Pattern Recognition 180 (2026) 114034

Contents lists available at ScienceDirect

Pattern Recognition

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

Graph theory-based dynamic unsupervised feature selection method for interval-valued datasets

Xuan Shen, Xiaoyan Zhang *

College of Artificial Intelligence, Southwest University, Chongqing, 400715, PR China

ARTICLE INFO

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

Keywords:
Feature selection
Graph theory
Interval-valued information system(IVIS)
Dynamic data

ABSTRACT

With the growing demand for modeling uncertainty and dynamic environments, feature selection in interval-valued information systems (IVIS) has attracted increasing attention. However, as data scales expand and systems undergo continuous updates, the structure of information systems becomes highly dynamic. Most existing methods are designed for static scenarios and struggle to accommodate structural changes such as the addition or deletion of objects and attributes. To address this, this paper propose a unified unsupervised graph-based dynamic feature selection framework that efficiently handles four typical structural variations in IVIS. For object variations, multiple similarity graphs are constructed with objects as nodes, and graph fusion and feature weight learning are jointly optimized. For attribute variations, a fully connected weighted graph is built using attributes as nodes, and dynamic ranking is achieved via locally updated adjacency matrices. The proposed method leverages local updates to avoid costly full graph reconstruction. Extensive experiments on multiple benchmarks demonstrate that our approach consistently outperforms state-of-the-art baselines, improving ARI and NMI metrics by over 10% on datasets such as Dermatology. Furthermore, the local update mechanism yields approximately an 80% reduction in computational overhead compared to static reconstruction, making the framework highly suitable for large-scale and real-time dynamic applications.

1. Introduction

With the rapid development of big data and artificial intelligence, real-world data often suffers from measurement errors, subjective judgments, and inherent ambiguity, making it difficult to describe using precise numerical values [1,2]. To address such uncertainty, the Interval-Valued Information System (IVIS) [3,4] has emerged and been widely applied in medical diagnosis, environmental monitoring, and decision support. By representing attribute values as intervals, IVIS effectively preserves data variability and fuzziness, and has become an important paradigm for uncertainty modeling [5,6].

In high-dimensional scenarios such as IVIS, feature selection [7,8] plays a key role in improving model performance, reducing computational complexity, and alleviating overfitting . Especially in unsupervised settings, feature selection becomes more challenging due to the absence of class labels [9–11]. This difficulty is further increased in the presence of uncertainty introduced by interval-valued data. To this end, graph-based modeling methods have emerged as a significant research direction, owing to their robust structural representation capabilities. Graph-theoretic [12] approaches can ingeniously capture and utilize structural features within the data from a global perspective, intuitively reflecting complex data architectures [13–15].

However, real-world data systems are inherently dynamic, with continuous sample arrival and attribute expansion leading to evolving data structures [16–18]. Most graph-based methods rely on static assumptions and thus perform poorly in such settings. They require full model reconstruction after updates and depend on global optimization, resulting in high computational cost and low efficiency. This limits their applicability in dynamic IVIS, highlighting the need for graph-based feature selection methods with efficient updating. None of the existing graph-based methods simultaneously support both object and attribute dynamics in interval-valued systems.

To address the above issues, this paper proposes a graph-based dynamic feature selection framework for IVIS, capable of handling four types of structural variation: object addition, object deletion, attribute addition, and attribute deletion. The framework aims to tackle two key problems: (1) modeling the intrinsic structural relationships of interval-valued data in an unsupervised manner, and (2) enabling efficient updates without global reconstruction. The motivation of this work is illustrated in Fig. 1. The main contributions are as follows:

- This study models structural relationships by constructing feature-level and object-level correlation graphs using graph theory.

* Corresponding author.
E-mail addresses: sx18908529566@163.com (X. Shen), zxy19790915@163.com (X. Zhang).

https://doi.org/10.1016/j.patcog.2026.114034
Received 29 September 2025; Received in revised form 10 April 2026; Accepted 17 May 2026
Available online 25 May 2026

Table 1
The summary of existing methods.

Categories	Methods	Limitations
Graph-based	Akhiat et al. [19], Roffo et al. [20], Xie et al. [21], Xiang et al. [22], Zhou et al. [23]	It supports only static scenarios and cannot be directly applied to interval-valued datasets characterized by uncertainty.
	Zhang et al. [24]	Unable to adapt to dynamic environments.
Dynamic single-valued	Sang et al. [25], Yang et al. [26], Pan et al. [27], Xu et al. [28], Zhang et al. [29]	Only supports object dynamics, incurs high computational cost, and is not suitable for interval-valued data.
Dynamic interval-valued data	Feng et al. [30]	Highly dependent on supervisory information and decision consistency constraints.
	Shu et al. [17]	Does not support attribute updates, has limited structural modeling capability, and incurs high computational cost.

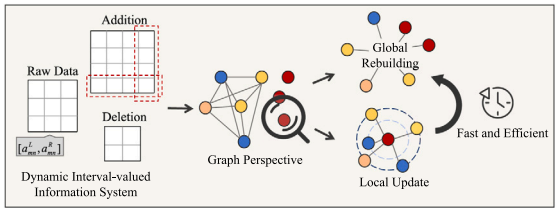


Fig. 1. The motivation for our work.

- A local optimization mechanism is developed to efficiently propagate local changes to global feature adjustments, significantly improving computational efficiency.
- The proposed method supports both object and attribute dynamics in IVIS, achieving strong real-time adaptability and scalability.

This work is organized into eight sections. After reviewing related work and preliminaries in Sections 2–3, we discuss object and attribute-level changes in Sections 4–5. Building on these, Section 6 presents four dynamic update algorithms. The paper concludes with experimental evaluations in Section 7 and a summary in Section 8.

2. Related work

Table 1 classifies and summarizes existing feature selection methods from the perspectives of method types and application scenarios. Existing feature selection methods still suffer from limitations in handling dynamic and uncertain data, including reliance on static assumptions, inadequate support for attribute evolution, weak structural modeling capability, and high computational cost. The proposed graph-based dynamic feature selection framework for IVIS effectively addresses the gaps and limitations in existing research.

3. Preliminaries

3.1. Interval-valued information system

Let $IVIS = (U, AT, F)$ be an information system, where $U = \{x_1, x_2, \ldots, x_m\}$ is a non-empty finite set of objects, $AT = \{a_1, a_2, \ldots, a_n\}$ is a non-empty finite set of attributes, and $\forall x_i \in U$, $a_k \in AT$, the $f(x_i, a_k)$ yields an interval-valued number, specifically denoted as $f(x_i, a_k) = [a_k^l(x_i), a_k^r(x_i)]$. $a_k^l(x_i)$ and $a_k^r(x_i)$ represent the left and right bounds of the interval, which can be abbreviated as a_{ik}^l and a_{ik}^r. Notably, if these bounds coincide, i.e., $a_k^l(x_i) = a_k^r(x_i)$, the interval collapses to a single value, showing that a conventional single-valued decision system is simply a special case of the more general IVIS framework [31].

3.2. Graph theory

A graph consists of nodes and edges, where edges represent relationships between nodes. When constructing graph theory models for interval-valued information systems, the graph can be categorized based on the type of nodes. It can either be a graph model with objects as nodes or a graph model with attributes as nodes.

When constructing a graph with objects as nodes [32], a similarity graph $G = \langle V, E \rangle$ is built, where the nodes represent objects $U = \{x_1, x_2, \ldots, x_m\}$ and the edges represent the similarity between samples. By defining the similarity between samples, an adjacency matrix S can be constructed to capture the complex relationships among objects.

When constructing a graph with attributes as nodes [33], a weighted undirected complete graph $G = \langle V, E \rangle$ is built, where the nodes represent attributes $AT = \{a_1, a_2, \ldots, a_n\}$ and the edges represent the correlation between attributes. The adjacency matrix H is generated by assigning weights to the edges, which captures the association information among attributes.

4. DIUO: Feature selection methods for object dynamics

To address dynamic object variation, we propose a graph-based local update strategy. The update procedure is illustrated in Fig. 2. In this section, we detail the proposed method under object dynamics.

4.1. Feature selection framework

We first construct a graph based on the interval-valued information system, where objects serve as nodes. For interval-valued features, each interval $[a_i^l, a_i^r]$ is represented by its midpoint $\frac{a_i^l + a_i^r}{2}$ so that the data can be mapped into a numerical space for subsequent processing.

For unsupervised feature selection, since label information is unavailable, data must be processed without supervision. Accordingly, the original data are mapped into the pseudo-label space Y, defined as

$$Y = P^T X, \tag{1}$$

where $P \in \mathbb{R}^{n \times c}$ denotes the projection matrix and $Y \in 0, 1^{c \times m}$ represents the label indicator matrix. Ideally, each column of Y contains exactly one entry equal to 1, while the remaining entries are 0 [34]. In the projection process described in Eq. (1), it is reasonable to assume that similar samples are more likely to belong to the same category. This assumption can be incorporated as a minimization problem:

$$\begin{aligned} &\min_{P,\Omega} \sum_{i,j=1}^{m} \left\| P^T \Omega x_i - P^T \Omega x_j \right\|_2^2 S_{ij} + \lambda \|P\|_F^2 \\ &\text{s.t.} \quad P^T X X^T P = I, \quad \sum_{i=1}^{n} \omega_i = 1, \quad \omega_i \geq 0. \end{aligned} \tag{2}$$

where S represents the similarity matrix capturing relationships between samples. To enable feature weighting, let $w = [\omega_1, \omega_2, \ldots, \omega_n]$ and define Ω as a diagonal matrix with entries $\sqrt{\omega_i}$. The discrete constraint on Y is relaxed by imposing $P^T X X^T P = I$ for tractable optimization.

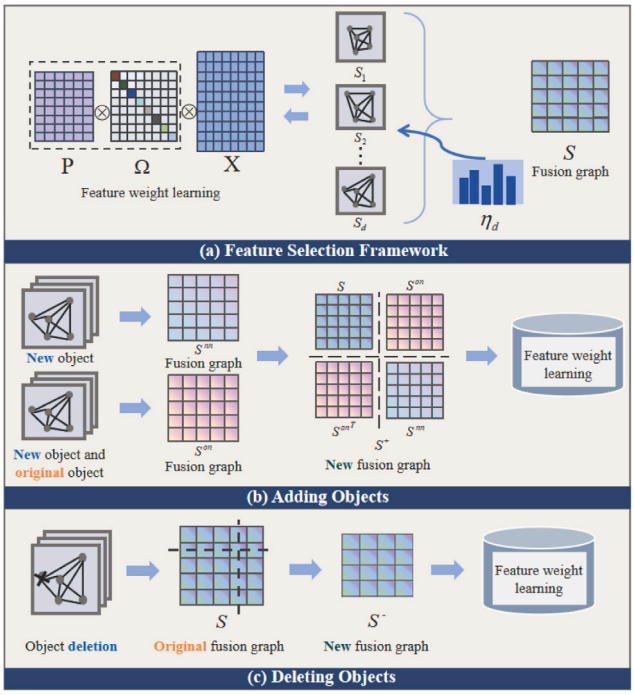


Fig. 2. The framework of feature selection methods for object dynamics. (a) Feature selection via multi-graph fusion. (b) Fusion graph update after object addition. (c) Fusion graph update after object deletion.

