Bi-Sparse Unsupervised Feature Selection

Xianchao Xiu, *Member, IEEE*, Chenyi Huang, Pan Shang, and Wanquan Liu, *Senior Member, IEEE*

***Abstract*—To deal with high-dimensional unlabeled datasets in many areas, principal component analysis (PCA) has become a rising technique for unsupervised feature selection (UFS). However, most existing PCA-based methods only consider the structure of datasets by embedding a single sparse regularization or constraint on the transformation matrix. In this paper, we introduce a novel bi-sparse method called BSUFS to improve the performance of UFS. The core idea of BSUFS is to incorporate $\ell_{2,p}$-norm and ℓ_q-norm into the classical PCA, which enables our method to select relevant features and filter out irrelevant noises, thereby obtaining discriminative features. Here, the parameters p and q are within the range of $[0, 1)$. Therefore, BSUFS not only constructs a unified framework for bi-sparse optimization, but also includes some existing works as special cases. To solve the resulting non-convex model, we propose an efficient proximal alternating minimization (PAM) algorithm using Stiefel manifold optimization and sparse optimization techniques. In addition, the computational complexity analysis is presented. Extensive numerical experiments on synthetic and real-world datasets demonstrate the effectiveness of our proposed BSUFS. The results reveal the advantages of bi-sparse optimization in feature selection and show its potential for other fields in image processing. Our code is available at https://github.com/xianchaoxiu/BSUFS.**

***Index Terms*—Unsupervised feature selection, bi-sparse, principal component analysis, proximal alternating minimization, manifold optimization.**

I. Introduction

HIGH-DIMENSIONAL features make data processing challenging due to the presence of redundant information and additional noises [1]. To address this challenge, feature selection techniques are proposed and widely studied. One can read the review paper [2] for more comprehensive backgrounds and advances. Among feature selection techniques, there is a special type called unsupervised feature selection (UFS), which is designed to select features for unlabeled datasets, i.e., label information is missing. Note that unlabeled datasets are cheap and easy to obtain in practice. Therefore, a large number of researchers have devoted themselves to developing UFS methods, which have shown outstanding performances in image processing [3], [4], [5], gene analysis [6], [7], [8], machine learning [9], [10], [11], and deep learning [12], [13], [14].

According to [2] and [3], existing UFS methods can be generally divided into three categories based on different search strategies, including filtering methods, wrapper methods, and embedded methods. It is worth pointing out that the embedded methods combine the advantages of filtering methods and wrapper methods by directly integrating feature selection into the learning procedure [15]. Under some graph techniques, there are many representative UFS methods that are proposed to optimize the similarity matrix, such as Laplacian score (LapScore) [16], multi-cluster feature selection (MCFS) [17], unsupervised discriminative feature selection (UDFS) [18], structured optimal graph feature selection (SOGFS) [19], robust neighborhood embedding (RNE) [20], and non-convex regularized graph embedding and self-representation (NLGMS) [21], to name a few. As one of the most straightforward and accessible embedded methods, principal component analysis (PCA) stands out from UFS methods [22]. By introducing a linear transformation matrix (also projection matrix), PCA tends to characterize this matrix and is different from graph-based embedded methods. Despite its excellent performance, one of the main drawbacks is the lack of feature interpretability [23]. To overcome this limitation, Li et al. [24] combined PCA with the $\ell_{2,p}$-norm regularization term on the transformation matrix. This novel PCA variant, referred to as SPCAFS, aims to achieve feature sparsity and still retain the principal information simultaneously. Here, $p \in (0, 1)$ and thus SPCAFS is a non-convex version of sparse PCA with $\ell_{2,1}$-norm [25]. In addition, Nie et al. [26] considered feature-sparsity constrained PCA (FSPCA), which is accomplished by enforcing the $\ell_{2,0}$-norm constraint. Note that $\ell_{2,0}$-norm is a direct choice to characterize the feature sparsity [27]. FSPCA enables selecting a subset of features that are most representative and intrinsic. Recently, Zheng et al. [28] proposed a sparse PCA method based on positive semidefinite projection (SPCA-PSD), which achieves excellent efficiency in clustering. Gao et al. [29] extended the $\ell_{2,p}$-norm to fuzzy elastic net for better characterizing the structure of the transformation matrix, which is called FEN-PCAFS. The above $\ell_{2,1}$-norm, $\ell_{2,p}$-norm, and $\ell_{2,0}$-norm can be used to capture row sparsity, which is closely related to the selected features. These sparse regularization terms attribute interpretability and robustness to PCA models, which is beneficial for understanding the data structure [30].

However, most works introduced a single row-wise sparse regularization term (e.g., $\ell_{2,1}$-norm) to the transformation matrix in PCA, the element-wise sparsity of this matrix is

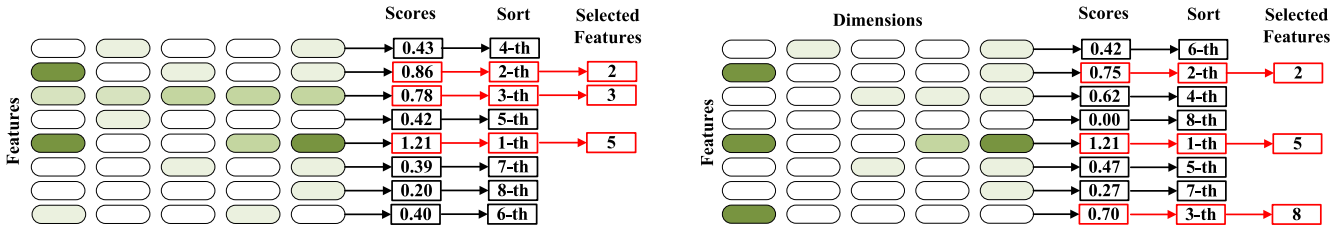
Received 23 October 2024; revised 1 July 2025; accepted 9 October 2025. Date of publication 17 October 2025; date of current version 13 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 12371306, Grant 12401430, and Grant 12271309. The associate editor coordinating the review of this article and approving it for publication was Dr. Charles Kervrann. *(Corresponding author: Pan Shang.)*

Xianchao Xiu and Chenyi Huang are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: xcxiu@shu.edu.cn; huangchenyi@shu.edu.cn).

Pan Shang is with the School of Mathematics and Statistics, Beijing Jiaotong University, Beijing 100044, China (e-mail: pshang@amss.ac.cn).

Wanquan Liu is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: liuwq63@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TIP.2025.3620667



(a) SPCAFS (b) Our proposed BSUFS

Fig. 1. Visualization of feature selection results by (*a*) SPCAFS [24] and (*b*) our proposed BSUFS. Feature selection is performed by first computing the tranformation matrix W, where the dimensions of W correspond to the original data features. Then, by calculating the row-wise norms of W, these values are sorted and the top-ranked values are thus selected. Finally, the features associated with these top-ranked values are chosen as the selected features.

not considered. Therefore, a bi-sparse optimization model with $\ell_{2,1}$-norm and ℓ_1-norm regularization terms was proposed in [31], where $\ell_{2,1}$-norm is used to select features and ℓ_1-norm is enforced to filter out noise. In fact, the combination of $\ell_{2,1}$-norm and ℓ_1-norm has been considered in multi-task feature selection and has shown excellent performances [32], [33]. Although the idea of bi-sparse or double sparse has been applied in image denoising [34], compressed sensing [35], gene expression [36], and radar imaging [37], the involved double sparsity terms are constrained to different variables. Note that $\ell_{2,p}$-norm with $p \in [0, 1)$ is a general form of $\ell_{2,p}$-norm in [24] and $\ell_{2,0}$-norm [26]. *A natural question is whether we can propose a bi-sparse non-convex PCA-based feature selection framework by inheriting the advantages of $\ell_{2,p}$-norm with $p \in [0, 1)$ and ℓ_q-norm with $q \in [0, 1)$, and verify the effectiveness of ℓ_q-norm for feature selection.*

Motivated by the above observations, we propose a unified bi-sparse UFS method called BSUFS, which is a PCA model with $\ell_{2,p}$-norm and ℓ_q-norm regularization terms, where p and q are within the range of $[0, 1)$. As can be seen from Fig. 1, compared with the benchmark SPCAFS which only contains a single $\ell_{2,p}$-norm regularization term with $p \in (0, 1)$ on the transformation matrix, adding an additional ℓ_q-norm regularization makes the feature selection results of BSUFS different. Actually, our proposed BSUFS takes SPCAFS and FSPCA as special cases and has higher flexibility than the convex relaxation form in [31]. Of course, the choice of p and q may be affected by the specific data structure, which will be discussed in Subsection IV-F, and it is confirmed that combining the values of 0 and $(0, 1)$ into $[0, 1)$ is of great significance. In addition, the idea of bi-sparse optimization is not limited to feature selection, but can be easily extended to other fields of image processing.

The main contributions of this paper can be summarized in the following three aspects:

- We develop a novel BSUFS method that can improve the performance of feature selection by regularizing $\ell_{2,p}$-norm and ℓ_q-norm. The values of p and q are set in the range of $[0, 1)$, which has not been considered before.
- We design an efficient proximal alternating minimization (PAM) algorithm and all subproblems admit closed-form proximal operators or can be solved by Stiefel manifold optimization. In addition, we provide its computational complexity for every iteration.
- We conduct sufficient numerical experiments to evaluate the performance of BSUFS. In particular, we analyze the parameter effects of $p, q \in [0, 1)$ and show that, in feature selection, p plays a dominant role, while q serves a complementary role, both of which are indispensable.

The structure of this paper is outlined as follows. Section II introduces the notations and related works. Section III presents the proposed model, optimization algorithm, and computational complexity. Section IV provides the numerical results, ablation experiments, statistical tests, effects of p and q, and discussions. Section V concludes this paper.

II. Preliminaries

This section first introduces the notations used throughout this paper and then gives some related preliminaries.

A. Notations

In this paper, matrices are represented by capital letters, vectors by boldface letters, and scalars by lowercase letters. Let $\mathbb{R}^d$ and $\mathbb{R}^{m\times n}$ be the sets of all d-dimensional vectors and $m \times n$-dimensional matrices, respectively. For any vector $\mathbf{x} \in \mathbb{R}^d$, its i-th element is denoted as x_i. The ℓ_2-norm of $\mathbf{x}$ is $\|\mathbf{x}\| = \left(\sum_{i=1}^d x_i^2\right)^{1/2}$. For any matrix $X \in \mathbb{R}^{m\times n}$, x_{ij} represents its ij-th element, $\mathbf{x}^i$ and $\mathbf{x}_i$ represent its i-th row and i-th column, respectively. The Frobenius norm of X is $\|X\|_{\mathrm{F}} = \left(\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2\right)^{1/2}$. For $p, q \in (0, 1)$, the $\ell_{2,p}$-norm of X is defined as $\|X\|_{2,p} = \left(\sum_{i=1}^m \|\mathbf{x}^i\|^p\right)^{1/p}$ and the ℓ_q-norm is $\|X\|_q = \left(\sum_{i=1}^m \sum_{j=1}^n |x_{ij}|^q\right)^{1/q}$. When $p, q = 1$, they are exactly $\ell_{2,1}$-norm and ℓ_1-norm. In addition, $\|X\|_{2,0}$ and $\|X\|_0$ count the numbers of non-zero rows and non-zero elements of X, respectively. A further notation will be introduced wherever it appears.

B. PCA Basis

Consider a data matrix $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in \mathbb{R}^{d\times n}$ with $\mathbf{x}_i \in \mathbb{R}^d$. Denote W as a transformation matrix and $W = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m) \in \mathbb{R}^{d\times m}$, where $\mathbf{w}_i \in \mathbb{R}^d$, $\|\mathbf{w}_i\| = 1$, and $\mathbf{w}_i^\top \mathbf{w}_j = 0$ for $i \neq j$. Here, it is supposed that $m < n$.

Mathematically, PCA can be represented in a trace formulation with an orthogonal constraint as

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}\left(W^\top X X^\top W\right) \\ \text{s.t.} \quad & W^\top W = I_m, \end{aligned} \tag{1}$$

where the data is assumed to be centralized. For the general case that the data is not centralized, PCA can be written as

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}\left(W^\top S W\right) \\ \text{s.t.} \quad & W^\top W = I_m, \end{aligned} \tag{2}$$

where $S = XHX^\top$ and $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ with $\mathbf{1} \in \mathbb{R}^n$ being a vector whose elements are all ones.

