test. In it, the critical difference $(\mathcal{CD})$ of average ordinal value is calculated by

$$\mathcal{CD}_\alpha = q_\alpha\sqrt{\frac{\mathcal{M}(\mathcal{M}+1)}{6\mathcal{N}}}. \quad \text{(Equation 30)}$$

Table 7. The clustering ARI of the k-Means algorithm on reduced dataset

Dataset	Original attribute	FRUAR	UFRFS	HKCMI	LS	FSFS	USFSM	SPEC	FMIUFS	UAR_mSMU	UAR_VPWND
Derm	0.6246 ± 0.1212	0.5666 ± 0.0709	0.6036 ± 0.1291	0.3304 ± 0.0415	**0.7327** ± **0.1158**	0.4866 ± 0.0825	0.3821 ± 0.0417	0.4808 ± 0.0402	0.5394 ± 0.1152	0.6057 ± 0.1237	0.6306 ± 0.0118
Diab	0.1023 ± 0.0001	0.0867 ± 0.0455	0.1023 ± 0.0001	0.0994 ± 0.0006	0.0779 ± 0.0355	0.0839 ± 0.0167	0.0555 ± 0.0396	0.1062 ± 0.0000	**0.1070** ± **0.0000**	0.1062 ± 0.0000	**0.1070** ± **0.0000**
Ecoli	**0.4811** ± **0.0881**	0.4359 ± 0.0655	0.4465 ± 0.0707	0.4536 ± 0.0748	0.3711 ± 0.0548	0.3925 ± 0.0618	0.4158 ± 0.0765	0.3821 ± 0.0547	0.3806 ± 0.0363	0.3496 ± 0.0236	0.3854 ± 0.0525
Glass	0.4891 ± 0.1137	0.4360 ± 0.1458	0.4648 ± 0.1080	0.3918 ± 0.1389	0.3514 ± 0.0706	0.5255 ± 0.1421	0.1914 ± 0.0309	0.3928 ± 0.0838	**0.5621** ± **0.1032**	0.1752 ± 0.0193	0.5213 ± 0.0676
Iono	0.1742 ± 0.0024	0.1771 ± 0.0015	0.1651 ± 0.0391	0.1822 ± 0.0091	0.1185 ± 0.0037	0.1876 ± 0.0682	0.0208 ± 0.0590	0.0207 ± 0.0075	0.1299 ± 0.0000	0.1131 ± 0.0247	**0.3981** ± **0.0000**
Move	0.3025 ± 0.0314	0.2740 ± 0.0335	0.1284 ± 0.0092	0.2451 ± 0.0199	0.1573 ± 0.0052	0.2288 ± 0.0112	0.1874 ± 0.0198	0.1364 ± 0.0046	0.2414 ± 0.0234	0.2249 ± 0.0117	**0.3072** ± **0.0251**
Page	0.1063 ± 0.0108	0.1042 ± 0.0101	0.1073 ± 0.0089	0.0984 ± 0.0102	0.0043 ± 0.0006	0.0217 ± 0.0004	0.2194 ± 0.0729	**0.3782** ± **0.0073**	0.0041 ± 0.0000	0.3767 ± 0.0064	0.3767 ± 0.0064
Park	0.0493 ± 0.0034	0.0801 ± 0.0184	0.2376 ± 0.0112	0.2321 ± 0.0190	0.0403 ± 0.1709	0.1724 ± 0.0713	−0.0702 ± 0.0034	−0.0675 ± 0.0000	−0.0254 ± 0.0359	0.1172 ± 0.1275	**0.2534** ± **0.0895**
Sonar	0.0031 ± 0.0053	0.0039 ± 0.0077	0.0109 ± 0.0000	0.0462 ± 0.0189	−0.0047 ± 0.0001	0.0070 ± 0.0009	0.0055 ± 0.0055	0.0227 ± 0.0084	0.0114 ± 0.0188	−0.0033 ± 0.0000	**0.0733** ± **0.0000**
WBC	0.8391 ± 0.0000	0.8284 ± 0.0000	0.8391 ± 0.0000	**0.8444** ± **0.0000**	0.8225 ± 0.0017	0.7825 ± 0.0035	0.8306 ± 0.0027	0.7866 ± 0.0000	0.7814 ± 0.0016	0.8236 ± 0.0017	0.8443 ± 0.0000
WDBC	0.7302 ± 0.0000	0.6726 ± 0.0054	0.6849 ± 0.0030	0.6521 ± 0.0041	0.6686 ± 0.0099	0.0418 ± 0.0010	0.0997 ± 0.0457	0.0318 ± 0.0461	0.6701 ± 0.0001	0.0023 ± 0.0028	**0.7575** ± **0.0232**
Wilt	−0.0074 ± 0.0141	−**0.0006** ± **0.0003**	−0.0074 ± 0.0141	−0.0008 ± 0.0000	−0.0043 ± 0.0002	−0.0014 ± 0.0001	−0.0344 ± 0.0120	−0.0349 ± 0.0121	−0.0043 ± 0.0002	−0.0387 ± 0.0000	−0.0014 ± 0.0001
Wine1	0.8456 ± 0.0102	**0.8670** ± **0.0233**	0.8058 ± 0.0138	0.8598 ± 0.0236	0.7028 ± 0.0109	0.4288 ± 0.0068	0.2327 ± 0.0519	0.6219 ± 0.0000	0.8149 ± 0.0000	0.4289 ± 0.0142	0.8110 ± 0.0185
Wine2	0.0441 ± 0.0033	0.0450 ± 0.0042	0.0429 ± 0.0032	0.0447 ± 0.0027	0.0451 ± 0.0022	0.0181 ± 0.0019	0.0139 ± 0.0048	0.0490 ± 0.0020	0.0397 ± 0.0021	**0.0493** ± **0.0022**	0.0126 ± 0.0035
Wine3	0.0664 ± 0.0096	0.0613 ± 0.0092	0.0555 ± 0.0150	0.0578 ± 0.0121	0.0748 ± 0.0062	0.0206 ± 0.0038	0.0372 ± 0.0095	0.0112 ± 0.0035	0.0291 ± 0.0130	**0.0794** ± **0.0051**	0.0302 ± 0.0019
WPBC	0.0338 ± 0.0017	0.0010 ± 0.0205	0.0210 ± 0.0277	0.0089 ± 0.0364	0.0243 ± 0.0219	−0.0142 ± 0.0095	−0.0231 ± 0.0014	**0.0510** ± **0.0418**	0.0207 ± 0.0024	0.0362 ± 0.0457	0.0508 ± 0.0470
Average	0.3053 ± 0.0259	0.2900 ± 0.0289	0.2943 ± 0.0283	0.2841 ± 0.0257	0.2614 ± 0.0319	0.2114 ± 0.0301	0.1603 ± 0.0298	0.2106 ± 0.0195	0.2689 ± 0.0220	0.2154 ± 0.0255	**0.3474** ± **0.0217**

