International Journal of Machine Learning and Cybernetics (2026) 17:61
https://doi.org/10.1007/s13042-025-02919-z

ORIGINAL ARTICLE

Unsupervised feature selection through preservation of locality relation

Fatih Aydın[1]

Received: 9 November 2024 / Accepted: 24 October 2025 / Published online: 2 February 2026

Abstract
Unsupervised feature selection (UFS) aims to find the most useful features from unlabeled data, and it remains an important research topic in machine learning. To preserve the data manifold, many UFS methods have been proposed. These methods utilize various techniques such as similarity preservation, spectral learning, sparse modeling, and matrix decomposition. However, many existing approaches encounter various challenges because they rely on explicit pairwise distance computations, are sensitive to noise, and suffer from problems arising from high dimensionality. Besides, they have limited capability in preserving the manifold structure, especially when neighborhood graphs are heuristically defined or when embeddings distort the geometry. To overcome these issues, we propose a novel unsupervised feature selection framework that uses a geometry-aware optimization model. The main purpose is to learn a diagonal transformation matrix that captures the features' importance while maintaining local and global manifold structures. While carrying out this process, we do not directly calculate pairwise distances. This strategy increasescomputational robustness. The diagonal structure enables interpretable feature suppression by scaling or collapsing dimensions. This feature suppression leads to better generalization. The results of the benchmark data sets indicates that the proposed approach provides a convenient trade-off between structure preservation and the effectiveness of feature selection, and the proposed method often achieves performance comparable to or better than six state-of-the-art UFS methods.

Keywords Machine learning · Unsupervised feature selection · Manifold structure preservation · Diagonal matrix optimization · Geometry-aware learning · Interior-point algorithms

1 Introduction

In recent years, in many fields of study, the growth in data collection and storage has generated a lot of information and knowledge. This increment in data size has become a crucial problem in the data mining field [1, 2]. In most cases, all the data sets' features are non-essential to discovering the knowledge within them. Therefore, decreasing the number of features is indispensable. The dimensionality reduction process includes feature extraction and feature selection [3, 4].

Feature selection determines the most beneficial attributes in supervised, unsupervised, or semi-supervised machine learning tasks [5, 6]. Visualization also employs feature selection to comprehend the abovementioned tasks and data [7]. Moreover, it is necessary to prevent overfitting and cope with the curse of dimensionality [8, 9]. Because of these qualities, feature selection is a significant preprocessing stage in machine learning [10]. The feature selection has three principal approaches: filter-based, wrapper-based, and embedded [11, 12]. The filter-based unsupervised feature selection approach tries to find an optimal feature subset provided such that this subset has the least interrelation [13, 14]. The wrapper-based approach consults machine learning algorithms to probe the subsets of features [8, 15, 16]. Embedded methods combine filter and wrapper paradigms by incorporating feature selection into the model-learning process [17, 18].

Applying the graph-based approaches to unsupervised feature selection problems has gained a lot of interest [19, 20]. Graph-based unsupervised feature selection methods use different strategies to preserve graph structure and learn

✉ Fatih Aydın
fatih.aydin@balikesir.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Balikesir University, 10145 Balıkesir, Turkey

from it [21, 22]. The traditional approaches construct only a graph from data. The feature selection with an adaptive multiple-graph learning approach selects features and retains the data's intrinsic structure by adapting a consensus graph from multiple base graphs [23]. In the case of assuming data independence, the relational information in the data may be lost. To overcome this problem, the Central Point link information and Sparse Latent Representation (CPSLR) method has been proposed, which obtains pseudo-label information by using a data graph and link graph to form a dual graph structure [24]. It is also vital to capture complicated nonlinear relationships in large-scale data sets. To this end, a new framework was introduced, which uses the autoencoder with a differentiable gating function [25].

Feature selection poses a severe hardship in machine learning [26], and it is undoubtedly regarded as an NP-hard problem [27, 28]. The strategies in the filter-based methods mainly focus on learning the sophisticated data relationships [29]. However, feature selection algorithms can underperform due to some reasons. They are the preservation of both graph structure and the sample distribution simultaneously [22], high sensitivity against outliers [30], inability to deal with large data sets [31], and exploiting uncorrelated, unrelated, and tight constraints does not lead to a significant improvement in the results of the work in [24]. Nonetheless, we recognize that using uncorrelated constraints is crucial for gaining insight [24].

This study introduces an innovative unsupervised feature selection (UFS) algorithm. The proposed framework aims to preserve the data manifold and address high-dimensional or noisy data. It uses a filter-based strategy that learns a diagonal matrix to weight features by relevance. Unlike many conventional UFS algorithms, the proposed method does not directly compute pairwise distances. Instead, it focuses on preserving geometry through the Gram matrix. The underlying contributions of this research can be stated as follows:

- Geometry-aware feature weighting: The method seeks an optimal diagonal matrix that minimizes the difference between the original and transformed manifolds so as to preserve both local and global structures.
- Implicit neighborhood preservation: The algorithm captures geometric relationships without constructing a neighborhood graph and computing distances.
- Interpretability and low hyperparameter burden: The diagonal structure facilitates the direct observation of the importance of each feature. Moreover, the method has only one hyperparameter to adjust.

The rest of this paper is organized as follows: Sect. 2 contains the literature review. Section 3 presents the details of the proposed algorithm and the optimization algorithm. Section 4 explains the experimental procedure. Section 5 reports empirical results and discusses the findings. Finally, the article is concluded in Sect. 6.

2 Literature review

Feature selection plays an important role in obtaining more efficient and interpretable machine learning models in high-dimensional data sets. Considering the growing complexity of data sets, feature selection has vital importance in enhancing model performance and reducing computational costs. This literature review provides an overview of recent advances in feature selection techniques and highlights differences between them and the approach adopted in our study. Besides, we aim to position our method thoughtfully within this evolving landscape.

Similarity Preserving Feature Selection (SPFS) [32] aims to select a subset of features in order to preserve similarity between the pairwise samples by using a predefined similarity matrix. It formulates a combinatorial optimization problem using a binary selection matrix to maximize similarity retention while eliminating redundant features. In order to address sample variation in dynamic learning systems, a group incremental feature selection approach, based on Triple Nested Equivalence Classes (TNEC), was proposed. This approach improves dependency computation and reduces redundancy in high-dimensional data using rough set theory [33]. To achieve parameter-free and efficient feature selection in multi-class classification, a Graph-Based filter method for Automatic Feature Selection (GB-AFS) was proposed, which leverages the Mean Simplified Silhouette (MSS) index to automatically identify a minimal and effective feature subset while maintaining high predictive performance [34]. An embedded feature selection method based on adaptive group lasso (AGL) was proposed to improve the feature set in high-dimensional neural network potentials for atomistic simulations. The AGL approach shows better performance over unsupervised filters by incorporating model predictions into the selection process [35]. In order to improve clustering performance in UFS, a new method called a low-rank structure-preserving method for unsupervised feature selection (LRPFS) was proposed, which introduces sample-specific attribute scoring through latent learning. I also includes a sparse penalty term to maintain the data structure and ensure solution uniqueness [36].

To address noise sensitivity and limited similarity modeling in graph-based UFS, a method named Unsupervised feature selection with High-order Similarity Learning (HSL) was proposed. This method jointly learns a projection matrix, first-order, and high-order similarities in a unified framework to better preserve intrinsic data structure in a denoised low-dimensional embedding space [37]. To enhance robustness in UFS under noisy conditions, a reconstruction-based framework was suggested, which jointly removes anomalous samples and features by using double binary weighting. Further, its integration with AutoEncoder Feature Selector (AEFS) [38] exhibits better performance over real-world data sets [39]. To address view heterogeneity in multi-view unsupervised feature selection, a novel method was introduced that builds mutually exclusive graphs to enhance view complementarity. This method integrates graph learning with consensus clustering to exploit indicator consistency for improved feature selection performance [40]. To overcome graph construction and feature redundancy issues in unsupervised feature selection, a new method, abbreviated as DLSEO was proposed. This method combines dual space-based low redundancy scores with extended orthogonal least square discriminant analysis (OLSDA). This approach generates non-negative clustering labels and selects salient features without depending on Laplacian graph construction [41]. To improve unsupervised feature selection by addressing both redundancy and relevance, the Orthogonal EncoderDecoder factorization for unsupervised Feature Selection (OEDFS) model was proposed. This model integrates self-representation and pseudo-supervised learning within an orthogonal encoder-decoder framework, and thus, it preserves local structure and achieves superior performance across the benchmarks [42].

To address the problems with only relevance criteria in unsupervised feature selection, the unsupervised Feature Selection with Max-Relevance and Minimum Global Redundancy (MRMGRFS) algorithm was proposed, which integrates spectral clustering-based relevance evaluation with a Jensen-Shannon divergence-based Global Redundancy Minimization model (SJGRM). The SJGRM model is optimized via Alternating Direction Method of Multipliers (ADMM) to select maximally relevant and minimally redundant features [43]. To effectively capture both local and global data structures in unsupervised feature selection, the Jointly Subspace Learning and Subspace orthogonal basis Clustering (JSLSC) model was proposed. This model brings together robust subspace learning, self-representation, and orthogonal basis clustering to preserve structural consistency and reveal latent clustering patterns in unlabeled high-dimensional data [44]. To address the issues of missing instances and high computational cost in incomplete multi-view feature selection, the Confident Local Similarity Graphs for Unsupervised Feature Selection on Incomplete Multi-view Data (CLSG) method was developed. This method constructs confident dual-level similarity graphs by leveraging latent representations and cross-view information to recover missing data and identify important features efficiently [45].

The proposed method endeavors to cope with the common limitations encountered in the literature. Unlike some UFS methods that rely on local similarity preservation, pseudo-label generation, or graph-based representations, the proposed approach attempts to preserve global pairwise distance to maintain the intrinsic geometry of the data manifold. By minimizing the distortion between the original and transformed data spaces, it guarantees that neighborhood relationships between data instances are retained better. Furthermore, in contrast to the approaches that treat features independently or ignore inter-feature dependencies, the proposed algorithm incorporates a diagonal matrix-based optimization framework that reflects feature importance while inherently capturing linear dependencies and redundancy among features. Thus, this framework relaxes strict selection constraints by allowing continuous-valued importance scoring. In addition, unlike certain reconstruction-based or clustering-based methods that may struggle with non-convexity or heuristic approximations, it approximates global convergence properties and enhances scalability in high dimensionality. In summary, the proposed method provides a unified and mathematically grounded UFS strategy to effectively manage relevance, redundancy, and manifold preservation.

3 The proposed method

In this section, we introduce the notations used in the paper. Then, we present the proposed algorithm and the optimization algorithm in detail. Finally, we examine the asymptotic complexity of the proposed algorithm.

3.1 Notations and nomenclature

Let $\boldsymbol{X} \in \mathbb{R}^{m \times d}$ be a data matrix with m points and d features. Let $\boldsymbol{y} \in \{c_1, \cdots, c_p\}$ be a label vector with m observations and p classes. The data matrix's columns, rows, and elements in i th row j th column are referred to as features $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(d)}$, instances $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$, and $\boldsymbol{X}_{ij}$, respectively. Accordingly, the rest of the notations are as follows:

- The i th component of a vector $\boldsymbol{x}$ is symbolized by x_i.
- The values between i th and j th components of a vector $\boldsymbol{x}$ is indicated by $x_{(i:j)}$.
- The transpose and inverse of $\boldsymbol{X}$ is $\boldsymbol{X}^T$ and $\boldsymbol{X}^{-1}$, respectively.
- The trace, rank, determinant, and nullity of $\boldsymbol{X}$ are denoted as $Tr(\boldsymbol{X})$, $r(\boldsymbol{X})$, $det(\boldsymbol{X})$, and $\text{null}(\boldsymbol{X})$, respectively.
- The set of eigenvalues of $\boldsymbol{X}$ are denoted as $\Lambda(\boldsymbol{X})$.
- $diag(\boldsymbol{Q})$ refers to a column vector including the n diagonal elements of $\boldsymbol{Q}$.
- $Dg(\boldsymbol{q})$ refers to an $n \times n$ diagonal matrix whose entries are the n elements in the vector $\boldsymbol{q}$.
- $\boldsymbol{J}_{d\times m}$ denotes a $d \times m$ matrix of ones.
- The Frobenius norm of $\boldsymbol{X}$ is denoted as $\|\boldsymbol{X}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{d}|\boldsymbol{X}_{ij}|^2} = \sqrt{Tr\left(\boldsymbol{X}^T\boldsymbol{X}\right)}$.
- The identity matrix of order n is denoted as $\mathbb{I}_n$.
- The symbol $\oslash$ denotes the element-wise divide.

3.2 Objective function

The proposed method aims to preserve the local structure of the data while coping with high dimensionality. It particularly eliminates linearly dependent, redundant, or irrelevant features that distort the intrinsic geometry of the sample distribution. To this end, we try to seek a diagonal matrix $\boldsymbol{D}$ based on Corollary 1. Each diagonal element of the matrix $\boldsymbol{D}$ represents a corresponding feature importance. If the value of feature importance is zero, then this implies that the feature is discarded; otherwise, it indicates higher relevance. However, constraining the diagonal entries to be strictly binary (i.e., 0 or 1) results in a combinatorial problem, which is computationally difficult to solve. Therefore, we allow the entries to take continuous values in $(0, 1)$ to enable tractable optimization. The matrix $\boldsymbol{D}$ is learned to minimize the distortion of the pairwise relationships between samples, allowing the most informative features to be retained. While addressing this issue, we prioritize using diagonal and symmetric matrices. The reason is that a diagonal matrix helps extend each axis (i.e., feature) differently. Setting the corresponding values in the diagonal matrix to zero is sufficient for the dimensions we desire to eliminate. Figure 1 illustrates how to scale a data matrix along any axis using a rectangular prism.

First of all, the fundamental insight of the proposed method is that there exists a diagonal matrix $\boldsymbol{D}$, which maximizes the similarity between the pair-wise distance matrices of $\boldsymbol{X}$ and $\boldsymbol{XD}$ matrices. In other words, it is to minimize the Frobenius distance between the pair-wise distance matrices. Thus, the neighborhood relationship between instances is preserved. Furthermore, the importance levels of features are kept in the diagonal entries of $\boldsymbol{D}$. Accordingly, based on Definition 1, the pair-wise distance matrices of $\boldsymbol{X}$ and $\boldsymbol{XD}$ are $2\left(\boldsymbol{J}_{m\times m} - \boldsymbol{X}\boldsymbol{X}^T\right)$ and $2\left(\boldsymbol{J}_{m\times m} - \boldsymbol{XD}(\boldsymbol{XD})^T\right)$, respectively. Accordingly, we express the objective function as follows:

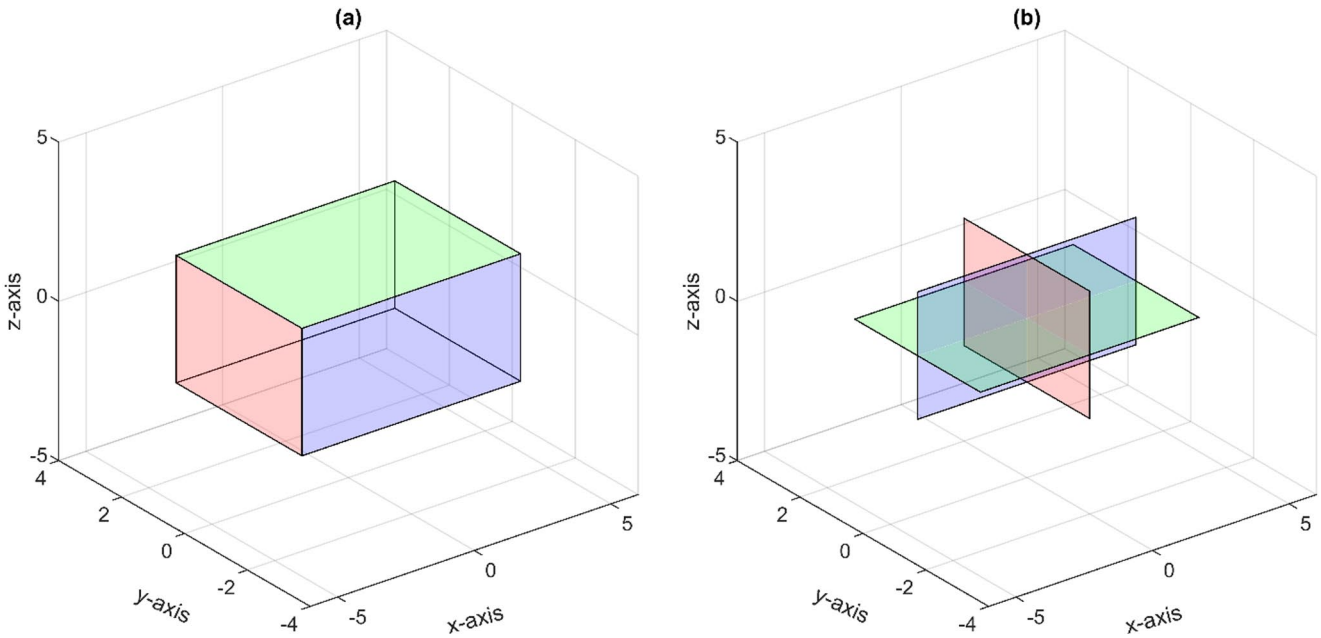

Fig. 1 Illustration of scaling a data matrix along all axes using a rectangular prism. (**a**) the original shape represents that multiplying a data matrix by the identity matrix does not cause any change in it. (**b**) According to the axes, the illustration represents multiplying a data matrix by $\boldsymbol{Dg}([\ 1\ \ 0\ \ 1\ ])$, $\boldsymbol{Dg}([\ 0\ \ 1\ \ 1\ ])$, and $\boldsymbol{Dg}([\ 1\ \ 1\ \ 0\ ])$, respectively and thus, some dimensions collapse

$$\begin{aligned} &\min_{\boldsymbol{D}} 2\left(\boldsymbol{J}_{m\times m}-\boldsymbol{X}\boldsymbol{X}^T\right)-2\left(\boldsymbol{J}_{m\times m}-\boldsymbol{X}\boldsymbol{D}\left(\boldsymbol{X}\boldsymbol{D}\right)^T\right)_F \\ &\min_{\boldsymbol{D}} 2\boldsymbol{X}\boldsymbol{D}\left(\boldsymbol{X}\boldsymbol{D}\right)^T-2\boldsymbol{X}\boldsymbol{X}_F^T \end{aligned} \tag{1}$$