Without loss of generality, denote $\mathbf{w}^{i\top}$ as the transpose of $\mathbf{w}^i$, where $\mathbf{w}^i$ is the i-th row of W. In feature selection, it can be found that $\mathbf{w}^{i\top}$ represents the transformation vector associated with the i-th feature in X. In details, the matrix W is denoted as

$$W = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m) = \begin{pmatrix} \mathbf{w}^1 \\ \mathbf{w}^2 \\ \vdots \\ \mathbf{w}^d \end{pmatrix} \in \mathbb{R}^{d\times m}. \tag{3}$$

By transforming the vector

$$\mathbf{x}_i = \begin{pmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{d,i} \end{pmatrix} \in \mathbb{R}^d \tag{4}$$

via the transformation matrix W, we get the transformation vector $\mathbf{z}_i$ as

$$\mathbf{z}_i = W^\top \mathbf{x}_i = (\mathbf{w}^{1\top},\ \mathbf{w}^{2\top},\ \ldots,\ \mathbf{w}^{d\top}) \begin{pmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{d,i} \end{pmatrix}. \tag{5}$$

Accordingly, $\|\mathbf{w}^i\|$ can be used to measure the importance of the i-th feature [38].

C. Sparse PCA

In recent years, non-convex optimization has been rapidly developed, which can provide more possibilities than convex optimization [39], [40], [41]. With this advantage, Li et al. [24] proposed a sparse PCA model called SPCAFS, which is given as the form of

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) + \lambda\|W\|_{2,p}^p \\ \text{s.t.} \quad & \mathrm{W}^\top\mathrm{W} = \mathrm{I_m}, \end{aligned} \tag{6}$$

where $\lambda \geq 0$ is the regularization parameter and $p \in (0, 1)$. By introducing $\ell_{2,p}$-norm into the objective function of (2), SPCAFS can obtain a row-wise sparse transformation matrix W, thereby improving the performance of feature selection. It is also numerically demonstrated that its performance is better than the convex relaxation when $p = 1/2$.

FSPCA is another popular UFS method, whose mathematical model is given by

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) \\ \text{s.t.} \quad & \|\mathrm{W}\|_{2,0} \leq \mathrm{s},\ \mathrm{W}^\top\mathrm{W} = \mathrm{I_m}, \end{aligned} \tag{7}$$

where $s > 0$ is the sparsity level. Recall Fig. 1 and (5), it is seen that s corresponds to the number of selected features.

III. The Proposed Method

This section first presents our proposed bi-sparse and non-convex UFS model, and then shows an optimization algorithm.

A. Model Construction

As is demonstrated in [31] that, different from these single structured sparse PCA methods, bi-sparse regularized models can better select representative features due to the introduction of $\ell_{2,1}$-norm and ℓ_1-norm on the transformation matrix. Therefore, to inherit the advantages of non-convex optimization and bi-sparse optimization, we propose a sparse PCA variant called BSUFS, which is reformulated as

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) + \lambda_1\|W\|_{2,p}^p + \lambda_2\|W\|_q^q \\ \text{s.t.} \quad & W^\top W = I_m, \end{aligned} \tag{8}$$

where $p, q \in [0, 1)$ and $\lambda_1, \lambda_2 \geq 0$ are the parameters to control the bi-sparse regularization terms.

Compared with the existing PCA-based UFS methods, the following conclusions can be made:

- When $\lambda_2 = 0$ in (8), our proposed BSUFS unifies SPCAFS [24] and the Lagrangian form of FSPCA [26].
- When p and q tend to 1 in (8), our proposed BSUFS equals to the model in [31].

In summary, BSUFS can provide flexible sparse solutions by choosing different p and q in the range of $[0, 1)$, thus facilitating feature selection.

B. Optimization Algorithm

Besides the Stiefel manifold constraint $W^\top W = I_m$, the introduced $\ell_{2,p}$-norm and ℓ_q-norm in (8) makes our proposed BSUFS more difficult. Below, we provide an efficient optimization algorithm based on the proximal alternating minimization (PAM) technique.

By utilizing auxiliary variables $W = V$, $W = U$, (8) can be rewritten as

$$\begin{aligned} \min_{W,U,V\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) + \lambda_1\|V\|_{2,p}^p + \lambda_2\|U\|_q^q \\ \text{s.t.} \quad & W^\top W = I_m,\ W = V,\ W = U. \end{aligned} \tag{9}$$

Algorithm 1 Proximal Alternating Minimization (PAM) Algorithm for BSUFS

Input: Data $X \in \mathbb{R}^{d\times n}$, parameters p, q, λ_1, λ_2, β_1, β_2, τ_1, τ_2, τ_3, calculate $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, $S = XHX^\top$

Output: V

Initialize: $k = 0$, W^k, U^k, V^k

While not converged **do**

1: Update W^{k+1} by (12)
2: Update U^{k+1} by (13)
3: Update V^{k+1} by (14)
4: Check convergence: If

$$\frac{|f^{k+1} - f^k|}{\max\{|f^k|, 1\}} < 10^{-4} \text{ or } k > 500$$

then stop. Otherwise, $k = k + 1$

End While

Then, introduce the indicator function

$$\Phi(W) = \begin{cases} 0, & W^\top W = I_m, \\ +\infty, & \text{otherwise}, \end{cases} \tag{10}$$

and rewrite (9) as its penalized form

$$\begin{aligned} \min_{W,U,V\in\mathbb{R}^{d\times m}} \; & -\mathrm{Tr}(W^\top S W) + \lambda_1\|V\|_{2,p}^p + \lambda_2\|U\|_q^q \\ & + \frac{\beta_1}{2}\|W-U\|_{\mathrm{F}}^2 + \frac{\beta_2}{2}\|W-V\|_{\mathrm{F}}^2 + \Phi(W), \end{aligned} \tag{11}$$

where $\beta_1, \beta_2 > 0$ are the penalty parameters. Denote the objective function of (11) as $f(W, U, V)$. Under the PAM framework, each variable can be updated alternately via the following scheme

$$W^{k+1} \in \arg\min_{W\in\mathbb{R}^{d\times m}} f(W, U^k, V^k) + \frac{\tau_1}{2}\|W - W^k\|_{\mathrm{F}}^2, \tag{12}$$

$$U^{k+1} \in \arg\min_{U\in\mathbb{R}^{d\times m}} f(W^{k+1}, U, V^k) + \frac{\tau_2}{2}\|U - U^k\|_{\mathrm{F}}^2, \tag{13}$$

$$V^{k+1} \in \arg\min_{V\in\mathbb{R}^{d\times m}} f(W^{k+1}, U^{k+1}, V) + \frac{\tau_3}{2}\|V - V^k\|_{\mathrm{F}}^2, \tag{14}$$

where $\tau_1, \tau_2, \tau_3 > 0$ and k is the iteration number. The overall scheme is presented in Algorithm 1, and the update rules for W, U, and V are analyzed as follows.

1) Update W by Fixing U and V: (12) can be expressed as

$$\begin{aligned} \min_{W^\top W = I_m} \; & g(W) = -\mathrm{Tr}(W^\top S W) + \frac{\beta_1}{2}\|W - U^k\|_{\mathrm{F}}^2 \\ & + \frac{\beta_2}{2}\|W - V^k\|_{\mathrm{F}}^2 + \frac{\tau_1}{2}\|W - W^k\|_{\mathrm{F}}^2. \end{aligned} \tag{15}$$

The Euclidean gradient of $g(W)$ is

$$\begin{aligned} \nabla g(W) = & -2SW + \beta_1(W - U^k) \\ & + \beta_2(W - V^k) + \tau_1(W - W^k), \end{aligned} \tag{16}$$

and the Euclidean Hessian is

$$\nabla^2 g(W) = -2I_m \otimes S + (\beta_1 + \beta_2 + \tau_1) I_{dm}, \tag{17}$$

where $\otimes$ represents the Kronecker product.

Denote

$$\mathrm{St}(d, m) = \{W \in \mathbb{R}^{d\times m} \mid W^\top W = I_m\}. \tag{18}$$

Then, (15) is a Stiefel manifold optimization problem and can be reformulated as

$$\min_{W\in\mathrm{St}(d,m)} g(W), \tag{19}$$

where the trust-region Riemannian manifold optimization algorithm [42] can be applied to solve this problem.

To implement this algorithm, there are two basic elements, i.e., the search direction and the trust-region ratio. The search direction relies on the Riemannian gradient and Riemannian Hessian of the objective function $g(W)$ in (19). Specifically, the Riemannian gradient can be obtained by projecting its Euclidean gradient onto the tangent space of the Stiefel manifold, i.e.,

$$\begin{aligned} \mathrm{grad} g(W) &= \mathcal{P}_W(\nabla g(W)) \\ &= \nabla g(W) - W\mathrm{sym}(W^\top \nabla g(W)), \end{aligned} \tag{20}$$

Algorithm 2 Trust-Region Riemannian Manifold Optimization Algorithm for Solving (12)

Input: Data $S \in \mathbb{R}^{d\times d}$, $U^k \in \mathbb{R}^{d\times m}$, $V^k \in \mathbb{R}^{d\times m}$, parameters $\beta_1, \beta_2, \tau_1, \varepsilon, \Delta' > 0$, $\rho' \in \left[0, \frac{1}{4}\right)$

Output: W^k

Initialize: $i = 0$, $W_i^k \in \mathrm{St}(d, m)$, $\Delta_i \in (0, \Delta')$**While** not converged **do**

Obtain M_i by solving (22) with $W = W_i^k$ and $\Delta = \Delta_i$
Compute ρ_i from (23) with $W = W_i^k$
if $\rho_i < \frac{1}{4}$ **then**
  $\Delta_{i+1} = \frac{1}{4}\Delta_i$
else if $\rho_i > \frac{3}{4}$ and $\|M_i\| = \Delta_i$ **then**
  $\Delta_{i+1} = \min(2\Delta_i, \Delta')$
else
  $\Delta_{i+1} = \Delta_i$
end if
if $\rho_i > \rho'$ **then**
  $W_{i+1}^k = R_W(M_i)$
else
  $W_{i+1}^k = W_i^k$
end if
Check convergence: If

$$\mathrm{grad} g(W_{i+1}^k) < 10^{-6} \text{ or } i > 100$$

then stop. Otherwise, $i = i + 1$
End While

where $\mathcal{P}_W(\nabla g(W))$ is the projection of the Euclidean gradient onto the tangent space of the Stiefel manifold, and $\mathrm{sym}(X) = (X + X^\top)/2$ extracts the symmetric part of a square matrix X. The Riemannian Hessian can be obtained by projecting its Euclidean Hessian onto the tangent space of the Stiefel manifold, i.e.,

$$\begin{aligned} \mathrm{Hess} g(W)[M] = & \mathcal{P}_W(\nabla^2 g(W)[M] \\ & - M\mathrm{sym}(W^\top \nabla g(W)) \\ & - W\mathrm{sym}(M^\top \nabla g(W)) \\ & - W\mathrm{sym}(W^\top \nabla^2 g(W)[M])), \end{aligned} \tag{21}$$

where $\nabla^2 g(W)[M]$ is the Euclidean Hessian-vector product. Then, the search direction of the trust region algorithm is derived by solving the following problem

$$\begin{aligned} \min_{M\in \mathrm{T_W St}(d,m)} \; & m_W(M) = g(W) + \left\langle \mathrm{grad} g(W), M \right\rangle_W \\ & + \frac{1}{2}\left\langle \mathrm{Hess} g(W)[M], M \right\rangle_W \\ \text{s.t.} \quad & \langle M, M \rangle_W \le \Delta^2, \end{aligned} \tag{22}$$

where Δ is the trust-region radius, and $\mathrm{T_W St}(d, m)$ is the tangent space of the manifold at W.

Finally, the trust region ratio is determined by

$$\rho = \frac{g(W) - g(R_W(M))}{m_W(0) - m_W(M)}, \tag{23}$$

where $R_W(M)$ is the retraction of M onto the Stiefel manifold.