Since a single similarity graph cannot effectively capture complex relationships and is sensitive to noise, we integrate multiple graphs to construct a unified graph that better reflects the intrinsic data structure. Suppose we have d different similarity graphs $S_1, S_2, \ldots, S_d$ obtained using different distance metrics, where $S_g \in \mathbb{R}^{m\times m}$. We then learn the feature weights and the unified similarity graph by solving the following optimization problem:

$$\min_{P,S,\Omega,\eta} \sum_{i,j=1}^{m} \left\| P^T\Omega x_i - P^T\Omega x_j \right\|_2^2 S_{ij} + \lambda \|P\|_F^2 + \gamma \sum_{g=1}^{d} \eta_g \|S - S^g\|_F^2,$$
$$\text{s.t.} \quad P^T\Omega X X^T \Omega P = I, \sum_{i=1}^{n} \omega_i = 1, \omega_i \geq 0, \tag{3}$$
$$\sum_{j=1}^{m} S_{ij} = 1, 0 \leq S_{ij} \leq 1, S_{ii} = 0.$$

where η_g denotes the weight assigned to the gth graph, reflecting its varying contribution to the final unified graph. Since graphs constructed from different distance metrics may differ in scale, normalization is applied prior to fusion. Specifically, each graph is normalized as $S^g \leftarrow (D^g)^{-1}S^g$, where D^g is a diagonal matrix whose ith diagonal entry is $D^g_{ii} = \sum_j S^g_{ij}$.

As presented in Eq. (3), the proposed approach jointly incorporates feature weight learning, multi-graph fusion, and graph-regularized label-space projection into a unified optimization framework for effective feature selection. To address evolving data, we further design dynamic local update strategies capable of adapting to object additions and deletions without requiring global re-optimization.

4.2. A-DIUO: Updating mechanism for object addition

In a dynamic data environment, as the number of objects in the dataset increases, the similarity graph constructed with objects as nodes expands accordingly. To avoid global reconstruction each time, the fused graph obtained by multi-graph fusion should be updated locally in time.

Definition 1. Let $DIVIS = (U \cup U', AT, F)$ be a dynamic interval-valued information system, where $U = \{x_1, x_2, \ldots, x_m\}$ is a set of existing objects and $U' = \{x_{m+1}, x_{m+2}, \ldots, x_{m+m'}\}$ indicates the dynamically added object set. In the static stage, a fused similarity graph $S_{\text{prior}} \in \mathbb{R}^{m\times m}$ is learned for U by solving the global optimization problem. In the incremental stage, instead of re-optimizing S_{prior}, we directly introduce it as prior knowledge and apply a row-wise scaling with budget parameters ρ, then

$$S = (1-\rho)S_{\text{prior}}, \tag{4}$$

which ensures that the row sum of each existing object becomes $1-\rho$.

Definition 2. For the cross-relations from existing to new objects, d candidate graphs $\{S^{on}_1, \ldots, S^{on}_d\}$ are predefined, where $S^{on}_g \in \mathbb{R}^{m\times m'}$. The fused S^{on} is obtained by solving the following local update problem:

$$\min_{S^{on},\,\eta^{on}} \sum_{i=1}^{m}\sum_{r=1}^{m'} \left\| P^T\Omega x_i - P^T\Omega x_{m+r} \right\|_2^2 S^{on}_{ir} + \gamma \sum_{g=1}^{d} \eta^{on}_g \|S^{on} - S^{on}_g\|_F^2,$$
$$\text{s.t.} \quad \sum_{r=1}^{m'} S^{on}_{ir} = \rho, 0 \leq S^{on}_{ir} \leq 1. \tag{5}$$

Similarly, for the relations among new objects, d candidate graphs $\{S^{nn}_1, \ldots, S^{nn}_d\}$ are predefined, with $S^{nn}_g \in \mathbb{R}^{m'\times m'}$. The fused S^{nn} is learned by

$$\min_{S^{nn},\,\eta^{nn}} \sum_{t=1}^{m'}\sum_{q=1}^{m'} \left\| P^T\Omega x_{m+t} - P^T\Omega x_{m+q} \right\|_2^2 S^{nn}_{tq} + \gamma \sum_{g=1}^{d} \eta^{nn}_g \|S^{nn} - S^{nn}_g\|_F^2,$$
$$\text{s.t.} \quad \sum_{q=1}^{m'} S^{nn}_{tq} = 1 - a_t, 0 \leq S^{nn}_{tq} \leq 1, S^{nn}_{tt} = 0. \tag{6}$$

where a_t denotes the mass assigned from the new object x_{m+t} to the old objects, which is determined by the column sums of S^{on}.

After performing local updates, two fused similarity graphs S^{on} and S^{nn} are obtained. The complete fused similarity graph can be constructed as follows:

$$S^{+}_{(m+m')\times(m+m')} = \begin{bmatrix} S_{m\times m} & S^{on}_{m\times m'} \\ S^{on\,T}_{m\times m'} & S^{nn}_{m'\times m'}. \end{bmatrix}$$

The fused graph S^+ is then incorporated into the feature weight learning framework. The feature weights are updated locally by solving the following optimization problem.

$$\min_{P,\Omega} \sum_{i,j=1}^{m+m'} \left\| P^T\Omega x_i - P^T\Omega x_j \right\|_2^2 S^{+}_{ij} + \lambda \|P\|_F^2,$$
$$\text{s.t.} \quad P^T\Omega X X^T \Omega P = I, \sum_{i=1}^{n} \omega^{+}_i = 1, \omega^{+}_i \geq 0. \tag{7}$$

In the local update process, the above optimization problem is solved using the block coordinate descent method.

4.3. D-DIUO: Updating mechanism for object deletion

As the number of objects in the dataset decreases, the similarity graph constructed with these objects as nodes is correspondingly reduced. Therefore, the fused graph obtained through multi-graph fusion also requires local updates.

Definition 3. Let $DIVIS = (U - U', AT, F)$ be an dynamic interval-valued information system, where $U = \{x_1, x_2, \ldots, x_m\}$ is a set of objects and $x_k \in U (1 \leq k \leq m)$. $U' = \{x_1, x_2, \ldots, x_{m'}\}$ indicates the dynamically deleted object set, with each object $x_r \in U' (1 \leq r \leq m')$. For any $i, j \in [1, m-m']$, we have $S = S_{prior}[1 : m-m', 1 : m-m']$. To maintain the constraints, we define the normalized matrix S^- as

$$S^{-}_{ij} = \frac{S_{ij}}{\sum_{q=1}^{m-m'} S_{iq}}, \quad i, j \in [1, m-m'], \tag{8}$$

The fused graph S^- is incorporated into the feature weight learning framework, and the feature weights are obtained through local updates by solving the following optimization problem.

$$\begin{aligned} &\min_{P,\Omega} \sum_{i,j=1}^{m-m'} \left\| P^T \Omega x_i - P^T \Omega x_j \right\|_2^2 S_{ij}^- + \lambda \|P\|_F^2, \\ &\text{s.t.} \quad P^T \Omega X X^T \Omega P = I, \sum_{i=1}^{n} \omega_i^- = 1, \omega_i^- \geq 0. \end{aligned} \tag{9}$$

In the local update process, the above optimization problem is still solved using the block coordinate descent method.

4.4. Optimization

Since the matrix P is coupled with the diagonal matrix Ω, we introduce matrix Φ to replace ΩP in order to simplify the optimization process. On this basis, the objective function can be reformulated as:

$$\begin{aligned} &\min_{\Phi,S,\Omega,\eta} \sum_{i,j=1}^{m} \|\Phi^T x_i - \Phi^T x_j\|_2^2 S_{ij} \\ &+\lambda \mathrm{Tr}\left(\Phi^T \mathrm{diag}(\omega)^{-1} \Phi\right) + \gamma \sum_{g=1}^{d} \eta_g \|S - S^g\|_F^2 \\ &\text{s.t.} \Phi^T X X^T \Phi = I, \sum_{i=1}^{n} \omega_i = 1, \ \omega_i \geq 0, \\ &\sum_{j=1}^{m} S_{ij} = 1, \ 0 \leq S_{ij} \leq 1, \ S_{ii} = 0. \end{aligned} \tag{10}$$

4.4.1. Optimization of S with fixed variables

After fixing Φ, Ω, and η, and replacing B_{ij} with $\|\Phi^T x_i - \Phi^T x_j\|_2^2$, the optimization problem of S can be simplified as follows:

$$\begin{aligned} &\min_{S} \sum_{i,j=1}^{m} B_{ij} S_{ij} + \gamma \sum_{g=1}^{d} \eta_g \|S - S^g\|_F^2 \\ &\text{s.t.} \sum_{j=1}^{m} S_{ij} = 1, \ 0 \leq S_{ij} \leq 1, \ S_{ii} = 0. \end{aligned} \tag{11}$$

The optimization objective of each row s_j is to minimize the discrepancy with candidate graphs and the reconstruction error between samples. This problem can be formulated as a quadratic optimization with linear constraints, and a closed-form solution is derived by constructing the Lagrangian function and applying the KKT conditions. The final update form of s_j is:

$$s_j^* = u + \xi^* e, \tag{12}$$

where u and ξ^* are obtained iteratively. The parameter ξ^* can be efficiently solved by the Newton method:

$$\xi_{t+1} = \xi_t - \frac{f(\xi_t)}{f'(\xi_t)}. \tag{13}$$

Note that the above solution is derived under the constraint $\sum_{j=1}^{m} S_{ij} = 1$. For a more general constant-budget constraint, the same optimization procedure still applies, with only minor adjustments to the corresponding multiplier term. In summary, the optimization of S can be achieved by solving each row separately and iteratively updating ξ^* until convergence.

4.4.2. Optimization of Φ with fixed variables

By holding the other variables fixed, the optimization problem with respect to Φ is given as:

$$\begin{aligned} &\min_{\Phi} \mathrm{Tr}\left(\Phi^T X L X^T \Phi\right) + \lambda \mathrm{Tr}\left(\Phi^T \mathrm{diag}(\omega)^{-1} \Phi\right) \\ &\text{s.t.} \Phi^T X X^T \Phi = I. \end{aligned} \tag{14}$$

where L denotes the Laplacian matrix associated with S. This problem can be equivalently expressed as:

$$\begin{aligned} &\min_{\Phi} \mathrm{Tr}\left(\Phi^T \left(X L X^T + \lambda \, \mathrm{diag}(\omega)^{-1}\right) \Phi\right) \\ &\text{s.t.} \ \Phi^T X X^T \Phi = I. \end{aligned} \tag{15}$$

The solution is obtained through generalized eigenvalue decomposition.