Table 8. The clustering ARI of the FCM algorithm on reduced dataset

Dataset	Original attribute	FRUAR	UFRFS	HKCMI	LS	FSFS	USFSM	SPEC	FMIUFS	UAR_mSMU	UAR_VPWND
Derm	0.6047 ± 0.0939	0.2717 ± 0.0032	0.4720 ± 0.0253	0.3013 ± 0.0032	0.6190 ± 0.0289	0.4098 ± 0.0344	0.2394 ± 0.0074	0.4920 ± 0.0219	0.5381 ± 0.0240	**0.6831** ± **0.0000**	0.4172 ± 0.0016
Diab	0.1069 ± 0.0000	**0.1138** ± **0.0000**	0.1069 ± 0.0000	0.1099 ± 0.0000	0.0988 ± 0.0000	0.0881 ± 0.0009	0.0697 ± 0.0008	0.0995 ± 0.0000	0.1104 ± 0.0000	0.0995 ± 0.0000	0.1104 ± 0.0000
Ecoli	0.3594 ± 0.0034	0.3586 ± 0.0012	0.3585 ± 0.0016	0.3594 ± 0.0038	0.2853 ± 0.0028	0.3076 ± 0.0044	0.3348 ± 0.0048	0.3118 ± 0.0037	0.3322 ± 0.0046	0.2700 ± 0.0004	**0.3639** ± **0.0182**
Glass	0.3590 ± 0.0458	0.3703 ± 0.0450	0.3947 ± 0.0053	0.3697 ± 0.0458	0.3574 ± 0.0216	0.4613 ± 0.0164	0.1985 ± 0.0000	0.3243 ± 0.0182	0.4690 ± 0.0182	0.1748 ± 0.0151	**0.4877** ± **0.0000**
Iono	0.1727 ± 0.0000	0.1776 ± 0.0000	0.1196 ± 0.0000	0.1495 ± 0.0000	0.1163 ± 0.0000	0.2092 ± 0.0000	−0.0054 ± 0.0000	0.0134 ± 0.0000	0.1299 ± 0.0000	0.1024 ± 0.0012	**0.2983** ± **0.0000**
Move	0.0747 ± 0.0042	0.0436 ± 0.0013	0.1224 ± 0.0042	0.1205 ± 0.0101	0.1472 ± 0.0053	0.2088 ± 0.0085	0.1640 ± 0.0079	0.1148 ± 0.0083	**0.2268** ± **0.0092**	0.2115 ± 0.0058	0.2266 ± 0.0077
Page	0.0641 ± 0.0000	0.0641 ± 0.0000	0.0641 ± 0.0000	0.0641 ± 0.0000	0.0035 ± 0.0000	0.0242 ± 0.0000	0.2317 ± 0.0171	**0.4786** ± **0.0009**	0.0035 ± 0.0000	0.4784 ± 0.0009	0.4757 ± 0.0097
Park	0.1109 ± 0.0000	0.1187 ± 0.0000	**0.2418** ± **0.0356**	0.2359 ± 0.0000	0.0584 ± 0.0706	0.1645 ± 0.0000	−0.0742 ± 0.0000	−0.0675 ± 0.0000	0.0869 ± 0.0000	0.0896 ± 0.0000	0.1723 ± 0.0000
Sonar	0.0067 ± 0.0016	0.0121 ± 0.0078	0.0134 ± 0.0000	0.0475 ± 0.0182	−0.0047 ± 0.0000	0.0254 ± 0.0000	0.0221 ± 0.0000	0.0086 ± 0.0000	0.0333 ± 0.0059	−0.0033 ± 0.0000	**0.0733** ± **0.0000**
WBC	0.8178 ± 0.0000	0.8284 ± 0.0000	0.8178 ± 0.0000	0.8178 ± 0.0000	0.8230 ± 0.0000	0.7705 ± 0.0000	0.8074 ± 0.0000	0.7814 ± 0.0000	0.7757 ± 0.0000	0.8230 ± 0.0000	**0.8443** ± **0.0000**
WDBC	0.7305 ± 0.0000	0.6947 ± 0.0000	0.6950 ± 0.0000	0.6718 ± 0.0000	0.6456 ± 0.0000	0.1790 ± 0.0000	0.2609 ± 0.0000	0.2984 ± 0.0000	0.6824 ± 0.0000	0.0038 ± 0.0000	**0.7543** ± **0.0000**
Wilt	0.0000 ± 0.0000	0.0001 ± 0.0000	0.0000 ± 0.0000	**0.0004** ± **0.0000**	−0.0038 ± 0.0000	−0.0013 ± 0.0000	−0.0418 ± 0.0027	−0.0446 ± 0.0000	−0.0038 ± 0.0000	−0.0446 ± 0.0000	−0.0013 ± 0.0000
Wine1	0.8498 ± 0.0000	**0.8499** ± **0.0000**	0.7845 ± 0.0000	0.8349 ± 0.0000	0.7071 ± 0.0039	0.4595 ± 0.0000	0.2763 ± 0.0090	0.6373 ± 0.0000	0.8149 ± 0.0000	0.4243 ± 0.0000	0.8185 ± 0.0000
Wine2	0.0485 ± 0.0022	**0.0537** ± **0.0012**	0.0491 ± 0.0015	0.0486 ± 0.0012	0.0439 ± 0.0007	0.0142 ± 0.0004	0.0254 ± 0.0008	0.0098 ± 0.0026	0.0382 ± 0.0010	0.0411 ± 0.0029	0.0406 ± 0.0009
Wine3	0.0312 ± 0.0009	0.0318 ± 0.0033	0.0297 ± 0.0016	0.0326 ± 0.0003	0.0611 ± 0.0009	0.0192 ± 0.0001	0.0423 ± 0.0010	0.0602 ± 0.0003	0.0543 ± 0.0008	**0.0675** ± **0.0001**	0.0580 ± 0.0001
WPBC	0.0222 ± 0.0012	0.0212 ± 0.0191	0.0314 ± 0.0300	**0.0649** ± **0.0000**	0.0351 ± 0.0000	−0.0047 ± 0.0000	−0.0100 ± 0.0000	0.0370 ± 0.0000	0.0166 ± 0.0000	0.0424 ± 0.0031	0.0538 ± 0.0020
Average	0.2725 ± 0.0096	0.2507 ± 0.0051	0.2688 ± 0.0066	0.2643 ± 0.0052	0.2496 ± 0.0084	0.2085 ± 0.0041	0.1588 ± 0.0032	0.2222 ± 0.0035	0.2693 ± 0.0040	0.2165 ± 0.0018	**0.3246** ± **0.0025**