2) Update U by Fixing W and V: After updating W, (13) can be solved by

$$\min_{U\in\mathbb{R}^{d\times m}} \lambda_2\|U\|_q^q + \frac{\beta_1}{2}\|W^{k+1}-U\|_{\mathrm{F}}^2 + \frac{\tau_2}{2}\|U-U^k\|_{\mathrm{F}}^2. \quad (24)$$

By expanding the norm terms in (24), this subproblem can be rewritten as the form of

$$\min_{U\in\mathbb{R}^{d\times m}} \lambda_2\|U\|_q^q + \frac{\beta_1+\tau_2}{2}\|U-Y\|_{\mathrm{F}}^2, \quad (25)$$

where

$$Y = \frac{\beta_1}{\beta_1+\tau_2}W^{k+1} + \frac{\tau_2}{\beta_1+\tau_2}U^k.$$

By considering each element of the matrix separately, the optimization problem can be reformulated as

$$\min_{u_{ij}\in\mathbb{R}} \lambda_2|u_{ij}|^q + \frac{\beta_1+\tau_2}{2}(u_{ij}-y_{ij})^2. \quad (26)$$

The solution of this problem is the proximal operator of the $|\cdot|^q$, whose result is reviewed in the following lemma.

Lemma 1: Consider the proximal operator

$$\begin{aligned}\mathrm{Prox}_{\lambda|\cdot|^q}(a) &= \operatorname*{argmin}_{x\in\mathbb{R}} \lambda|x|^q + \frac{1}{2}(x-a)^2 \\ &= \begin{cases}\{0\}, & |a| < \kappa(\lambda,q),\\ \{0, \mathrm{sgn}(a)c(\lambda,q)\}, & |a| = \kappa(\lambda,q),\\ \left\{\mathrm{sgn}(a)\varpi_q(|a|)\right\}, & |a| > \kappa(\lambda,q),\end{cases}\end{aligned} \quad (27)$$

where

$$\begin{aligned} &c(\lambda,p) = (2\lambda(1-q))^{\frac{1}{2-q}} > 0,\\ &\kappa(\lambda,q) = (2-q)\lambda^{\frac{1}{2-q}}(2(1-q))^{\frac{q+1}{q-2}},\\ &\varpi_q(a) \in \{x \mid x-a+\lambda q\,\mathrm{sgn}(x)x^{q-1} = 0, x > 0\}. \end{aligned} \quad (28)$$

More details can be found in [43].

According to Lemma 1, the solution of (26) can be easily obtained by

$$u_{ij} \in \mathrm{Prox}_{\frac{\lambda_2}{\beta_1+\tau_2}|\cdot|^q}(y_{ij}). \quad (29)$$

Based on [44], the proximal operator in (27) admits a closed-form when $q = 0$. This closed-form also exists when $q = 1/2$ and $q = 2/3$ as illustrated in [45] and [46]. For other choices $q \in (0,1)$, efficient algorithms proposed in [43] and [47] can be considered.

3) Update V by Fixing W and U: Once W and U have been updated, (14) can be calculated via

$$\min_{V\in\mathbb{R}^{d\times m}} \lambda_1\|V\|_{2,p}^p + \frac{\beta_2}{2}\|W^{k+1}-V\|_{\mathrm{F}}^2 + \frac{\tau_3}{2}\|V-V^k\|_{\mathrm{F}}^2. \quad (30)$$

It is easy to reformulate (30) as

$$\min_{V\in\mathbb{R}^{d\times m}} \lambda_1\|V\|_{2,p}^p + \frac{\beta_2+\tau_3}{2}\|V-Z\|_{\mathrm{F}}^2, \quad (31)$$

where

$$Z = \frac{\beta_2}{\beta_2+\tau_3}W^{k+1} + \frac{\tau_3}{\beta_2+\tau_3}V^k.$$

It can be further decomposed into a series of vector optimization problems as

$$\min_{\mathbf{v}^i\in\mathbb{R}^m} \lambda_1\|\mathbf{v}^i\|^p + \frac{\beta_2+\tau_3}{2}\|\mathbf{v}^i-\mathbf{z}^i\|^2, \quad (32)$$

where $i \in \{1,2,\cdots,d\}$. The solution of (32) is the proximal operator of $\|\cdot\|^p$ as in [27] and [44], which is reviewed in the next lemma.

Lemma 2: Consider the proximal operator

$$\begin{aligned}\mathrm{Prox}_{\lambda\|\cdot\|^p}(\mathbf{a}) &= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^m} \lambda\|\mathbf{x}\|^p + \frac{1}{2}\|\mathbf{x}-\mathbf{a}\|^2\\ &= \begin{cases}\mathrm{Prox}_{\lambda|\cdot|^p}(\|\mathbf{a}\|)\cdot\dfrac{\mathbf{a}}{\|\mathbf{a}\|}, & \mathbf{a}\neq\mathbf{0},\\ \{\mathbf{0}\}, & \mathbf{a}=\mathbf{0}.\end{cases}\end{aligned} \quad (33)$$

Here, $\|\mathbf{x}\|^0 = 1$ when $\mathbf{x} \neq \mathbf{0}$, and $\|\mathbf{x}\|^0 = 0$ when $\mathbf{x} = \mathbf{0}$.

From results in Lemma 2, the solution of (32) is

$$\mathbf{v}^i \in \begin{cases}\mathrm{Prox}_{\frac{\lambda_1}{\beta_2+\tau_3}|\cdot|^p}(\|\mathbf{z}^i\|)\cdot\dfrac{\mathbf{z}^i}{\|\mathbf{z}^i\|}, & \mathbf{z}^i\neq\mathbf{0},\\ \{\mathbf{0}\}, & \mathbf{z}^i=\mathbf{0}.\end{cases} \quad (34)$$

Remark 1: For Algorithm 1, it is suggested to update U^{k+1} first for eliminating the interference caused by noise and then update V^{k+1} for selecting discriminative features.

Remark 2: To establish the convergent result of Algorithm 1, there is a common pattern similar to [48], consisting of a sufficient decent lemma and the Kurdyka-Lojasiewicz (K-L) property of the objective function. For this algorithm, the main challenge is to obtain a sufficient decent lemma, because of the fact that the update of W^{k+1} relies on Algorithm 2. In [42], it is proved that Algorithm 2 converges to the zero gradient point, i.e., the point satisfies $\mathrm{grad}g(W) = 0$. However, to prove the sufficient decent lemma of Algorithm 1, there is a requirement that Algorithm 2 converges to the global optimizer, which is a more strict result. Therefore, it is deferred as a future work to tackle this challenge.

C. Computational Complexity Analysis

At the end of this section, we provide the computational complexity of Algorithm 1 (including Algorithm 2), which can be decomposed into the following four aspects.

- When initializing Algorithm 1, there require $O(n^2)$ and $O(dn^2)$ to compute H and S, respectively, which leads to the computational complexity of initialization is $O(dn^2)$.
- When updating W by Algorithm 2, the computational complexity is that of solving (22) and (23). As pointed out in [42], (22) is approximated solved by the truncated (Steihaug-Toint) conjugate-gradient method with caching, where the computational cost relies on computations of the Hessian-vector product and inner product. Here, the computational complexity of the Hessian-vector product is $O(d^2m + dm^2)$ and the inner product is $O(dm^2)$. For (23), the computational complexity is $O(d^2m + dm^2)$. By summing up these results, the computational complexity of each iteration in Algorithm 2 is $O(d^2m + dm^2)$.
- When updating U and V, the computational complexity only relies on the proximal operators, which are proved as closed-form functions in our numerical studies, and their computational complexity is $O(dm)$.
- The convergent check is based on the loss function f, which has a computational complexity of $O(d^2m)$.

Overall, the computational complexity for every iteration of Algorithm 1 is $O((\kappa+1)d^2m + \kappa dm^2 + dm)$, where κ is the iteration number of Algorithm 2.

TABLE I
STATISTICAL INFORMATION OF SELECTED DATASETS

Type	Datasets	Features	Samples	Classes
Synthetic	Diamond9	9	3000	9
	Dartboard1	9	1000	4
Real-world	COIL20	1024	1440	20
	Isolet	617	1560	26
	USPS	256	1000	10
	umist	644	575	20
	GLIOMA	4434	50	4
	pie	1024	1166	53
	LUNG	325	73	7
	MSTAR	1024	2425	10

IV. NUMERICAL STUDIES

To illustrate the effectiveness of the proposed BSUFS, this section presents extensive comparisons with several benchmark UFS methods, including graph-based methods as LapScore [16], SOGFS [19], RNE [20], and UDFS [18], and PCA-based methods as SPCAFS [24], FSPCA [49], SPCA-PSD [28], and FEN-PCAFS [29]. Note that LapScore, SOGFS, RNE, and UDFS are directly implemented by the AutoUFSTool toolbox,[1] while SPCAFS,[2] FSPCA[3], SPCA-PSD,[4] and FEN-PCAFS[5] are from the authors' GitHub repositories.

To verify whether BSUFS is better than these UFS methods and test every component of BSUFS, this section is organized as follows. Subsection IV-A describes the datasets, parameter settings, and evaluation metrics. Subsection IV-B and Subsection IV-C present numerical experiments on synthetic and real-world datasets, respectively. Subsection IV-D provides ablation experiments. Subsection IV-E shows the statistical test results. Subsection IV-F analyzes the effects of p and q. Subsection IV-G gives more discussions.

A. Experimental Setup

1) Dataset Description: There are two synthetic datasets and eight real-world datasets that are used to validate the performance of our proposed BSUFS. For more details about these datasets, please refer to Table I.

The two synthetic datasets[6] are generated by assigning specific distributions to the first two features, while the remaining seven features are filled with Gaussian noise. The eight real-world datasets encompass a diverse range of domains, such as Isolet[7] for spoken letter recognition, MSTAR_SOC_CNN[8] (referred to as MSTAR) for deep learning, GLIOMA[7] and lung_discrete[7] (referred to as LUNG) for biological information, COIL20,[7] USPS,[7] pie[9], and umist[10] for image processing.

[1]https://github.com/farhadabedinzadeh/AutoUFSTool
[2]https://github.com/quiter2005/algorithm
[3]https://github.com/tianlai09/FSPCA
[4]https://github.com/zjj20212035/SPCA-PSD
[5]https://github.com/gaoyl-group/FEN-PCAFS
[6]https://github.com/milaan9/Clustering-Datasets
[7]https://jundongl.github.io/scikit-feature/datasets.html
[8]https://github.com/zjj20212035/SPCA-PSD
[9]https://data.nvision2.eecs.yorku.ca/PIE_dataset/
[10]https://github.com/saining/PPSL/blob/master/Platform/Data/UMIST/UMIST.mat

2) Parameter Settings: For LapScore, SOGFS, and RNE, the value of k-neighbors is chosen as 5. For SOGFS, SPCAFS, SPCA-PSD, FEN-PCAFS, and BSUFS, their regularization parameters are selected from the candidate set $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$. As suggested in [43] and [45], the values of p and q for BSUFS are selected from $\{0, 1/2, 2/3\}$. Although other values in $[0, 1)$ are also possible in principle, they are not considered in this paper because there is no closed-form proximal operator and the corresponding solution must be obtained by iterative calculations. For other parameters, the default values or the best parameters provided by the authors are used.

3) Evaluation Metrics: Two key metrics are applied to evaluate these compared methods, including clustering accuracy (ACC) and normalized mutual information (NMI). Here, ACC is defined as

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(y_i, c_i), \tag{35}$$

where n is the number of samples, y_i is the true label of the i-th sample, and c_i is the cluster label of the i-th sample. The function $\delta(y_i, c_i)$ is the Kronecker delta function, which equals 1 if $y_i = c_i$ and 0 otherwise. Besides, NMI is defined as

$$\mathrm{NMI} = \frac{I(\mathbf{y}, \mathbf{c})}{\sqrt{H(\mathbf{y})H(\mathbf{c})}}, \tag{36}$$

where $\mathbf{y} = (y_1, y_2, \cdots, y_n) \in \mathbb{R}^n$, $\mathbf{c} = (c_1, c_2, \cdots, c_n) \in \mathbb{R}^n$, $I(\mathbf{y}, \mathbf{c})$ is the mutual information between the true label vector $\mathbf{y}$ and the cluster label $\mathbf{c}$, $H(\mathbf{y})$ and $H(\mathbf{c})$ are the entropies of the true label and the cluster label, respectively.