If $\boldsymbol{D}$ is the identity matrix, note that the solution in (1) is trivial. Applying a mathematical trick that does not change the trivial solution, i.e., we multiply the first term of (1) by $\boldsymbol{X}\boldsymbol{D}$ and the second term by $\boldsymbol{X}$, we obtain the following objective function:

$$\min_{\boldsymbol{D}}\|2\boldsymbol{X}\boldsymbol{D}(\boldsymbol{X}\boldsymbol{D})^T\boldsymbol{X}\boldsymbol{D}-2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{X}\|_F \tag{2}$$

Let us assume that the expression in (2) is zero. In this case, we have $2\boldsymbol{X}\boldsymbol{D}(\boldsymbol{X}\boldsymbol{D})^T\boldsymbol{X}\boldsymbol{D}=2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{X}$. Let us multiply both sides of the equation by $\boldsymbol{X}^{-1}$ and we attain Eq. (3). We assume that $\boldsymbol{X}$ is invertible for illustrative purposes to provide an intuitive derivation of the objective function. We note that the use of $\boldsymbol{X}^{-1}$ is only symbolic, introduced to illustrate a conceptual transformation under the assumption of ideal invertibility for derivation. If $\boldsymbol{X}$ is rank-deficient or not square, then it may not be invertible. In order to maintain mathematical rigor, the Moore–Penrose pseudoinverse should be exploited. Since the final optimization problem is constructed based on the matrix $\boldsymbol{X}^T\boldsymbol{X}$, this does not exclude the validity of the solution.

$$\begin{aligned} &\boldsymbol{X}^{-1}\left(2\boldsymbol{X}\boldsymbol{D}\left(\boldsymbol{X}\boldsymbol{D}\right)^T\boldsymbol{X}\boldsymbol{D}\right)=\boldsymbol{X}^{-1}\left(2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{X}\right) \\ &\boldsymbol{D}\left(\boldsymbol{X}\boldsymbol{D}\right)^T\boldsymbol{X}\boldsymbol{D}=\boldsymbol{X}^T\boldsymbol{X} \\ &\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{D}=\boldsymbol{X}^T\boldsymbol{X} \end{aligned} \tag{3}$$

To avoid the trivial solution, we redefine (3) by using a slack variable α as in Eq. (4). In Eq. (3), the expression.

$\boldsymbol{D}(\boldsymbol{X}\boldsymbol{D})^T\boldsymbol{X}\boldsymbol{D}=\boldsymbol{X}^T\boldsymbol{X}$ may lead to a trivial solution when $\boldsymbol{D}=\mathbb{I}_d$. To avoid this and to introduce flexibility in aligning the left-hand term with $\boldsymbol{X}^T\boldsymbol{X}$, we incorporate α as a scalar scaling variable in Eq. (4). This allows the model to adjust the magnitude of the target matrix (i.e., $\boldsymbol{X}^T\boldsymbol{X}$) to better match the transformed structure $\boldsymbol{D}(\boldsymbol{X}\boldsymbol{D})^T\boldsymbol{X}\boldsymbol{D}$, which depends on the selected features. Thus, α plays two important roles: (i) It scales discrepancies between the original matrix $\boldsymbol{X}^T\boldsymbol{X}$ and the one generated through the learned feature weights in $\boldsymbol{D}$, (ii) It ensures the optimization problem remains bounded and non-trivial by decoupling the scaling of $\boldsymbol{D}$ from that of $\boldsymbol{X}^T\boldsymbol{X}$. Crucially, in Eq. (6) and the final non-convex in Eq. (7), α becomes a learnable parameter constrained in the range $(0,1)$, similar to the entries of $\boldsymbol{D}$, to remain interpretability and optimization feasibility.

$$\min_{\boldsymbol{D}}\|\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{D}-\alpha\boldsymbol{X}^T\boldsymbol{X}\|_F \tag{4}$$

Since $\boldsymbol{D}$ is a diagonal matrix, we rewrite Eq. (4) as follows:

$$\min_{\boldsymbol{D}}\|\boldsymbol{D}^2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{D}-\alpha\boldsymbol{X}^T\boldsymbol{X}\|_F \tag{5}$$

For simplification, let $\boldsymbol{G}=\boldsymbol{X}^T\boldsymbol{X}$, i.e., Gram matrix. Equation (5) is transformed into the following trace form:

$$\min_{\boldsymbol{D}}\|\boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D}-\alpha\boldsymbol{G}\|_F \tag{6}$$

Accordingly, we redefine the non-convex program as in Eq. (7). We note that the optimization problem is strictly non-convex because of the cubic term in $\boldsymbol{D}$. Therefore, we refer to the formulation as a convex-like program in the relaxed sense of being solvable using interior-point methods.

$$\begin{aligned} &\min_{\boldsymbol{D}} \boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D}-\alpha\boldsymbol{G}_F \\ &s.t. 0+\leq\alpha,\boldsymbol{D}_{i,i}\leq 1-\epsilon; i=1,\cdots,d; \epsilon=1.0e-7 \end{aligned} \tag{7}$$

3.3 Optimization process

The convex optimization problem in Eq. (7) consists of a constrained nonlinear multivariable objective function. Interior-point algorithms can deal with non-linear convex optimization problems varying from small, dense ones to large, sparse ones. It satisfies bounds at all iterations and is a large-scale algorithm that uses exclusive strategies for large-scale problems. Hence, we employ the interior-point optimization algorithm to solve the convex optimization problem in Eq. (7). In the interior-point technique, the form of the original problem is $\min_{\boldsymbol{a}} f(\boldsymbol{a})=\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj}-\alpha\right)^2\cdot\left(\boldsymbol{G}_{ij}\right)^2\right)$, where $\boldsymbol{a}\in\mathbb{R}^{(d+1)}=\begin{bmatrix}\boldsymbol{D}_{11}\\ \vdots\\ \boldsymbol{D}_{dd}\\ \alpha\end{bmatrix}$ represents variables to be

optimized, i.e., a vector consisting of the diagonal entries of $\boldsymbol{D}$ and α. $f(\boldsymbol{a})$ denotes an objective function, i.e., $\|\boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D} - \alpha\boldsymbol{G}\|_F$. Accordingly, the form of the approximation problem under loose constraints is $\min_{\boldsymbol{a}} f(\boldsymbol{a})$. The system of non-linear equation for the objective function is as follows:

$$
\begin{aligned}
&\boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D} - \alpha\boldsymbol{G}_F = Tr\left(\left(\boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D} - \alpha\boldsymbol{G}\right)^T\left(\boldsymbol{D}^2\boldsymbol{G}\boldsymbol{D} - \alpha\boldsymbol{G}\right)\right)\\
&= \sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\boldsymbol{D}^2\cdot\boldsymbol{G}\cdot\boldsymbol{D} - \alpha\boldsymbol{G}\right)_{ij}\right)^2\\
&= \sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{G}_{ij}\cdot\boldsymbol{D}_{jj} - \alpha\boldsymbol{G}_{ij}\right)^2\\
&= \sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)\cdot\boldsymbol{G}_{ij}\right)^2\\
&= \sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)^2\cdot\left(\boldsymbol{G}_{ij}\right)^2\right)\\
&= \left[\left(\boldsymbol{G}_{11}\right)^2\left(\left(\boldsymbol{D}_{11}\right)^3 - \alpha\right)^2 + \cdots + \left(\boldsymbol{G}_{1d}\right)^2\left(\left(\boldsymbol{D}_{11}\right)^2\boldsymbol{D}_{dd} - \alpha\right)^2\right] + \cdots\\
&+ \left[\left(\boldsymbol{G}_{d1}\right)^2\left(\left(\boldsymbol{D}_{dd}\right)^2\boldsymbol{D}_{11} - \alpha\right)^2 + \cdots + \left(\boldsymbol{G}_{dd}\right)^2\left(\left(\boldsymbol{D}_{dd}\right)^3 - \alpha\right)^2\right]\epsilon =^2
\end{aligned}
\tag{7}
$$

where ϵ is expected to be a value close to zero, we impose constraints in Eq. (9). The interior-point method solves constrained optimization by transforming it into an unconstrained problem using a barrier function. We define the log-barrier in Eq. (10) for each constraint.

$$
\begin{aligned}
&0 + \epsilon < \alpha, \boldsymbol{D}_{i,i} < 1 - \epsilon; i = 1, \cdots, d; \epsilon = 1.0e-7\\
&a_k \langle 1- \Rightarrow -a_k + 1 - \epsilon\rangle\, 0 \; and \; a_k > 0+ \Rightarrow a_k - \epsilon > 0
\end{aligned}
\tag{9}
$$

$$
B(\boldsymbol{a}) = -\sum_{k=1}^{d+1}\left(\ln\left(a_k - \epsilon\right) + \ln\left(1 - \epsilon - a_k\right)\right) \tag{10}
$$

After defining the barrier function, we define the full objective function as in Eq. (11). Herein, the goal is to minimize the full objective function.

$$
F(\boldsymbol{a}, \delta) = \delta\cdot f(\boldsymbol{a}) + B(\boldsymbol{a}) \tag{11}
$$

While figuring out the approximate problem, the interior-point algorithm operates a two-step strategy at each iteration. By default, the necessary optimality conditions are firstly derived using Lagrange multipliers as in $\mathcal{L}(\boldsymbol{a}) = f(\boldsymbol{a})$. In other words, the Karush–Kuhn–Tucker (KKT) conditions are fulfilled. Accordingly, the gradient of $f(\boldsymbol{a})$ is calculated as in Eq. (12) for each element.

$$
\begin{aligned}
&\frac{\partial f}{\partial \boldsymbol{D}_{kk}} = \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)^2\cdot\left(\boldsymbol{G}_{ij}\right)^2\right), k = 1, \cdots, d\\
&\frac{\partial f}{\partial \boldsymbol{D}_{kk}} = \sum_{i=1}^{d}\sum_{j=1}^{d}\left(2\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)\cdot\left(\boldsymbol{G}_{ij}\right)^2\cdot\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj}\right)\right)
\end{aligned}
\tag{12}
$$

$\boldsymbol{D}_{kk}$ does not appear in the cases $i \neq k$ and $j \neq k$. In other words, the term's derivative is zero. In the case $i \neq k$, we can fix $i = k$ and sum over j. Let u be $\left(\left(\boldsymbol{D}_{kk}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)$. In this case, $\frac{\partial u}{\partial \boldsymbol{D}_{kk}} = 2\boldsymbol{D}_{kk}\boldsymbol{D}_{jj}$.

$$
\begin{aligned}
&= \sum_{j=1}^{d}\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(\left(\left(\boldsymbol{D}_{kk}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)^2\cdot\left(\boldsymbol{G}_{kj}\right)^2\right)\\
&= \sum_{j=1}^{d}\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(u^2\cdot\left(\boldsymbol{G}_{kj}\right)^2\right)\\
&= \sum_{j=1}^{d}2u\cdot\frac{\partial u}{\partial \boldsymbol{D}_{kk}}\cdot\left(\boldsymbol{G}_{kj}\right)^2\\
&= \sum_{j=1}^{d}2u\cdot\left(2\boldsymbol{D}_{kk}\boldsymbol{D}_{jj}\right)\cdot\left(\boldsymbol{G}_{kj}\right)^2\\
&= \sum_{j=1}^{d}4\boldsymbol{D}_{kk}\boldsymbol{D}_{jj}\cdot\left(\left(\boldsymbol{D}_{kk}\right)^2\cdot\boldsymbol{D}_{jj} - \alpha\right)\cdot\left(\boldsymbol{G}_{kj}\right)^2
\end{aligned}
\tag{13}
$$

In the case $j \neq k$, we can fix $j = k$ and sum over i. Let u be $\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{kk} - \alpha\right)$. In this case, $\frac{\partial u}{\partial \boldsymbol{D}_{kk}} = \left(\boldsymbol{D}_{ii}\right)^2$.

$$
\begin{aligned}
&= \sum_{i=1}^{d}\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{kk} - \alpha\right)^2\cdot\left(\boldsymbol{G}_{ik}\right)^2\right)\\
&= \sum_{i=1}^{d}\frac{\partial}{\partial \boldsymbol{D}_{kk}}\left(u^2\cdot\left(\boldsymbol{G}_{ik}\right)^2\right)\\
&= \sum_{i=1}^{d}2u\cdot\left(\boldsymbol{D}_{ii}\right)^2\cdot\left(\boldsymbol{G}_{ik}\right)^2\\
&= \sum_{i=1}^{d}2\left(\boldsymbol{D}_{ii}\right)^2\cdot\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{kk} - \alpha\right)\cdot\left(\boldsymbol{G}_{ik}\right)^2
\end{aligned}
\tag{14}
$$

Let us combine both cases and reach the following term:

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{D}_{kk}} &= \sum_{i=1}^{d}\left[2\left(\boldsymbol{D}_{ii}\right)^2\cdot\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{kk}-\alpha\right)\cdot\left(\boldsymbol{G}_{ik}\right)^2\right]\\ &+\sum_{j=1}^{d}\left[4\boldsymbol{D}_{kk}\boldsymbol{D}_{jj}\cdot\left(\left(\boldsymbol{D}_{kk}\right)^2\cdot\boldsymbol{D}_{jj}-\alpha\right)\cdot\left(\boldsymbol{G}_{kj}\right)^2\right]\end{aligned} \tag{15}$$

We calculate the gradients for α and the barrier function as in Eq. (16).

$$\begin{aligned}\frac{\partial f}{\partial \alpha} &= -\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\left(\left(\boldsymbol{D}_{ii}\right)^2\cdot\boldsymbol{D}_{jj}-\alpha\right)\cdot\left(\boldsymbol{G}_{ij}\right)^2\right)\\ \frac{\partial B}{\partial a_k} &= -\frac{1}{a_k-\epsilon}+\frac{1}{1-\epsilon-a_k}\end{aligned} \tag{16}$$

Finally, we calculate the total gradient of the full objective function $F\left(\boldsymbol{a},\delta\right)$ as in Eq. (17).

$$\nabla L\left(\boldsymbol{a}\right)=\nabla F\left(\boldsymbol{a},\delta\right)=\delta\cdot\nabla f\left(\boldsymbol{a}\right)+\nabla B\left(\boldsymbol{a}\right) \tag{17}$$

Then, the interior-point algorithm iterates around the optimal point using Newton's method. To this end, the Hessian of the Lagrangian of $f\left(\boldsymbol{a}\right)$ should be calculated as in $H=\nabla^2 f\left(\boldsymbol{a}\right)$, which includes $\frac{\partial^2 \mathrm{f}}{\partial \boldsymbol{D}_{kk}\partial \boldsymbol{D}_{ll}}$, $\frac{\partial^2 \mathrm{f}}{\partial \boldsymbol{D}_{kk}\partial \alpha}$, and $\frac{\partial^2 \mathrm{f}}{\partial \alpha^2}$. The Hessian of the barrier function is also calculated in Eq. (18).

$$\begin{aligned}\frac{\partial^2 B}{\partial a_k^2} &= \frac{1}{\left(a_k-\right)^2}+\frac{1}{\left(1-\epsilon-a_k\right)^2}\\ \nabla^2 B\left(\boldsymbol{a}\right) &= Dg\left(\frac{1}{\left(a_1-\epsilon\right)^2}+\frac{1}{\left(1-\epsilon-a_1\right)^2},\ldots,\right.\\ &\left.\frac{1}{\left(a_{d+1}-\epsilon\right)^2}+\frac{1}{\left(1-\epsilon-a_{d+1}\right)^2}\right)\end{aligned} \tag{18}$$

The algorithm first tries to take a direct step. If it cannot, this means that the approximate problem is locally concave around the current iteration. In this case, the interior-point technique tries a conjugate gradient step. The Conjugate Gradient algorithm acquires Lagrange multipliers by roughly solving the KKT condition equations in $\nabla L=\nabla f\left(\boldsymbol{a}\right)$. Consequently, the Hessian of the whole objective function is as follows:

$$H_L=\nabla^2 F\left(\boldsymbol{a},\delta\right)=\delta\cdot\nabla^2 f\left(\boldsymbol{a}\right)+\nabla^2 B\left(\boldsymbol{a}\right) \tag{19}$$

While the interior-point algorithm solves the problem in the least-squares sense, a step, i.e., $\Delta\boldsymbol{a}$ is taken to reach an approximate solution. To this end, the statement $H_L\cdot\Delta\boldsymbol{a}=-\nabla L\left(\boldsymbol{a}\right)$ is solved. We update the optimization variables in an iterative block scope at each iteration k. Herein, we need to decide how far to move in the related direction using a step size $\varphi\in(0,1]$. The value to be optimized (i.e., the new candidate point) is updated by $\boldsymbol{a}^{(t+1)}=\boldsymbol{a}^{(t)}+\varphi\cdot\Delta\boldsymbol{a}$. Further, the step size φ is calculated as in Eq. (20).

$$\begin{aligned}\varphi_k^{max} &= \begin{cases}\frac{1-\epsilon-a_k^{(t)}}{\Delta a_k}, \Delta a_k>0\\ \frac{-a_k^{(t)}}{\Delta a_k}, \Delta a_k<0\end{cases}\\ \varphi &= \left(\min_k \varphi_k^{max}\right)\cdot\tau\end{aligned} \tag{20}$$

where $\tau\in(0,1)$. This multiplication operation ensures that all updated variables strictly remain within $(0,1)$.

