Journal of Machine Learning Research 27 (2026) 1-44 Submitted 2/23; Revised 10/25; Published 1/26

Unsupervised Feature Selection via Nonnegative Orthogonal Constrained Regularized Minimization

Yan Li LIYAN2024@AMSS.AC.CN
State Key Laboratory of Mathematical Sciences
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing 100190, China

Defeng Sun DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Liping Zhang[*] LIPINGZHANG@TSINGHUA.EDU.CN
Department of Mathematical Sciences
Tsinghua University, Beijing, China
Yanqi Lake Beijing Institute of Mathematical Sciences and Applications
Beijing, China

Editor: Silvia Villa

Abstract

Unsupervised feature selection has drawn wide attention in the era of big data, since it serves as a fundamental technique for dimensionality reduction. However, many existing unsupervised feature selection models and solution methods are primarily designed for practical applications, and often lack rigorous theoretical support, such as convergence guarantees. In this paper, we first establish a novel unsupervised feature selection model based on regularized minimization with nonnegative orthogonality constraints, which has advantages of embedding feature selection into the nonnegative spectral clustering and preventing overfitting. To solve the proposed model, we develop an effective inexact augmented Lagrangian multiplier method, in which the subproblems are addressed using a proximal alternating minimization approach. We rigorously prove the algorithm's sequence converges to a stationary point of the model. Extensive numerical experiments on popular datasets demonstrate the stability and robustness of our method. Moreover, comparative results show that our method outperforms some existing state-of-the-art methods in terms of clustering evaluation metrics. The code is available at https://github.com/liyan-amss/NOCRM_code.

Keywords: unsupervised feature selection, nonnegative orthogonality constraints, augmented Lagrangian multiplier method, alternating minimization method

1. Introduction

Due to large amounts of data produced by rapid development of technology, processing high-dimensional data is one of the most challenging problems in many fields, such as action

[*] Liping Zhang is the corresponding author (lipingzhang@tsinghua.edu.cn).

recognition (Klaser et al., 2011), image classification (Gui et al., 2014), computational biology (Chen et al., 2020b). Generally, not all the features are equally important for the data with high-dimensional features. There are some redundant, irrelevant and noisy features, which not only increase computational cost and storage burden, but also reduce the performance of learning tasks. Dimensionality reduction methods can be roughly divided into two types: feature extraction (Lee and Seung, 1999; Charte et al., 2022; Lian et al., 2018) and feature selection (Kittler, 1986; Li et al., 2021; Roffo et al., 2020; Yu et al., 2019). They project the high-dimensional feature space to a low-dimensional space to squeeze features. The low-dimensional space generated by the former is usually composed of linear or nonlinear combinations of original features, but irrelevant, redundant and even noisy features are involved in the process of reducing dimension, which may affect the subsequent learning tasks to some extent. However, the latter evaluates each dimension feature of high dimensional data and directly select the optimal feature subset from the original high-dimensional feature set by using certain criteria to achieve compact and accurate data representation (Liu et al., 2004; Molina et al., 2002). Compared with the former, the latter has better interpretability. Feature selection maintains the semantic information of the original features and it aims to select valuable and discriminative feature subsets from the original high-dimensional feature set, while feature extraction changes the original meanings of the feature and the new features usually lose the physical meanings of the original features. Therefore, feature selection enjoys tremendous popularity in a wide range of applications from data mining to machine learning. Many feature selection methods (Nie et al., 2016; Hou et al., 2013; Nie et al., 2019) are proposed to better explore the properties of high-dimensional data.

According to whether the class label information is available or not, feature selection methods can be roughly grouped into two categories, i.e., supervised feature selection, and unsupervised feature selection (Dash et al., 1997; He et al., 2005). Benefiting from the sample-wise annotations, supervised feature selection algorithms, e.g., Fisher score (Duda and Hart, 2001), robust regression (Nie et al., 2010), minimum redundancy maximum relevance (Peng et al., 2005) and trace radio (Nie et al., 2008), are able to select discriminative features and achieve superior classification accuracy and reliability. With the fact that the labeled data is often inadequate or completely unobtainable in many practical applications, traditional supervised feature selection methods cannot deal with such problems. In addition, annotating the unlabeled data requires an excessive cost in human resources and is time-consuming. Therefore, for the high-dimensional data with missing labels, it is an effective means to solve above mentioned problems by using unsupervised approaches to reduce the feature dimension. Compared to supervised feature selection, unsupervised feature selection is a more challenging task since the label information of the training data is unavailable (He et al., 2005). Many studies have been conducted on unsupervised feature selection methods, such as spectral analysis (Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012), matrix factorization (Wang et al., 2015; Qian and Zhai, 2013), dictionary learning (Zhu et al., 2016) and so on.