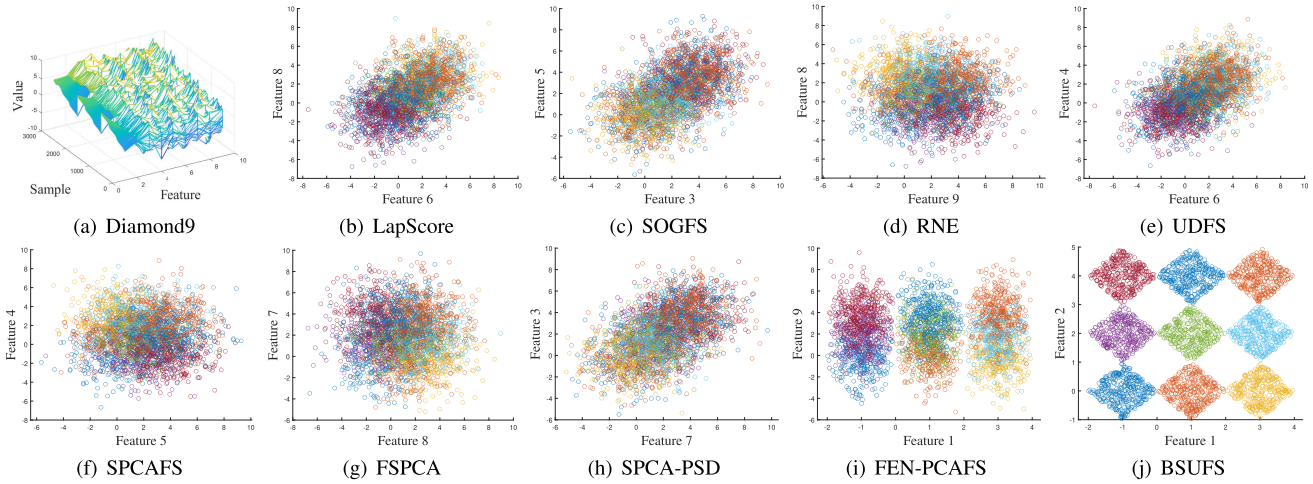
Following a similar way in [24], we select the number of features across all datasets in increments of 10, ranging from 10 to 100. To be fair and counteract the variability introduced by different initial conditions, 50 repetitions of the k-means clustering algorithm are conducted. This means that the final results are reported as the mean and standard deviation of the 50 repetitions.

B. Synthetic Results

In this experiment, various UFS methods are conducted on two synthetic datasets, i.e., Diamond9 and Dartboard1. Each UFS method is applied to obtain scores for all nine features and the top two features are selected. After that, the selected features are visualized in scatter diagrams alongside the samples.

Fig. 2 shows the feature selection results of the Diamond9 dataset, where (a) is the dataset distribution and (b)-(j) are the feature selection results. It is demonstrated that BSUFS is the only method that can successfully identify the two most discriminative features.

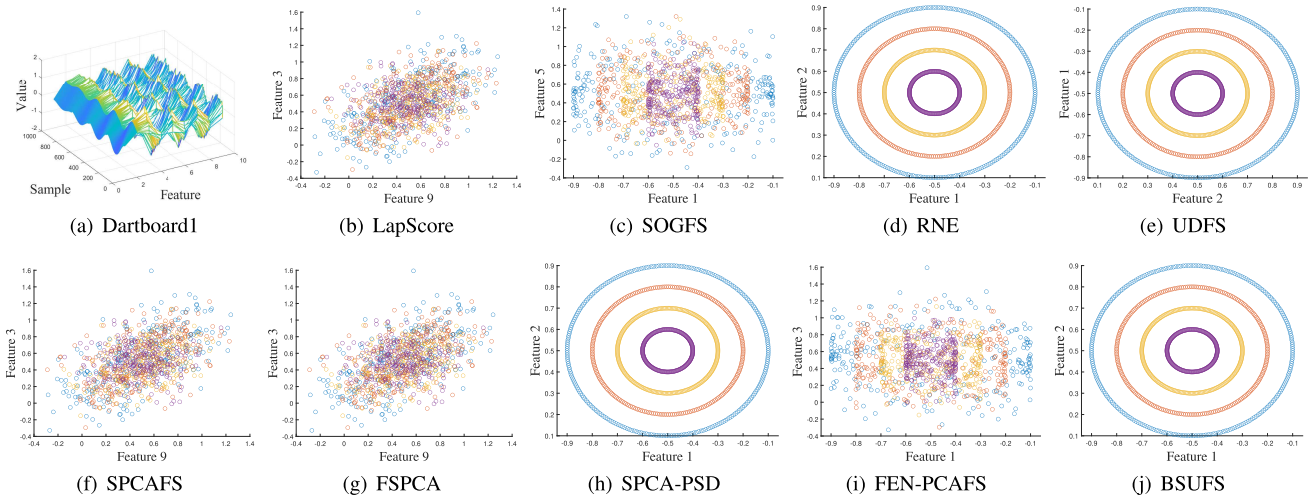
For the Dartboard1 dataset, Fig. 3 shows that RNE, UDFS, SPCA-PSD, and BSUFS are all capable of identifying the appropriate features, compared to the other methods. In Fig. 4, by adding Gaussian noise with mean 0 and standard deviation 0.01 on this dataset, UDFS, SPCA-PSD, and BSUFS identify the right features, but RNE fails. As for Fig. 5, the 0.03 salt-and-pepper noise is added that is 3% of the pixels are affected by this noise. From this figure, only UDFS and BSUFS are capable of selecting the right features. Therefore,



(a) Diamond9 (b) LapScore (c) SOGFS (d) RNE (e) UDFS

(f) SPCAFS (g) FSPCA (h) SPCA-PSD (i) FEN-PCAFS (j) BSUFS

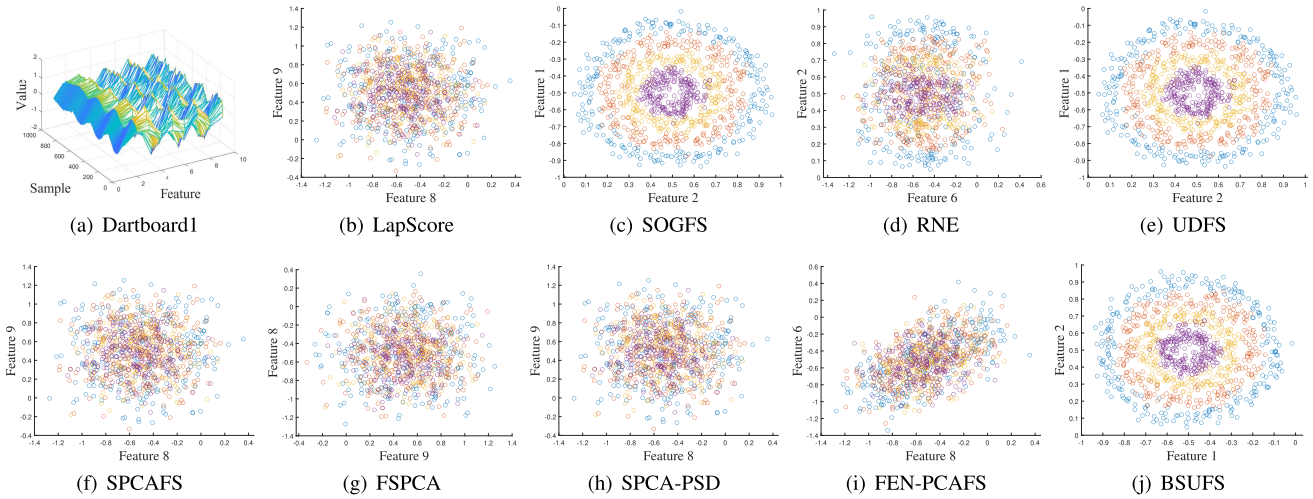
Fig. 2. Visual comparisons on the Diamond9 dataset, where (a) is the dataset distribution and (b)-(j) are the feature selection results.



(a) Dartboard1 (b) LapScore (c) SOGFS (d) RNE (e) UDFS

(f) SPCAFS (g) FSPCA (h) SPCA-PSD (i) FEN-PCAFS (j) BSUFS

Fig. 3. Visual comparisons on the Dartboard1 dataset, where (a) is the dataset distribution and (b)-(j) are the feature selection results.

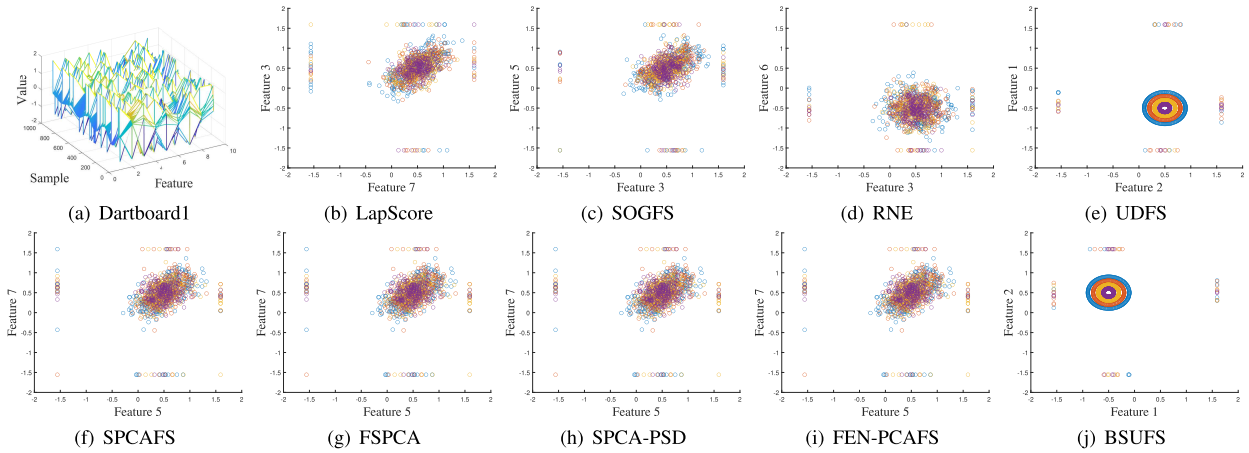


(a) Dartboard1 (b) LapScore (c) SOGFS (d) RNE (e) UDFS

(f) SPCAFS (g) FSPCA (h) SPCA-PSD (i) FEN-PCAFS (j) BSUFS

Fig. 4. Visual comparisons on the Dartboard1 dataset corrupted by 0.01 Gaussian noise, where (a) is the dataset distribution and (b)-(j) are the feature selection results.

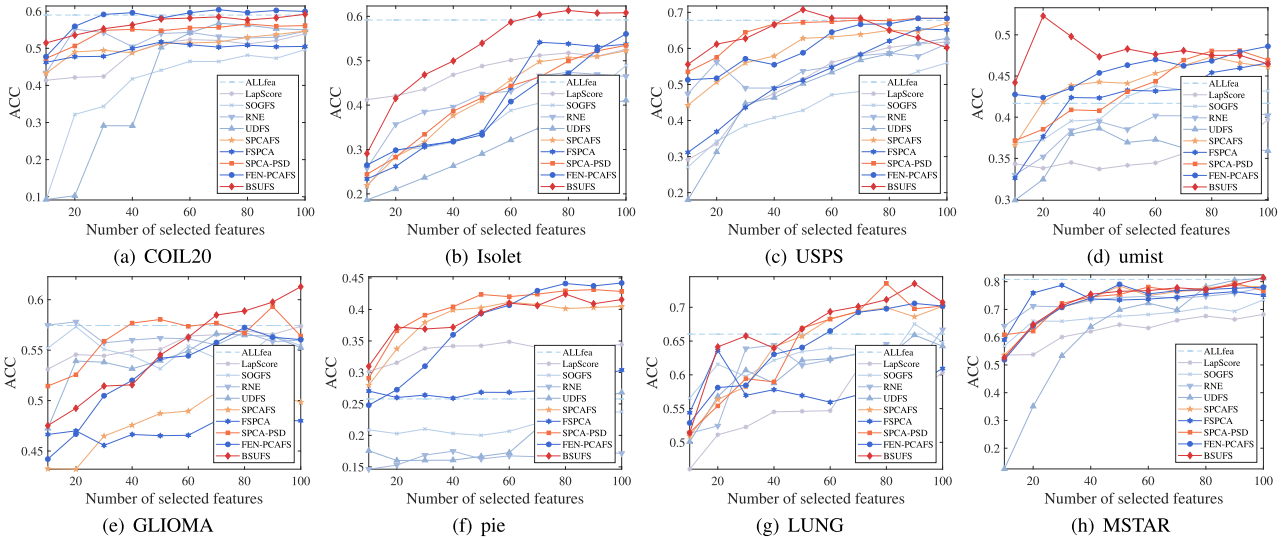
it can be concluded that UDFS and BSUFS perform better and are more robust on the Dartboard1 dataset. Besides, between graph-based methods and PCA-based methods, it is difficult to say which ones perform better.