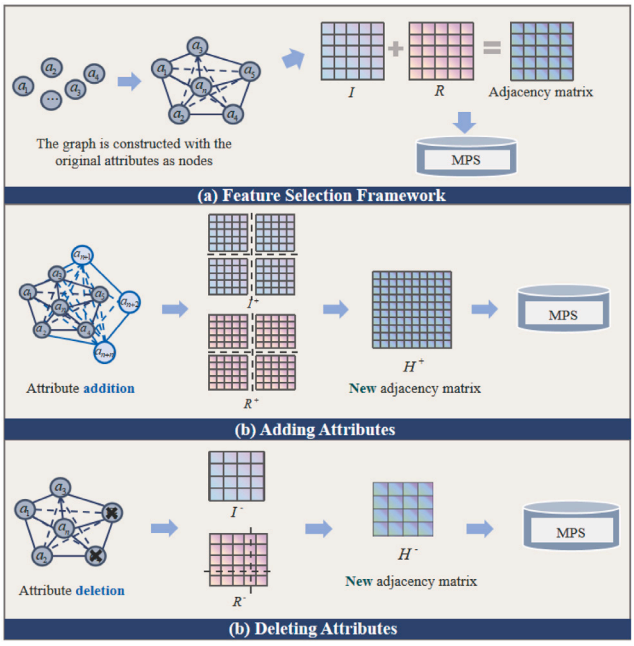


Fig. 3. The framework of feature selection methods for attribute dynamics. (a) Feature selection via weighted graph construction. (b) Adjacency matrix update after attribute addition. (c) Adjacency matrix update after attribute deletion.

4.4.3. Optimization of ω with fixed variables

For optimizing ω, the problem is:

$$\begin{aligned} &\min_{\omega} \mathrm{Tr}\left(\Phi^T \mathrm{diag}(\omega)^{-1} \Phi\right) \\ &\text{s.t.} \sum_{i=1}^{n} \omega_i = 1, \omega_i \geq 0. \end{aligned} \tag{16}$$

By utilizing the properties of diagonal matrices together with the Cauchy–Schwarz inequality, the closed-form solution can be derived as:

$$\omega_i = \frac{\sqrt{\sum_j \Phi_{ij}^2}}{\sum_i \sqrt{\sum_j \Phi_{ij}^2}}. \tag{17}$$

4.4.4. Optimization of η with fixed variables

With Φ, S, and Ω fixed, the graph weights are updated by an adaptive reweighting strategy according to the discrepancy between the fused graph S and each candidate graph S^g. Specifically, the weight of the gth graph is computed as

$$\eta_g = \frac{1}{2\sqrt{\|S - S^g\|_F^2}}. \tag{18}$$

This update assigns a smaller weight to a candidate graph with a larger deviation from the fused graph, and a larger weight to a graph that is more consistent with the current fused structure. In this way, the contributions of different candidate graphs can be automatically adjusted during the fusion process.

5. DIUA: Feature selection methods for attribute dynamics

To address dynamic attribute variation, we propose a graph-based local update strategy that constructs a weighted undirected connected graph over attributes. The update process for handling attribute variation is illustrated in Fig. 3. In this section, we detail the proposed feature selection method under attribute variation.

5.1. Feature selection framework

We first construct a graph based on the interval-valued information system, where attributes serve as nodes.

Definition 4. In interval-based feature selection, we construct a weighted undirected complete graph $G = \langle V, E\rangle$, where each node corresponds to a feature set $A = a_1, a_2, \ldots, a_n$, and edge weights are defined by the function $w(i,j)$. The adjacency matrix $H(i,j)$ is computed as

$$w(i,j) = \alpha I_{ij} + (1-\alpha)R_{ij}. \tag{19}$$

I_{ij} represents the maximum normalized standard deviation between the distributions of two features. A higher I_{ij} value indicates greater feature importance. R_{ij} represents the non-redundancy between features, we set $R_{ij} = 1 - \mid \rho(a_i, a_j) \mid$.

5.2. A-DIUA: Updating mechanism for attribute addition

In a dynamic data environment, as the number of attributes in the dataset increases, the similarity graph constructed with objects as nodes also expands accordingly. To avoid global reconstruction each time, the attribute-weighted graph needs to be updated locally in time. To support the update of I and R, we first introduce the necessary definitions for dominance relation, statistics and redundancy measure.

Definition 5. Let $b = (b^l, b^r)$ and $c = (c^l, c^r)$ be two interval values. The distance between the two interval values b and c is redefined as follows:

$$\begin{aligned} d(b,c) = \frac{1}{\sqrt{3}} \Big(\mid b^l - c^l \mid^2 + \mid \frac{b^l + b^r}{2} - \frac{c^l + c^r}{2} \mid^2 \\ + \mid b^r - c^r \mid^2 \Big)^{\frac{1}{2}}. \end{aligned} \tag{20}$$

Specifically, the proposed interval distance simultaneously measures the differences in the lower bound, midpoint, and upper bound, thereby capturing both the central tendency and the variation range of interval-valued data, and providing a more comprehensive characterization of interval uncertainty information.

Definition 6. let X be an interval-valued random variable. Its sample mean $\overline{X}$ and sample variance σ^2 can be defined as

$$\overline{X} = (\overline{X^l}, \overline{X^r}) = (\frac{1}{m}\sum_{i=1}^{m} {x_i}^l, \frac{1}{m}\sum_{i=1}^{m} {x_i}^r), \tag{21}$$

$$\sigma^2 = \frac{1}{m-1}\sum_{i=1}^{m} (d(x_i, \overline{X}))^2, \tag{22}$$

$$\sigma(X) = \sqrt{\sigma^2(X)} = (\frac{1}{m-1}\sum_{i=1}^{m} (d(x_i, \overline{X}))^2)^{\frac{1}{2}}, \tag{23}$$

where m represents the sample size. By computing sample statistics, population parameters can be estimated, leading to the determination of the relevant metrics I_{ij}.

Next, we give the definition of dominance relationship under interval valued information system.

Definition 7. Given an interval-valued information system $IVIS = (U, AT, F)$, for any $A \subseteq AT$ and $a_k \in A$, the mapping $f(x_i, a_k) = [a_{ik}^l, a_{ik}^r]$ denotes an interval-valued number. The dominance relation $D_A^{\leq}$ is defined as

$$D_A^{\leq} = \{(x_i, x_j) \in U \times U \mid \frac{a_{jk}^l + a_{jk}^r}{2} - \frac{a_{ik}^l + a_{ik}^r}{2} \geq \theta, \theta \geq 0\}.$$

where θ denotes a nonnegative number representing the allowable extent of deviation, serving as a tolerance threshold.

The ranking of object x_i in attribute a_j can be determined using the dominance relationship. To measure the redundancy between features, we employ the SRCC for representation and further define its computation method.

Definition 8. Let $IVIS = (U, AT, F)$ be an interval-valued information system, $\forall a_i, a_j \in AT$, SRCC of a_i and a_j is defined as

$$\rho_{i,j} = 1 - \frac{6\sum_{k=1}^{m}(\mid D_{\{a_i\}}^{-}(x_k) \mid - \mid D_{\{a_j\}}^{-}(x_k) \mid)^2}{m(m^2-1)}. \tag{24}$$

The redundancy metrics in the weighting function is denoted as $R_{ij} = 1 - \mid \rho(a_i, a_j) \mid$.

Definition 9. Let the dynamic interval-valued information system be defined as $DIVIS = (U, AT \cup A', F)$, where A' denotes the set of newly added attributes. A weighted undirected complete graph $G^+ = \langle V^+, E^+\rangle$ is constructed, the elements of the adjacency matrix $H^+(i,j)$ of this graph can be defined using a weight function $w^+(.,.)$ as follows:

$$w^+(i,j) = \alpha I_{ij}^+ + (1-\alpha)R_{ij}^+, \tag{25}$$

where I_{ij}^+ denotes the normalized maximum standard deviation between features, and $R_{ij}^+ = 1 - \mid \rho^+(a_i, a_j) \mid$ measures the non-redundancy between features.

The update of the matrix I^+ is divided into two cases:

- **Case 1:** When either the maximum or minimum value of the standard deviation remain unchanged after attribute expansion:
  - The original part remains unchanged, i.e., $I^+[1:n, 1:n] = I$;
  - For newly added attributes:
    $$I_{ij}^+ = \frac{max(\sigma(a_i), \sigma(a_j)) - min\{\sigma(a_k)\}}{max\{\sigma(a_k)\} - min\{\sigma(a_k)\}}, \tag{26}$$
    where $1 \leq k \leq n + n'$.
- **Case 2:** When either the maximum or minimum value of the standard deviation changes, update the extreme values accordingly and recalculate all I_{ij}^+ using Eq. (21).

For redundancy matrix R^+ updates:

- The original part remains unchanged, i.e., $R^+[1:n, 1:n] = R$;
- For any $i, j \in [n+1, n+n']$ or $i \in [1, n]$ and $j \in [n+1, n+n']$:
  $$R_{ij}^+ = 1 - \mid \rho(a_i, a_j) \mid \tag{27}$$

Finally, matrices I^+ and R^+ are combined to construct the updated adjacency matrix $H^+ \in \mathbf{R}^{(n+n')\times(n+n')}$, and the Matrix Power Series (MPS) method is employed to perform feature ranking.

5.3. D-DIUA: Updating mechanism for attribute deletion

As the number of attributes in the dataset decreases, the weighted graph built with these attributes as nodes will also decrease accordingly, so local updates are required. Similar to the addition case, the update relies on the dominance relation, statistics and redundancy measure introduced above, which provide the basis for recomputing I and R after attribute deletion.

Definition 10. Let $DIVIS = (U, AT - A', F)$ denote a dynamic interval-valued information system with the reduced attribute set, where A' contains the deleted attributes $a_r \in A'$ $(1 \leq r \leq n')$. A new graph $G^- = \langle V^-, E^-\rangle$ is constructed, with its adjacency matrix $H^-(i,j)$ defined by the weight function:

$$w^-(i,j) = \alpha I_{ij}^- + (1-\alpha)R_{ij}^-. \tag{28}$$

The update of the matrix I^- is divided into two cases:

- **Case 1:** When either the maximum or minimum value of the standard deviation remain unchanged after attribute deletion, then $I^- = I[1:n-n', 1:n-n']$;
- **Case 2:** When either the maximum or minimum value of the standard deviation changes, update the extreme values accordingly and recalculate all I_{ij}^- using Eq. (24).

$$I_{ij}^- = \frac{max(\sigma(a_i), \sigma(a_j)) - min\{\sigma(a_k)\}}{max\{\sigma(a_k)\} - min\{\sigma(a_k)\}}, \tag{29}$$

Similarly, the matrix R^- is updated by truncating the original R to the remaining attributes: $R^- = R[1:n-n', 1:n-n']$.