Unsupervised feature selection methods (Nie et al., 2019; Chen et al., 2023) generally select features according to the intrinsic structural characteristics of data and have achieved pretty good performance, which can alleviate the undesirable influence of noise and redundant features in the original data. For example, MaxVar (Krzanowski, 1987) is a statistical method, which selects features corresponding to the maximum variance. Laplacian Score (He et al., 2005) is a similarity preserving method, which evaluates the importance of a feature by its power of locality preservation; SPEC (Zhao and Liu, 2007) selects features using spectral regression. RSR (Zhu et al., 2015) is a data reconstruction method, which uses the $l_{2,1}$-norm to measure the fitting error and to promote sparsity; CPFS (Masaeli et al., 2010) relaxes the feature selection problem into a continuous convex optimization problem; REFS (Li et al., 2017) embeds the reconstruction function learning process to feature selection. MCFS (Cai et al., 2010) selects features based on spectral analysis and sparse regression problem, UDFS (Yang et al., 2011a) which selects features by preserving the structure based on discriminative information; UDPFS (Wang et al., 2022) introduces fuzziness into subspace learning to learn a discriminative projection for feature selection; NDFS (Li et al., 2012) selects features by leveraging a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularization. However, numerical algorithms proposed in these unsupervised feature selection methods are often presented without convergence analysis and then lack of theoretical support (Shi et al., 2016). Furthermore, these methods may be greatly affected by disturbance and do not have good performance, and then they may not have good stability and strong robustness.

Motivated by these observations, we propose a novel unsupervised feature selection framework formulated as a regularized minimization problem with nonnegative orthogonality constraints. Our model offers two key benefits: through sparse regression, it seamlessly embeds feature selection into nonnegative spectral clustering to fully leverage the data's intrinsic geometric structure, and it explicitly guards against overfitting. In particular, the nonnegative orthogonality constraints facilitate the acquisition of pseudo class label indicators, which in turn guide the feature selection process. The $l_{2,1}$-regularization terms ensure that the subproblems in our algorithm admit the closed-form optimal solutions, while the Frobenius norm regularization explicitly controls the overfitting. These design choices distinguish it from related approaches such as NDFS (Li et al., 2012). However, in general, handing nonnegative orthogonality constraints is inherently challenging; even optimization problems with orthogonality constraints alone are known to be difficult (Absil et al., 2009). For example, Lai and Osher (Lai and Osher, 2014) proposed the method of splitting orthogonality constraints using Bregman iteration, where each subproblem admits a closed-form solution. However, this approach lacks rigorous convergence guarantees, and whether it enjoys subsequential convergence remains an open question. Several other works have explored optimization over the Stiefel manifold, such as the retraction-based proximal gradient method (Chen et al., 2020a) and Riemannian ADMM algorithm (Li et al., 2024). Nevertheless, introducing nonnegativity constraints destroys the smooth structure of the Stiefel manifold (Jiang et al., 2023) and further significantly complicates the solving process, making these approaches inapplicable in our setting. Jiang et al. (Jiang et al., 2023) showed that when objective function is continuously differentiable , one can reformulate the nonnegative orthogonality constraints via an exact penalty transformation to obtain an equivalent problem with the same optimality conditions, and they established the asymptotic convergence of the algorithm. Under the same assumption, Qian et al. (Qian et al., 2024) designed a penalty algorithm that solves a sequence of smooth penalty subproblems approximately using a retraction-based nonmonotone line-search proximal gradient method, proving that any cluster point of the generated sequence is a stationary point. However, these methods critically rely on smoothness assumptions of the objective function, which do not hold in our setting due to the presence of the nonsmooth $l_{2,1}$ regularization. Moreover, other popular solution strategies, such as the multiplicative update method (Ding et al., 2006; Yoo and Choi, 2008) and the greedy orthogonal pivoting algorithm (Zhang et al., 2019), require the objective function to be differentiable and follow a special formulation, rendering them unsuitable for our model. These limitations motivate us to develop an effective solution strategy tailored to our proposed model. Drawing inspiration from the augmented Lagrangian method (ALM) (Andreani et al., 2008) and the proximal alternating minimization (PAM) framework (Attouch et al., 2013), we propose an inexact ALM method that incorporates PAM to solve subproblems in each iteration. We establish that the sequence generated by our method converges to a stationary point of the proposed model. Extensive numerical experiments conducted on multiple benchmark datasets demonstrate the stability and robustness of the proposed method. Furthermore, the comparative results show that our method performs better than some existing state-of-the-art unsupervised feature selection methods in terms of clustering evaluation metrics.