(a) Dartboard1 (b) LapScore (c) SOGFS (d) RNE (e) UDFS

(f) SPCAFS (g) FSPCA (h) SPCA-PSD (i) FEN-PCAFS (j) BSUFS

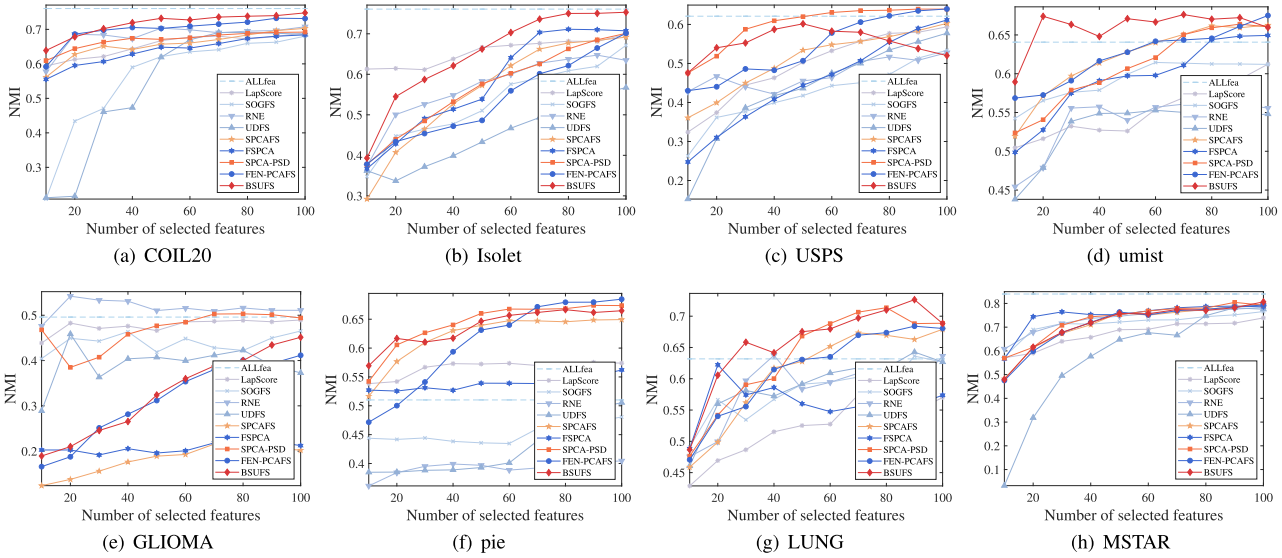
Fig. 5. Visual comparisons on the Dartboard1 dataset corrupted by 0.03 salt-and-pepper noise, where (a) is the dataset distribution and (b)-(j) are the feature selection results.



(a) COIL20 (b) Isolet (c) USPS (d) umist

(e) GLIOMA (f) pie (g) LUNG (h) MSTAR

Fig. 6. Visual comparisons of the ACC metric under different real-world datasets with different number of selected features.



(a) COIL20 (b) Isolet (c) USPS (d) umist

(e) GLIOMA (f) pie (g) LUNG (h) MSTAR

Fig. 7. Visual comparisons of the NMI metric under different real-world datasets with different number of selected features.

To sum up, our proposed BSUFS consistently selects discriminative features and performs robustly on different noises, while the other methods fail to select the correct features in some cases. All those results suggest

TABLE II

AVERAGE ACC (MEAN % ± STD %) AND NUMBER OF SELECTED FEATURES (IN BRACKETS) OF K-MEANS CLUSTERING. THE TOP TWO VALUES ARE MARKED AS RED AND BLUE

Datasets	ALLfea	LapScore	SOGFS	RNE	UDFS	SPCAFS	FSPCA	SPCA-PSD	FEN-PCAFS	BSUFS
COIL20	58.97±4.99 (10)	53.91±3.61 (100)	56.77±3.09 (70)	49.66±3.63 (100)	55.16±3.35 (20)	51.71±3.05 (50)	54.63±3.64 (100)	56.57±4.08 (80)	**60.41±4.41** (70)	**59.18±3.49** (100)
Isolet	59.18±3.19 (10)	52.55±2.83 (100)	41.11±1.71 (100)	48.93±2.69 (100)	47.39±2.91 (80)	54.15±2.69 (70)	52.26±2.81 (100)	53.45±2.82 (100)	**56.04±3.50** (100)	**61.34±3.33** (80)
USPS	67.79±4.96 (10)	61.76±4.52 (100)	62.83±3.79 (100)	56.00±3.48 (100)	61.28±3.46 (100)	65.43±4.90 (90)	66.98±3.92 (100)	**68.38±3.85** (100)	68.36±4.62 (90)	**70.77±3.73** (50)
umist	41.68±2.46 (10)	39.71±3.28 (100)	38.64±1.61 (40)	43.81±2.98 (60)	41.01±2.25 (90)	46.58±2.34 (100)	47.32±3.48 (80)	48.08±3.06 (90)	**48.61±3.23** (100)	**52.29±3.61** (20)
GLIOMA	57.44±6.40 (10)	57.36±3.60 (100)	56.64±6.47 (70)	57.32±6.47 (20)	57.80±2.98 (20)	48.04±5.26 (90)	52.08±3.64 (80)	**59.32±6.27** (90)	57.24±8.16 (80)	**61.28±9.01** (100)
pie	25.79±1.39 (10)	34.86±1.43 (60)	26.82±1.32 (100)	23.78±1.19 (100)	17.49±0.76 (40)	30.39±1.43 (100)	41.16±2.46 (60)	**43.16±2.38** (90)	**44.21±2.03** (100)	42.45±1.74 (80)
LUNG	66.03±7.23 (10)	60.93±8.02 (70)	65.89±7.43 (90)	67.53±7.73 (90)	66.68±8.32 (100)	63.62±5.45 (20)	70.16±7.71 (100)	**73.53±8.91** (80)	70.58±6.88 (90)	**73.51±6.80** (90)
MSTAR	80.81±8.76 (10)	68.21±4.57 (100)	**81.25±7.48** (100)	73.46±5.61 (100)	77.82±6.16 (100)	78.74±5.20 (30)	78.63±8.68 (90)	79.53±6.75 (90)	79.03±6.02 (50)	**81.43±6.89** (100)
Average	57.21±4.92	53.66±3.98	53.74±4.11	52.56±4.22	53.08±3.77	54.83±3.79	57.90±4.54	60.25±4.76	**60.56±4.86**	**62.78±4.83**

TABLE III

AVERAGE NMI (MEAN % ± STD %) AND NUMBER OF SELECTED FEATURES (IN BRACKETS) OF K-MEANS CLUSTERING. THE TOP TWO VALUES ARE MARKED AS RED AND BLUE

Datasets	ALLfea	LapScore	SOGFS	RNE	UDFS	SPCAFS	FSPCA	SPCA-PSD	FEN-PCAFS	BSUFS
COIL20	76.04±1.69 (10)	69.01±1.53 (100)	69.12±1.17 (80)	68.03±1.59 (100)	70.76±2.07 (100)	68.41±1.60 (100)	70.29±1.31 (100)	69.21±1.41 (100)	**73.23±1.31** (90)	**74.78±1.79** (100)
Isolet	76.09±1.77 (10)	69.86±1.26 (100)	56.73±1.05 (100)	67.15±1.45 (100)	64.74±1.28 (90)	**71.12±1.11** (80)	69.18±1.33 (100)	70.11±1.11 (100)	70.14±1.56 (100)	**75.32±1.22** (100)
USPS	62.11±2.24 (10)	59.37±1.98 (100)	57.76±2.02 (100)	53.36±1.83 (100)	52.77±2.01 (100)	61.14±1.87 (100)	60.28±2.17 (100)	**63.93±2.06** (90)	**63.96±2.24** (100)	60.16±1.68 (50)
umist	64.07±1.76 (10)	61.23±2.15 (100)	55.43±1.50 (80)	61.46±2.03 (60)	56.08±1.80 (70)	64.94±1.65 (100)	66.26±1.74 (100)	66.39±1.93 (90)	**67.51±1.92** (100)	**67.62±1.91** (70)
GLIOMA	49.59±6.76 (10)	48.96±3.59 (100)	45.86±8.08 (20)	46.51±9.11 (100)	**54.21±2.23** (20)	22.17±5.17 (90)	22.01±4.88 (80)	**50.31±6.65** (80)	41.16±7.66 (100)	45.14±8.66 (100)
pie	51.01±1.02 (10)	57.53±0.73 (90)	50.55±1.03 (100)	48.05±0.76 (100)	40.45±0.79 (100)	56.21±0.90 (100)	64.94±1.30 (100)	**67.40±1.21** (90)	**68.47±1.15** (100)	66.66±1.14 (80)
LUNG	63.18±5.48 (10)	57.44±6.44 (70)	64.27±5.35 (90)	63.62±5.41 (90)	63.74±5.30 (40)	62.23±4.80 (20)	67.91±6.23 (100)	**71.36±6.71** (80)	68.40±5.34 (90)	**72.64±4.69** (90)
MSTAR	83.96±3.14 (10)	73.90±1.62 (100)	78.18±3.64 (90)	76.56±1.54 (100)	78.26±2.51 (100)	78.87±2.52 (90)	79.62±2.30 (100)	**80.44±2.04** (90)	79.34±3.27 (100)	**80.66±2.68** (100)
Average	65.76±2.98	62.16±2.41	59.74±2.98	60.59±2.96	60.13±2.25	60.64±2.45	62.56±2.66	**67.39±2.89**	66.53±3.06	**67.87±2.97**

the effectiveness of our proposed BSUFS on synthetic datasets.

C. Real-World Results

This section presents numerical results on eight real-world datasets. Here is another compared method called ALLfea, which stands for all features used for clustering and is the gold standard for comparison. Fig. 6 and Fig. 7 show the average values of ACC and NMI for 50 repetitions under different selected feature numbers. Table II and Table III summarize the detailed results, with the top two values marked in **red** and **blue** except for ALLfea. To be specific, we set the number of selected features from 10 to 100 and report the best result with the number of features shown in brackets.