Consequently, we can say that the importance of coefficients for each feature is formed from $\boldsymbol{a}$, i.e., the set $\Lambda\left(\boldsymbol{D}\right)$. While values close to zero indicate unimportant features, the importance of the feature increases as it moves away from zero. As a result, the set $\Lambda\left(\boldsymbol{D}\right)$ is sorted from largest to smallest, and the index values of the features corresponding to these values are returned by the algorithm developed. Finally, the pseudo-code of the proposed algorithm is given in Algorithm 1. The proposed method is called PLRU (the Preservation of Locality Relation for Unsupervised feature selection).

Input: Data matrix $X \in R^{m \times d}$, the slack parameter $\alpha \in [0,1]$.

Output: Indices of sorted features

$G \leftarrow X^T X$

$N \leftarrow Dg(1 \oslash diag(G))$

$G \leftarrow N \cdot G$

$a \in R^{(d+1)} \leftarrow \left(\underbrace{1/2, \cdots, 1/2}_{d}, \alpha \right)$

Define the objective $f(a) = \left\| Dg(a_{(1:d)})^2 \cdot G \cdot Dg(a_{(1:d)}) - a_{(d+1)} \cdot G \right\|_F$

Define the barrier $B(a) = -\sum_{k=1}^{d+1} (\ln a_k + \ln(1 - a_k))$

Define the complete objective $F(a, \delta) = \delta \bullet f(a) + B(a)$

while $(1/\delta >$ barrier tolerance)

  repeat

    Calculate $\nabla L(a) = \nabla F(a, \delta)$

    Calculate $H_L = \nabla^2 L(a)$

    Calculate $\Delta a \leftarrow -H_L^{-1} \cdot \nabla L(a)$

    Update $a^{(t+1)} \leftarrow a^{(t)} + \varphi \bullet \Delta a$

  until converge

  Update δ

end

Sort $a_{(1:d)}$ in descending order

return indices of sorted features

Algorithm 1 PLRU

Table 1 The asymptotic computational complexities of the algorithms

#	Algorithms	Asymptotic complexities
1	Laplacian Score (LS) [46]	$O\left(m^2 d + d\log_2 d\right)$
2	Robust Neighborhood Embedding for unsupervised feature selection (RNE) [47]	$O\left(m^2 d + s_1\left(l^2 d + m d^2 + l d^2\right) + s_2 m + d\log_2 d\right)$
3	feature Dependency-based Unsupervised Feature Selection (DUFS) [48]	$O\left(d^3 + m^2\right)$
4	Unsupervised Soft-label Feature Selection (USFS) [49]	$O\left(s\left(dl + mr + dr\right)\right)$
5	Soft-Label guided Non-negative Matrix Factorization (SLNMF) [30]	$O\left(s\left(m^2 d + m^2 r + d^2 l + m d^2\right)\right)$
6	unsupervised feature selection with High-order Similarity Learning (HSL) [37]	$O\left(d^3 + mdl\right)$
7	Preservation of Locality Relation for Unsupervised feature selection (PLRU)	$O\left(N_{outer} \cdot N_{inner}\left(d^3 + m d^2\right)\right)$

Table 2 The data sets used in the experiments

Data sets (#)	Names	Samples (m)	Features (d)	Classes (k)	Imbalance rate
#1	CNAE-9	1080	856	9	1.00
#2	Connectionist Bench (Sonar, Mines vs. Rocks)	208	60	2	1.14
#3	Gait Classification	48	321	16	1.00
#4	Smartphone Dataset for Human Activity Recognition (HAR) in Ambient Assisted Living (AAL)	5744	561	6	1.42
#5	Hill-Valley	1212	100	2	1.01
#6	Libras Movement	360	90	15	1.00
#7	LSVT Voice Rehabilitation	126	310	2	2.00
#8	Madelon	2000	500	2	1.00
#9	Musk (Version 1)	476	166	2	1.30
#10	Optical Recognition of Handwritten Digits	3823	64	10	1.03
#11	Semeion Handwritten Digit	1593	256	10	1.05
#12	TUANDROMD (Tezpur University Android Malware Dataset)	4464	241	2	4.00
#13	COIL-20 (Columbia Object Image Library)	1440	1024	20	1.00

3.4 Asymptotic complexity

First, we state that mathematical operation time complexities are calculated herein according to the naive approaches. In Algorithm 1, the time complexity of calculating the gram matrix $\boldsymbol{G}$ in Step 1 is $O\left(md^2\right)$. The time complexity of calculating the matrix $\boldsymbol{N}$ in Step 2 is $O\left(d\right)$**.** The operation $\boldsymbol{N}\cdot\boldsymbol{G}$ in Step 3 is calculated by $O\left(d^2\right)$. The objective function $\nabla F\left(\boldsymbol{a},\delta\right)$ at each step in Step 10 is computed by the time complexity $O\left(d^2+2d\right)$. The Hessian H_L of the full objective function at each step in Step 11 is calculated by $O\left(5d^2+d\right)$. The time complexity required to solve $H_L\cdot\Delta\boldsymbol{a}=-\nabla L\left(\boldsymbol{a}\right)$ in Step 12 is $O\left(d^3+2d^2\right)$. To update $\boldsymbol{a}^{(t+1)}\leftarrow\boldsymbol{a}^{(t)}+\varphi\cdot\Delta\boldsymbol{a}$ at each step, the time complexity is $O\left(3d\right)$. Furthermore, the convergence-determining operation in Step 14 is calculated by $O\left(d\right)$. Finally, the algorithmic complexity needed to update δ in Step 15 is $O\left(1\right)$. In consequence, the time complexity at each step is $O\left(d^3+8d^2+md^2+8d+1\right)$. Let N_{inner} and N_{outer} be the numbers of iterations for converging the solution and updating δ, respectively. the overall algorithmic complexity is $O(N_{outer}\cdot N_{inner}\cdot\ (d^3+8d^2+md^2+8d)+N_{outer})$. But we neglect lower-order terms because they have a minor influence on the upper-order ones. Accordingly, the time complexity of the PLRU is $O(N_{outer}\cdot N_{inner}\cdot(d^3+md^2))$. Further, we examine the computational complexity of the PLRU in three cases, according to m and d. The computational complexity of the PLRU is found as $O\left(N_{outer}\cdot N_{inner}\cdot m\right)$, $O\left(N_{outer}\cdot N_{inner}\cdot d^3\right)$,and$O\left(N_{outer}\cdot N_{inner}\cdot d^3\right)$,respectively if $m\gg d$, $d\gg m$, and $m\approx d$.

The computationally expensive steps arise during each Newton iteration of the interior-point optimization process, where the objective function's gradient and Hessian are computed. The computation of the full gradient vector has a time complexity of $O\left(d^2\right)$, due to the pair-wise interactions in the Frobenius norm involving the Gram matrix. At the same time, the asymptotic growth of the Hessian matrix also requires $O\left(d^2\right)$ time. However, the most computationally intensive operation is solving the Newton system $H_L\cdot\Delta\boldsymbol{a}=-\nabla L\left(\boldsymbol{a}\right)$, which incurs a time complexity of $O\left(d^3\right)$ per iteration due to matrix factorization. As such, the computational difficulty in the PLRU algorithm is due to solving this linear system at each step.

Finally, Table 1 shows the asymptotic computational complexities of the PLRU and other algorithms. In Table 1, s, r, and l denote the number of iterations, the number of clusters, and the number of selected features, respectively. s_1 is the total number of iterations for solving the indicator matrix. s_2 is the sum of the number of iterations for solving the auxiliary matrix.

4 Experimental procedure

This section clarifies the experimental methodology, twelve benchmark data sets, unsupervised feature selection algorithms, evaluation metrics, and implementations.

4.1 Benchmark data sets

We measure the performance of the PLRU by using 12 real-world data sets and one artificial data set (i.e., Madelon) from the UCI Machine Learning Repository.[1] The descriptive information of the benchmark data sets is shown in

[1] https://archive.ics.uci.edu/

Table 3 The default hyperparameter values of the compared algorithms and proposed PLRU algorithm

#	Algorithms	Parameters
1	LS [46]	$Metric$=′$Cosine$′ $NeighborMode = $′$KNN$′ $WeightMode$=′$Cosine$′ $k = 3$
2	RNE [47]	$m = d$ $numNeighbor = 5$
3	DUFS [48]	$class_{size} = numberofclusters$
4	USFS [49]	$lambda1 = 1$ $lambda2 = 1$ $lambda3 = 1$ $class_{num} = numberofclusters$ $m = d$ $alpha = 10000$
5	SLNMF [30]	$beta = 1$ $gamma = 10000$ $r = numberofclusters$ $W = J_{d,m}$ $H = NSSRD_{init_{S(X^T,r)}}$ $G = J_{m,r}$ $alpha = 0.1$ $beta = 0.001$
6	HSL [37]	$beta1 = 10000$ $gamma = 0.01$ $l = d$ $NIter = 10$ $alpha = 0.001$ $beta = 0.01$ $m = numberofclusters$ $O = 6$
7	PLRU$_1$ (adjusted α and limited search)	$\alpha = [0, 1]$ Interior-point parameters have default values as below: $OptimalityTolerance = 1e-6$ $StepTolerance$: $1e-10$ $MaxIterations$: 1000 $MaxFunctionEvaluations$: 3000 $HessianApproximation = $′$bfgs$′
8	PLRU$_2$ (adjusted α and a relatively detailed search)	$\boldsymbol{\alpha} = [0, 1]$ Interior-point parameters: $OptimalityTolerance = 1e-10$ $StepTolerance$: $1e-12$ $MaxIterations$: 3000 $MaxFunctionEvaluations$: 10000 $HessianApproximation$=′$bfgs$′

Table 2. Table 2 contains 13 data sets. However, we note that the COIL-20 data set is used to demonstrate the robustness of the proposed method for the feature visualization experiment. In other words, the COIL-20 data set is not employed in the classification or clustering experiments. Meanwhile, the data sets belong to various domains and each has different learning challenges. Thus, they allow for a robust and comprehensive assessment of the proposed UFS algorithm. A summary of the data sets and the justification for their selection are specified below:

- CNAE-9 and Madelon are high-dimensional and sparse data sets commonly used to test the performance of feature selection under complex distributions and irrelevant/noisy features.
- Sonar, Musk, and LSVT Voice Rehabilitation are real-world biomedical and physical measurement data sets with redundancy and non-linearity.
- Gait Classification, HAR-AAL, and TUANDROMD offer multi-class classification settings in behavioral and cybersecurity domains, assessing robustness to class imbalance and noise.
- Hill-Valley, Libras Movement, Optical Digits, and Semeion Handwritten Digit provide time-series or image-like data with structured patterns, suitable for evaluating the method's ability to preserve locality relations.
- COIL-20 is an object recognition data set with visual similarity across classes, which challenges a method's capacity to retain discriminative spatial information. In the meantime, this data set has been barely employed for the feature visualization experiment.

The aforementioned data sets are selected to assess the proposed algorithm under properties such as data type (e.g., numeric, image-based, time-series), dimensionality, noise level, and class. Thereby, it is shown that the experimental results cover multiple domains.

4.2 Comparison methods

We compare the proposed PLRU method with the six cutting-edge algorithms. The algorithms used in the experiments and their parameters are shown in Table 3. The methods are conducted using their default parameter values in all the experiments. We have assigned the compared algorithms' hyperparameters to their default values in the corresponding work for a fair comparison. In other words, the values of the UFS algorithms' hyperparameters in Table 3 have been determined with respect to their papers and code. No additional tuning is performed in order to ensure a fair and unbiased comparison. The proposed PLRU algorithm is evaluated under two settings: $PLRU_1$ and $PLRU_2$. $PLRU_1$ and $PLRU_2$ rely on a roughly suitable α value. Further, $PLRU_1$ and $PLRU_2$ rely on non-exhaustive and comprehensive searches, respectively. Unless stated otherwise, 'PLRU' refers to '$PLRU_1$' throughout the section 'Results and Discussion'. Finally, we herein provide a concise explanation of these methods:

- Laplacian Score (LS) [46] assesses features based on their Laplacian scores, which indicate their ability to preserve local structure.
- Robust Neighborhood Embedding for unsupervised feature selection (RNE) [47] takes into account the neighborhood relationships and similarity of data points to preserve the data's local structure.
- feature Dependency-based Unsupervised Feature Selection (DUFS) [48] considers features' relationships and interactions and evaluates their degree of independence and information-carrying ability.

Table 4 The comparative results of the algorithms' maximum NMIs on the benchmark data sets

#	Algorithms							
	LS	RNE	DUFS	USFS	SLNMF	HSL	$PLRU_1$	$PLRU_2$
#1	0.5041±0.0	0.4716±0.1	0.4771±0.1	0.5553±0.1	0.4910±0.1	0.4629±0.1	**0.5679±0.1**	0.4695±0.1
#2	0.0386±0.0	0.0190±0.0	0.0456±0.0	0.0481±0.0	0.0175±0.0	0.0177±0.0	**0.1067±0.0**	**0.1038±0.0**
#3	0.6836±0.0	0.6678±0.0	0.6754±0.0	0.6678±0.0	0.6692±0.0	0.6767±0.0	**0.8152±0.0**	**0.7397±0.0**
#4	0.5267±0.0	0.5288±0.0	0.5237±0.0	0.5222±0.0	0.5337±0.0	0.5241±0.0	**0.5540±0.0**	**0.5348±0.0**
#5	0.0002±0.0	0.0001±0.0	0.0001±0.0	0.0002±0.0	0.0001±0.0	0.0001±0.0	**0.0003±0.0**	0.0001±0.0
#6	0.5811±0.0	0.5900±0.0	0.5905±0.0	0.5863±0.0	0.5882±0.0	0.5867±0.0	**0.6017±0.0**	**0.5924±0.0**
#7	0.0396±0.0	0.1529±0.0	0.0534±0.0	0.0624±0.0	0.0817±0.0	0.0411±0.0	**0.1554±0.0**	0.1492±0.0
#8	0.0128±0.0	0.0122±0.0	0.0373±0.0	0.0342±0.0	0.0105±0.0	0.0299±0.0	**0.0403±0.0**	0.0096±0.0
#9	0.0213±0.0	0.0364±0.0	0.0374±0.0	0.0242±0.0	0.0210±0.0	0.0281±0.0	**0.0704±0.0**	**0.0459±0.0**
#10	0.7580±0.0	0.7562±0.0	0.7570±0.0	0.7469±0.0	0.7532±0.0	0.7535±0.0	**0.7619±0.0**	**0.7619±0.0**
#11	0.5681±0.0	0.5632±0.0	0.5465±0.0	0.5435±0.0	0.5520±0.0	0.5248±0.0	**0.5720±0.0**	**0.5796±0.0**
#12	0.1352±0.1	0.1520±0.1	0.2073±0.0	0.2659±0.1	0.3527±0.1	0.1561±0.1	**0.5779±0.2**	0.2271±0.1
Avg	0.3224±0.3	0.3292±0.3	0.3293±0.3	0.3381±0.3	0.3392±0.3	0.3168±0.3	**0.4020±0.3**	**0.3511±0.3**

The highest values are highlighted in bold. The second highest values are highlighted in underlined

- Unsupervised Soft-label Feature Selection (USFS) [49] evaluates the contribution of each feature with a soft-labeling approach. This approach expresses the degree to which features belong to a particular class, thus providing a more flexible feature evaluation process.
- Soft-Label guided Non-negative Matrix Factorization (SLNMF) [30] softly incorporates the data's class labels into the factorization process. This approach considers the possibility that each data point belongs to more than one class.
- unsupervised feature selection with High-order Similarity Learning (HSL) [37] focuses on pairwise similarities and multiple similarity relationships. This strategy provides a deeper and more comprehensive analysis of the data.

4.3 Evaluation criteria

The accuracy rate criteria will be used to measure the classification performance of the proposed algorithm. According to the confusion matrix, the ACCuracy rate (ACC) equation is given in Eq. (21). The value is desired to be close to 1 as a ratio. It can be said that the model's performance decreases as it approaches 0.

		Actual	
		Positive	Negative
Prediction	Positive	TP	FP
	Negative	FN	TN

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

Regarding clustering problems, we use the Normalized Mutual Information (NMI) criterion while measuring the performance of clustering algorithms according to the features selected by the unsupervised feature selection algorithms. The equation for calculating this criterion is given in Eq. (22):

$$NMI\left(Y, C\right) = \frac{2 \times I\left(Y, C\right)}{H\left(Y\right) + H\left(C\right)} \tag{22}$$

where $H\left(\cdot\right)$ and $I\left(\boldsymbol{Y}; \boldsymbol{C}\right)$ are denoted as the entropy of a random variable and the mutual information of two random variables $\boldsymbol{Y}$ and $\boldsymbol{C}$, respectively.

When testing the quality of feature subsets selected by the unsupervised feature selection algorithms on data sets, the K-means algorithm is used as a clustering algorithm. In terms of classification, the k-Nearest Neighbors algorithm (KNN) is considered to be used as a classification algorithm. We have repeated K-means and KNN algorithms thirty times and have reported their average performances. Finally, the experiments performed in this work have been conducted in the MATLAB R2023b on a CPU at 1.6 GHz with 8 GB of RAM.