Finally, the updated H^- is used in the MPS to generate the new feature ranking. This local update strategy avoids global reconstruction, thus significantly reducing computational cost and enabling real-time adaptation to attribute variations.

5.4. Feature ranking process

After calculating an adjacency matrix constructed from the weight function, the MPS approach is applied for path selection to derive feature score rankings. The procedure can be outlined as follows. Let E be the identity matrix and $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ denote the eigenvalues of H. The parameter $\delta = 0.9/\varphi(H)$ is introduced as a scalar regularization factor, where $\varphi(H) = \max_{\epsilon_i \in \epsilon_1, \epsilon_2, \dots, \epsilon_n}(|\epsilon_i|)$ represents the spectral radius, which guarantees convergence of the infinite series. The regularized feature score is then defined as

$$\breve{\omega}(i) = [\breve{Y}\beta]_i, \tag{30}$$

where $\breve{Y} = (E - \delta H)^{-1} - E$, and β is a vector of ones.

Finally, features are ranked in descending order of $\breve{\omega}(i)$, with higher scores indicating stronger relevance and lower redundancy, making them more representative for constructing an optimized feature subset.

6. Feature selection algorithm

In this subsection, we propose four dynamic feature selection algorithms corresponding to the addition and deletion of instances and attributes, respectively.

6.1. Feature selection algorithm with adding objects

Algorithm 1 addresses the scenario of object addition and presents a complete dynamic feature selection process. First, the fused graph S_{prior} and feature weights ω obtained from the static phase are incorporated as prior knowledge. Then, similarity graphs are constructed both among new instances and between new and original instances as inputs for subsequent updates. Let M be the number of original instances, M' the number of newly added instances, and N the number of features. The overall time complexity is $\mathcal{O}\left(t_1 M M' + t_2 M'^2 + t_3 (M + M')^2 N\right)$.

6.2. Feature selection algorithm with deleting objects

Algorithm 2 addresses the scenario of object deletion and introduces an efficient dynamic feature selection strategy. Let M be the number of original instances, M' the number of deleted instances, and N the number of features. The time complexity of the algorithm is $\mathcal{O}\left((m-m')^2 + t(m-m')^2 N\right)$.

6.3. Feature selection algorithm with adding attributes

Algorithm 3 tackles attribute addition with an efficient dynamic feature selection strategy. Let M be the number of samples, N the number of original features, and N' the number of added features. The algorithm has a time complexity of $\mathcal{O}((N+N')^3)$.

Algorithm 1: A-DIUO

Input:

1. A dynamic interval-valued information system $DIVIS = (U \cup U', AT, F)$;
   Given the d similarity graphs of the new data and the d similarity graphs between the new and original data;
   The original fusion graph S_{prior} and original feature weight ω are already available.
2. The parameters λ and γ, $\varepsilon = 0.000001$;
3. A cutoff proportion p, where $p \in (0,1)$, representing the proportion of features to select.

Output: The optimal feature subset B.

begin
  Update S according to Definition 1; The iteration time t_1 is initialized to 0;
  while *not converged* **do**
    Update S^{on} and η^{on} by solving (5) according to Definition 2;
    Check the convergence condition $\left|\frac{\text{obj}^{t_1-1} - \text{obj}^{t_1}}{\text{obj}^{t_1}}\right| < \varepsilon$;
    Update $t_1 = t_1 + 1$;
  end
  The iteration time t_2 is initialized to 0;
  while *not converged* **do**
    Update S^{nn} and η^{nn} by solving (6) according to Definition 2;
    Check the convergence condition $\left|\frac{\text{obj}^{t_2-1} - \text{obj}^{t_2}}{\text{obj}^{t_2}}\right| < \varepsilon$;
    Update $t_2 = t_2 + 1$;
  end
  Incremental update S^+;
  The iteration time t_3 is initialized to 0;
  while *not converged* **do**
    Update ω^+ by solving (7) ;
    Check the convergence condition $\left|\frac{\text{obj}^{t_3-1} - \text{obj}^{t_3}}{\text{obj}^{t_3}}\right| < \varepsilon$;
    Update $t_3 = t_3 + 1$;
  end
  Sort ω^+ in descending order;
  Select the top p proportion of features as the feature subset B;
  return B.
end

Algorithm 2: D-DIUO

Input:

1. A dynamic interval-valued information system $DIVIS = (U - U', AT, F)$, where $U = \{x_1, x_2, \dots, x_m\}$. U' represents the set of removed objects, where $x_r \in U' (1 \le r \le m')$. The original fusion graph S_{prior} and original feature weight ω are already available.
2. The parameters λ and γ, $\varepsilon = 0.000001$;
3. A cutoff proportion p, where $p \in (0,1)$, representing the proportion of features to select.

Output: The optimal feature subset B.

begin
  Update $S = S_{prior}[1:n-n', 1:n-n']$;
  Row-normalize S to get S^- according to (8); The iteration time t is initialized to 0;
  while *not converged* **do**
    Update ω^- by solving (9);
    Check the convergence condition $\left|\frac{\text{obj}^{t-1} - \text{obj}^{t}}{\text{obj}^{t}}\right| < \varepsilon$;
    Update $t = t + 1$;
  end
  Sort ω^- in descending order;
  Select the top p proportion of features as the feature subset B;
  return B.
end

Algorithm 3: A-DIUA

Input:

1. A dynamic interval-valued information system $DIVIS = (U, AT \cup A', F)$, where $U = \{x_1, x_2, \dots, x_m\}$, $AT = \{a_1, a_2, \dots, a_n\}$. A' indicates the dynamically increased attribute set, where $a_r \in A'(n+1 \le r \le n+n')$. Already owned $\{\sigma(a_k) \mid 1 \le k \le n\}$ and $\{R_{ij} \mid 1 \le i, j \le n\}$;
2. The parameters α and θ, where $\alpha \in [0,1]$, $\theta \ge 0$;
3. A cutoff proportion p, where $p \in (0,1)$, representing the proportion of features to select.

Output: The optimal feature subset B.

begin
  for $k \leftarrow (n+1)$ *to* $(n+n')$ **do**
    Calculate $\sigma(a_k)$ according to Definition 6;
  end
  if $\max\{\sigma(a_k) \mid 1 \le k \le n\} = \max\{\sigma(a_k) \mid n+1 \le k \le n+n'\}$ ***and*** $\min\{\sigma(a_k) \mid 1 \le k \le n\} = \min\{\sigma(a_k) \mid n+1 \le k \le n+n'\}$ **then**
    for $i \leftarrow (n+1)$ **to** $(n+n')$ **do**
      for $j \leftarrow (n+1)$ **to** $(n+n')$ **do**
        Calculate I_{ij}^{+} according to Equation(27);
      end
    end
  end
  else
    for $i \leftarrow 1$ **to** $(n+n')$ **do**
      for $j \leftarrow 1$ **to** $(n+n')$ **do**
        Update I_{ij}^{+} according to Equation(27);
      end
    end
  end
  for $i \leftarrow (n+1)$ *to* $(n+n')$ **do**
    for $j \leftarrow (n+1)$ *to* $(n+n')$ **do**
      Calculate R_{ij}^{+} according to Equation(28);
    end
  end
  for $i \leftarrow 1$ *to* n **do**
    for $j \leftarrow (n+1)$ *to* $(n+n')$ **do**
      Calculate R_{ij}^{+} according to Equation(28);
    end
  end
  Update matrix I^{+} and R^{+};
  $H^{+}(i,j) = \alpha I_{ij}^{+} + (1-\alpha)R_{ij}^{+}$;
  Update adjacency matrix H^{+};
  $\delta = 0.9/\varphi(QH^{+})$;
  $\breve{\Psi} = (E - \delta H^{+})^{-1} - E$;
  $\breve{\omega}^{+}(i) = [\breve{\Psi}\beta]_i$;
  Sort $\breve{\omega}^{+}$ in descending order;
  Select the top p proportion of features as the feature subset B;
  return B.
end

Algorithm 4: D-DIUA

Input:

1. A dynamic interval-valued information system $DIVIS = (U, AT - A', F)$, where $U = \{x_1, x_2, \dots, x_m\}$, $AT = \{a_1, a_2, \dots, a_n\}$. A' represents the set of removed attributes, where $a_r \in A'(1 \le r \le n')$. Already owned $\{\sigma(a_k) \mid 1 \le k \le n\}$ and $\{R_{ij} \mid 1 \le i, j \le n\}$;
2. The parameters α and θ, where $\alpha \in [0,1]$, $\theta \ge 0$;
3. A cutoff proportion p, where $p \in (0,1)$, representing the proportion of features to select.

Output: The optimal feature subset B.

begin
  if $\max\{\sigma(a_k) \mid 1 \le k \le n\} = \max\{\sigma(a_k) \mid 1 \le k \le n-n'\}$ ***and*** $\min\{\sigma(a_k) \mid 1 \le k \le n\} = \min\{\sigma(a_k) \mid 1 \le k \le n-n'\}$ **then**
    $I^{-} = I[1 : n-n', 1 : n-n']$
  end
  else
    for $i \leftarrow 1$ **to** $(n-n')$ **do**
      for $j \leftarrow 1$ **to** $(n-n')$ **do**
        Update I_{ij}^{-} according to Equation(30);
      end
    end
  end
  Update matrix $R^{-} = R[1 : n-n', 1 : n-n']$;
  $H^{-}(i,j) = \alpha I_{ij}^{-} + (1-\alpha)R_{ij}^{-}$;
  Update adjacency matrix H^{-};
  $\delta = 0.9/\varphi(QH^{-})$;
  $\breve{\Psi} = (E - \delta H^{-})^{-1} - E$;
  $\breve{\omega}^{-}(i) = [\breve{\Psi}\beta]_i$;
  Sort $\breve{\omega}^{-}$ in descending order;
  Select the top p proportion of features as the feature subset B;
  return B.
end

6.4. Feature selection algorithm with deleting attributes

Algorithm 4 deals with attribute deletion using an efficient dynamic feature selection strategy. Let M be the number of samples, N the number of original features, and N' the number of deleted features. The algorithm has a time complexity of $\mathcal{O}((N-N')^3)$.

7. Experimental analysis

To evaluate the effectiveness and performance of the proposed unsupervised graph-based dynamic feature selection algorithm, a series of comprehensive experiments were conducted.