For the ACC metric, our proposed BSUFS obtains good results on most datasets, even performs better than the latest SPCA-PSD and FEN-PCAFS. In Fig. 6, almost all BSUFS lines are higher than other lines under different numbers of selected features, which implies that BSUFS performs better than others in terms of ACC. In Table II, BSUFS has the largest average ACC value on these real-world datasets, followed by FEN-PCAFS and SPCA-PSD. In addition, compared with graph-based methods (such as RNE and UDFS), PCA-based methods (such as SPCAFS and SPCA-PSD) show good performance. Due to the introduction of bi-sparse regularization, BSUFS achieves at least an average improvement of 7.95% than SPCAFS. Of course, in some cases, BSUFS is slightly inferior to SPCA-PSD, we believe that low-rank priors can also improve the performance of UFS. For the Isolet dataset, the ACC value is significantly improved. The reason may be that the fine-grained noise and speaker variability in this dataset can be effectively handled

TABLE IV

ACC (%) COMPARISONS FOR FOUR CASES IN ABLATION EXPERIMENTS. THE TOP TWO VALUES ARE MARKED AS RED AND BLUE

Datasets	Case I	Case II	Case III	Case IV
COIL20	54.09	57.33	**58.76**	**59.18**
Isolet	51.77	58.78	**56.19**	**61.34**
USPS	67.06	66.42	**68.11**	**70.77**
umist	47.16	47.57	**49.23**	**52.29**
GLIOMA	49.76	58.40	**60.12**	**61.28**
pie	40.98	41.05	**41.15**	**42.45**
LUNG	71.34	71.51	**72.33**	**73.51**
MSTAR	79.25	74.67	**80.08**	**81.43**

TABLE V

NMI (%) COMPARISONS FOR FOUR CASES IN ABLATION EXPERIMENTS. THE TOP TWO VALUES ARE MARKED AS RED AND BLUE

Datasets	Case I	Case II	Case III	Case IV
COIL20	69.94	72.12	**74.57**	**74.78**
Isolet	66.84	73.30	**72.73**	**75.32**
USPS	**58.86**	35.66	**61.14**	60.16
umist	66.48	67.77	**69.45**	**67.62**
GLIOMA	20.64	**52.11**	43.13	**45.14**
pie	65.02	65.13	**65.23**	**66.66**
LUNG	69.31	69.17	**71.94**	**72.64**
MSTAR	79.92	73.14	**79.97**	**80.66**

by the additional ℓ_q-norm, thereby achieving more effective features.

For the corresponding NMI metric, similar results can be obtained, that is, the proposed BSUFS generally outperforms other competitors. Note that the values of NMI here adopt these parameters corresponding to the best ACC, which may cause NMI to be not such good in some cases. Although not all BSUFS lines in Fig. 7 are higher than others, they are still the most lines. As shown in Table III, on average, BSUFS improves the NMI value by at least 0.48% compared with other methods. For the GLIOMA dataset with a smaller sample size and a larger number of features, it achieves a higher ACC value but a relatively lower NMI value. This may be because this dataset has only four categories, which easily leads to an imbalance in ACC and NMI.

Overall, our proposed BSUFS outperforms compared methods on many real-world datasets by achieving higher ACC and NMI values. On the other hand, achieving higher accuracy with a smaller number of selected features makes BSUFS a very practical method for real-world applications.

D. Ablation Experiments

To investigate the effect of bi-sparse regularization terms in BSUFS, this section conducts ablation experiments on

- Case I: BSUFS without $\ell_{2,p}$-norm and ℓ_q-norm, i.e.,

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) \\ \text{s.t.} \quad & W^\top W = I_m. \end{aligned} \tag{37}$$

- Case II: BSUFS without $\ell_{2,p}$-norm, i.e.,

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) + \lambda_2 \|W\|_q^q \\ \text{s.t.} \quad & W^\top W = I_m. \end{aligned} \tag{38}$$

- Case III: BSUFS without ℓ_q-norm, i.e.,

$$\begin{aligned} \min_{W\in\mathbb{R}^{d\times m}} \quad & -\mathrm{Tr}(W^\top S W) + \lambda_1 \|W\|_{2,p}^p \\ \text{s.t.} \quad & W^\top W = I_m. \end{aligned} \tag{39}$$

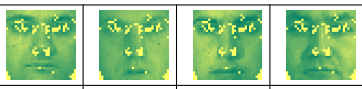
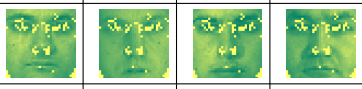
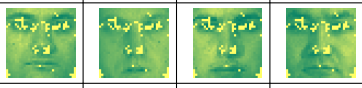
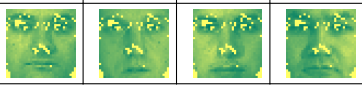
- Case IV: BSUFS.

Note that $p, q \in [0, 1)$ for all cases.

1) Clustering Efficacy: Table IV and Table V list the clustering performance in terms of ACC and NMI, respectively. These results indicate that Case IV consistently achieves the top performance. In particular, on the USPS and GLIOMA datasets, the ACC values of Case IV increase from 67.06% to 70.77% and from 49.76% to 61.28% compared to Case I, respectively. In addition, compared with Case III, almost all ACC and NMI results of Case IV are improved, which shows that the introduction of ℓ_q-norm to Case III is meaningful for feature selection.

TABLE VI

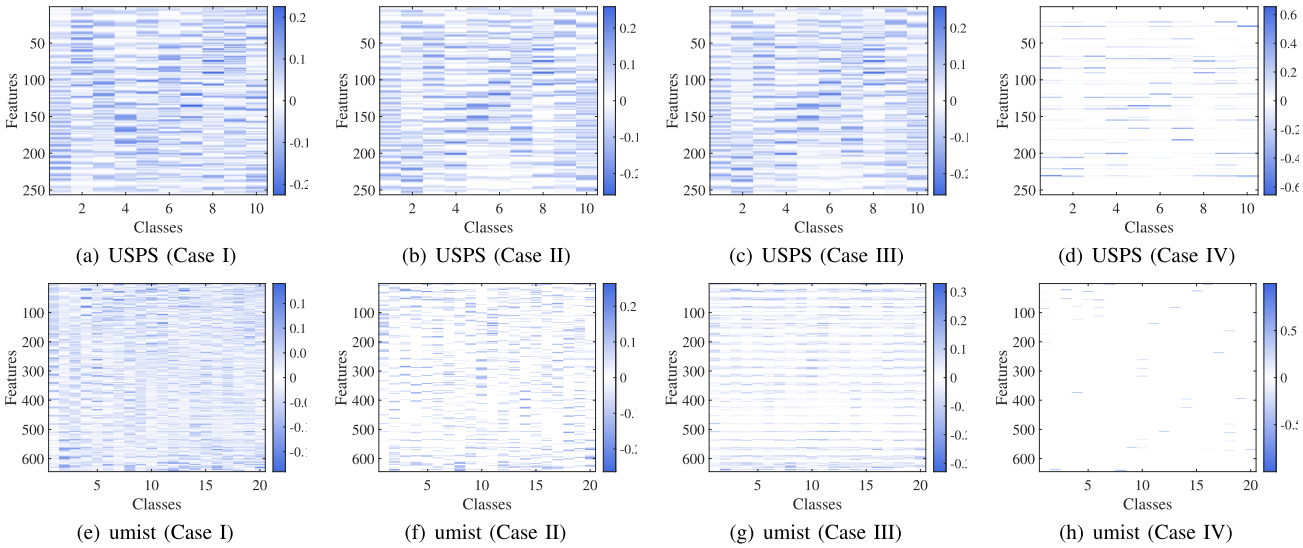
VISUAL COMPARISONS OF SELECTED IMAGE SAMPLES FROM THE PIE DATASET WITH THE CORRESPONDING ACC (%) AND NMI (%). THE TOP TWO VALUES ARE MARKED AS RED AND BLUE

Methods	Samples	ACC	NMI
Case I		40.98	65.02
Case II		41.05	65.13
Case III		**41.15**	**65.23**
Case IV		**42.45**	**66.66**

2) Feature Visualization: Table VI presents visual comparisons of feature selection results of Cases I to IV on the pie dataset. In this study, we set the number of selected features to 80 and randomly select 4 image samples to highlight the effectiveness of feature selection. Although all methods can select the basic facial features (eyes, mouth, nose, lips), Case IV selects more additional features, such as eyebrows. This diverse selection helps to maintain a more complete geometric structure of the face, potentially making better use of smaller regions of an image and reducing redundant features. Consequently, Case IV achieves higher ACC and NMI values, which reflects its superior effectiveness.

3) Sparse Analysis: Fig. 8 shows the sparse visualization of the transformation matrix W on the USPS and umist datasets. Since both USPS and umist are image datasets, they usually contain a lot of noise. Obviously, Case IV combines the basic sparsity of Case II and the row sparsity of Case III to achieve a more sparse W, that is, more focused on effective features. This is because the introduction of ℓ_q-norm regularization can eliminate noise in the datasets, thereby affecting feature selection and clustering.

In summary, by comparing Cases I, II, III with Case IV in different measurements, the introduced bi-sparse term



(a) USPS (Case I) (b) USPS (Case II) (c) USPS (Case III) (d) USPS (Case IV)

(e) umist (Case I) (f) umist (Case II) (g) umist (Case III) (h) umist (Case IV)

Fig. 8. Sparse visualization of the transformation matrix, where (a)-(d) are the results on the USPS dataset and (e)-(h) are the results on the umist dataset.

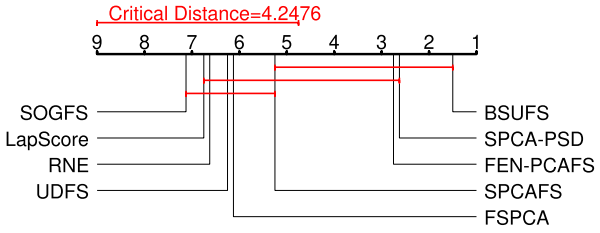


Fig. 9. Post-hoc Nemenyi test in terms of ACC.

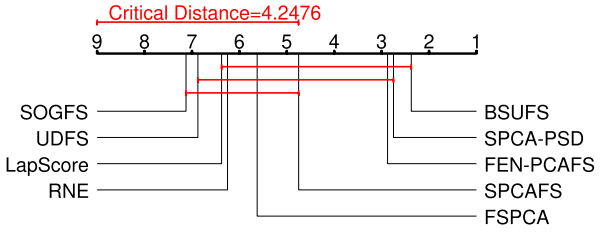


Fig. 10. Post-hoc Nemenyi test in terms of NMI.

improves the performance of PCA in feature selection, i.e., our proposed BSUFS is promising.

E. Statistical Analysis

To evaluate the pairwise differences between all compared methods, we employ the post-hoc Nemenyi test with the critical difference value as a metric. The test outcomes for ACC and NMI are shown in Fig. 9 and Fig. 10, respectively.

It can be found that BSUFS demonstrates statistically significant differences when compared to SOGFS, LapScore, UDFS, and RNE, while no significant differences are observed in the performance of BSUFS relative to FSPCA, SPCAFS, SPCA-PSD, and PEN-PCAFS. This result shows that PCA-based methods are significantly different from graph-based methods, while our proposed BSUFS has no significant difference from other PCA-based methods.

F. Effects of p and q

There are three common choices of p and q in BSUFS, that are 0, $1/2$, and $2/3$. In order to evaluate the importance of p and q, Fig. 11 and Fig. 12 present the clustering performance in terms of ACC and NMI under these different values of p and q, respectively. In these two figures, the x-axis represents various values of p and the color variations in the bars indicate different values of q.

According to the clustering results, the following conclusions can be made.

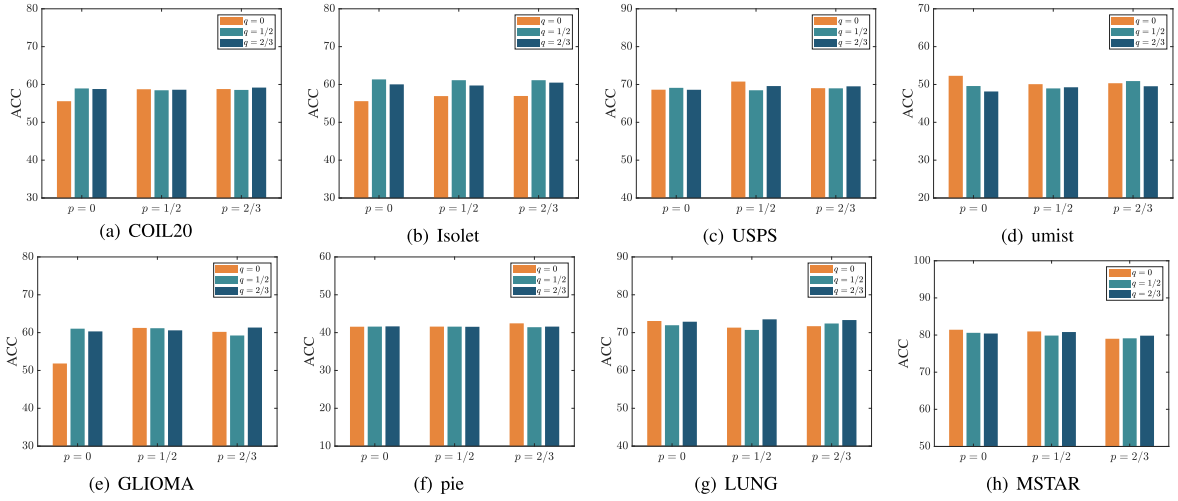
- First, the optimal choices of p and q are different for different datasets. In specific, for the Isolet dataset, the optimal values of p and q are 0 and $1/2$, respectively, while for the LUNG dataset, the optimal values are $1/2$ and $2/3$, respectively.
- Second, under different values of p and q, their ACC values and corresponding NMI values are not the same. For example, for the GLIOMA dataset, the ACC value varies greatly at $p = 0$, which also shows that the selection of q also affects the results.
- Finally, for the umist and MSTAR datasets, the best clustering performance can be observed when both p and q are set to 0, which illustrates that the extension from $(0, 1)$ to $[0, 1)$ is of importance.