5 Experimental results and discussion

In this section, we perform a variety of experiments to show the effectiveness of the proposed PLRU algorithm on the benchmark data sets. To this end, we concentrate on those empirical analyses: (i) the clustering performance analysis, (ii) the classification performance analysis, (iii) the feature visualization, (iv) the hyperparameter sensitivity analysis, (v) the convergence analysis, and (vi) the computational complexity analysis.

5.1 The clustering performance analysis

This section compares the proposed framework with the six cutting-edge unsupervised feature selection algorithms on 12 real-world data sets and one synthetic data set. Table 4 shows the maximum values of the clustering results of the unsupervised feature selection algorithms used in the experiments. The results obtained by PLRU_1 are approximately optimal with respect to the selected α parameter. Moreover, the interior-point method seems to have converged without performing an in-depth exploration of the search space. However, the results of PLRU_2 were obtained by a comprehensive search using the same α value. In the meantime, the roughly suitable α initial values corresponding to the data sets are 0.0, 1.0, 0.3, 0.4, 0.5, 1.0, 0.6, 0.1, 0.0, 0.8, 0.0, and 0.4, respectively. According to the results, PLRU_1 outperforms the other algorithms in all the benchmark data sets.

Furthermore, the average NMI result of the proposed PLRU algorithm on the benchmark data sets is higher than the other unsupervised feature selection algorithms. In this regard, the SLNMF algorithm ranks second after the proposed PLRU. Finally, the PLRU algorithm demonstrates its superiority in different data sets where the number of features exceeds the number of samples, the imbalance rate is high, the number of classes is larger than two, and so on. No algorithms outperform the other unsupervised feature selection algorithms in data sets.

The USFS algorithm surpasses the compared algorithms on 2 data sets. Consequently, different types of data sets exist on which each unsupervised feature selection algorithm is better. However, from the point of view of the average result, the performance of the algorithms can vary. In this respect, PLRU ranks first compared to the other algorithms in terms of the average NMI outcome on the data sets. Finally, we would like to underline that the DUFS, USFS, SLNMF, and HSL algorithms need the number of

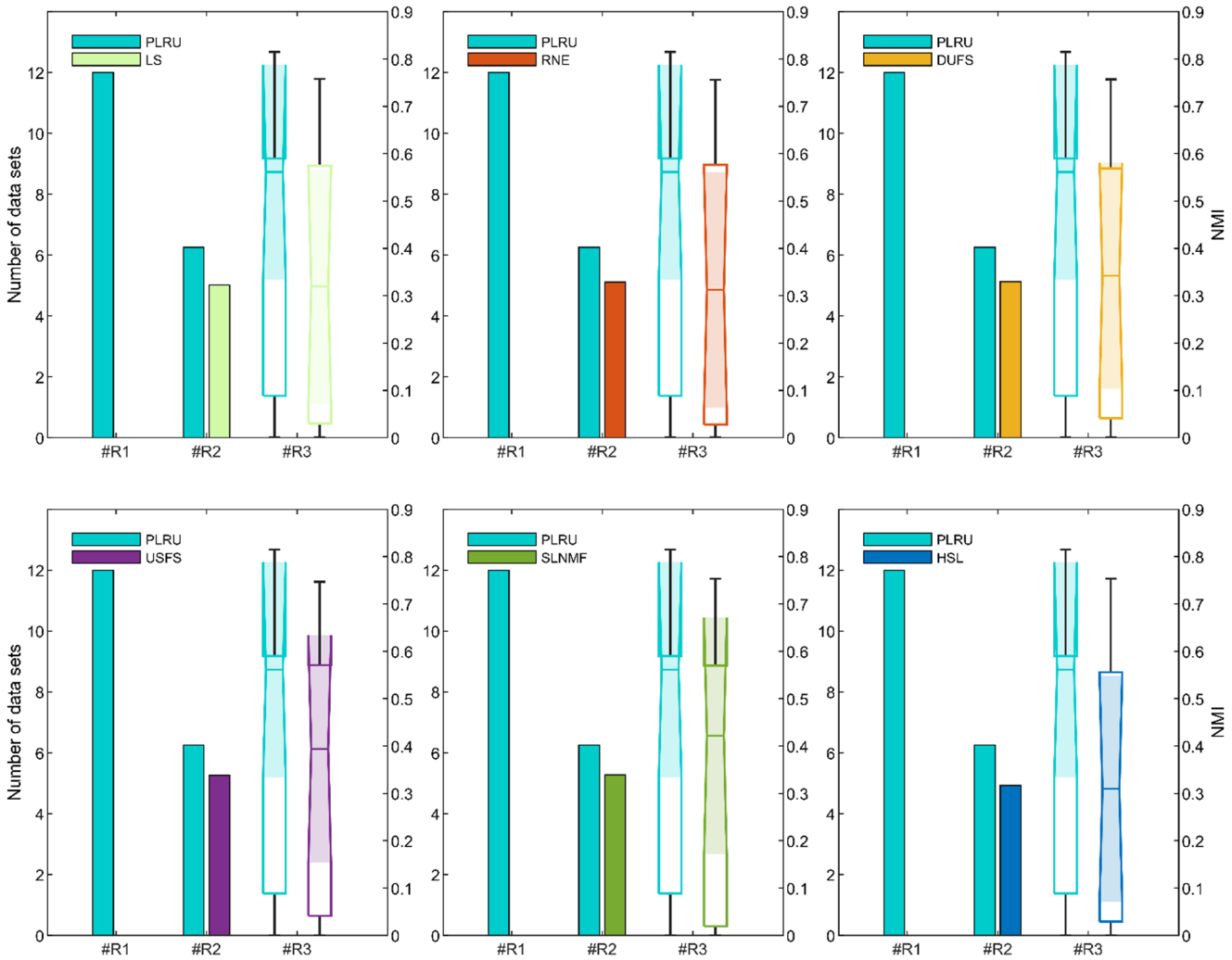

Fig. 2 The comparative results of the $PLRU_1$ and other algorithms on the benchmark data sets. The #R1 denotes the comparative results of the number of data sets with maximum NMI. The #R2 represents comparative results in terms of the average NMI. The #R3 points out three quartile values (i.e., median, lower quartile, upper quartile), maximum, and minimum values of the NMIs over the data sets. The median value is shown as the line inside the box. The whiskers extending below and above the box indicate the endpoints corresponding to the minimum and maximum NMIs. Box charts with notches hold different medians at 5% significance

clusters as the hyperparameter. In this respect, we emphasize that the number of clusters is unknown for some data sets, i.e., unsupervised ones. Note that the LS, RNE, and PLRU algorithms do not require the number of clusters.

The NMI scores of the PLRU algorithm are calculated using two optimization strategies: a coarse search ($PLRU_1$) and a more exhaustive fine-tuned search ($PLRU_2$). The results show that $PLRU_2$ mostly obtains higher NMI values across several data sets. However, there are data sets where the fine-tuned search does not yield a performance gain. As a result, the findings point out that a detailed optimization may boost the clustering performance. However, it can also lead to marginal degradation due to overfitting or the sensitivity of the objective landscape.

Figure 2 shows the comparative results of the $PLRU_1$ and other algorithms on the benchmark data sets. The #R1 designates the number of data sets where the compared algorithms have maximum NMI. The #R2 denotes the comparison of the average NMIs of the compared algorithms over the benchmark data sets. The #R3 indicates the median, maximum, minimum, and quartile values (i.e., lower and upper quartiles) of the compared algorithms' NMIs over the benchmark data sets. The line inside the box signifies the median value. The endpoints corresponding to the minimum and maximum NMIs of the compared algorithms are shown as the whiskers extending below and above the box. Box charts with notches hold different medians at 5% significance. Concerning the comparisons between the PLRU and other methods, the proposed PLRU algorithm yields maximum NMI on all data sets. In point of the average NMI, the result of the PLRU algorithm is larger than that of the other algorithms. The proposed method also offers higher

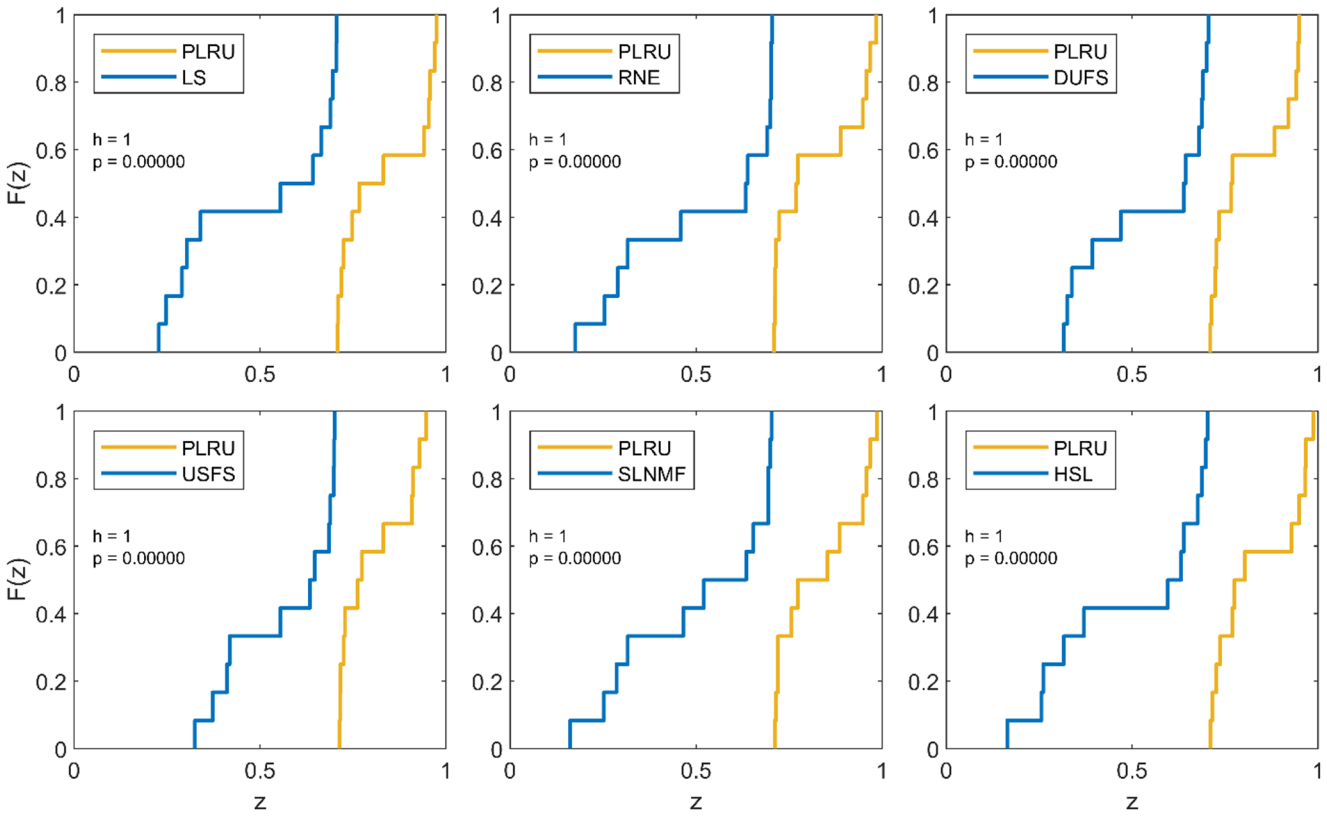

Fig. 3 The comparison of the statistical significance between the proposed $PLRU_1$ and the methods compared over the benchmark data sets. The symbol $\boldsymbol{h}$ indicates that whether the Kolmogorov–Smirnov test rejects the null hypothesis at the 1% significance level or not. The symbol $\boldsymbol{p}$ denotes the probability of observing test statistics as large as, or larger than, the observed value under the null hypothesis. The x-axis represents the normalized sample data vector. The y-axis denotes the cumulative distribution function of the normalized sample data vector distribution

results than other algorithms in terms of median, minimum, and quartile values. Consequently, we can remark that the proposed PLRU algorithm gains better results than the other algorithms in all the comparisons.

Figure 3 depicts the comparative results of the statistical significance between the proposed $PLRU_1$ algorithm and compared algorithms over the benchmark data sets. We used the Kolmogorov–Smirnov test as the significant test to analyze whether two samples came from the same distribution. The two-sample Kolmogorov–Smirnov test is a nonparametric test that appraises the difference between the cumulative distribution functions of the distributions of the two samples over a particular range in data sets. First, note that the experimental results are normalized by Euclidean norm for a fair test. The symbol h denotes whether the Kolmogorov–Smirnov test rejects the null hypothesis at a certain significance level (α), or not. The symbol p indicates the probability of observing test statistics as large as, or larger than, the observed value under the null hypothesis. We have determined that the significance level (α) is 1%. Accordingly, the null hypothesis is rejected if $h = 1$ or $p < \alpha$, and vice versa. The null hypothesis is that two samples are from the same continuous distribution. As a result, the NMI results between the proposed PLRU and other methods that have been run on the benchmark data sets have statistically significant differences at the 1% significance level.

Figure 4 represents the NMI performances of the unsupervised feature selection algorithms used in the experiments depending on the equally spaced feature numbers. This experiment divided the number of features into ten equal parts, and the results showed the maximum NMI values within the corresponding range. Thus, this approach ensures that a fair experiment is carried out and that the performances of the algorithms are observed across the selected features. Besides, the results somewhat present a detailed view of Table 4. In light of the results, the NMI performances of the algorithms mainly vary as the number of features increases. First of all, note that the proposed PLRU algorithm does not need a hyperparameter such as the number of clusters, unlike the other compared algorithms, excluding the LS and RNE methods. Further, the NMI performances of the algorithms differ between the benchmark data sets. The overall NMI behavior of some algorithms, according to the number of features, resemble each other

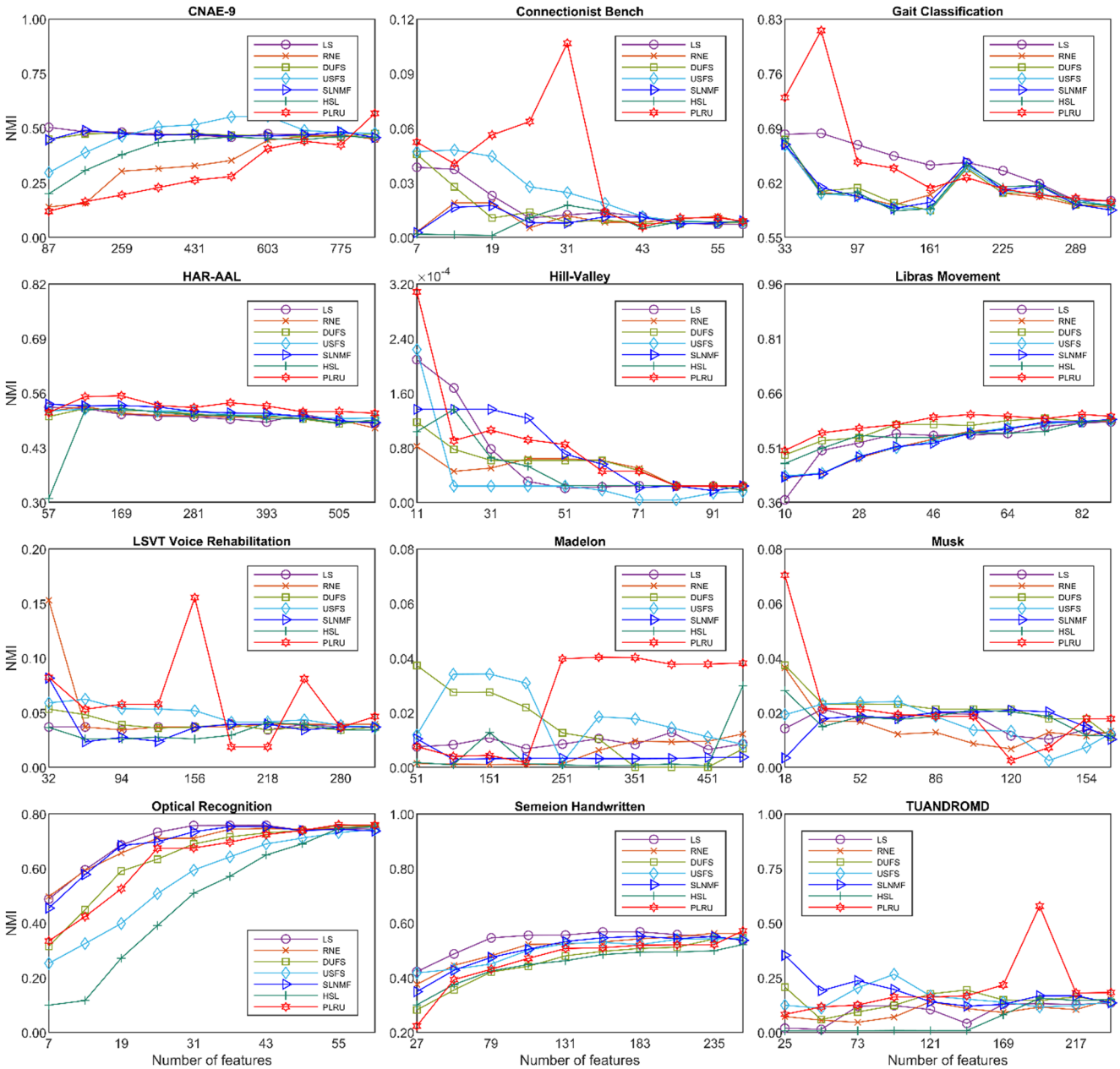

Fig. 4 The comparative NMI results of the unsupervised feature selection algorithms used in the experiments according to the equally spaced feature numbers

in some data sets. Consequently, the proposed PLRU algorithm performs well in capturing the maximum NMI on most of the data sets.