Table 9. τ_F of different clustering learning algorithms

Clustering learning algorithmsτ_F	Critical value (α = 0.1)$\mathcal{CD}_{\alpha = 0.1}$	
k-Means	2.53281.6783	3.1257
FCM	3.0868–	–

Where q_α is the critical value from the Tukey's distribution, which can be found in.[63]

Given a significant level α, the corresponding $\mathcal{CD}_\alpha$ can be obtained. By calculating the difference of average ordinal vale between any two AR algorithms on the same clustering learning algorithm, which indicates the significant performance differences between the two AR algorithms on the clustering learning algorithm.

However, the aforementioned method does not visually represent the significant differences between the two AR algorithms. To this end, we describe the previous test results using the Nemenyi's test figure.[64] In the Nemenyi test figure, the average ordinal values of 10 AR algorithms are plotted at corresponding positions on the number line. If a group of algorithms is connected by a horizontal line segment, which indicates that there is no significant difference between these algorithms. Conversely, it means there is a significant difference between them.

By observing Tables 5 and 6, we can get that $\mathcal{M}$ = 10 and $\mathcal{N}$ = 16, the $\tau_{\mathcal{F}}$ distribution has 9 and 135 degrees of freedom. According to the Friedman's test, the $\tau_{\mathcal{F}}$ and $\mathcal{CD}_\alpha$ for different clustering learning algorithms are shown in Table 7. According to Table 9, when α = 0.1, the each values of $\tau_{\mathcal{F}}$ on two clustering learning algorithms is greater than the critical value of 1.6783. Therefore, the null hypothesis that "all AR algorithms have the same performance" is rejected on two clustering learning algorithms. This indicates that there are significant performance differences among all UAR algorithms in clustering learning algorithms. At this point, the Nemenyi's post hoc test is used to further distinguish these AR algorithms on two clustering learning algorithms.

From Table 9, for the significant level α = 0.1, we can obtain $\mathcal{CD}_\alpha$ = 3.1257. The Nemenyi's test figures on two clustering learning algorithms are shown in Figure 1. From Figure 1, it can be seen that algorithm UAR_VPWND has statistically significant differences compared to most other UAR algorithms. From Figure 1A, it can be see that algorithm UAR_VPWND is connected by horizontal line segment to algorithms HKCMI, FRUAR, SPEC, UAR_mSMU, and UFRFS; From Figure 1B, it can be see that algorithm UAR_VPWND is connected by horizontal line segment to algorithms HKCMI, UFRFS, FRUAR, SPEC, and FMIUFS. It indicates that the algorithm UAR_VPWND has statistically significant differences from the remaining AR algorithms on two clustering learning algorithms. Besides, from Figure 1, we can see that the algorithm UAR_VPWND has the highest average ordinal values on two clustering learning algorithms, which also shows the effectiveness of the algorithm UAR_VPWND.

Parameter analyses

The parameter β plays a very important role in the algorithm UAR_VPWND. It can be used to control the neighborhood granules of data analysis.

Figure 2 describes the variation in the number of attribute in $\mathcal{S}et$ and ACC with the parameter β for UAR_VPWND. β takes values from 0.5 to 1.0 in steps of 0.05. As can be seen from Figure 2, when $\beta < 0.65$, few attributes are found by the algorithm UAR_VPWND, and the corresponding ACC is also relatively low; when $\beta > 0.85$, the attributes found by the algorithm are also very few. Relatively speaking, when β is set between $[0.65, 0.85]$, the algorithm UAR_VPWND identifies attributes more appropriately, and the corresponding ACC is also higher.

Next, we will analyze from the perspectives of the number of attributes in the $\mathcal{S}et$ and ACC.

(1) From Figure 2, it shows that on some datasets, as β increases, the number of attributes in the reduction set remains relatively stable, such as datasets Ecoli, Page, Wine1, and Wine2. However, for datasets Derm, Diab, Park, WBC, Wilt, and WPBC, as β increases, the number of attributes in the reduction set is in a state of fluctuation. Besides, for datasets Move, and Sonar, as β increases, the number of attributes in the reduction set first remains relatively stable, and then decreases. For datasets WDBC, and Wine3, as β increases, the number of attributes in the reduction set first gradually increases, and then decreases.