A total of twelve benchmark datasets were drawn from the UCI Machine Learning Repository and Kaggle, as summarized in Table 2. The algorithms were implemented in Python and executed on a workstation configured with an AMD Ryzen 7 5800H processor, 16 GB of memory, and the Windows 11 operating system. The overall experimental workflow is outlined below:

(1) *Data preprocessing*: Because all twelve datasets contain real-valued attributes, they are first transformed into interval-valued data. For each attribute $a \in AT$, let σ_a denote the standard deviation of attribute a over the whole dataset, and let $B_a(x)$ be the normalized attribute value of object x. Then the interval corresponding to $a(x)$ is defined as $a(x)^{l} = a(x) - \sigma_a\big(0.5 + 0.5B_a(x)\big)$ and $a(x)^{r} = a(x) + \sigma_a\big(0.5 + 0.5B_a(x)\big)$.

(2) *Dynamic Experimental Design*: Each dataset is evenly split into two parts: one as the initial data and the other for dynamic simulation. In object addition and deletion experiments, the dynamic part is processed in five stages with ratios of 0.1, 0.2, 0.3, 0.4, and 0.5. In the multi-graph fusion module, four candidate graphs (d = 4) are used: Euclidean KNN, cosine similarity, SNN, and adaptive local scaling, with k = 10.

(3) *Performance evaluation:*We compare the proposed dynamic feature selection algorithm with eight representative feature selection algorithms: GRRS [35], W-MGMN [28], GLSFS [24], Inf-UFS [20], BPNN, KNCMI [33], MGF^{2}WL [34], and the IGUFS [36]. For the baseline methods, we conducted our own re-implementations strictly following the original mathematical frameworks, with hyperparameters consistent with the settings reported in the original publications. Evaluation metrics include both feature reduction time and clustering performance. Clustering performance is assessed using KMeans and DBSCAN. For KMeans, the number of clusters c is set to the true number of classes for each dataset. For DBSCAN, the parameters are set to $\varepsilon = 0.9$ and *min_samples* = 4. The clustering results are reported in terms of the average Adjusted Rand Index (ARI) and Normalized

Table 2
Summary of the experimental datasets

No.	Datasets	Objects	Attributes	Classes
1	Gait Classfication	48	321	16
2	TOX 171	171	5748	4
3	Ultrasonic flowmeter diagnostics	181	43	4
4	Ionosphere	351	34	2
5	Dermatology	358	35	6
6	Glioma Grading Clinical and Mutation Features	839	23	2
7	Autism screening data for toddlers	1054	19	2
8	Customer Churn	3150	13	2
9	Travel Review Ratings	5456	24	9
10	Parkinsons Telemonitoring	5875	19	42
11	Shill Bidding	6321	10	2
12	Dry Bean	13611	16	7

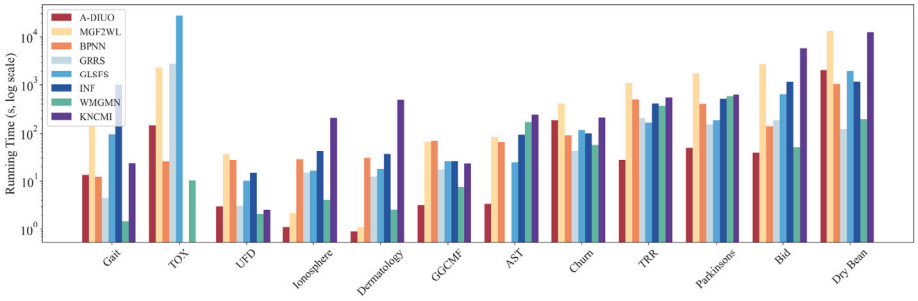


Fig. 4. Running time of algorithms with adding objects.

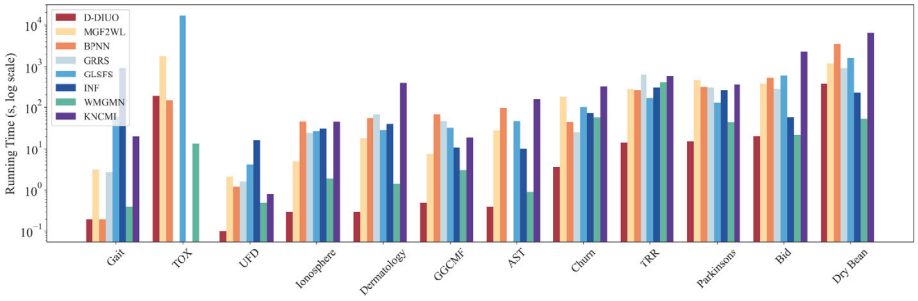
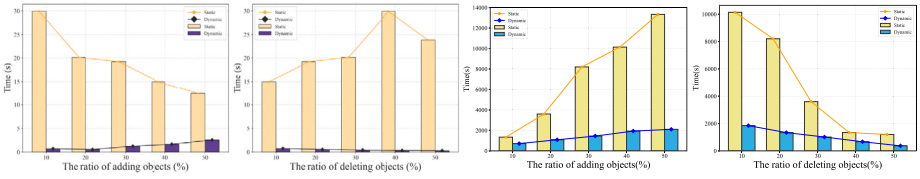


Fig. 5. Running time of algorithms with deleting objects.



(a) GGCMF (b) GGCMF (c) Dry Bean (d) Dry Bean

Fig. 6. Comparison of static and dynamic execution time under object dynamics on GGCMF and Dry Bean datasets.

Mutual Information (NMI). All experiments are repeated 10 times using random seeds from 42 to 51.

(4) *Parameter Sensitivity Analysis*: We test four key parameters. For the object variation algorithm, we evaluate $\lambda \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1\}$. For the attribute variation algorithm, we analyze the sensitivity of the parameter α, with its values ranging from 0 to 1 in increments of 0.1, and $\theta \in \{0.001, 0.01, 0.1, 1\}$.

7.1. Effectiveness of DIUO under object variations

7.1.1. Analysis of the time efficiency

To evaluate efficiency under object-level dynamics, we report the runtime of various algorithms during adding or removing 50% of the objects, as shown in Figs. 4 and 5. For algorithms with run times exceeding 24 h, we set their values to 0, excluding them from the comparison. Results show that our dynamic algorithms A-DIUO and D-DIUO maintain clear efficiency advantages on most datasets. This advantage becomes more evident particularly on large datasets like Bid, Dry Bean, along with TRR, in which the runtime is substantially reduced compared with other methods. These findings indicate that our dynamic strategies are both effective and scalable in handling object-level changes.

We further compare the runtime of static and dynamic algorithms under object dynamics on GGCMF and the large-scale Dry Bean dataset, as shown in Fig. 6. The results indicate that the dynamic algorithms

Table 3
Average ARI of algorithms on datasets with adding objects (50%).

Datasets	Original	GRRS	GLSFS	Inf-UFS	W-MGMN	KNCMI	BPNN	MGF^{2}WL	A-DIUO
Gait	25.59 ± 2.93	27.25 ± 5.45	26.58 ± 6.34	27.38 ± 2.78	16.63 ± 7.06	21.87 ± 7.53	26.52 ± 3.08	22.13 ± 4.05	**27.69 ± 3.42**
TOX	7.86 ± 1.28	11.85 ± 2.42	**11.97 ± 3.47**	–	4.69 ± 2.91	–	6.86 ± 1.10	10.28 ± 2.16	8.63 ± 0.70
UFD	10.14 ± 0.00	4.48 ± 1.45	11.70 ± 1.38	9.81 ± 0.00	6.58 ± 3.22	13.59 ± 3.15	**16.18 ± 2.37**	10.14 ± 0.00	11.57 ± 2.37
Ionosphere	44.08 ± 0.07	17.61 ± 4.67	**51.78 ± 1.39**	34.27 ± 0.00	24.81 ± 10.59	13.22 ± 5.21	22.93 ± 2.60	39.76 ± 0.20	40.25 ± 00.58
Dermatology	51.77 ± 2.42	30.68 ± 1.44	53.11 ± 2.18	58.26 ± 0.45	25.34 ± 12.43	27.31 ± 3.67	31.57 ± 4.34	57.47 ± 2.68	**60.88 ± 7.32**
GGCMF	25.52 ± 0.00	**29.88 ± 5.53**	1.96 ± 0.41	25.78 ± 0.10	5.01 ± 11.76	13.28 ± 2.51	17.84 ± 11.2	26.34 ± 0.00	**29.69 ± 1.57**
AST	30.04 ± 0.00	–	23.01 ± 0.00	32.81 ± 0.09	18.25 ± 13.24	21.96 ± 4.53	24.94 ± 2.34	29.55 ± 0.06	**34.18 ± 0.54**
Customer Churn	33.62 ± 0.00	36.69 ± 0.00	36.72 ± 0.00	33.61 ± 0.00	22.17 ± 18.51	14.41 ± 8.16	13.55 ± 20.47	29.96 ± 0.00	**39.77 ± 0.00**
TRR	27.08 ± 1.97	18.94 ± 1.23	12.68 ± 0.01	21.18 ± 0.73	11.00 ± 3.07	22.41 ± 1.33	17.78 ± 2.33	23.52 ± 0.54	**26.54 ± 0.97**
Parkinsons	18.59 ± 0.38	31.11 ± 5.08	18.91 ± 3.64	19.02 ± 0.33	3.03 ± 2.00	18.70 ± 0.51	18.17 ± 0.38	29.31 ± 0.47	**32.27 ± 0.42**
Bid	0.10 ± 0.00	18.19 ± 19.6	**42.06 ± 1.02**	0.17 ± 0.04	18.76 ± 20.19	8.30 ± 3.47	7.36 ± 2.92	1.12 ± 0.02	19.26 ± 0.30
Dry Bean	31.32 ± 0.04	33.62 ± 2.60	31.62 ± 5.04	22.98 ± 0.01	22.63 ± 9.01	22.27 ± 1.93	32.92 ± 1.34	27.85 ± 1.06	**34.05 ± 0.03**
Average	25.48 ± 0.76	23.66 ± 4.50	26.84 ± 2.07	25.93 ± 0.41	14.91 ± 9.50	17.94 ± 3.82	19.72 ± 4.54	25.62 ± 0.94	**30.40 ± 1.52**

Table 4
Average ARI of algorithms on datasets with deleting objects (50%).