Figure 5 shows (a) t-SNE visualization on the whole features; (b) t-SNE manifestation on the subspace after applying PLRU; (c) t-SNE representation of cluster assignments obtained by k-means on the selected features. Several indicators may give rise to the low NMI results on the two-class data sets. These are inter-class overlapping, the loss of discriminant dimensionality in feature selection, the global cluster assumption of k-means, the low sample size, and intra-class scatter. Analysis of the two-class data sets shows that data points are intertwined in both the original space and the subspace after feature selection. In particular, in cases #7 and #8, the entire data is distributed almost as a single cloud, with no clear decision boundary distinguishing the two classes. In this case, the clusters assigned by k-means are inconsistent with the true labels, and mutual information approaches 0. In cases #2 and #9, the data distribution in t-SNE becomes more compact and homogeneous along the axes as it moves from the full feature space to the feature subspace. In other words, the selected features

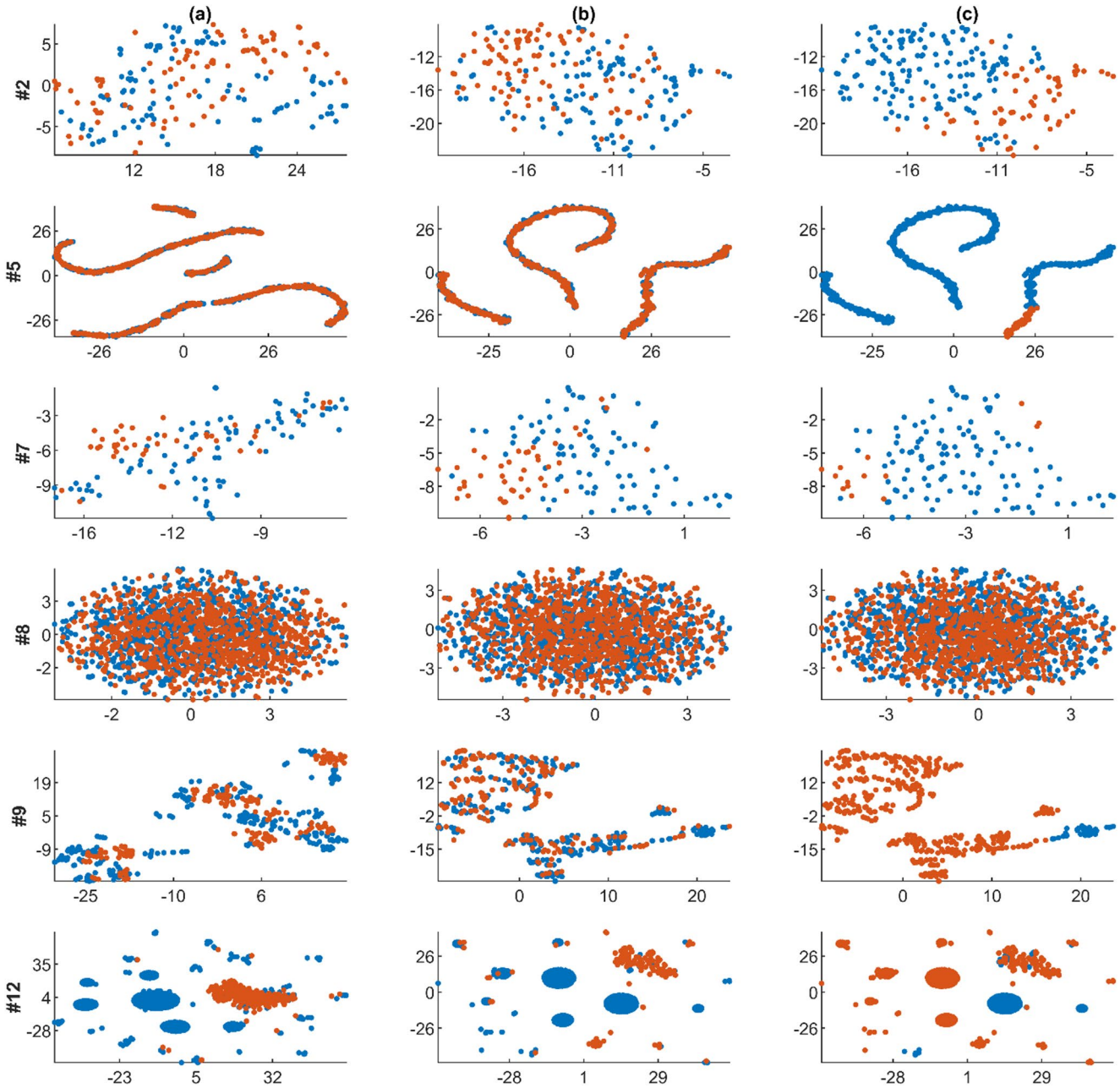

Fig. 5 t-SNE visualization on the data sets with binary class: (**a**) t-SNE visualization on the whole features; (**b**) t-SNE manifestation on the subspace after applying PLRU; (**c**) t-SNE representation of cluster assignments obtained by k-means on the selected features.

partially reflect the distinction between the classes in the new dimension.

Furthermore, in #9, one class in particular is very scattered, with small samples randomly clustered. Because even t-SNE cannot provide a clear distinction, the true labels become nearly independent of k-means' random assignments. Therefore, the k-means algorithm clusters two different classes. In #5 and #12, the t-SNE output displays complex clustering patterns, including rings and curved-spiral structures. K-means, however, can only properly separate circle-like spherical clusters. In these complex patterns, k-means assignments are located in regions completely different from the true class.

5.2 The classification performance analysis

Table 5 shows the maximum values of the classification results of the unsupervised feature selection algorithms used in the experiments. According to the ACC results, the proposed PLRU algorithm outmatches the other unsupervised

Table 5 The comparative maximum ACC results of the algorithms on the benchmark data sets

#	Algorithms								
	LS	RNE	DUFS	USFS	SLNMF	HSL	$PLRU_1$	$PLRU_2$	$PLRU_{1,\ tuned}$
#1	0.8764±0.0	0.8868±0.0	0.8737±0.0	**0.9029±0.0**	0.8777±0.0	0.8675±0.0	0.8982±0.0	0.8886±0.0	0.8982±0.0
#2	0.8313±0.0	0.8322±0.0	0.8279±0.0	0.8596±0.0	0.8279±0.0	0.8500±0.0	0.8542±0.0	**0.8686±0.0**	**0.8734±0.0**
#3	0.3458±0.0	0.3542±0.0	0.3542±0.0	0.3625±0.0	0.3563±0.0	0.3542±0.0	**0.6250±0.0**	**0.4236±0.0**	**0.6250±0.0**
#4	0.9460±0.0	0.9504±0.0	0.9433±0.0	0.9498±0.0	0.9507±0.0	0.9505±0.0	**0.9519±0.0**	**0.9519±0.0**	**0.9519±0.0**
#5	0.6549±0.0	0.6461±0.0	0.6407±0.0	0.6532±0.0	0.6452±0.0	0.6460±0.0	**0.6554±0.0**	**0.6658±0.0**	**0.6617±0.0**
#6	0.8758±0.0	0.8692±0.0	0.8750±0.0	0.8628±0.0	0.8633±0.0	0.8664±0.0	0.8713±0.0	**0.8815±0.0**	**0.8787±0.0**
#7	0.5706±0.0	0.7000±0.0	0.6865±0.0	0.6762±0.0	0.7802±0.0	0.7349±0.0	**0.7857±0.0**	**0.8042±0.0**	**0.7910±0.0**
#8	**0.8741±0.0**	0.6527±0.0	0.7883±0.0	0.7957±0.0	0.8602±0.0	0.6441±0.0	0.6483±0.0	0.6685±0.0	0.8177±0.0
#9	0.8635±0.0	0.8756±0.0	**0.8954±0.0**	0.8599±0.0	0.8872±0.0	0.8792±0.0	0.8747±0.0	0.8831±0.0	0.8922±0.0
#10	0.9860±0.0	0.9859±0.0	0.9863±0.0	0.9859±0.0	0.9860±0.0	0.9856±0.0	**0.9865±0.0**	**0.9874±0.0**	**0.9865±0.0**
#11	0.9166±0.0	**0.9212±0.0**	0.9159±0.0	0.9181±0.0	0.9192±0.0	0.9146±0.0	0.9167±0.0	0.9161±0.0	0.9201±0.0
#12	0.9801±0.0	0.9802±0.0	0.9802±0.0	0.9801±0.0	0.9802±0.0	0.9801±0.0	**0.9803±0.0**	**0.9804±0.0**	**0.9803±0.0**
Avg	0.8101±0.2	0.8045±0.2	0.8139±0.2	0.8172±0.2	0.8278±0.2	0.8061±0.2	**0.8374±0.1**	0.8266±0.2	**0.8564±0.1**

The highest values are highlighted in bold. The second highest values are highlighted in underlined

feature selection algorithms in 8 out of 12 benchmark data sets. The results obtained by $PLRU_1$ are approximately optimal with respect to the chosen α parameter. Additionally, the interior-point method appears to have converged without a comprehensive search. The results of $PLRU_2$ have been obtained with respect to the same α value at the end of the detailed search. The α values used in the clustering experiments are likewise used in the classification experiments. In other words, the fit α values are not separately searched for classification. The results of $PLRU_{1,\ tuned}$ have been obtained by α parameter tuned roughly. In the meantime, the roughly suitable α initial values corresponding to the data sets are 0.0, 0.0, 0.3, 0.4, 0.8, 0.4, 0.5, 0.0, 0.3, 0.8, 1.0, and 0.4, respectively. The average ACC result of the proposed PLRU algorithm on the benchmark data sets ranks first. Each of the LS, RNE, DUFS, and USFS algorithms exceeds the other unsupervised feature selection algorithms in 1 of 12 benchmark data sets. The SLNMF and HSL algorithms surpass the other algorithms in none of the 12 benchmark data sets. As a result, in terms of classification and clustering performance, there also exist data sets on which each unsupervised feature selection algorithm is superb or poor. Nevertheless, in terms of the average ACC result, the proposed PLRU algorithm is superior to the other unsupervised feature selection algorithms in point of the average ACC result on the data sets. Consequently, PLRU appears preferable to the other algorithms over the benchmark data sets from the different descriptive information and domains.

The classification accuracy results of $PLRU_1$ and $PLRU_2$ provide further insights into the effects of detailed optimization on downstream supervised tasks. According to the results, $PLRU_2$ has lower accuracy in some data sets compared to $PLRU_1$. This suggests that the fine-tuned optimization may lead to feature subsets that better preserve the local geometric structure of the data but not necessarily the information most relevant for separating class labels. On the other hand, small improvements are observed in several data sets, where $PLRU_2$ performs slightly better or on par with $PLRU_1$. These results point out a trade-off introduced by the detailed interior-point search. Although $PLRU_2$ may enhance clustering consistency, it does not turn to higher classification performance. The reason is possibly due to overfitting or a shift in the feature selection bias towards preserving locality rather than maximizing class separation.

The results indicate that improved clustering performance does not necessarily imply enhanced classification performance. Namely, the $PLRU_2$ can reach higher NMI results over some data sets, but its classification accuracy may be lower than $PLRU_1$. This situation shows that optimizing cluster structure (i.e., preserving local neighborhood relationships) and maximizing class separation may simultaneously conflict. Therefore, it is recommended that the proposed algorithm be configured and executed independently for clustering and classification tasks. Treating these tasks as separate objectives allows for more targeted optimization and it avoids possible conflicts between structure preservation in unsupervised tasks and discriminative performance in supervised tasks.

Figure 6 displays the comparative results of the PLRU and other algorithms on the benchmark data sets. The #R1 denotes the number of data sets where the compared algorithms have maximum ACC. The #R2 designates comparing the average ACCs of the compared algorithms over the benchmark data sets. The #R3 represents the quartile values (i.e., median, lower, and upper quartiles), maximum, and minimum values of the compared algorithms' ACCs over the benchmark data sets. The line inside the box shows the median value. The endpoints that correspond to the minimum and maximum ACCs of the compared algorithms are indicated as the whiskers that extend below and above the

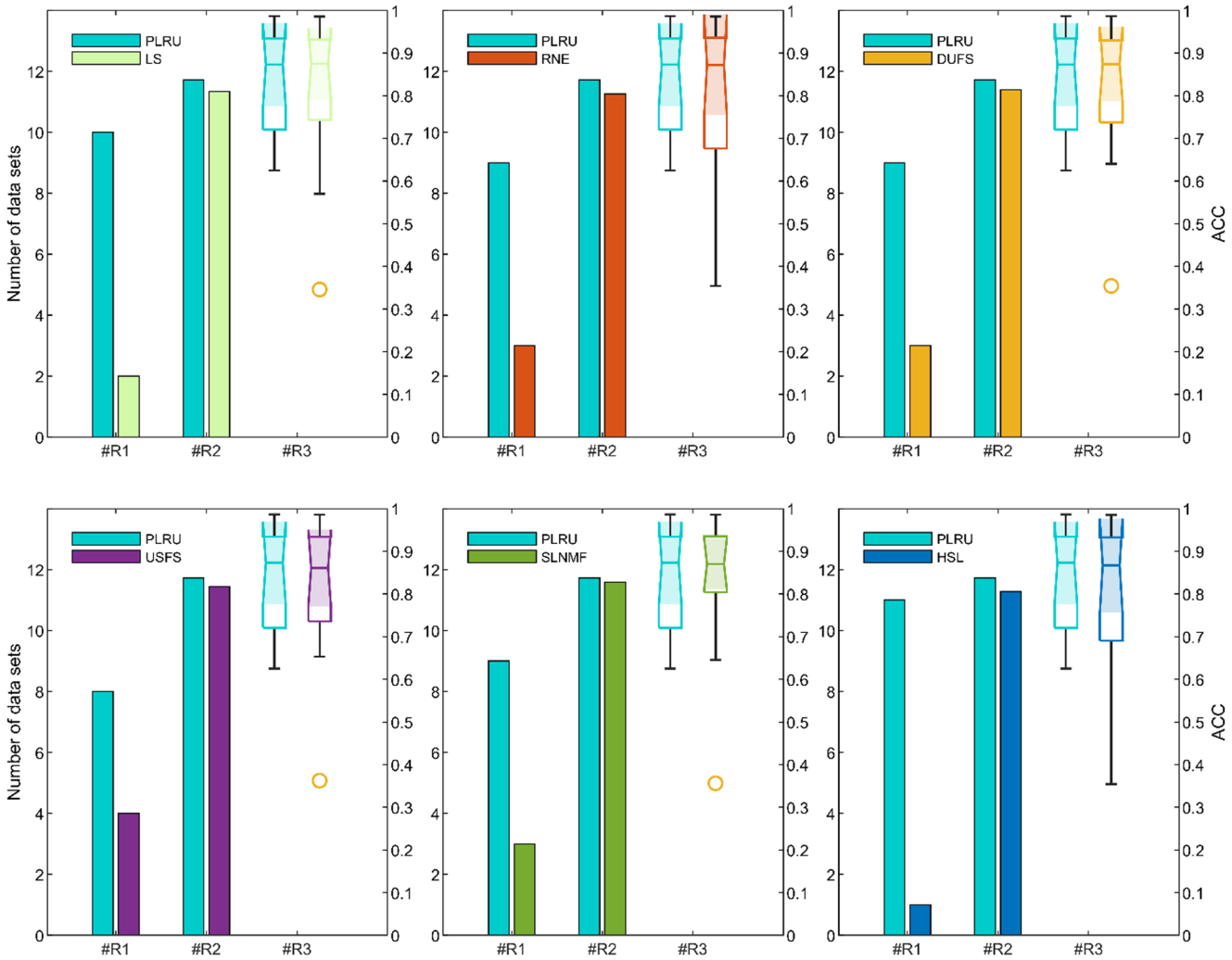

Fig. 6 Comparative results of the proposed $PLRU_1$ and other unsupervised feature selection methods on the benchmark data sets. The #R1 indicates the comparative results of the number of data sets with maximum ACC. The #R2 denotes the comparative results in terms of the average ACC. The #R3 represents the median, maximum, minimum, and other quartile values of the ACCs over the benchmark data sets. The median value is shown as the line inside the box. The whiskers extending below and above the box indicate the endpoints corresponding to the minimum and maximum ACCs. The yellow circles indicate outliers. Box charts with notches hold different medians at 5% significance

box. Box charts with notches hold different medians at 5% significance. Considering the comparative results between the proposed PLRU and the LS, RNE, DUFS, and USFS algorithms, the PLRU delivers maximum ACC in 8 out of 12 data sets. These four algorithms have the highest ACC result over only one data set. Relating to the results between the proposed PLRU, the compared HSL and SLNMF algorithms, the PLRU obtains maximum ACC on all the data sets. Consequently, the proposed PLRU algorithm acquires higher results than the other algorithms for all the comparisons in terms of the average ACC. All in all, we can state that the proposed PLRU algorithm gives remarkable ACC results.