The main contribution of this paper is summarized as follows:

- We establish a novel $l_{2,1}$-regularized optimization model with nonnegative orthogonality constraints for unsupervised feature selection, which has two advantages of embedding feature selection into the nonnegative spectral clustering and preventing overfitting. Specifically, we use the spectral clustering technique to learn pseudo class labels, and then select features which are most discriminative to pseudo class labels.

- We propose an effective inexact ALM method to solve our model, where the subproblems are solved at each iteration using the PAM method. A key advantage of this approach is that each subproblem arising in PAM admits a closed-form solution, which significantly simplifies the overall computation. This structure enables us to show that the algorithm's sequence converges to a stationary point of the model without any further assumptions.

- Numerical results on popular datasets are reported to demonstrate the efficiency, stability, and robustness of our method, as well as its computational advantages in high-dimensional settings.

The rest of this paper is organized as follows. Section 2 reviews essential preliminaries on nonsmooth optimization. In Section 3, we establish a novel model for unsupervised feature selection. Section 4 presents an inexact ALM method for solving the proposed model, and Section 5 provides a detailed convergence analysis. Numerical experiments and concluding remarks are given in the final two sections.

2. Preliminaries

In this section, we provide some preliminaries on nonsmooth optimization and introduce relevant notations. Throughout the paper, matrices are denoted by capital letters (e.g., $A, B, \cdots$) , and vectors are represented by boldface lowercase letters (e.g., $\boldsymbol{x}, \boldsymbol{y}, \cdots$). For any positive integer n, we define the set $[n] := \{1, 2, \ldots, n\}$. Given a matrix $Y \in \mathbb{R}^{n\times m}$, its maximum (elementwise) norm is denoted as

$$\|Y\|_\infty := \max\{|Y_{ij}| : \; i \in [n], \, j \in [m]\},$$

where Y_{ij} represents the entry in the i-th row and j-th column of Y. The Frobenius norm of Y is given by

$$\|Y\|_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} Y_{ij}^2},$$

and its $l_{2,1}$-norm is defined as

$$\|Y\|_{2,1} := \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} Y_{ij}^2} = \sum_{i=1}^{n}\|Y_{i:}\|_2,$$

where $Y_{i:}$ denotes the i-th row of Y and $\|\cdot\|_2$ represents the Euclidean norm. Let $\text{Vec}(Y) \in \mathbb{R}^{mn}$ be the vectorized form of Y, defined as

$$\text{Vec}(Y) := (Y_{:1}^{\intercal}, Y_{:2}^{\intercal}, \cdots, Y_{:m}^{\intercal})^{\intercal},$$

where $Y_{:j}$ denotes the j-th column of Y. Given the matrices $X, Y \in \mathbb{R}^{n\times m}$,

$$\text{Vec}([X|Y]) := (X_{:1}^{\intercal}, X_{:2}^{\intercal}, \cdots, X_{:m}^{\intercal}, Y_{:1}^{\intercal}, Y_{:2}^{\intercal}, \cdots, Y_{:m}^{\intercal})^{\intercal}.$$