(2) As for ACC, we can see from Figure 2, for most datasets, optimal values can be obtained under values of multiple parameter β. For each dataset, we can select the appropriate β based on Figure 2. Therefore, the algorithm UAR_VPWND is feasible for AR in clustering learning algorithms. Based on the above analyses, it is known that the algorithm UAR_VPWND proposed in this paper has the certain sensitivity to parameter β. Therefore, it is necessary for us to adjust the parameter β in the proposed algorithm. With appropriate parameter values, the algorithm UAR_VPWND can achieve better results in most cases.

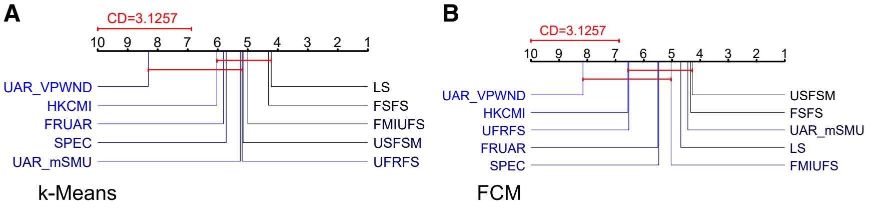


Figure 1. Nemenyi's test figures on two clustering learning algorithms

(A) Ten UAR algorithms on Nemenyi's test figure on k-Means learning algorithm.

(B) Ten UAR algorithms on Nemenyi's test figure on k-Means learning algorithm.

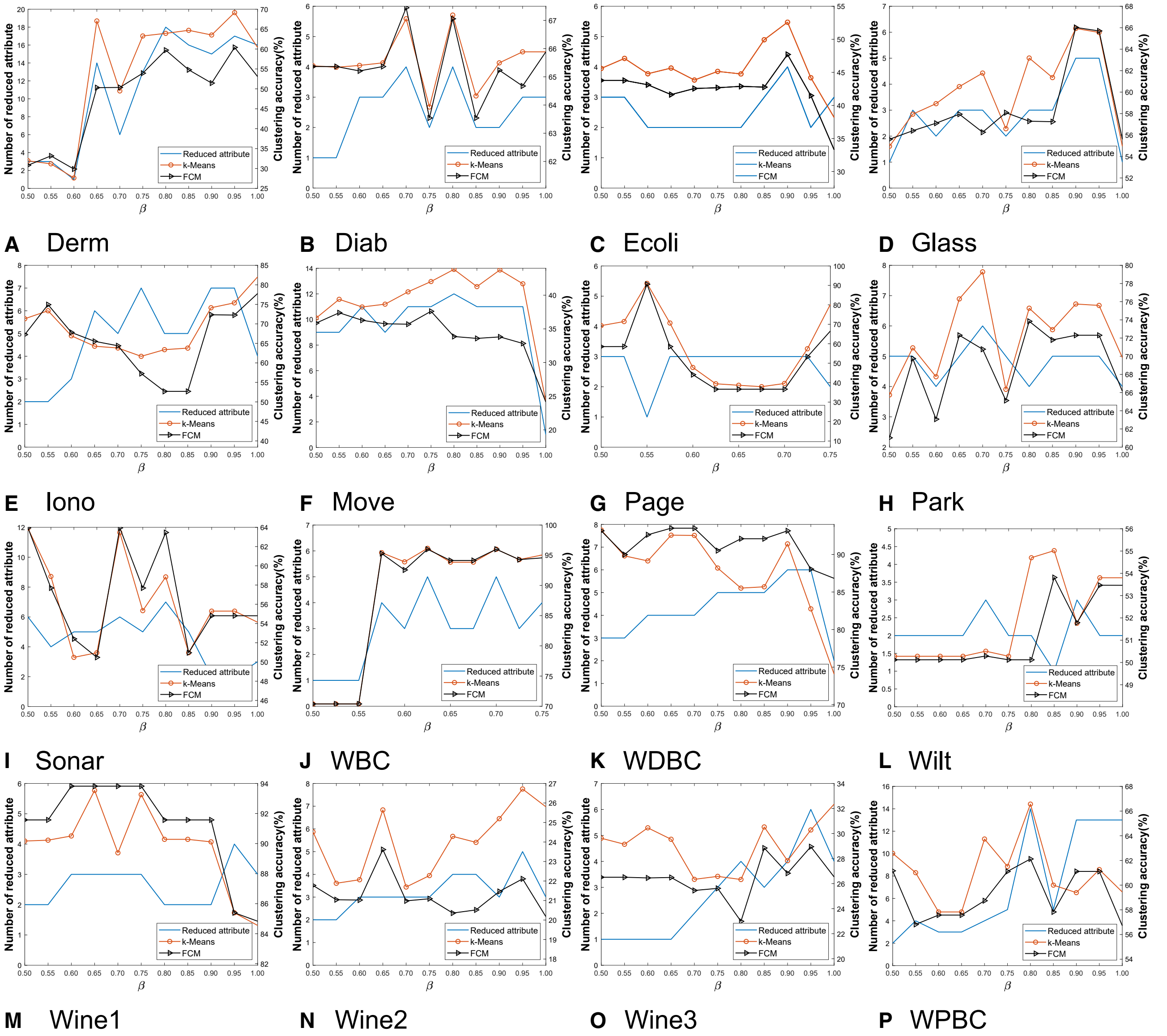


Figure 2. The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on numerical datasets

(A) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Derm dataset.
(B) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Diab dataset.
(C) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Ecoli dataset.
(D) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Glass dataset.
(E) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Iono dataset.
(F) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Move dataset.
(G) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Page dataset.
(H) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Park dataset.
(I) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Sonar dataset.
(J) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the WBC dataset.
(K) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the WDBC dataset.
(L) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Wilt dataset.
(M) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Wine1 dataset.
(N) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Wine2 dataset.
(O) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the Wine3 dataset.
(P) The number of reduced attributes and clustering ACC of the algorithm UAR_VPWND using different β values on the WPBC dataset.