Figure 7 exhibits the comparative results of the statistical significance between the proposed $PLRU_1$ algorithm and compared algorithms over the benchmark data sets in terms of ACC. First, the experimental results are normalized according to the Euclidean norm for a fair test. The symbol h indicates whether the Kolmogorov–Smirnov test rejects the null hypothesis at a particular significance level (α), or not. The symbol p denotes the probability of observing test statistics as large as, or larger than, the observed value under the null hypothesis. We have determined that the significance level (α) is 7%. Accordingly, the null hypothesis is accepted if $h = 0$ or $p \geq \alpha$, and vice versa. The null hypothesis is that two samples are from the same continuous distribution. Consequently, the ACC results between the proposed PLRU and other algorithms excluding USFS conducted over the benchmark data sets possess statistically significant differences at the 7% significance level. In this respect, we can

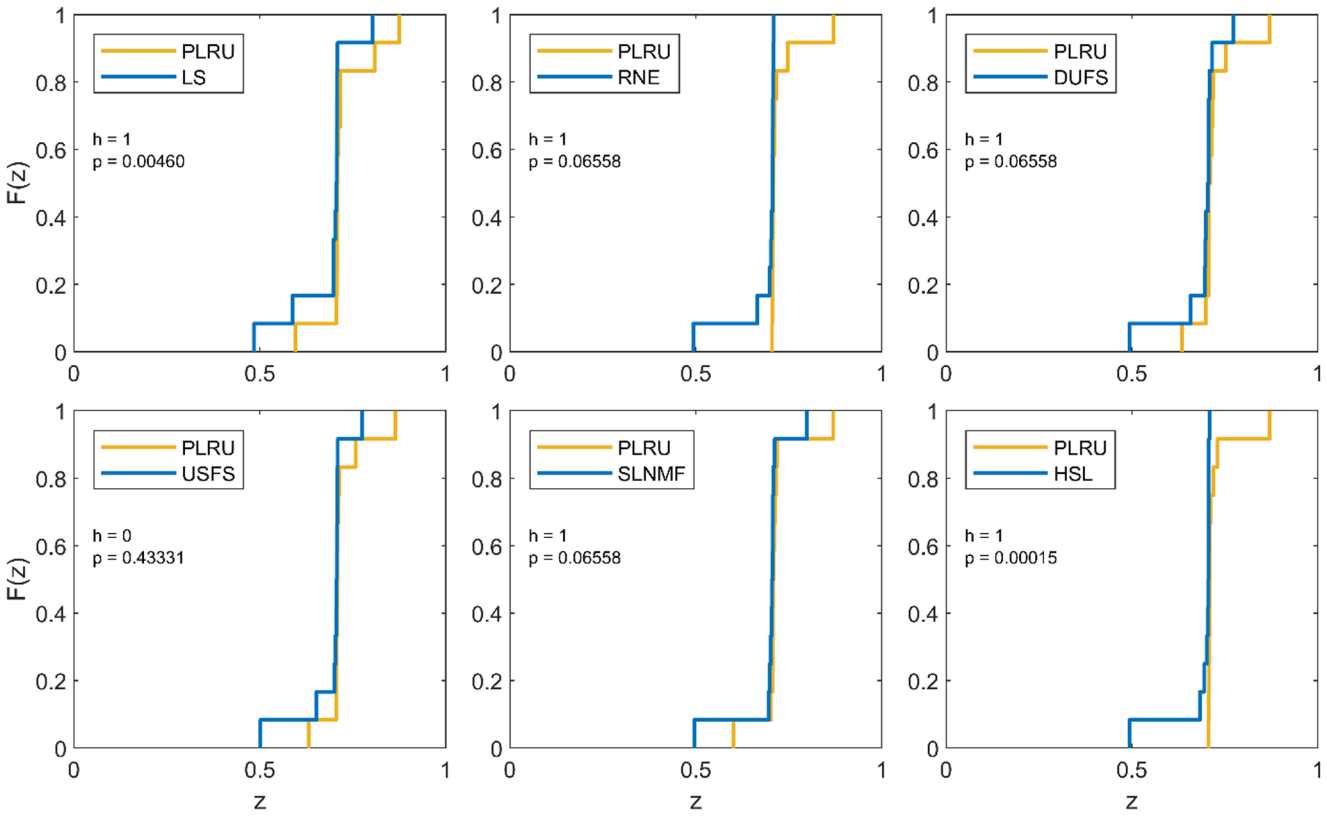

Fig. 7 The comparison of the statistical significance between the proposed $PLRU_1$ and the methods compared over the benchmark data sets. The symbol $\boldsymbol{h}$ indicates whether or not the Kolmogorov–Smirnov test accepts the null hypothesis at the 7% significance level. The symbol $\boldsymbol{p}$ denotes the probability of observing test statistics as large as, or larger than, the observed value under the null hypothesis. The x-axis indicates the normalized sample data vector. The y-axis denotes the cumulative distribution function of the normalized sample data vector distribution

report that the performance of the proposed PLRU is different than that of other algorithms in terms of ACC.

Figure 8 illustrates the ACC performances of the proposed $PLRU_1$ and the compared unsupervised feature selection methods depending on the equally spaced feature numbers. In this experiment, the number of features has been divided into ten equal parts. This operation guarantees that a fair experiment has been realized and that the performances of the existing methods are monitored across the selected features. Furthermore, the results put forward a little detailed view of Table 5. Considering the empirical results, the ACC performances of the algorithms vary as the number of features increases. Besides, the ACC performances of the algorithms generally differ between the benchmark data sets. In most data sets, the proposed PLRU algorithm achieves high ACC starting from low feature numbers, while in some, the opposite occurs. This situation is also actual for some other algorithms. Thus, it has been demonstrated once again that there are data sets where the algorithms are better and data sets where they produce lousy performance. Consequently, the proposed PLRU algorithm performs well in capturing the maximum ACC over many data sets.

When the ACC and NMI results are considered, it is seen that the ACC values are relatively higher than the NMI values. We interpret this situation as follows: K-means can capture global structure but may miss local structure. If the data has a manifold-based or non-linear separable structure, the cluster centers may not be homogeneous even though K-means seems to put each class in a separate cluster. ACC looks for the largest overlap. However, NMI takes the entire distribution into account. Especially in a context where local structure is preserved such as the proposed PLRU, the global nature of K-means may have neglected some structures. In particular, some clusters may exhibit label mixing or capture dominant classes disproportionately, which may lead to a decrease in mutual information despite high assignment accuracy. Such behaviors are common in data sets with class imbalance or where certain features favor dominant clusters.

5.3 Feature visualization

The feature visualization experiment was explicitly designed to illustrate qualitatively how each method

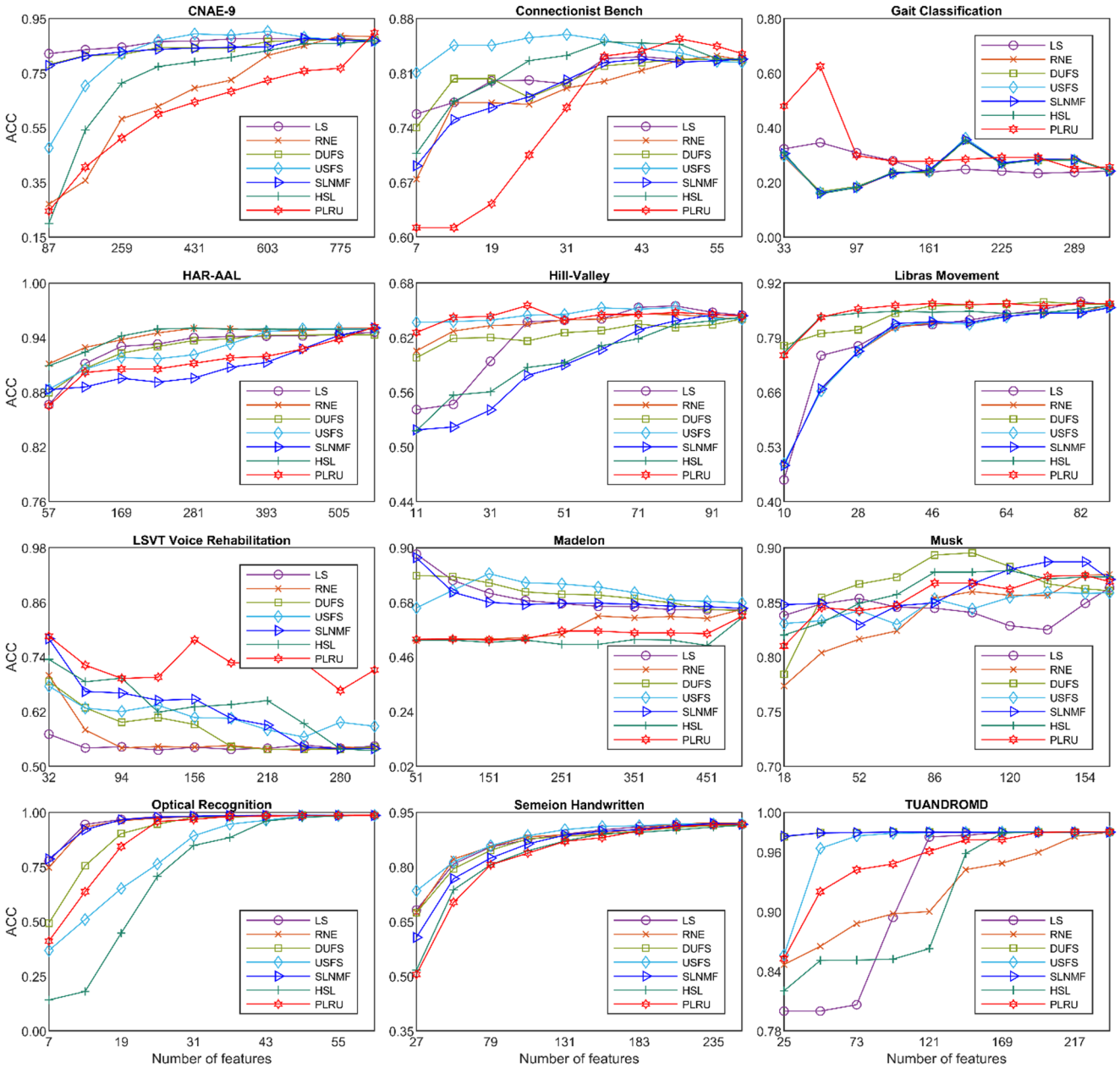

Fig. 8 The comparative ACC results of the unsupervised feature selection algorithms used in the experiments according to the number of features

preserves or distorts the spatial structure of a single image instance after feature selection. In other words, this experiment aims to provide an intuitive, visual understanding of how different feature selection methods retain or suppress discriminative visual information in a localized setting. Figure 9 visualizes the features of five objects on the COIL20 data set by LS, RNE, DUFS, USFS, SLNMF, and PLRU. The columns show the original image and ones with feature numbers of 50, 100, 150, 200, 250, and 300, respectively. Further, HSL does not work on visual data because the number of clusters is selected as one. Hence, we cannot present the visual results of HSL. Several observations on feature visualization are as follows: PLRU enables the selection of local and global features for different objects. PLRU identifies the duck's mouth, the building block's point, the car's hood, the cat's head, and the mug's rim for local features. It captures interior and bounding aspects representing objects such as duck shape, building block silhouette, car contour, cat form, and mug figure for global features. Even though the number of features is low, PLRU can uncover the outline of objects like the duck, the building block, the car, the cat, and the mug. SLNMF and USFS concentrate more on global features. LS and DUFS are adequately unable to focus on local and global features. RNE cannot adequately identify

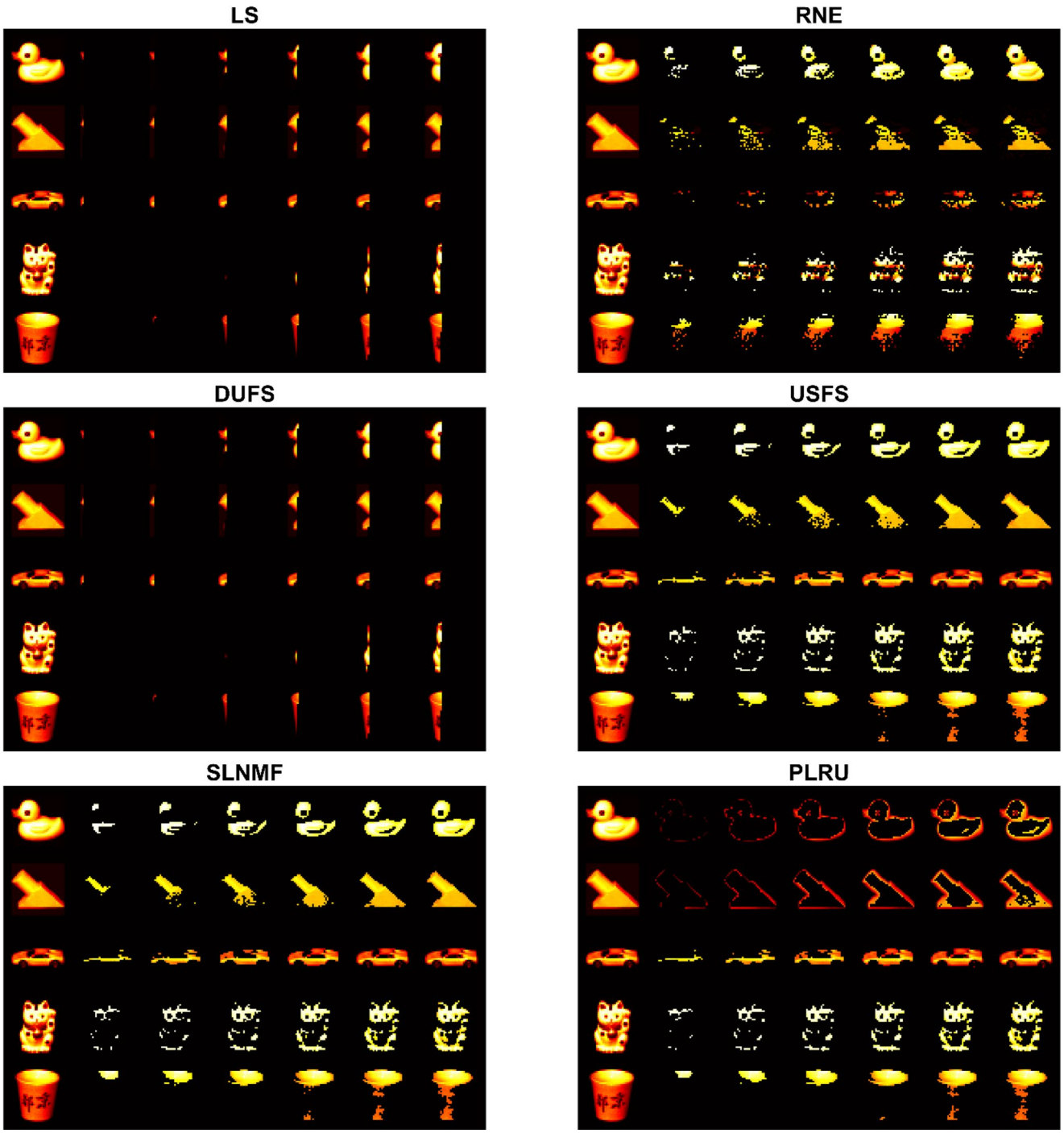

Fig. 9 Feature visualization of the COIL-20 data set by LS, RNE, DUFS, USFS, SLNMF, and PLRU. The columns show the original image and ones with feature numbers of 50, 100, 150, 200, 250, and 300, respectively

the duck's mouth/tail, car, and mug's outline. Besides, it can partly pinpoint the building block and cat's outline. Overall, RNE is unable to efficiently capture the local and global features of the objects. Hence, it is poorer than PLRU, SLNMF, and USFS. Further, SLNMF and USFS cannot efficiently capture global features well, but the PLRU method can. As the number of features ascends, PLRU detects the features in the global and local regions of the objects. Like SLNMF and USFS, PLRU cannot only identify the car and mug's outline when the low number of features. However, unlike SLNMF and USFS, PLRU can capture the duck and the building block's contours even with a few features.