Obviously, p and q should be tuned carefully. In practice, it is recommended to determine p first and then q.

G. Discussion

This section first visualizes the feature correlations between SPCAFS and our proposed BSUFS, then presents the model stability of all compared methods, and finally discusses the parameter sensitivity and convergence in the numerical perspective.

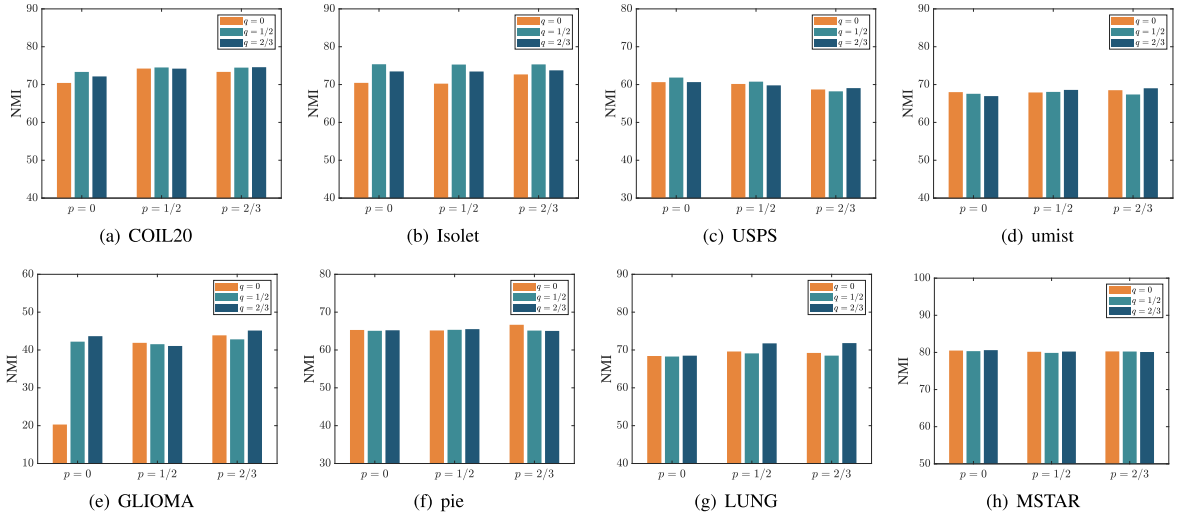
1) Feature Correlation: Fig. 13 shows the feature correlation results using SPCAFS and BSUFS on the COIL20 and USPS datasets. Here, 10 features are selected, denoted as $F_1, F_2, \cdots, F_{10}$, and the correlation among these features is examined. It can be concluded that compared with SPCAFS, the features extracted by our proposed BSUFS are more discriminative. This means that the newly introduced ℓ_q-norm can eliminate redundant features and improve feature selection results.



(a) COIL20 (b) Isolet (c) USPS (d) umist

(e) GLIOMA (f) pie (g) LUNG (h) MSTAR

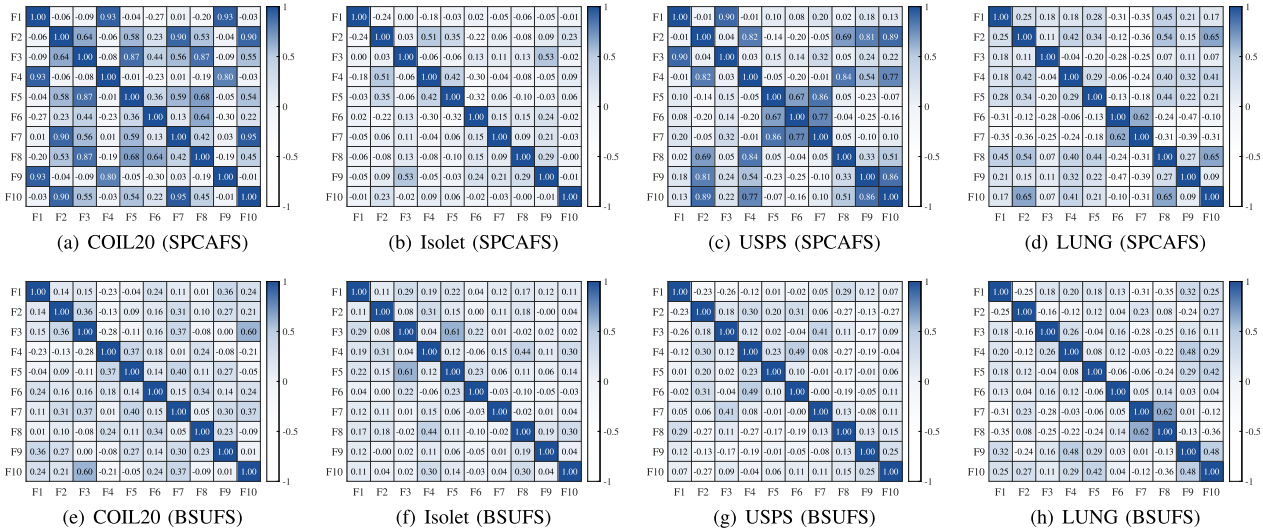
Fig. 11. Effects of p and q on eight real-world datasets in terms of ACC (%).



(a) COIL20 (b) Isolet (c) USPS (d) umist

(e) GLIOMA (f) pie (g) LUNG (h) MSTAR

Fig. 12. Effects of p and q on eight real-world datasets in terms of NMI (%).

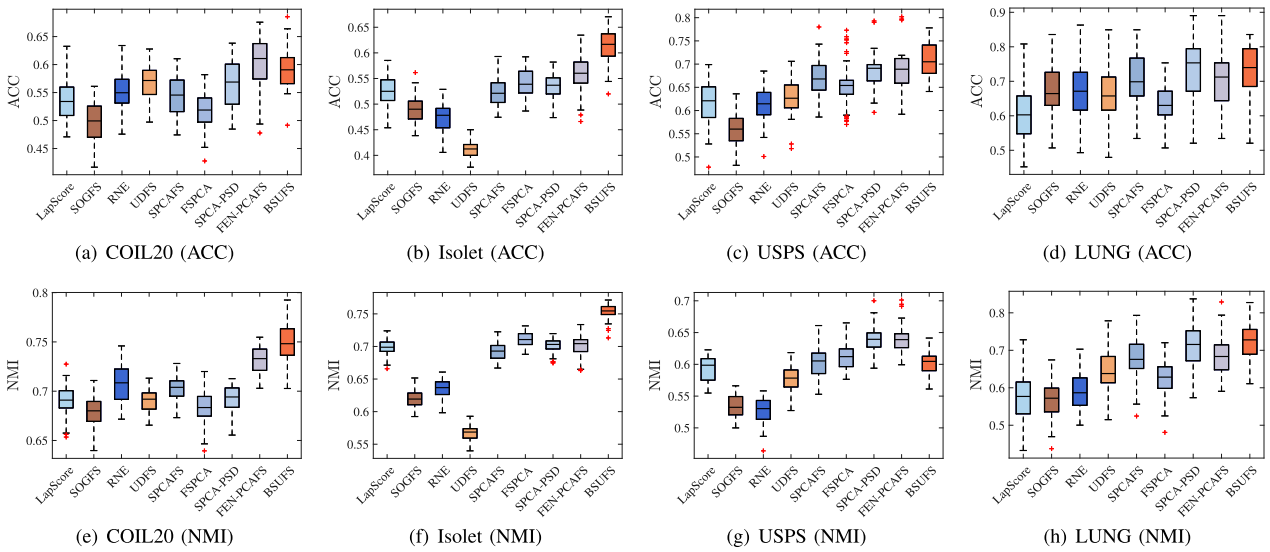


(a) COIL20 (SPCAFS) (b) Isolet (SPCAFS) (c) USPS (SPCAFS) (d) LUNG (SPCAFS)

(e) COIL20 (BSUFS) (f) Isolet (BSUFS) (g) USPS (BSUFS) (h) LUNG (BSUFS)

Fig. 13. Heatmap visualizations of correlations for 10 selected features, where (a)-(d) are the results of SPCAFS and (e)-(h) are the results of BSUFS.

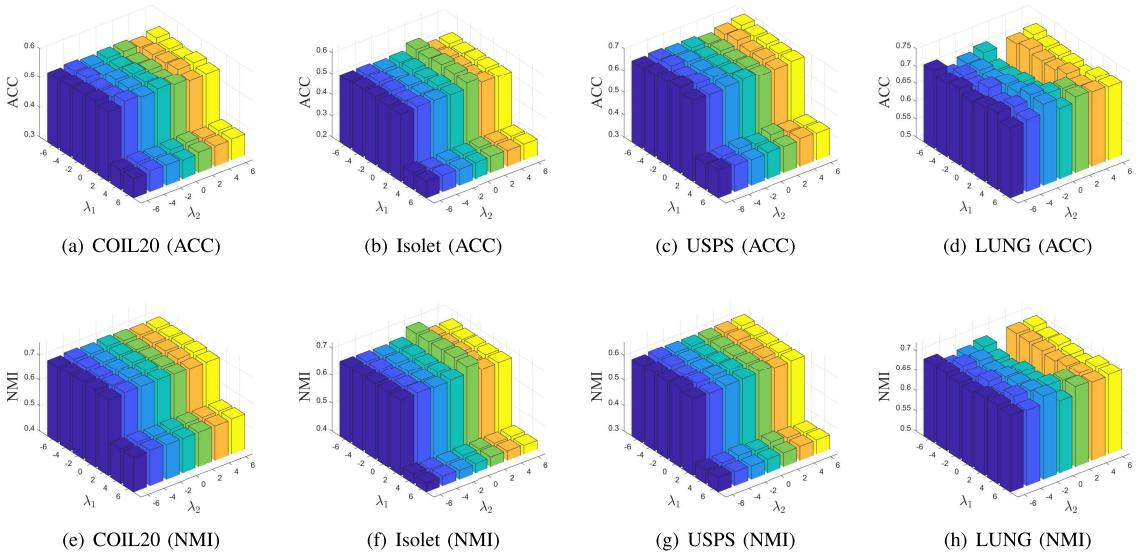
2) Model Stability: In this experiment, box plots of the 50 clustering results are shown in Fig. 14. It is obvious that in terms of ACC and NMI, the average values of BSUFS are generally larger than other methods. Especially for the Isolet data,



(a) COIL20 (ACC) (b) Isolet (ACC) (c) USPS (ACC) (d) LUNG (ACC)

(e) COIL20 (NMI) (f) Isolet (NMI) (g) USPS (NMI) (h) LUNG (NMI)

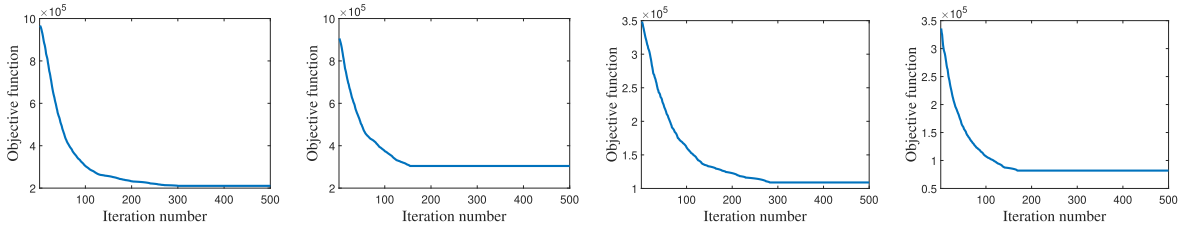
Fig. 14. Model stability comparisons of all compared methods on four real-world datasets, where (a)-(d) are the ACC results and (e)-(h) are the NMI results.