In summary, the algorithm UAR_VPWND is feasible and effective.

DISCUSSION

Aiming at the problem of UAR, This paper proposes an UAR method based on VPWND that can deal with uncertainty and incomplete data without decision information.

The algorithm UAR_VPWND does not require discretization processing, which can reduce the time for data processing while avoiding the loss of data information. For this method, we have designed the corresponding algorithm UAR_VPWND. Based on 16 datasets, the algorithm UAR_VPWND is compared with existing nine classical UAR algorithms. The experimental results show that the algorithm UAR_VPWND can select fewer attributes to maintain or improve the performance of clustering learning algorithms. Besides, the hypothesis statistical test show that the algorithm UAR_VPWND has significant differences compared to most classical UAR algorithms.

Limitations of the study

However, the the algorithm UAR_VPWND has some limitations. It is only suitable for processing numerical data and requires the adjustment of parameter. Some traditional methods can not only handle mixed data but also do not require parameter adjustment. In the future, we can further research the UAR of mixed data based on NRS.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Yi Li (scmzly@scun.edu.cn).

Materials availability

This study proposes a new unsupervised attribute reduction algorithm UAR_VPWND, which is suitable for numerical data.

Data and code availability

- Data have been deposited at WeiYun and are publicly available as of the date of publication.
- Code have been deposited at WeiYun and are publicly available as of the date of publication.
- Any additional information required to reanalyze the data reported in this study is available from the lead contact upon request.

ACKNOWLEDGMENTS

The authors thank both the editors and reviewers for their valuable suggestions, which substantially improved this paper. This work was supported by the Key Research and Development Projects in Sichuan Province (2023YFG0303), the Natural Science Foundation of Sichuan Province (2022NSFSC1830), the Project of Sichuan Province Department of Science and Technology (2023ZHCG0009), the Science and Technology Project in Ganzi Prefecture Sichuan Province (23KJJH00016), and the Academic Innovation Team of Sichuan Minzu College (TD202302).

AUTHOR CONTRIBUTIONS

Y.L. and B.Z.: conceived and performed experiments and wrote the manuscript. H.M. and J.H.: performed experiments and analyzed data. Y.L.: wrote algorithmic code. X.T.: constructed theoretical algorithm and secured funding.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
- METHOD DETAILS
- QUANTIFICATION AND STATISTICAL ANALYSIS

Received: April 22, 2024
Revised: August 18, 2024
Accepted: October 24, 2024
Published: October 29, 2024

REFERENCES

1. Wan, J., Chen, H., Li, T., Li, M., and Yang, X. (2023). High-order interaction feature selection for classification learning: A robust knowledge metric perspective. Pattern Recogn. *143*, 109733. https://doi.org/10.1016/j.patcog.2023.109733.
2. Wan, J., Chen, H., Li, T., Yuan, Z., Liu, J., and Huang, W. (2023). Interactive and complementary feature selection via fuzzy multigranularity uncertainty measures. IEEE Trans. Cybern. *53*, 1208–1221. https://doi.org/10.1109/TCYB.2021.3112203.
3. Zhu, P., Xu, Q., Hu, Q., and Zhang, C. (2018). Co-regularized unsupervised feature selection. Neurocomputing *275*, 2855–2863. https://doi.org/10.1016/j.neucom.2017.11.061.
4. Yuan, Z., Chen, H., Li, T., Yu, Z., Sang, B., and Luo, C. (2021). Unsupervised attribute reduction for mixed data based on fuzzy rough sets. Inform. Sci. *572*, 67–87. https://doi.org/10.1016/j.ins.2021.04.083.
5. Wang, H., Zhang, Y., Zhang, J., Li, T., and Peng, L. (2019). A factor graph model for unsupervised feature selection. Inform. Sci. *480*, 144–159. https://doi.org/10.1016/j.ins.2018.12.034.
6. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., and Ziarko, W. (1995). Rough sets. Commun. ACM *38*, 88–95. https://doi.org/10.1145/219717.219791.
7. Lian Jie, D., De Gang, C., Ning Ling, W., and Zhan Hui, L. (2020). Key energy-consumption feature selection of thermal power systems based on robust attribute reduction with rough sets. Inform. Sci. *532*, 61–71. https://doi.org/10.1016/j.ins.2020.03.085.
8. Zhao, J., Liang, J.M., Dong, Z.N., Tang, D.Y., and Liu, Z. (2020). Nec: A nested equivalence class-based dependency calculation approach for fast feature selection using rough set theory. Inform. Sci. *536*, 431–453. https://doi.org/10.1016/j.ins.2020.03.092.
9. Li, D., Zhang, B., and Leung, Y. (2004). On knowledge reduction in inconsistent decision information systems. Int. J. Unc. Fuzz. Knowl. Based. Syst. *12*, 651–672. https://doi.org/10.1142/s0218488504003132.
10. Salama, M.A., Hassanien, A.E., and Mostafa, A. (2016). The prediction of virus mutation using neural networks and rough set techniques. EURASIP J. Bioinform. Syst. Biol. *2016*, 10. https://doi.org/10.1186/s13637-016-0042-0.
11. Teng, S.H., Lu, M., Yang, A.F., Zhang, J., Nian, Y., and He, M. (2016). Efficient attribute reduction from the viewpoint of discernibility. Inform. Sci. *326*, 297–314. https://doi.org/10.1016/j.ins.2015.07.052.
12. Zhang, X., Mei, C., Chen, D., and Li, J. (2016). Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy. Pattern Recogn. *56*, 1–15. https://doi.org/10.1016/j.patcog.2016.02.013.