Datasets	Original	GRRS	GLSFS	Inf-UFS	W-MGMN	KNCMI	BPNN	MGF^{2}WL	D-DIUO
Gait	21.83 ± 6.06	20.47 ± 8.77	28.49 ± 4.62	29.12 ± 9.09	20.34 ± 8.42	21.13 ± 6.72	24.09 ± 9.57	24.40 ± 6.57	**29.32 ± 9.54**
TOX	6.88 ± 1.26	–	10.82 ± 0.84	–	2.83 ± 3.38	–	7.43 ± 2.11	**15.79 ± 6.51**	11.41 ± 0.69
UFD	14.62 ± 2.73	6.20 ± 2.67	13.97 ± 1.74	11.56 ± 2.36	8.45 ± 3.98	10.88 ± 2.41	**14.85 ± 4.33**	13.35 ± 3.70	**14.20 ± 3.96**
Ionosphere	39.10 ± 5.64	25.62 ± 10.23	40.09 ± 4.09	33.79 ± 6.27	11.42 ± 3.45	26.15 ± 5.66	10.40 ± 3.59	**43.74 ± 3.65**	**43.52 ± 6.67**
Dermatology	46.55 ± 3.83	30.51 ± 5.98	41.57 ± 4.90	41.21 ± 5.81	16.82 ± 6.08	35.93 ± 5.46	26.16 ± 9.30	59.31 ± 4.89	**60.48 ± 7.20**
GGCMF	21.01 ± 8.19	45.17 ± 5.93	36.74 ± 2.36	21.00 ± 8.94	3.99 ± 2.37	3.34 ± 4.07	10.20 ± 3.10	24.43 ± 13.70	**28.58 ± 2.27**
AST	28.50 ± 2.57	–	30.68 ± 4.97	35.99 ± 3.04	18.01 ± 8.24	22.72 ± 5.39	21.04 ± 7.78	**48.28 ± 5.52**	31.85 ± 5.09
Customer Churn	33.84 ± 1.21	40.31 ± 1.23	27.98 ± 12.22	33.95 ± 2.95	14.91 ± 8.09	15.04 ± 11.14	18.11 ± 8.90	26.30 ± 2.30	**36.24 ± 4.59**
TRR	27.11 ± 1.76	19.51 ± 2.50	22.77 ± 1.82	11.16 ± 1.29	17.32 ± 2.90	17.36 ± 1.70	17.92 ± 2.10	22.50 ± 2.32	**26.14 ± 1.65**
Parkinsons	18.61 ± 0.40	30.53 ± 1.94	17.02 ± 0.04	18.56 ± 0.37	13.60 ± 1.13	19.22 ± 0.36	6.33 ± 6.12	**31.81 ± 2.93**	**31.54 ± 1.55**
Bid	0.15 ± 0.09	42.03 ± 0.41	42.16 ± 0.52	0.11 ± 0.03	9.87 ± 7.97	0.85 ± 1.88	0.95 ± 2.59	26.38 ± 0.47	**48.17 ± 0.13**
Dry Bean	31.15 ± 0.28	35.07 ± 0.92	22.80 ± 0.28	23.66 ± 0.36	22.33 ± 2.33	21.18 ± 6.26	31.44 ± 4.39	30.95 ± 4.31	**33.91 ± 0.09**
Average	24.11 ± 2.84	29.54 ± 4.06	27.92 ± 3.20	23.65 ± 3.68	13.32 ± 4.86	17.62 ± 4.64	15.74 ± 5.32	30.60 ± 4.74	**32.95 ± 3.62**

consistently outperform the static ones in all cases, especially at higher update rates. This improvement is attributed to the local update strategy adopted by the dynamic algorithms, which avoids full graph reconstruction and greatly reduces computation time, thereby demonstrating superior efficiency and scalability.

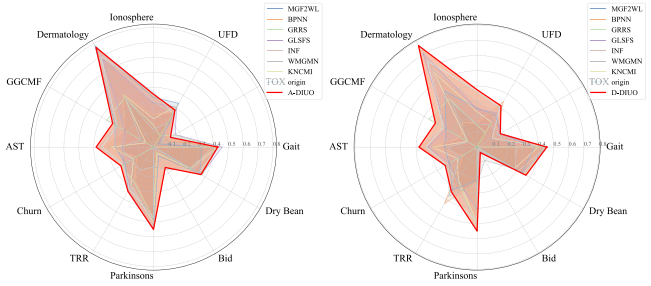
7.1.2. Analysis of the clustering performance

Tables 3 and 4, together with Fig. 7, present the average ARI and NMI when adding or deleting 50% of objects. As observed in Table 3, when 50% of objects are added, A-DIUO achieves the highest average ARI (30.40%), notably outperforming the static MGF^{2}WL (25.62%). A similar trend is evident in the deletion scenarios shown in Table 4, where D-DIUO maintains a leading average ARI of 32.95%. Furthermore, the radar charts in Fig. 7 visually confirm these advantages across multiple evaluation dimensions. The red solid lines, representing our proposed dynamic algorithms, consistently encompass a larger area than other comparative methods across most datasets. It can be seen that our proposed dynamic algorithms A-DIUO and D-DIUO achieve competitive or superior performance compared with other methods in most cases. This shows that the local update strategy can greatly reduce computational cost without harming clustering quality.

7.2. Effectiveness of DIUA under attribute variations

7.2.1. Analysis of the time efficiency

To evaluate efficiency under attribute-level dynamics, we report the running time of different algorithms when adding or deleting 50% of objects as shown in Figs. 8 and 9. It is clear that our proposed methods A-DIUA and D-DIUA achieve the lowest execution time across most datasets, especially on large-scale ones such as Bid and Dry Bean. Compared with other competitive algorithms, the dynamic methods consistently maintain a computational advantage, which highlights the efficiency of the local update mechanism in attribute-level scenarios.



(a) Addition objects (b) Deletion objects

Fig. 7. Average NMI of algorithms when updating objects (50%) on datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 10 further compares the static and dynamic execution time under attribute dynamics on GGCMF and the large-scale Dry Bean dataset. The results show that the static baseline requires re-executing the entire selection process, leading to a rapid increase in runtime as the number of attributes grows. In contrast, the dynamic methods scale smoothly and achieve substantial time savings. This confirms that the proposed dynamic update strategy significantly improves computational efficiency without sacrificing effectiveness, making it more suitable for large-scale or frequently changing datasets.

7.2.2. Analysis of the clustering performance

Tables 5 and 6 together with Fig. 11 show the clustering performance in terms of ARI and NMI under attribute dynamics. When adding attributes, our proposed algorithm A-DIUA achieves the best average ARI (32.44%), outperforming all competing methods. On several datasets such as Dermatology, GGCMF and TRR, it delivers substantial gains, showing that the dynamic update strategy effectively captures

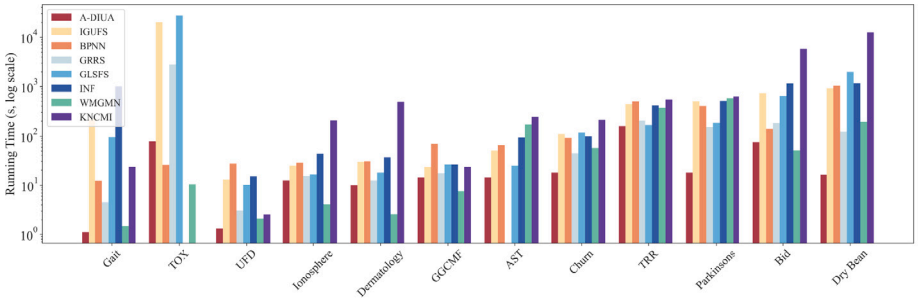


Fig. 8. Running time of algorithms with adding attributes.

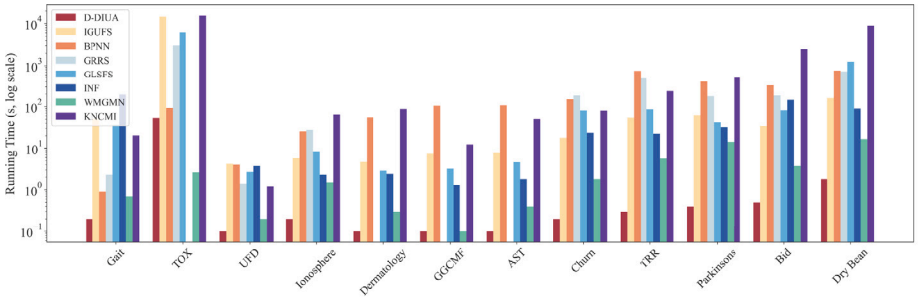
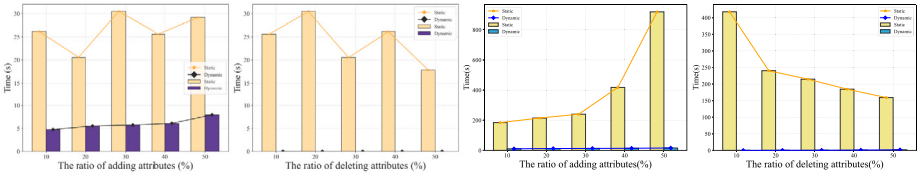


Fig. 9. Running time of algorithms with deleting attributes.



(a) GGCMF (b) GGCMF (c) Dry Bean (d) Dry Bean

Fig. 10. Comparison of static and dynamic execution time under attribute dynamics on GGCMF and Dry Bean datasets.

Table 5
Average ARI of algorithms on datasets with adding attributes (50%).