(a) COIL20 (ACC) (b) Isolet (ACC) (c) USPS (ACC) (d) LUNG (ACC)

(e) COIL20 (NMI) (f) Isolet (NMI) (g) USPS (NMI) (h) LUNG (NMI)

Fig. 15. Effects of λ_1 and λ_2 on four real-world datasets, where (a)-(d) are the ACC results and (e)-(h) are the NMI results.



(a) COIL20 (b) Isolet (c) USPS (d) LUNG

Fig. 16. Convergence curve of BSUFS on four real-world datasets.

the improvement is more obvious. This is because this dataset is highly sparse, and the advantages of double sparsity are fully demonstrated.

3) Parameter Sensitivity: Because there are two regularization terms, BSUFS needs to tune two regularization parameters. Fig. 15 investigates the effects of two regularization parameters, i.e., λ_1 and λ_2, for ACC and NMI metrics. Although the improvement under different λ_2 is not as significant as that under different λ_1, there are still differences, which also proves the necessity of the bi-sparse term in BSUFS. In general, $\ell_{2,p}$-norm plays a vital role in BSUFS, while ℓ_q-norm is a complementary choice in feature selection.

4) Convergence Analysis: Fig. 16 shows the objective value curves for Algorithm 1 on four real-world datasets. It is seen that the objective function of our proposed BSUFS shows a consistent pattern of continuous decrease and reaches stability

within finite iteration numbers. Although the convergence theorem cannot be derived as in Remark 2, the algorithm has good convergence in the numerical perspective.

V. Conclusion

In this study, a novel method called BSUFS is introduced for UFS, which integrates both $\ell_{2,p}$-norm and ℓ_q-norm to PCA with $p, q \in [0, 1)$. Technically, $\ell_{2,p}$-norm facilitates feature selection and ℓ_q-norm serves to remove the impact of redundant features. To our best knowledge, this is the first UFS method in a unified bi-sparse optimization framework. In algorithms, an efficient PAM optimization scheme is designed and its computational complexity is also analyzed. The feature selection results of BSUFS on synthetic and real-world datasets, respected to ACC and NMI, are more excellent than other competitors. Numerical studies also illustrate that the wide range of p and q is essential, and the best selection of them needs to be determined by the dataset. Furthermore, $\ell_{2,p}$-norm performs a leading role and ℓ_q-norm enhances feature selection. Obviously, this bi-sparse framework can be applied to other related image processing fields.

In the future, we are interested in extending the proposed method to tensor cases [50] for better clustering performance. Besides, developing efficient optimization algorithms based on elegant proximal operator results [51] and applying neural network methodologies [52] to automatically learn parameters are worth investigating.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015.

[2] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 1, pp. 56–70, May 2020.

[3] V. Bolón-Canedo and B. Remeseiro, "Feature selection in image analysis: A survey," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2905–2931, Apr. 2020.

[4] Z. Li and J. Tang, "Unsupervised feature selection via nonnegative spectral analysis and redundancy control," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5343–5355, Dec. 2015.

[5] Z. Hu, J. Wang, K. Zhang, W. Pedrycz, and N. R. Pal, "Bi-level spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 6597–6611, Apr. 2025.

[6] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 5, pp. 971–989, Sep. 2016.

[7] P. García-Díaz, I. Sánchez-Berriel, J. A. Martínez-Rojas, and A. M. Diez-Pascual, "Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-seq data," *Genomics*, vol. 112, no. 2, pp. 1916–1925, Mar. 2020.

[8] M. M. Rahman, O. L. Usman, R. C. Muniyandi, S. Sahran, S. Mohamed, and R. A. Razak, "A review of machine learning methods of feature selection and classification for autism spectrum disorder," *Brain Sci.*, vol. 10, no. 12, p. 949, Dec. 2020.

[9] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[10] J. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Aug. 2004.

[11] Y. Guo, Y. Zhang, Y. Chen, and S. X. Yu, "Unsupervised feature learning with emergent data-driven prototypicality," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23199–23208.

[12] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 907–948, Feb. 2020.

[13] Q. Zhou, Q. Wang, Q. Gao, M. Yang, and X. Gao, "Unsupervised discriminative feature selection via contrastive graph learning," *IEEE Trans. Image Process.*, vol. 33, pp. 972–986, 2024.

[14] D. Shi, L. Zhu, J. Li, Z. Zhang, and X. Chang, "Unsupervised adaptive feature selection with binary hashing," *IEEE Trans. Image Process.*, vol. 32, pp. 838–853, 2023.

[15] J. Li et al., "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, p. 94, 2017.

[16] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2005, pp. 507–514.

[17] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2010, pp. 333–342.

[18] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "*ℓ2,1*-norm regularized discriminative feature selection for unsupervised learning," in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[19] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 1302–1308.

[20] Y. Liu, D. Ye, W. Li, H. Wang, and Y. Gao, "Robust neighborhood embedding for unsupervised feature selection," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105462.

[21] H. Bai, M. Huang, and P. Zhong, "Precise feature selection via non-convex regularized graph embedding and self-representation for unsupervised learning," *Knowl.-Based Syst.*, vol. 296, Jul. 2024, Art. no. 111900.

[22] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Rev. Methods Primers*, vol. 2, no. 1, p. 100, 2022.

[23] H. Zou and L. Xue, "A selective overview of sparse principal component analysis," *Proc. IEEE*, vol. 106, no. 8, pp. 1311–1320, Aug. 2018.

[24] Z. Li, F. Nie, J. Bian, D. Wu, and X. Li, "Sparse PCA via $\ell_{2,p}$-norm regularization for unsupervised feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5322–5328, Apr. 2023.

[25] Y. Tian and Y. Zhang, "A comprehensive survey on regularization strategies in machine learning," *Inf. Fusion*, vol. 80, pp. 146–166, Apr. 2022.

[26] F. Nie, L. Tian, R. Wang, and X. Li, "Learning feature-sparse principal subspace," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4858–4869, Apr. 2023.

[27] X. Zhang, J. Zheng, D. Wang, G. Tang, Z. Zhou, and Z. Lin, "Structured sparsity optimization with non-convex surrogates of $\ell_{2,0}$-norm: A unified algorithmic framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6386–6402, May 2023.

[28] J. Zheng, X. Zhang, Y. Liu, W. Jiang, K. Huo, and L. Liu, "Fast sparse PCA via positive semidefinite projection for unsupervised feature selection," 2023, *arXiv:2309.06202*.

[29] Y. Gao et al., "Principal component analysis with fuzzy elastic net for feature selection," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 12, pp. 6878–6890, Dec. 2024.

[30] Y. Hu, C. Li, K. Meng, J. Qin, and X. Yang, "Group sparse optimization via $\ell_{p,q}$ regularization," *J. Mach. Learn. Res.*, vol. 18, no. 30, pp. 1–52, 2017.

[31] Y. Zhu, X. Zhang, G. Wen, W. He, and D. Cheng, "Double sparse-representation feature selection algorithm for classification," *Multimedia Tools Appl.*, vol. 76, no. 16, pp. 17525–17539, Aug. 2017.

[32] H. Wang, F. Nie, H. Huang, S. L. Risacher, A. J. Saykin, and L. Shen, "Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning," *Bioinformatics*, vol. 28, no. 12, pp. i127–i136, Jun. 2012.

[33] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 352–360.

[34] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, Mar. 2010.

[35] X. Bian, W. Xu, and S. Wang, "Double-sparsity recovery for ADC-distorted compressive sensing," in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2022, pp. 1215–1220.

[36] Y. Hu, J.-X. Liu, Y.-L. Gao, and J. Shang, "DSTPCA: Double-sparse constrained tensor principal component analysis method for feature selection," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 4, pp. 1481–1491, Jul. 2021.

[37] S. Zhang, Y. Liu, and X. Li, "Micro-Doppler effects removed sparse aperture ISAR imaging via low-rank and double sparsity constrained ADMM and linearized ADMM," *IEEE Trans. Image Process.*, vol. 30, pp. 4678–4690, 2021.
[38] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2138–2150, Sep. 2014.
[39] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Found. Trends Mach. Learn.*, vol. 10, nos. 3–4, pp. 142–363, 2017.
[40] S. Zhou, "Sparse SVM for sufficient data reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5560–5571, Sep. 2022.
[41] J. Bai, L. Jia, and Z. Peng, "A new insight on augmented Lagrangian method with applications in machine learning," *J. Scientific Comput.*, vol. 99, no. 2, p. 53, May 2024.
[42] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[43] S. Zhou, X. Xiu, Y. Wang, and D. Peng, "Revisiting $L_q(0\leq$ q <1) norm regularized optimization," 2023, *arXiv:2306.14394*.
[44] A. Beck, *First-order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.
[45] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$L_{1/2}$ regularization: A thresholding representation theory and a fast solver," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1013–1027, Jul. 2012.
[46] W. Cao, J. Sun, and Z. Xu, "Fast image deconvolution using closed-form thresholding formulas L_q $(q=\frac{1}{2}, \frac{1}{2})$ of regularization," *J. Vis. Commun. Image Represent.*, vol. 24, no. 1, pp. 31–41, Jan. 2013.
[47] J. Liu, M. Feng, X. Xiu, W. Liu, and X. Zeng, "Efficient and robust sparse linear discriminant analysis for data classification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 1, pp. 617–629, Feb. 2025.
[48] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, nos. 1–2, pp. 459–494, Aug. 2014.
[49] L. Tian, F. Nie, R. Wang, and X. Li, "Learning feature sparse principal subspace," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 14997–15008.
[50] J. Zheng, X. Zhang, W. Jiang, X. Qiu, and M. Ren, "Sparse tensor PCA via tensor decomposition for unsupervised feature selection," 2024, *arXiv:2407.16985*.
[51] S. Liao, C. Han, T. Guo, and B. Li, "Subspace Newton method for sparse group $\ell 0$} optimization problem," *J. Global Optim.*, vol. 90, no. 1, pp. 93–125, Sep. 2024.
[52] J. Zhang, B. Chen, R. Xiong, and Y. Zhang, "Physics-inspired compressive sensing: Beyond deep unrolling," *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 58–72, Jan. 2023.

Xianchao Xiu (Member, IEEE) received the Ph.D. degree in operations research from Beijing Jiaotong University, China, in 2019. From June 2019 to May 2021, he worked as a Post-Doctoral Researcher with Peking University, China. He is currently an Associate Professor with the School of Mechatronic Engineering and Automation, Shanghai University, China. His current research interests include machine learning, pattern recognition, and optimization.

Chenyi Huang received the B.S. degree in automation from Hangzhou Dianzi University, China, in 2023. He is currently pursuing the M.S. degree in control science and engineering with the School of Mechatronic Engineering and Automation, Shanghai University, China. His current research interests include sparse optimization and compression of large language models.

Pan Shang received the Ph.D. degree in statistics from Beijing Jiaotong University, China, in 2023. From October 2023 to September 2025, she worked as a Post-Doctoral Researcher with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China. She is currently an Assistant Professor with the School of Mathematics and Statistics, Beijing Jiaotong University. Her current research interests include tuning parameter selection, high-dimensional matrix regression, and optimization.

Wanquan Liu (Senior Member, IEEE) received the B.S. degree in applied mathematics from Qufu Normal University, China, in 1985, the M.S. degree in control theory and operation research from Chinese Academy of Science in 1988, and the Ph.D. degree in electrical engineering from Shanghai Jiaotong University in 1993. He has held the ARC Fellowship, U2000 Fellowship, and JSPS Fellowship, and has attracted research funds from different resources over 2.4 million dollars. He is currently a Full Professor with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include large-scale pattern recognition, signal processing, machine learning, and control systems.