13. Hu, Q., Yu, D., Liu, J., and Wu, C. (2008). Neighborhood rough set based heterogeneous feature subset selection. Inform. Sci. *178*, 3577–3594. https://doi.org/10.1016/j.ins.2008.05.024.

14. Wan, J., Chen, H., Yuan, Z., Li, T., Yang, X., and Sang, B. (2021). A novel hybrid feature selection method considering feature interaction in neighborhood rough set. Knowledge-Based Syst. *227*, 107167. https://doi.org/10.1016/j.knosys.2021.107167.

15. Yang, X., Chen, H., Li, T., Wan, J., and Sang, B. (2021). Neighborhood rough sets with distance metric learning for feature selection. Knowledge-Based Syst. *224*, 107076. https://doi.org/10.1016/j.knosys.2021.107076.

16. Pang, Q.Q., and Zhang, L. (2020). Semi-supervised neighborhood discrimination index for feature selection. Knowledge-Based Syst. *204*, 106224. https://doi.org/10.1016/j.knosys.2020.106224.

17. Ibrahim, R.A., Abd Elaziz, M., Oliva, D., and Lu, S. (2020). An improved runner-root algorithm for solving feature selection problems based on rough sets and neighborhood rough sets. Appl. Soft Comput. *97*, 105517. https://doi.org/10.1016/j.asoc.2019.105517.

18. Sun, L., Yin, T., Ding, W., Qian, Y., and Xu, J. (2020). Multilabel feature selection using ml-relieff and neighborhood mutual information for multilabel neighborhood decision systems. Inform. Sci. *537*, 401–424. https://doi.org/10.1016/j.ins.2020.05.102.

19. Liu, J., Lin, Y., Li, Y., Weng, W., and Wu, S. (2018). Online multi-label streaming feature selection based on neighborhood rough set. Pattern Recogn. *84*, 273–287. https://doi.org/10.1016/j.patcog.2018.07.021.

20. Lin, Y., Hu, Q., Liu, J., Chen, J., and Duan, J. (2016). Multi-label feature selection based on neighborhood mutual information. Appl. Soft Comput. *38*, 244–256. https://doi.org/10.1016/j.asoc.2015.10.009.

21. Liu, D., and Li, J. (2019). Safety monitoring data classification method based on wireless rough network of neighborhood rough sets. Safety Sci. *118*, 103–108. https://doi.org/10.1016/j.ssci.2019.05.004.

22. Chu, X., Sun, B., Li, X., Han, K., Wu, J., Zhang, Y., and Huang, Q. (2020). Neighborhood rough set-based three-way clustering considering attribute correlations: An approach to classification of potential gout groups. Inform. Sci. *535*, 28–41. https://doi.org/10.1016/j.ins.2020.05.039.

23. Sun, L., Wang, L., Ding, W., Qian, Y., and Xu, J. (2020). Neighborhood multi-granulation rough sets-based attribute reduction using lebesgue and entropy measures in incomplete neighborhood decision systems. Knowledge-Based Syst. *192*, 105373. https://doi.org/10.1016/j.knosys.2019.105373.

24. Chen, Y., Zhang, Z., Zheng, J., Ma, Y., and Xue, Y. (2017). Gene selection for tumor classification using neighborhood rough sets and entropy measures. J. Biomed. Inform. *67*, 59–68. https://doi.org/10.1016/j.jbi.2017.02.007.

25. Sun, L., Zhang, X., Qian, Y., Xu, J., and Zhang, S. (2019). Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification. Inform. Sci. *502*, 18–41. https://doi.org/10.1016/j.ins.2019.05.072.

26. Hu, C., Zhang, L., Wang, B., Zhang, Z., and Li, F. (2019). Incremental updating knowledge in neighborhood multigranulation rough sets under dynamic granular structures. Knowledge-Based Syst. *163*, 811–829. https://doi.org/10.1016/j.knosys.2018.10.010.

27. Liu, Y., Huang, W.L., Jiang, Y.L., and Zeng, Z.Y. (2014). Quick attribute reduct algorithm for neighborhood rough set model. Inform. Sci. *271*, 65–81. https://doi.org/10.1016/j.ins.2014.02.093.

28. Hu, Q., Pedrycz, W., Yu, D., and Lang, J. (2010). Selecting discrete and continuous features based on neighborhood decision error minimization. IEEE Trans. Syst. Man Cybern. B Cybern. *40*, 137–150. https://doi.org/10.1109/tsmcb.2009.2024166.

29. Solorio-Fernández, S., Carrasco-Ochoa, J.A., and Martínez-Trinidad, J.F. (2020). A review of unsupervised feature selection methods. Artif. Intell. Rev. *53*, 907–948. https://doi.org/10.1007/s10462-019-09682-y.

30. Alelyani, S., Tang, J., and Liu, H. (2018). Feature selection for clustering: A review. Data Cluster. Algor. Appl. *1*, 29–60. https://doi.org/10.1201/9781315373515-2.

31. Rao, V.M., and Sastry, V. (2012). Unsupervised Feature Ranking Based on Representation Entropy. In 2012 1st International Conference on Recent Advances in Information Technology (RAIT), pp. 421–425. https://doi.org/10.1109/rait.2012.6194631.

32. Solorio-Fernández, S., Martínez-Trinidad, J.F., and Carrasco-Ochoa, J.A. (2017). A new unsupervised spectral feature selection method for mixed data: a filter approach. Pattern Recogn. *72*, 314–326. https://doi.org/10.1016/j.patcog.2017.07.020.

33. Mitra, P., Murthy, C., and Pal, S.K. (2002). Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Mach. Intell. *24*, 301–312. https://doi.org/10.1109/34.990133.

34. Tabakhi, S., Moradi, P., and Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. Eng. Appl. Artif. Intell. *32*, 112–123. https://doi.org/10.1016/j.engappai.2014.03.007.