Datasets	Original	GRRS	GLSFS	Inf-UFS	W-MGMN	KNCMI	BPNN	IGUFS	A-DIUA
Gait	25.59 ± 2.93	27.25 ± 5.45	26.58 ± 6.34	27.38 ± 2.78	21.46 ± 7.21	21.87 ± 7.53	26.52 ± 3.08	33.12 ± 1.32	**39.77 ± 2.81**
TOX	7.86 ± 1.28	11.85 ± 2.42	11.97 ± 3.47	–	4.78 ± 4.41	–	6.86 ± 1.10	5.94 ± 5.66	**16.73 ± 4.01**
UFD	10.14 ± 0.00	4.48 ± 1.45	11.70 ± 1.38	9.81 ± 0.00	6.68 ± 2.86	13.59 ± 3.15	**16.18 ± 2.37**	14.21 ± 0.01	14.29 ± 0.90
Ionosphere	44.08 ± 0.07	17.61 ± 4.67	**51.78 ± 1.39**	34.27 ± 0.00	15.89 ± 9.19	13.22 ± 5.21	22.93 ± 2.60	33.80 ± 0.04	40.72 ± 0.54
Dermatology	51.77 ± 2.42	30.68 ± 1.44	53.11 ± 2.18	58.26 ± 0.45	22.01 ± 19.60	27.31 ± 3.67	31.57 ± 4.34	50.55 ± 1.79	**60.94 ± 0.18**
GGCMF	25.52 ± 0.00	29.88 ± 5.53	1.96 ± 0.41	25.78 ± 0.10	7.09 ± 11.83	13.28 ± 2.51	17.84 ± 11.2	25.84 ± 0.00	**29.90 ± 0.02**
AST	30.04 ± 0.00	–	23.01 ± 0.00	32.81 ± 0.09	16.99 ± 14.02	21.96 ± 4.53	24.94 ± 2.34	24.26 ± 0.66	**33.68 ± 0.20**
Customer Churn	33.62 ± 0.00	36.69 ± 0.00	**36.72 ± 0.00**	33.61 ± 0.00	8.24 ± 8.24	14.41 ± 8.16	13.55 ± 20.47	33.56 ± 0.00	35.41 ± 3.37
TRR	**27.08 ± 1.97**	18.94 ± 1.23	12.68 ± 0.01	21.18 ± 0.73	14.95 ± 6.09	22.41 ± 1.33	17.78 ± 2.33	21.02 ± 1.57	**25.66 ± 1.32**
Parkinsons	18.59 ± 0.38	**31.11 ± 5.08**	18.91 ± 3.64	19.02 ± 0.33	7.57 ± 6.18	18.70 ± 0.51	18.17 ± 0.38	19.71 ± 0.01	19.47 ± 0.33
Bid	0.10 ± 0.00	18.19 ± 19.6	**42.06 ± 1.02**	0.17 ± 0.04	9.88 ± 17.11	8.30 ± 3.47	7.36 ± 2.92	19.05 ± 0.11	37.86 ± 12.61
Dry Bean	31.32 ± 0.04	33.62 ± 2.60	31.62 ± 5.04	22.98 ± 0.01	21.08 ± 1.74	22.27 ± 1.93	32.92 ± 1.34	25.68 ± 2.56	**34.87 ± 0.73**
Average	25.48 ± 0.76	23.66 ± 4.50	26.84 ± 2.07	25.93 ± 0.41	13.05 ± 9.04	17.94 ± 3.82	19.72 ± 4.54	25.56 ± 1.14	**32.44 ± 2.24**

useful information from newly added attributes. For attribute deletion, D-DIUA still maintains competitive performance and performs better than most baselines. The expansive coverage of our methods in Fig. 11 further confirms their superior stability and adaptability across diverse datasets. These results indicate that by locally updating the graph structures, our dynamic framework can effectively preserve essential feature information and yield more accurate clustering results than traditional global retraining approaches.

7.3. Statistical testing

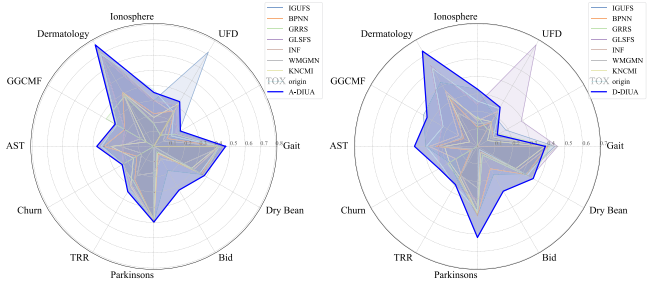
We performed the Friedman and Nemenyi post-hoc tests to evaluate the statistical significance of the results. Algorithms that exceeded the 24-hour execution limit on large datasets were assigned the last place (lowest rank) for those cases. With the significance level α set to 0.05, the results in Table 7 show that all p-values are below the threshold, leading to the rejection of the null hypothesis. These findings

Table 6
Average ARI of algorithms on datasets with deleting attributes (50%).

Datasets	Original	GRRS	GLSFS	Inf-UFS	W-MGMN	KNCMI	BPNN	IGUFS	D-DIUA
Gait	**26.19 ± 4.29**	9.85 ± 3.27	22.56 ± 3.83	20.11 ± 4.87	17.93 ± 7.21	16.65 ± 3.94	**24.09 ± 9.57**	16.306 ± 3.33	17.12 ± 3.31
TOX	7.27 ± 1.48	8.28 ± 2.57	8.76 ± 3.81	–	6.32 ± 2.22	4.65 ± 3.28	7.43 ± 2.11	8.56 ± 0.73	**9.46 ± 0.60**
UFD	9.21 ± 1.36	4.61 ± 2.83	5.19 ± 1.79	3.74 ± 0.38	5.20 ± 4.04	4.91 ± 2.77	**16.85 ± 4.33**	9.66 ± 0.21	10.11 ± 2.62
Ionosphere	28.64 ± 2.28	20.02 ± 4.50	22.06 ± 4.64	15.02 ± 3.18	13.59 ± 8.55	6.16 ± 2.39	18.99 ± 3.84	38.03 ± 0.27	**39.98 ± 0.05**
Dermatology	34.40 ± 9.18	–	33.79 ± 5.44	26.21 ± 4.35	14.38 ± 9.29	23.43 ± 10.27	13.08 ± 14.65	36.02 ± 1.48	**52.53 ± 1.23**
GGCMF	20.15 ± 13.93	–	15.77 ± 15.69	9.91 ± 11.92	3.10 ± 3.79	11.44 ± 2.29	10.20 ± 3.11	16.48 ± 0.19	**33.49 ± 0.61**
AST	21.92 ± 4.70	–	19.68 ± 5.33	20.45 ± 4.43	19.35 ± 12.44	17.85 ± 10.72	21.04 ± 4.25	29.73 ± 1.53	**30.51 ± 2.37**
Customer Churn	34.16 ± 10.98	32.62 ± 14.05	31.33 ± 19.08	18.33 ± 23.42	10.04 ± 20.17	24.38 ± 24.40	8.11 ± 19.01	31.11 ± 17.52	**37.21 ± 10.51**
TRR	14.70 ± 2.93	17.72 ± 2.43	10.77 ± 2.87	11.16 ± 1.29	11.88 ± 5.25	12.90 ± 2.78	17.92 ± 2.06	16.08 ± 2.11	**18.09 ± 1.46**
Parkinsons	13.16 ± 4.80	17.64 ± 23.82	20.67 ± 15.66	11.24 ± 10.61	6.50 ± 6.09	13.75 ± 5.60	6.33 ± 7.18	18.38 ± 9.94	**23.59 ± 3.91**
Bid	1.35 ± 2.33	10.46 ± 18.16	13.33 ± 19.90	2.47 ± 3.24	4.40 ± 13.23	2.57 ± 2.98	0.95 ± 2.63	19.16 ± 7.23	**33.70 ± 16.72**
Dry Bean	28.30 ± 5.30	31.98 ± 3.17	23.69 ± 6.28	20.61 ± 6.74	25.77 ± 5.01	18.19 ± 5.97	31.44 ± 4.39	25.62 ± 5.75	**34.87 ± 0.73**
Average	19.95 ± 5.30	17.02 ± 8.31	18.67 ± 8.69	14.48 ± 6.77	11.54 ± 8.11	13.07 ± 6.45	14.70 ± 6.43	22.09 ± 4.19	**28.39 ± 3.68**

Table 7
Result of the Friedman test.

Scenario	χ_F^2	P value
Adding objects	42.06	5.06×10^{-7}
Deleting objects	27.80	2.38×10^{-4}
Adding attributes	36.78	5.15×10^{-6}
Deleting attributes	105.22	<.001



(a) Addition attributes (b) Deletion attributes

Fig. 11. Average NMI of algorithms when updating attributes (50%) on datasets.

confirm statistically significant differences in performance among the algorithms, which warrants the use of the Nemenyi test. As illustrated in Fig. 12, the four proposed dynamic algorithms achieve the lowest average ranks and consistently occupy the top-tier position. This confirms their statistically significant superiority over the compared methods at the significance level of $\alpha = 0.05$.

7.4. Parameter sensitivity analysis

To evaluate the impact of key parameters on clustering performance, Figs. 13 and 14 illustrate the average ARI trends under 50% object and attribute addition, respectively. The results show that the proposed method exhibits parameter sensitivity on certain datasets (e.g., TOX, UFD, TRR), where performance surfaces fluctuate significantly, indicating the need for careful tuning. In contrast, datasets such as Gait, Dermatology, and GGCMF under attribute perturbation show relatively smooth surfaces, reflecting strong robustness. Overall, the method achieves better ARI when γ or θ is set to moderate values, demonstrating good adaptability and stability across most scenarios.

8. Conclusions and future work

This paper proposes a unified graph-based dynamic feature selection framework for interval-valued information systems (IVIS), addressing four types of structural changes: the addition and deletion of both objects and attributes. For object variations, we integrate graph fusion with feature weight optimization, employing local updates to bypass costly global re-optimization. For attribute variations, dynamic ranking is achieved via locally updated adjacency matrices on an attribute-based graph. By performing local incremental updates to the corresponding matrices, the proposed algorithm efficiently adapts to structural dynamics with significantly reduced computational costs. Experimental results on multiple public datasets demonstrate that our approach maintains high feature selection accuracy while substantially validating its effectiveness in handling large-scale interval-valued data and in real-time applications.

Despite these results, challenges remain. Future research will focus on enhancing the algorithm's robustness for class-imbalanced data and incorporating adaptive parameter tuning to improve model flexibility. Furthermore, we plan to extend this dynamic framework to more complex data environments and explore additional techniques to further optimize the selection process and improve the interpretability of the results.

CRediT authorship contribution statement

Xuan Shen: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Investigation, Formal analysis, Data curation. **Xiaoyan Zhang:** Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

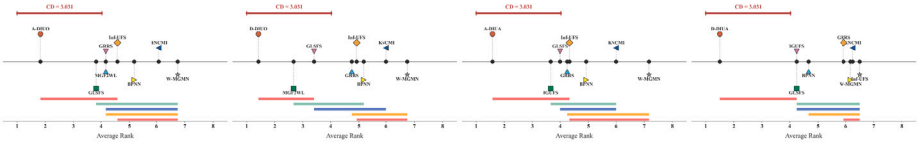
We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

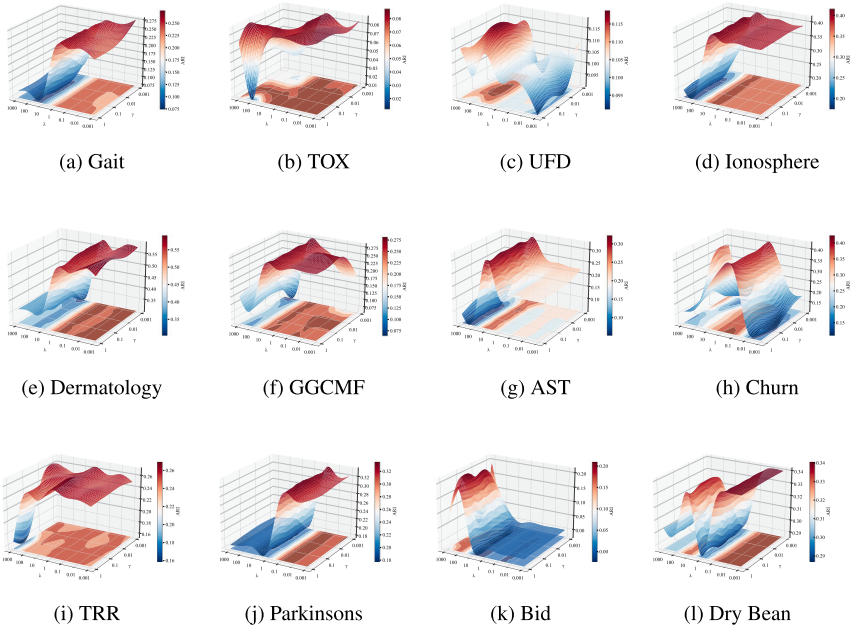
Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant NO. 12371465) and the Chongqing Natural Science Foundation of China (Grant NO. CSTB2023NSCQ-MSX1063).