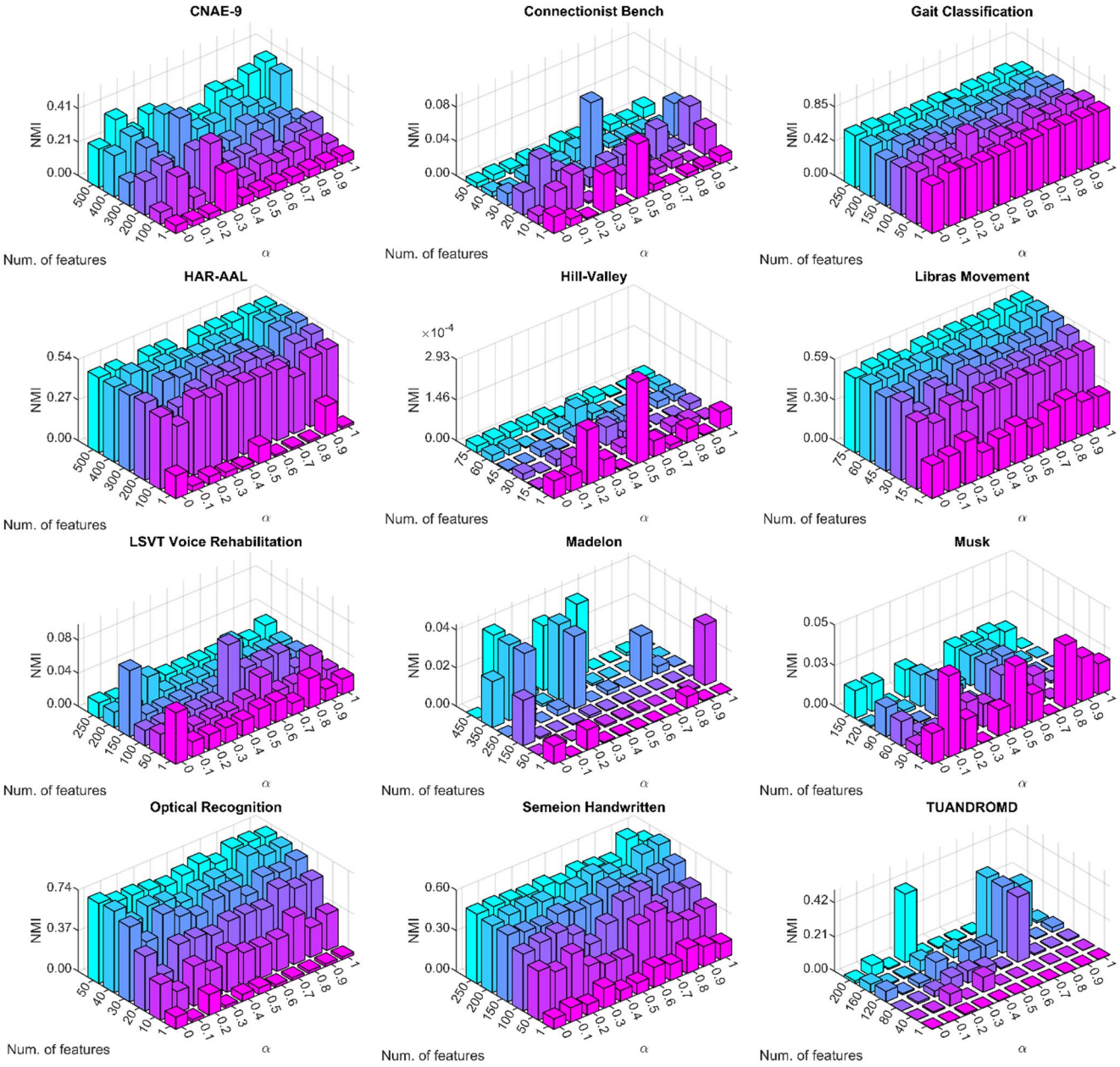

Fig. 10 The NMI results of PLRU on the benchmark data sets in terms of the different numbers of features and the various values of α

5.4 The hyperparameter sensitivity analysis

In this sub-section, we evaluate the role of α on 12 benchmark data sets by considering $\alpha \in [0, 1]$ and the number of selected features. The hyperparameter α manages the balance between structure preservation and feature selection sparsity. Figure 10 shows the NMI results of PLRU on the benchmark data sets in terms of the different numbers of features and the distinct values of α. According to the NMI results, the hyperparameter α clearly influences the performance of the proposed algorithm. This effect is more apparent in some data sets, such as Hill-Valley. However, the impact of α is moderate and consistent in some data sets, such as Gait Classification. The NMI results do not consistently rise across some data sets, such as Gait Classification, as the number of features increases. Further, there is no steady α value to provide the highest NMI values. Therefore, it is necessary to search for an appropriate α value to gain the highest NMI value. The proper α value may differ between data sets, so the appropriate α value must be searched to obtain the highest NMI result. In addition, the number of features yielding the highest NMI result varies between data sets. However, the proposed algorithm's limited search range facilitates this optimization process.

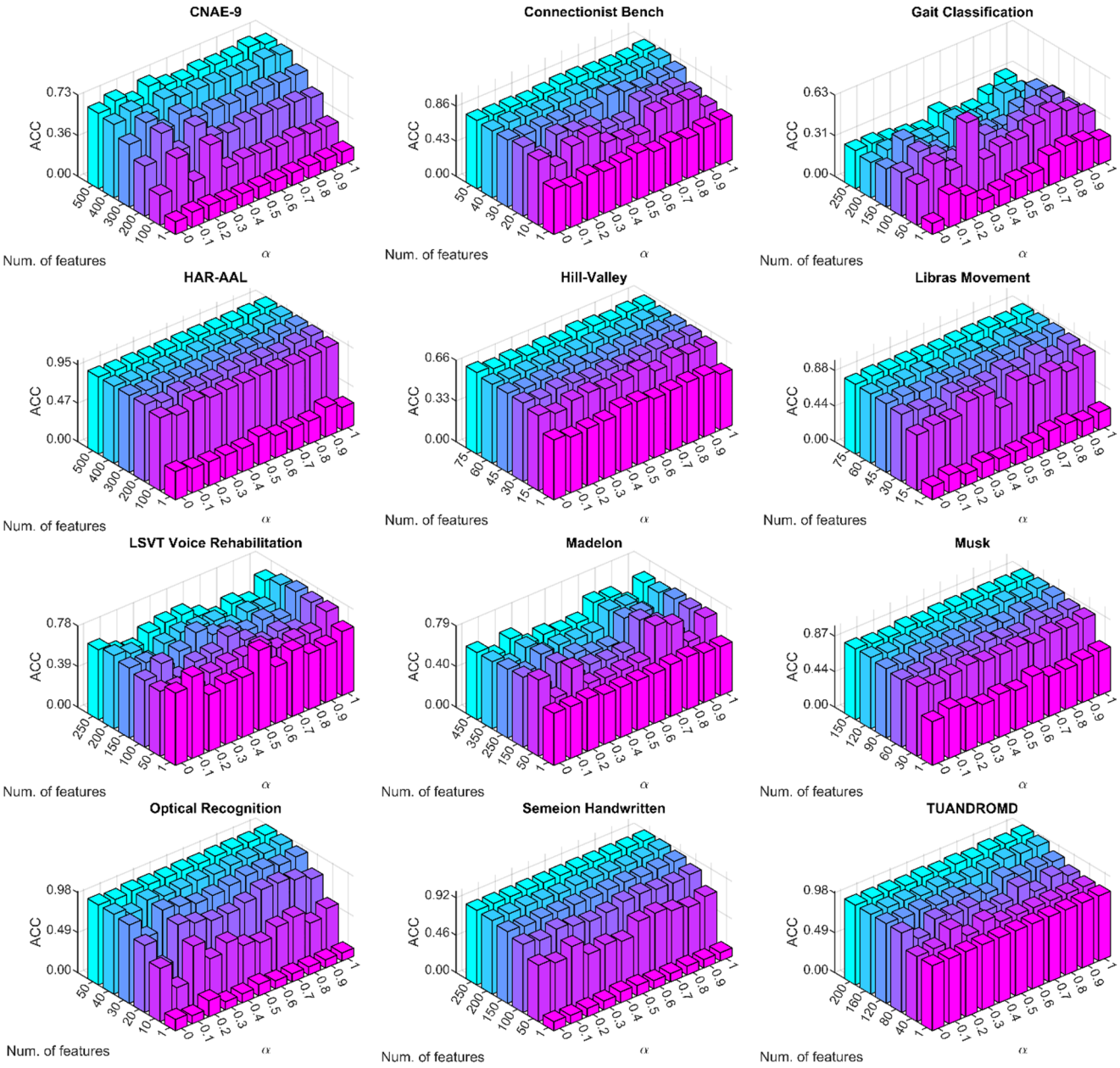

Fig. 11 The ACC results of PLRU on the benchmark data sets in terms of the different numbers of features and the various values of α

The optimal α value significantly varies across data sets. This situation indicates that the effect of α is closely related to intrinsic data characteristics such as sample size, class separability, and feature redundancy. For instance, in Gait Classification and Libras Movement, moderate values of α consistently yield strong performance, likely due to the proposed method's ability to balance smooth local manifold preservation with discriminative feature sparsity. Low or moderate α values perform better when fewer features are selected as they enforce stronger sparsity. Conversely, higher α values can become more effective as the number of the selected features increases and this situation can result in better performance in some data sets like HAR-AAL and Semeion Handwritten when more features are retained. Some data sets such as Madelon, Hill-Valley, and TUANDROMD exhibit very low and stable performance regardless of α. This situation is likely due to their inherent noise and irrelevance in features, making them less sensitive to α but more sensitive to feature quality overall.

As for practical guidance on α selection, for data sets with well-defined manifolds and high feature redundancy, $\alpha \in [0.3, 0.6]$ is generally optimal. For noisy or high-dimensional synthetic data sets, smaller α values (e.g., < 0.2) are recommended. If the number of selected features is large

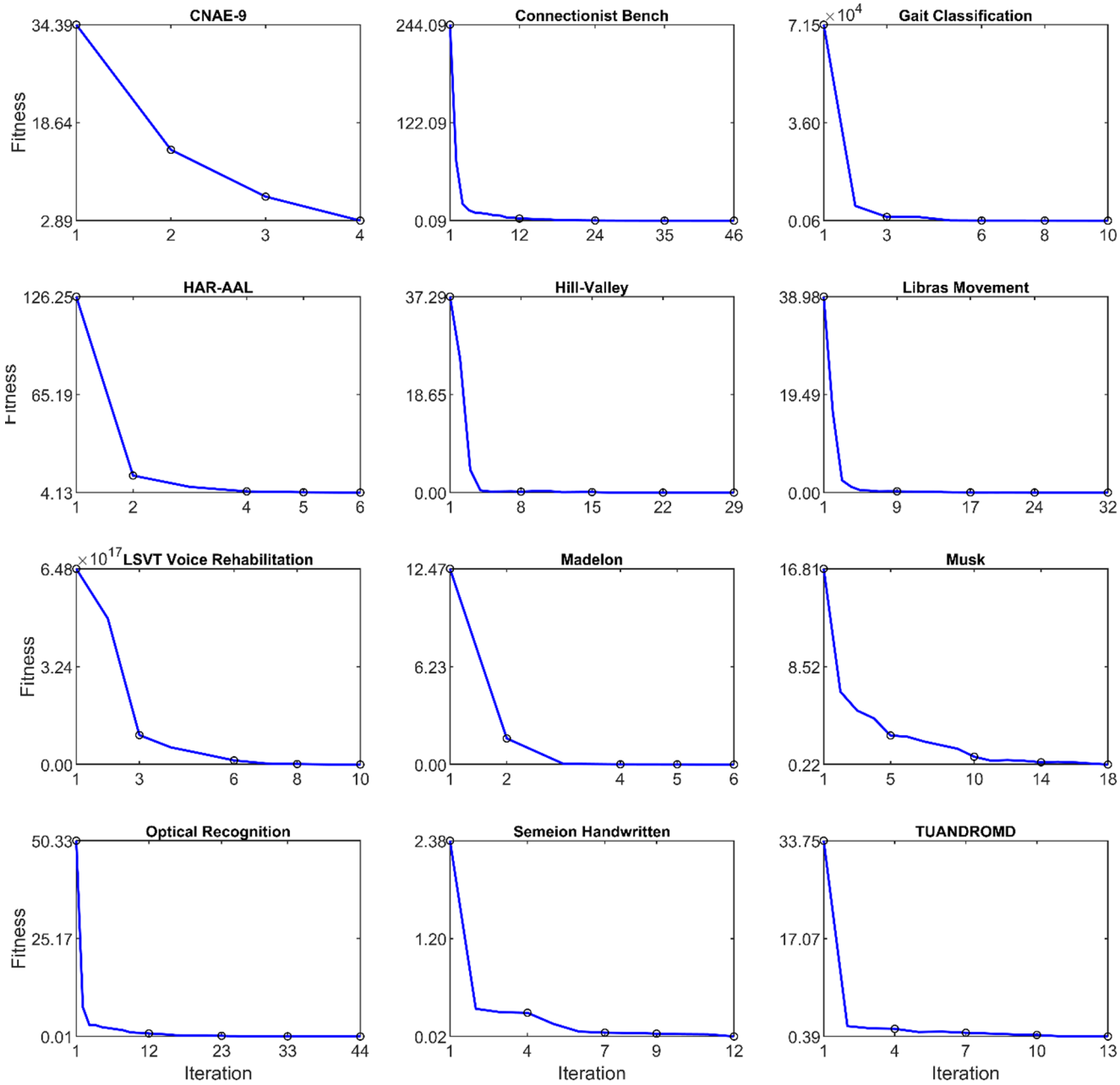

Fig. 12 The convergence behavior of the proposed algorithm across 12 benchmark data sets

(e.g., > 300), higher α values (0.7–1.0) tend to stabilize performance in data sets like Optical Recognition, where neighborhood preservation dominates.

Figure 11 shows the ACC results of PLRU on the benchmark data sets in terms of the different numbers of features and the distinct values of α. According to the ACC results, the hyperparameter α implicitly affects the performance of the proposed algorithm on some data sets. This influence is more apparent in some data sets, such as Gait Classification. The ACC results consistently increase in the rest of the data sets, excluding Gait Classification and LSVT Voice Rehabilitation data sets, as the number of features rises. Further, there does not exist a sharp difference between α values to obtain the highest ACC values. As a result, it is important to note that the appropriate alpha value may vary between data sets, so searching for the proper alpha value is necessary to achieve the highest ACC result. Additionally, the number of features producing the highest ACC result can differ between data sets. However, the proposed algorithm's restricted search range simplifies this optimization process.

Table 6 The running times (second) of the unsupervised feature selection algorithms on the data sets

Data sets	Algorithms						
	LS	RNE	DUFS	USFS	SLNMF	HSL	PLRU
CNAE-9	**0.3045**	275.5556	27.4009	10.9619	9.6383	11.0317	243.6054
Connectionist Bench	**0.0161**	0.5947	0.1074	0.1421	0.1179	0.1310	0.5469
Gait Classification	**0.0093**	15.1445	1.5446	0.3093	0.2083	0.3410	12.2077
HAR-AAL	**2.0484**	112.4321	37.7344	24.4736	128.7683	266.3334	82.6951
Hill-Valley	**0.0903**	1.1529	15.0824	0.4728	1.2759	4.7722	0.8509
Libras Movement	**0.0142**	1.0290	0.3346	0.5497	0.3563	0.2521	0.7977
LSVT Voice Rehabilitation	**0.0114**	13.0259	4.2905	0.4938	0.1521	0.3475	10.7816
Madelon	**0.2446**	72.3685	17.0521	3.4922	11.9957	18.4970	48.5709
Musk	**0.0129**	2.8052	1.2761	0.2286	0.3838	0.7664	2.2965
Optical Recognition	**0.3851**	2.2244	0.6326	3.2746	9.3653	73.8873	0.5511
Semeion Handwritten Digit	**0.1167**	9.7880	4.1991	4.5409	4.6806	9.9345	6.3675
TUANDROMD	**0.6907**	16.2160	7.4476	4.3664	34.4407	119.7836	5.7124
Avg	**0.3287**	43.5281	9.7585	4.4422	16.7819	42.1731	34.5820

The highest values are highlighted in bold. The second highest values are highlighted in underlined

The optimal α value varies among data sets and this variety implies differences in structural complexity, feature redundancy, and noise levels. In data sets such as CNAE-9, Madelon, and Gait Classification, low-to-moderate α values (e.g., 0.1–0.3) tend to yield higher classification accuracy when a small number of features is selected (e.g., ≤ 100). In data sets such as Gait Classification, Libras Movement, and HAR-AAL, moderate α values (e.g., 0.2–0.6) generally perform best. For data sets (e.g., the number of features > 200), higher α values (e.g., 0.7–1.0) are particularly more effective in the data sets TUANDROMD, Optical Recognition, and Semeion Handwritten, where maintaining local relationships significantly contributes to classification performance. On the other hand, in data sets such as Hill-Valley and Madelon, classification accuracy relatively remains low and inconsistent regardless of α. This situation specifies that these data sets are more sensitive to the quality of selected features than preserving the neighborhood. These observations show that the effectiveness of α is closely tied to the data set characteristics and the dimensionality of the selected feature subset. Therefore, adjusting α is important for maximizing classification performance.

As for practical guidance on α selection, $\alpha \in [0.1, 0.3]$ is recommended when the number of selected features is low (e.g., ≤ 100) and the data set is sparse or has high inter-feature redundancy (e.g., Gait Classification, CNAE-9). $\alpha \in [0.30.6]$ is generally effective for balanced data sets where preserving manifold structure and sparsity are critical (e.g., HAR-AAL, Libras Movement). $\alpha \in [0.7, 1]$ is preferable when more features are retained, and local geometry is critical (e.g., TUANDROMD, Optical Recognition).

5.5 The convergence analysis

The convergence behavior of the proposed algorithm was analyzed across 12 benchmark data sets by tracking the fitness value over successive iterations of optimization. As shown in Fig. 12, most of the data sets initially exhibit a rapid drop in the objective function. A gradual convergence towards stability is then followed in these data sets. This situation shows that the algorithm can escape local minima and progressively refine the solution.

In the data sets CNAE-9, HAR-AAL, Libras Movement, and Optical Recognition, the convergence profile is achieved within a small number of iterations. In contrast, more complex data sets such as Connectionist Bench and Hill-Valley significantly require more iterations (up to 40–50) to reach a plateau. Despite some data sets exhibiting large initial fitness values (e.g., Gait Classification and LSVT Voice Rehabilitation), the algorithm mainly demonstrates stable and monotonic convergence patterns. Notably, even in cases where the final objective values remained relatively high, such as in gait classification, the algorithm managed to reduce the error by several orders of magnitude from its starting point, which supports its numerical robustness.

To sum up, NMI and ACC scores support all these observations. Although the NMI values significantly vary in data sets (ranging from 0.0003 to 0.8147), the ACC scores are high (mostly above 0.85) in general. This situation indicates that the algorithm maintains strong classification consistency even in complex clustering scenarios. Overall, the convergence behaviors demonstrate that the optimization procedure is capable of adapting to various data complexities. This circumstance makes the proposed method trustworthy across a broad range of unsupervised learning tasks.

5.6 The computational complexity analysis

Table 6 compares the running time results of the unsupervised feature selection algorithms. The time complexity of the proposed PLRU algorithm is cubic to the number of dimensions or linear to the number of samples. However, the number of inner and outer iterations is important. Regarding the running time, the empirical results specify that the proposed PLRU algorithm processes data at an average speed. According to the results, the proposed PLRU algorithm runs faster on average than the RNE and HSL algorithms. The LS algorithm works more rapidly on all the data sets. The proposed PLRU algorithm is slower on the high-dimensional data sets. However, for the large number of samples, the PLRU algorithm is faster. Nonetheless, the most decisive factor is the number of iterations required for convergence. Consequently, we would like to express that the proposed PLRU algorithm spends more time optimizing the objective function than calculating it.