35. Nie, F., Zhu, W., and Li, X. (2016). Unsupervised Feature Selection with Structured Graph Optimization. Proceedings of the AAAI Conference on Artificial Intelligence *30*. https://doi.org/10.1609/aaai.v30i1.10168.

36. Dutta, D., Dutta, P., and Sil, J. (2013). Simultaneous feature selection and clustering with mixed features by multi objective genetic algorithm. Int. J. Hybrid Intell. Syst. *11*, 41–54. https://doi.org/10.3233/his-130182.

37. Law, M.H.C., Figueiredo, M.A.T., and Jain, A.K. (2004). Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. *26*, 1154–1166. https://doi.org/10.1109/tpami.2004.71.

38. Solorio-Fernández, S., Carrasco-Ochoa, J.A., and Martínez-Trinidad, J.F. (2016). A new hybrid filter–wrapper feature selection method for clustering based on ranking. Neurocomputing *214*, 866–880. https://doi.org/10.1016/j.neucom.2016.07.026.

39. Hruschka, E.R., Covoes, T.F., and Ebecken, N.F. (2005). Feature Selection for Clustering Problems: A Hybrid Algorithm that Iterates between K-Means and a Bayesian Filter. In Fifth International Conference on Hybrid Intelligent Systems (HIS'05), p. 6. https://doi.org/10.1109/ichis.2005.42.

40. Wang, C., Huang, Y., Shao, M., Hu, Q., and Chen, D. (2020). Feature selection based on neighborhood self-information. IEEE Trans. Cybern. *50*, 4031–4042. https://doi.org/10.1109/tcyb.2019.2923430.

41. Chen, H., Li, T., Fan, X., and Luo, C. (2019). Feature selection for imbalanced data based on neighborhood rough sets. Inf. Sci. *483*, 1–20. https://doi.org/10.1016/j.ins.2019.01.041.

42. Wang, C., Shi, Y., Fan, X., and Shao, M. (2019). Attribute reduction based on k-nearest neighborhood rough sets. Intern. J. Approx. Reason. *106*, 18–31. https://doi.org/10.1016/j.ijar.2018.12.013.

43. Guo, Y., Tsang, E.C., Xu, W., and Chen, D. (2020). Adaptive weighted generalized multi-granulation interval-valued decision-theoretic rough sets. Knowledge-Based Syst. *187*, 104804. https://doi.org/10.1016/j.knosys.2019.06.012.

44. Tsang, E.C.C., Hu, Q., and Chen, D. (2016). Feature and instance reduction for pnn classifiers based on fuzzy rough sets. Int. J. Mach. Learn. Cybern. *7*, 1–11. https://doi.org/10.1007/s13042-014-0232-6.

45. Vluymans, S., Mac Parthaláin, N., Cornelis, C., and Saeys, Y. (2019). Weight selection strategies for ordered weighted average based fuzzy rough sets. Inform. Sci. *501*, 155–171. https://doi.org/10.1016/j.ins.2019.05.085.

46. Li, W., Zhai, S., Xu, W., Pedrycz, W., Qian, Y., Ding, W., and Zhan, T. (2023). Feature selection approach based on improved fuzzy c-means with principle of refined justifiable granularity. IEEE Trans. Fuzzy Syst. *31*, 2112–2126. https://doi.org/10.1109/TFUZZ.2022.3217377.

47. Liu, K., Li, T., Yang, X., Ju, H., Yang, X., and Liu, D. (2022). Hierarchical neighborhood entropy based multi-granularity attribute reduction with application to gene prioritization. Intern. J. Approx. Reason. *148*, 57–67. https://doi.org/10.1016/j.ijar.2022.05.011.

48. Li, W., Zhou, H., Xu, W., Wang, X.-Z., and Pedrycz, W. (2023). Interval dominance-based feature selection for interval-valued ordered data. IEEE Trans. Neural Netw. Learn. Syst. *34*, 6898–6912. https://doi.org/10.1109/TNNLS.2022.3184120.

49. Zhang, P., Li, T., Yuan, Z., Deng, Z., Wang, G., Wang, D., and Zhang, F. (2023). A possibilistic information fusion-based unsupervised feature selection method using information quality measures. IEEE Trans. Fuzzy Syst. *31*, 2975–2988. https://doi.org/10.1109/TFUZZ.2023.3238803.

50. Li, W., Deng, C., Pedrycz, W., Castillo, O., Zhang, C., and Zhan, T. (2024). Double-quantitative feature selection approach for multi-granularity ordered decision systems. IEEE Trans. Artif. Intell. *5*, 2385–2396. https://doi.org/10.1109/TAI.2023.3319301.

51. Sheikhpour, R., Berahmand, K., and Forouzandeh, S. (2023). Hessian-based semi-supervised feature selection using generalized uncorrelated constraint. Knowledge-Based Syst. *269*, 110521. https://doi.org/10.1016/j.knosys.2023.110521.

52. Berahmand, K., Bahadori, S., Abadeh, M.N., Li, Y., and Xu, Y. (2024). Sdac-da: Semi-supervised deep attributed clustering using dual autoencoder. IEEE Trans. Knowl. Data Eng. *36*, 6989–7002. https://doi.org/10.1109/TKDE.2024.3389049.

53. Hu, Q., Yu, D., and Xie, Z. (2008). Neighborhood classifiers. Expert Syst. Appl. *34*, 866–876. https://doi.org/10.1016/j.eswa.2006.10.043.

54. Hu, Q., Liu, J., and Yu, D. (2008). Mixed feature selection based on granulation and approximation. Knowledge-Based Syst. *21*, 294–304. https://doi.org/10.1016/j.knosys.2007.07.001.

55. Yuan, Z., Zhang, X., and Feng, S. (2018). Hybrid data-driven outlier detection based on neighborhood information entropy and its developmental measures. Expert Syst. Appl. *112*, 243–257. https://doi.org/10.1016/j.eswa.2018.06.013.