(a) (b) (c) (d)

Fig. 12. Critical difference (CD) diagram of the compared algorithms based on the Nemenyi test ($\alpha = 0.05$).

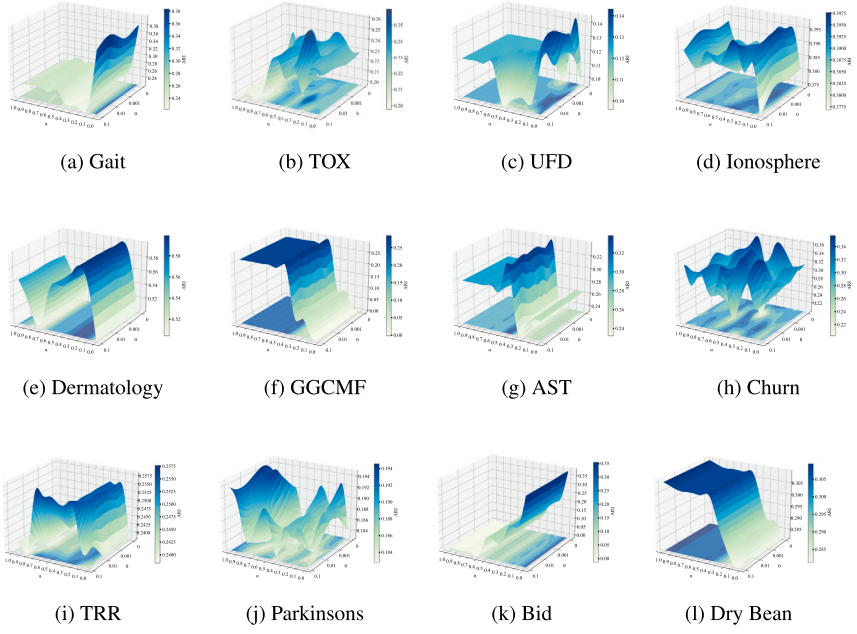


(a) Gait (b) TOX (c) UFD (d) Ionosphere

(e) Dermatology (f) GGCMF (g) AST (h) Churn

(i) TRR (j) Parkinsons (k) Bid (l) Dry Bean

Fig. 13. The average ARI for different parameters λ and γ when adding objects (50%).



(a) Gait (b) TOX (c) UFD (d) Ionosphere

(e) Dermatology (f) GGCMF (g) AST (h) Churn

(i) TRR (j) Parkinsons (k) Bid (l) Dry Bean

Fig. 14. The average ARI for different parameters α and θ when adding attributes (50%).

Data availability

The link of used data for the research described is http://archive.ics.uci.edu/ in our paper.

References

[1] J. Dai, Z. Wang, W. Huang, Interval-valued fuzzy discernibility pair approach for attribute reduction in incomplete interval-valued information systems, Inform. Sci. 642 (2023) 119215.

[2] G. Ma, J. Lu, Z. Fang, F. Liu, G. Zhang, Multiview classification through learning from interval-valued data, IEEE Trans. Neural Netw. Learn. Syst. (2024).

[3] H. Zapata, H. Bustince, S. Montes, B. Bedregal, G.P. Dimuro, Z. Takáč, M. Baczyński, J. Fernández, Interval-valued implications and interval-valued strong equality index with admissible orders, Internat. J. Approx. Reason. 88 (2017) 91–109.

[4] J. Liu, B. Huang, H. Li, X. Bu, X. Zhou, Optimization-based three-way decisions with interval-valued intuitionistic fuzzy information, IEEE Trans. Cybern. 53 (6) (2022) 3829–3843.

[5] X. Qi, H. Guo, Z. Artem, W. Wang, An interval-valued data classification method based on the unified representation frame, IEEE Access 8 (2020) 17002–17012.

[6] J. Dai, W. Wang, J.-S. Mi, Uncertainty measurement for interval-valued information systems, Inform. Sci. 251 (2013) 63–78.

[7] Y. Xue, Y. Tang, X. Xu, J. Liang, F. Neri, Multi-objective feature selection with missing data in classification, IEEE Trans. Emerg. Top. Comput. Intell. 6 (2) (2021) 355–364.

[8] C. Wang, C. Wang, Y. Qian, Q. Leng, Feature selection based on weighted fuzzy rough sets, IEEE Trans. Fuzzy Syst. (2024).

[9] C. Wang, J. Wang, Z. Gu, J.-M. Wei, J. Liu, Unsupervised feature selection by learning exponential weights, Pattern Recognit. 148 (2024) 110183.

[10] Y. Mi, H. Chen, C. Luo, S.-J. Horng, T. Li, Unsupervised feature selection with high-order similarity learning, Knowl.-Based Syst. 285 (2024) 111317.

[11] M. Banerjee, N.R. Pal, Unsupervised feature selection with controlled redundancy (UFeSCoR), IEEE Trans. Knowl. Data Eng. 27 (12) (2015) 3390–3403.

[12] R. Zhang, Y. Zhang, X. Li, Unsupervised feature selection via adaptive graph learning and constraint, IEEE Trans. Neural Netw. Learn. Syst. 33 (3) (2020) 1355–1362.

[13] J. Ma, F. Xu, X. Rong, Discriminative multi-label feature selection with adaptive graph diffusion, Pattern Recognit. 148 (2024) 110154.

[14] Y. Zhang, W. Huo, J. Tang, Multi-label feature selection via latent representation learning and dynamic graph constraints, Pattern Recognit. 151 (2024) 110411.

[15] S. Kim, S. Yun, J. Kang, DyGRAIN: An incremental learning framework for dynamic graphs, in: IJCAI, 2022, pp. 3157–3163.

[16] X. Zhang, X. Huang, W. Xu, Matrix-based multi-granulation fusion approach for dynamic updating of knowledge in multi-source information, Knowl.-Based Syst. 262 (2023) 110257.

[17] W. Shu, T. Chen, D. Cao, W. Qian, Incremental feature selection based on uncertainty measure for dynamic interval-valued data, Int. J. Mach. Learn. Cybern. 15 (4) (2024) 1453–1472.

[18] S. Qiang, Y. Liang, J. Wan, D. Zhang, Dynamic feature learning and matching for class-incremental learning, 2024, arXiv preprint arXiv:2405.08533.

[19] Y. Akhiat, Y. Asnaoui, M. Chahhou, A. Zinedine, A new graph feature selection approach, in: 2020 6th IEEE Congress on Information Science and Technology (CiSt), IEEE, 2021, pp. 156–161.

[20] G. Roffo, S. Melzi, U. Castellani, A. Vinciarelli, M. Cristani, Infinite feature selection: a graph-based feature filtering approach, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4396–4410.

[21] X. Xie, Z. Cao, F. Sun, Joint learning of graph and latent representation for unsupervised feature selection, Appl. Intell. 53 (21) (2023) 25282–25295.

[22] L. Xiang, H. Chen, T. Yin, S.-J. Horng, T. Li, Unsupervised feature selection based on bipartite graph and low-redundant regularization, Knowl.-Based Syst. 302 (2024) 112379.

[23] Q. Zhou, Q. Wang, Q. Gao, M. Yang, X. Gao, Unsupervised discriminative feature selection via contrastive graph learning, IEEE Trans. Image Process. 33 (2024) 972–986.

[24] X. Zhang, X. Shen, Graph-driven feature selection via granular-rectangular neighborhood rough sets for interval-valued data sets, Appl. Soft Comput. 170 (2025) 112716.

[25] B. Sang, H. Chen, L. Yang, T. Li, W. Xu, C. Luo, Feature selection for dynamic interval-valued ordered data based on fuzzy dominance neighborhood rough set, Knowl.-Based Syst. 227 (2021) 107223.

[26] Y. Yang, D. Chen, X. Zhang, Z. Ji, Y. Zhang, Incremental feature selection by sample selection and feature-based accelerator, Appl. Soft Comput. 121 (2022) 108800.

[27] Y. Pan, W. Xu, Q. Ran, An incremental approach to feature selection using the weighted dominance-based neighborhood rough sets, Int. J. Mach. Learn. Cybern. 14 (4) (2023) 1217–1233.

[28] W. Xu, Q. Bu, Matrix-based incremental feature selection method using weight-partitioned multigranulation rough set, Inform. Sci. 681 (2024) 121219.

[29] X. Zhang, J. Wang, J. Hou, Matrix-based approximation dynamic update approach to multi-granulation neighborhood rough sets for intuitionistic fuzzy ordered datasets, Appl. Soft Comput. 163 (2024) 111915.

[30] Z. Feng, X. Zhang, Supervised incremental feature selection using regularization vector for dynamic multi-scale interval valued datasets, Pattern Recognit. 170 (2026) 111985.

[31] W. Xu, Ordered Information Systems and Rough Sets, Science Press, Beijing, 2013.

[32] G. Chartrand, G. Kubicki, M. Schultz, Graph similarity and distance in graphs, Aequationes Math. 55 (1) (1998) 129–145.

[33] W. Xu, Z. Yuan, Z. Liu, Feature selection for unbalanced distribution hybrid data based on k-nearest neighborhood rough set, IEEE Trans. Artif. Intell. 5 (1) (2023) 229–243.

[34] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, E. Zhu, Unsupervised feature selection via multiple graph fusion and feature weight learning, Sci. China Inf. Sci. 66 (5) (2023) 152101.

[35] S. Xia, S. Wu, X. Chen, G. Wang, X. Gao, Q. Zhang, E. Giem, Z. Chen, GRRS: Accurate and efficient neighborhood rough set for feature selection, IEEE Trans. Knowl. Data Eng. 35 (9) (2022) 9281–9294.

[36] W. Xu, M. Huang, Z. Jiang, Y. Qian, Graph-based unsupervised feature selection for interval-valued information system, IEEE Trans. Neural Netw. Learn. Syst. 35 (9) (2024) 12576–12589.