6 Conclusion

In this paper, we proposed a novel unsupervised feature selection algorithm named PLRU (Preservation of Locality Relation for Unsupervised feature selection). The main objective of PLRU is to eliminate linearly dependent, irrelevant, or redundant features while preserving the data's local and global manifold structure in the feature space. Unlike many methods that rely on spectral decomposition or explicit pair-wise distance computations, PLRU adopts a geometry-aware diagonal matrix model and avoids constructing neighborhood graphs. The core idea of PLRU is to learn a diagonal matrix that scales each feature independently, such that the distance structure of the transformed data remains as close as possible to that of the original data. To this end, we introduced a relaxed convex formulation to minimize the Frobenius norm difference between the Gram matrices before and after transformation. This formulation enables the principled and efficient learning of continuous feature weights without requiring hard binary selection.

Comprehensive experiments on 13 benchmark data sets demonstrate that PLRU consistently yields competitive or superior results across clustering and classification tasks. PLRU excels in high-dimensional data sets and feature selection tasks. Furthermore, hyperparameter sensitivity analysis shows that PLRU is robust across a range of α values. In addition to classification and clustering performance, PLRU possesses several desirable properties: (i) Unlike many spectral or graph-based methods, the number of clusters does not need to be specified in advance; (ii) It involves only a single hyperparameter that is bounded within an adjustable range; (iii) It supports visual interpretability by embedding the importance of features into the diagonal matrix; (iv) Thanks to its flexible mathematical formulation, the proposed method can be easily adapted to supervised or semi-supervised learning tasks by incorporating label information when it is available.

Although the proposed method has strong aspects, it presents several shortcomings. The most computationally demanding part is solving the optimization problem using an interior-point method, particularly during Newton iterations, which scale cubically with the number of features. Although we provided a complete complexity analysis and achieved good runtime performance in practice, scalability to huge feature spaces remains an open challenge. In future work, we plan to replace the interior-point algorithm with cutting-edge alternatives. We aim to learn α in a data-adaptive or parameter-free manner automatically.

Appendix. The basis for the proposed method

Definition 1 Let $\boldsymbol{P} \stackrel{\text{def}}{=} 2\left(\boldsymbol{J}_{\boldsymbol{m}\times\boldsymbol{m}} - \boldsymbol{L}\right)$, which is pair-wise distance matrix of $\boldsymbol{X}$ such that $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{m}\boldsymbol{y}\boldsymbol{X}^{\boldsymbol{T}}$ that is normalized provided that its diagonal elements are one, i.e., $\|\boldsymbol{x_i}\|_2 = 1,\ \boldsymbol{i} = 1, \cdots, \boldsymbol{m}$, which denotes the number of samples.

We define the Euclidean distance between two vectors x_i and x_j in a metric space in (A1).

$$\|\boldsymbol{x_i} - \boldsymbol{x}_j\|_2^2 = \|\boldsymbol{x}_i\|_2^2 + \|\boldsymbol{x}_j\|_2^2 - 2\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \tag{A1}$$

In this case, for the pair-wise distance matrix of $\boldsymbol{X}$, we obtain the following formula:

$$\boldsymbol{P} = diag\left(\boldsymbol{L}\right)\boldsymbol{J}_{m\times 1}^T + \boldsymbol{J}_{m\times 1} diag\left(\boldsymbol{L}\right)^T - 2\boldsymbol{L} \tag{A2}$$

To simplify the formula in (A2), we reach (A3) by normalizing $\boldsymbol{L}$ so that the diagonal elements of $\boldsymbol{L}$ are one, i.e., $diag\left(\boldsymbol{L}\right)$ is transformed into $\boldsymbol{J}_{m\times 1}$.

$$\boldsymbol{P} \stackrel{\text{def}}{=} 2\left(\boldsymbol{J}_{m\times m} - \boldsymbol{L}\right) \tag{A3}$$

Theorem 1 *Let $\boldsymbol{Q} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ be diagonalizable if and only if the eigenvectors of $\boldsymbol{Q}$ are linearly independent since $\boldsymbol{Q}$ has distinct eigenvalues. Then, $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}$, $\boldsymbol{D}$ and $\boldsymbol{V}$ are the diagonal matrix of $\boldsymbol{Q}$ and the matrix of eigenvectors of $\boldsymbol{Q}$, respectively.*

Let us assume that $\boldsymbol{Q}$ satisfies the eigenvalue equation $\boldsymbol{Q}\boldsymbol{v} = \lambda\boldsymbol{v}$ and its n th-order polynomial characteristic

equation $det(\boldsymbol{Q} - \lambda\mathbb{I}) = 0$. *It has* n *distinct solutions. Accordingly, the eigenvalue decomposition based on the matrices of eigenvalues and eigenvectors can be derived as* $\boldsymbol{QV} = \boldsymbol{VD}$. *When multiplying both sides with* $\boldsymbol{V}^{-1}$, *we obtain* $\boldsymbol{Q} = \boldsymbol{VDV}^{-1}$.

Theorem 2 *Let* $\boldsymbol{Q}, \boldsymbol{V}, \boldsymbol{D} \in \mathbb{R}^{n \times n}$. *Assuming* $\boldsymbol{V}$ *is a non-singular matrix, then* $\boldsymbol{Q} = \boldsymbol{VDV}^{-1}$ *and* $\Lambda(\boldsymbol{Q}) = \Lambda(\boldsymbol{D})$.

To prove Theorem 2, let $\lambda \in \Lambda(\boldsymbol{D})$ and $\boldsymbol{u}$ be the associated eigenvectors. In this case, let us assume that $\boldsymbol{Dv} = \lambda\boldsymbol{v}$. Accordingly,

$$\begin{aligned} &\boldsymbol{DV}^{-1}\mathbf{Vu} = \lambda\boldsymbol{u} \\ &\boldsymbol{VDV}^{-1}\mathbf{Vu} = \lambda\mathbf{V}\boldsymbol{u} \\ &\text{Let } \boldsymbol{VDV}^{-1} = \mathbf{Q} \\ &\boldsymbol{QV}\mathbf{u} = \lambda\mathbf{V}\boldsymbol{u} \end{aligned} \tag{A4}$$

In this case, the matrix $\boldsymbol{Q}$ also has the same eigenvalues, namely, $\Lambda(\boldsymbol{Q}) = \Lambda(\boldsymbol{D})$.

Definition 2. Based on Theorems 1 and 2, if there exists a non-singular matrix $\boldsymbol{V}$ such that $\boldsymbol{Q} = \boldsymbol{VDV}^{-1}$, then the matrices $\boldsymbol{Q}$ and $\boldsymbol{D}$ are said to be similar matrices.

Corollary 1 *Let* $\boldsymbol{Q}, \boldsymbol{V}, \boldsymbol{D} \in \mathbb{R}^{n \times n}$, *based on Definition 2, then* $\boldsymbol{Q} = \boldsymbol{VDV}^{-1} \Leftrightarrow \boldsymbol{D} = \boldsymbol{V}^{-1}\boldsymbol{QV}$

$$\begin{aligned} &\boldsymbol{Q} = \boldsymbol{VDV}^{-1} \\ &\boldsymbol{V}^{-1}\boldsymbol{QV} = \boldsymbol{V}^{-1}\boldsymbol{VDV}^{-1}\boldsymbol{V} \\ &\boldsymbol{D} = \boldsymbol{V}^{-1}\boldsymbol{QV} \end{aligned} \tag{A5}$$

Author contributions Fatih Aydın: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

Data availability We share all the public data sets and code on the website https://github.com/fatihaydin1/PLRU.

Declarations

Conflict interest The author declare that they have no conflict of interest.

References

1. Mullick SS, Datta S, Dhekane SG, Das S (2020) Appropriateness of performance indices for imbalanced data classification: an analysis. Pattern Recognit 102:107197. https://doi.org/10.1016/j.patcog.2020.107197
2. Sadeghian Z, Akbari E, Nematzadeh H, Motameni H (2023) A review of feature selection methods based on meta-heuristic algorithms. J Exp Theor Artif Intell. https://doi.org/10.1080/0952813X.2023.2183267
3. Li A-D, Xue B, Zhang M (2021) Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. Appl Soft Comput 106:107302. https://doi.org/10.1016/j.asoc.2021.107302
4. Li J, Othman MS, Chen H, Yusuf LM (2024) Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. J Big Data 11:36. https://doi.org/10.1186/s40537-024-00892-y
5. Teng L, Feng Z, Fang X et al (2019) Unsupervised feature selection with adaptive residual preserving. Neurocomputing 367:259–272. https://doi.org/10.1016/j.neucom.2019.05.097
6. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2020) A review of unsupervised feature selection methods. Artif Intell Rev 53:907–948. https://doi.org/10.1007/s10462-019-09682-y
7. Jeong DH, Jeong BK, Leslie N et al (2022) Designing a supervised feature selection technique for mixed attribute data analysis. Mach Learn with Appl 10:100431. https://doi.org/10.1016/j.mlwa.2022.100431
8. Bommert A, Sun X, Bischl B et al (2020) Benchmark for filter methods for feature selection in high-dimensional classification data. Comput Stat Data Anal 143:106839. https://doi.org/10.1016/j.csda.2019.106839
9. Gong H, Li Y, Zhang J et al (2024) A new filter feature selection algorithm for classification task by ensembling pearson correlation coefficient and mutual information. Eng Appl Artif Intell 131:107865. https://doi.org/10.1016/j.engappai.2024.107865
10. Degeest A, Frénay B, Verleysen M (2021) Reading grid for feature selection relevance criteria in regression. Pattern Recognit Lett 148:92–99. https://doi.org/10.1016/j.patrec.2021.04.031
11. Abiodun EO, Alabdulatif A, Abiodun OI et al (2021) A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. Neural Comput Appl 33:15091–15118. https://doi.org/10.1007/s00521-021-06406-8
12. Yu Y, Wan M, Qian J et al (2024) Feature selection for multi-label learning based on variable-degree multi-granulation decision-theoretic rough sets. Int J Approx Reason 169:109181. https://doi.org/10.1016/j.ijar.2024.109181
13. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40:16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024
14. Sun L, Wang L, Ding W et al (2021) Feature selection using fuzzy neighborhood entropy-based uncertainty measures for fuzzy neighborhood multigranulation rough sets. IEEE Trans Fuzzy Syst 29:19–33. https://doi.org/10.1109/TFUZZ.2020.2989098
15. Rostami M, Berahmand K, Nasiri E, Forouzandeh S (2021) Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell 100:104210. https://doi.org/10.1016/j.engappai.2021.104210
16. Nouri-Moghaddam B, Ghazanfari M, Fathian M (2023) A novel bio-inspired hybrid multi-filter wrapper gene selection method with ensemble classifier for microarray data. Neural Comput Appl 35:11531–11561. https://doi.org/10.1007/s00521-021-06459-9
17. Nguyen BH, Xue B, Zhang M (2020) A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol Comput 54:100663. https://doi.org/10.1016/j.swevo.2020.100663
18. Pudjihartono N, Fadason T, Kempa-Liehr AW, O'Sullivan JM (2022) A review of feature selection methods for machine learning-based disease risk prediction. Front Bioinforma. https://doi.org/10.3389/fbinf.2022.927312

19. Zhu P, Hou X, Tang K et al (2023) Unsupervised feature selection through combining graph learning and $\ell 2, 0$-norm constraint. Inf Sci 622:68–82. https://doi.org/10.1016/j.ins.2022.11.156
20. Moslemi A, Shaygani A (2024) Unsupervised feature selection through combining graph learning and $\ell 2, 0$-norm constraint. Int J Mach Learn Cybern 15:5361–5380. https://doi.org/10.1007/s13042-024-02243-y
21. Saberi-Movahed F, Rostami M, Berahmand K et al (2022) Dual regularized unsupervised feature selection based on matrix factorization and minimum redundancy with application in gene selection. Knowl Based Syst 256:109884. https://doi.org/10.1016/j.knosys.2022.109884
22. Wang C, Wang J, Gu Z et al (2024) Unsupervised feature selection by learning exponential weights. Pattern Recogn 148:110183. https://doi.org/10.1016/j.patcog.2023.110183
23. Zhou P, Du L, Li X et al (2020) Unsupervised feature selection with adaptive multiple graph learning. Pattern Recognit 105:107375. https://doi.org/10.1016/j.patcog.2020.107375
24. Kong J, Shang R, Zhang W et al (2024) Robust feature selection via central point link information and sparse latent representation. Pattern Recognit 154:110617. https://doi.org/10.1016/j.patcog.2024.110617
25. Chen Z, Bian J, Qiao B, Xie X (2024) Differentiable gated autoencoders for unsupervised feature selection. Neurocomputing 601:128202. https://doi.org/10.1016/j.neucom.2024.128202
26. Akinola OO, Ezugwu AE, Agushaka JO et al (2022) Multiclass feature selection with metaheuristic optimization algorithms: a review. Neural Comput Appl 34:19751–19790. https://doi.org/10.1007/s00521-022-07705-4
27. Gad AG, Sallam KM, Chakrabortty RK et al (2022) An improved binary sparrow search algorithm for feature selection in data classification. Neural Comput Appl 34:15705–15752. https://doi.org/10.1007/s00521-022-07203-7
28. Zhang H, Qin X, Gao X et al (2024) Improved salp swarm algorithm based on Newton interpolation and cosine opposition-based learning for feature selection. Math Comput Simul 219:544–558. https://doi.org/10.1016/j.matcom.2023.12.037
29. Khan GA, Hu J, Li T et al (2023) Multi-view clustering for multiple manifold learning via concept factorization. Dig Signal Process 140:104118. https://doi.org/10.1016/j.dsp.2023.104118
30. Zhou S, Song P, Song Z, Ji L (2023) Soft-label guided non-negative matrix factorization for unsupervised feature selection. Expert Syst Appl 216:119468. https://doi.org/10.1016/j.eswa.2022.119468
31. Cordeiro de Amorim R (2019) Unsupervised feature selection for large data sets. Pattern Recognit Lett 128:183–189. https://doi.org/10.1016/j.patrec.2019.08.017
32. Zhao Z, Wang L, Liu H, Ye J (2013) On similarity preserving feature selection. IEEE Trans Knowl Data Eng 25:619–632. https://doi.org/10.1109/TKDE.2011.222
33. Zhao J, Wu D, Zhou Y et al (2024) Rough set theory-based group incremental approach to feature selection. Inf Sci 675:120733. https://doi.org/10.1016/j.ins.2024.120733
34. Levin D, Singer G (2024) GB-AFS: graph-based automatic feature selection for multi-class classification via mean simplified silhouette. J Big Data 11:79. https://doi.org/10.1186/s40537-024-00934-5
35. Sandberg J, Voigtmann T, Devijver E, Jakse N (2024) Feature selection for high-dimensional neural network potentials with the adaptive group lasso. Mach Learn Sci Technol 5:025043. https://doi.org/10.1088/2632-2153/ad450e
36. Ma Z, Huang Y, Li H, Wang J (2023) Unsupervised feature selection with latent relationship penalty term. Axioms 13:6. https://doi.org/10.3390/axioms13010006
37. Mi Y, Chen H, Luo C et al (2024) Unsupervised feature selection with high-order similarity learning. Knowledge Based Syst 285:111317. https://doi.org/10.1016/j.knosys.2023.111317
38. Han K, Wang Y, Zhang C, et al (2018) Autoencoder inspired unsupervised feature selection. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 2941–2945
39. Ling Y, Nie F, Yu W et al (2024) Robust autoencoder feature selector for unsupervised feature selection. Inf Sci 660:120121. https://doi.org/10.1016/j.ins.2024.120121
40. Cao Z, Xie X, Li Y (2024) Multi-view unsupervised feature selection with consensus partition and diverse graph. Inf Sci (NY) 661:120178. https://doi.org/10.1016/j.ins.2024.120178
41. Li D, Chen H, Mi Y et al (2024) Unsupervised feature selection via dual space-based low redundancy scores and extended OLSDA. Inf Sci 662:120227. https://doi.org/10.1016/j.ins.2024.120227
42. Mozafari M, Seyedi SA, Pir Mohammadiani R, Akhlaghian Tab F (2024) Unsupervised feature selection using orthogonal encoder-decoder factorization. Inf Sci (NY) 663:120277. https://doi.org/10.1016/j.ins.2024.120277
43. Zuo X, Zhang W, Wang X et al (2025) Unsupervised feature selection via maximum relevance and minimum global redundancy. Pattern Recognit 164:111483. https://doi.org/10.1016/j.patcog.2025.111483
44. Xiao Z, Chen H, Mi Y et al (2025) Joint subspace learning and subspace clustering based unsupervised feature selection. Neurocomputing 635:129885. https://doi.org/10.1016/j.neucom.2025.129885
45. Yu H-W, Wu J-Y, Wu J-S, Min W (2025) Confident local similarity graphs for unsupervised feature selection on incomplete multi-view data. Knowl Based Syst 316:113369. https://doi.org/10.1016/j.knosys.2025.113369
46. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. In: NIPS'05: Proceedings of the 18th International Conference on Neural Information Processing Systems. Pp. 507–514
47. Liu Y, Ye D, Li W et al (2020) Robust neighborhood embedding for unsupervised feature selection. Knowl Based Syst 193:105462. https://doi.org/10.1016/j.knosys.2019.105462
48. Lim H, Kim D-W (2021) Pairwise dependence-based unsupervised feature selection. Pattern Recognit 111:107663. https://doi.org/10.1016/j.patcog.2020.107663
49. Wang F, Zhu L, Li J et al (2021) Unsupervised soft-label feature selection. Knowl Based Syst 219:106847. https://doi.org/10.1016/j.knosys.2021.106847