56. Mac Parthaláin, N., and Jensen, R. (2013). Unsupervised fuzzy-rough set-based dimensionality reduction. Inform. Sci. *229*, 106–121. https://doi.org/10.1016/j.ins.2012.12.001.

57. Yuan, Z., Chen, H., Yang, X., Li, T., and Liu, K. (2021). Fuzzy complementary entropy using hybrid-kernel function and its unsupervised attribute reduction. Knowledge-Based Syst. *231*, 107398. https://doi.org/10.1016/j.knosys.2021.107398.

58. He, X.F., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection. Adv. Neural Inform. Process. Syst. *18*, 507–514.

59. Zhao, Z., and Liu, H. (2007). Spectral Feature Selection for Supervised and Unsupervised Learning. In Proceedings of the 24th international conference on Machine learning, *227*, pp. 1151–1157. https://doi.org/10.1145/1273496.1273641.

60. Yuan, Z., Chen, H., Zhang, P., Wan, J., and Li, T. (2022). A novel unsupervised approach to heterogeneous feature selection based on fuzzy mutual information. IEEE Trans. Fuzzy Syst. *30*, 3395–3409. https://doi.org/10.1109/TFUZZ.2021.3114734.

61. Zhang, P., Wang, D., Yu, Z., Zhang, Y., Jiang, T., and Li, T. (2024). A multi-scale information fusion-based multiple correlations for unsupervised attribute selection. Inform. Fusion *106*, 102276. https://doi.org/10.1016/j.inffus.2024.102276.

62. Zhu, P., Zhu, W., Hu, Q., Zhang, C., and Zuo, W. (2017). Subspace clustering guided unsupervised feature selection. Pattern Recogn. *66*, 364–374. https://doi.org/10.1016/j.patcog.2017.01.016.

63. Li, Y., Zhang, B., Yuan, Z., Liu, Y., Lei, S., and Tan, X. (2024). Unsupervised attribute reduction based on neighborhood dependency. Appl. Intell. *54*, 10653–10670. https://doi.org/10.1007/s10489-024-05604-w.

64. Yuan, Z., Chen, H., Xie, P., Zhang, P., Liu, J., and Li, T. (2021). Attribute reduction methods in fuzzy rough set theory: An overview, comparative experiments, and new directions. Appl. Soft Comput. *107*, 107353. https://doi.org/10.1016/j.asoc.2021.107353.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Data	WeiYun	https://share.weiyun.com/XX6WN4oE
Software and algorithms		
This study	WeiYun	https://share.weiyun.com/SWExHboE
MATLAB2021a	MathWorks	https://ww2.mathworks.cn

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

This study proposed an UAR algorithm UAR_VPWND, which is applicable to numerical data.

This study developed an UAR algorithm UAR_VPWND considering conditional attributes based on VPNRS, weighted neighborhood, and UAR. At the same time, we selected nine classical UAR algorithms as the comparison algorithms. There are FRUAR,[4] UFRFS,[56] HKCMI,[57] LS,[58] FSFS,[33] USFSM,[32] SPEC,[59] FMIUFS,[60] and UAR_mSMU.[49] We use clustering learning algorithms k-Means and FCM, and hypothesis testing analyses to evaluate the performance of algorithms. The final results of the analysis are shown in Tables 4, 5, 6, 7, and 8, and Figure 1.

METHOD DETAILS

This study proposed an UAR algorithm UAR_VPWND, which is applicable to the processing of high-dimensional numerical data without decision information. we compared UAR_VPWND with FRUAR, UFRFS, HKCMI, LS, FSFS, USFSM, SPEC, FMIUFS, and UAR_mSMU using publicly available datasets Table 3. We use clustering learning algorithms and hypothesis testing analyses to evaluate the performance of algorithms. For clustering experiments, We called algorithms in MATLAB R2021a k-Means and FCM. The parameters of all clustering experiments are kept as initial values. The number of clusters is set to the number of real decision classes. The clustering algorithms return the predicted decisions for the dataset. This study uses ACC and adjusted ARI to evaluate the performance of the clustering algorithms. For hypothesis testing analyses, we use Nemenyi test figure Figure 1, the average ordinal values of 10 UAR algorithms are plotted at corresponding positions on the number line.

The algorithm UAR_VPWND does not require discretization processing, which can reduce the time for data processing while avoiding the loss of data information. Based on 16 datasets, the algorithm UAR_VPWND is compared with existing nine classical UAR algorithms. The experimental results show that the algorithm UAR_VPWND can select fewer attributes Table 4 to maintain or improve the performance of clustering learning algorithms Tables 5, 6, 7, and 8. Besides, the hypothesis statistical test show that the algorithm UAR_VPWND has significant differences compared to most classical UAR algorithms Figure 1. This fully proves the effectiveness and application value of the UAR algorithm proposed in this study.

QUANTIFICATION AND STATISTICAL ANALYSIS

From Tables 4, 5, 6, 7, and 8, we can see that the algorithm UAR_VPWND can select fewer attributes to maintain or improve the performance of clustering learning algorithms. At the same time, we adopt hypothesis testing analyses[4,57,61] to evaluate the statistical importance of algorithm.

From Table 9 and Figure 1, we can see that τ_F for the k-Means and FCM learning algorithms, respectively:2.5328, and 3.0868. Also, for the critical value $= 0.1$, we can get the corresponding critical difference $\mathcal{CD}_{\alpha = 0.1}$. The experimental results show that the algorithm UAR_VPWND has significant differences compared to most classical UAR algorithms. This fully indicates that the algorithm UAR_VPWND has some effectiveness and application value. From Figure 2, we can see that when variable precision threshold parameterβ is set between $[0.65,0.85]$, the algorithm UAR_VPWND identifies attributes more appropriately, and the corresponding ACC is also higher.