For any vector $\boldsymbol{v} \in \mathbb{R}^n$, let v_i represent its i-th entry. Additionally, $\text{diag}(\boldsymbol{v}) \in \mathbb{R}^{n\times n}$ denotes the diagonal matrix with the i-th diagonal entry given by v_i. For a square matrix Y, we write $Y \succ 0$ to indicate Y is positive definite, and the trace of Y, i.e., the sum of its diagonal elements, is denoted by $\text{Tr}(Y)$. The inner product between two matrices $X, Y \in \mathbb{R}^{n\times m}$ is given by $\langle X, Y\rangle := \text{Tr}(X^{\intercal}Y)$. E represents a matrix with all elements equal to 1, while O represents a matrix with all elements equal to 0. The notation $O \le X \le E$ implies that each element of X satisfies $0 \le X_{ij} \le 1$, while $\mathbf{0} \le \boldsymbol{v} \le \mathbf{1}$ means that each element of $\boldsymbol{v}$ satisfies $0 \le v_i \le 1$. Given a set Ω, $\Pi_\Omega Y$ denotes the projection of Y on Ω. For an index sequence $\mathcal{K} = \{k_0, k_1, k_2, \ldots\}$ with $k_{j+1} > k_j$ for all $j \ge 0$, we define $\lim_{k\in\mathcal{K}} x_k := \lim_{j\to+\infty} x_{k_j}$. Finally, for any set $\mathcal{S}$, $|\mathcal{S}|$ denotes its cardinality and its indicator function is defined as

$$\delta_{\mathcal{S}}(X) = \begin{cases} 0, & \text{if } X \in \mathcal{S}; \\ +\infty, & \text{otherwise.} \end{cases} \tag{1}$$

For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient vector at $\boldsymbol{x} \in \mathbb{R}^n$ denotes by $\nabla f(\boldsymbol{x}) := \left[\frac{\partial f}{\partial x_j}(\boldsymbol{x})\right]_{j=1}^{n}$. Next, let us recall some definitions of sub-differential calculus (see, e.g., Rockafellar and Wets, 2009).

Definition 1 *Let $\mathcal{C} \subseteq \mathbb{R}^n$ and $\bar{\boldsymbol{x}} \in \mathcal{C}$. A vector $\boldsymbol{v}$ is normal to $\mathcal{C}$ at $\bar{\boldsymbol{x}}$ in the regular sense, or a regular normal, written $\boldsymbol{v} \in \hat{N}_{\mathcal{C}}(\bar{\boldsymbol{x}})$, if*

$$\langle \boldsymbol{v}, \boldsymbol{x} - \bar{\boldsymbol{x}}\rangle \le \boldsymbol{o}(\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|) \textit{ for } \boldsymbol{x} \in \mathcal{C}.$$

A vector is normal to $\mathcal{C}$ at $\bar{\boldsymbol{x}}$ in the general sense, written $\boldsymbol{v} \in N_{\mathcal{C}}(\bar{\boldsymbol{x}})$, if there exists sequence $\{\boldsymbol{x}_k\}_k \subset \mathcal{C}, \{\boldsymbol{v}_k\}_k$ such that $\boldsymbol{x}_k \to \bar{\boldsymbol{x}}$ and $\boldsymbol{v}_k \to \boldsymbol{v}$ with $\boldsymbol{v}_k \in \hat{N}_{\mathcal{C}}(\boldsymbol{x}_k)$. The cone $N_{\mathcal{C}}(\bar{\boldsymbol{x}})$ is called the normal cone to $\mathcal{C}$ at $\bar{\boldsymbol{x}}$.

Definition 2 *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function.*

1) *The domain of f is defined and denoted by* $\operatorname{dom} f := \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty\}$.

2) *For each* $\boldsymbol{x} \in \operatorname{dom} f$*, the vector* $\boldsymbol{x}^* \in \mathbb{R}^n$ *is said to be a regular subgradient of* f *at* $\boldsymbol{x}$*, written* $\boldsymbol{x}^* \in \hat{\partial} f(\boldsymbol{x})$*, if* $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{x}^*, \boldsymbol{y} - \boldsymbol{x} \rangle + \boldsymbol{o}(\|\boldsymbol{x} - \boldsymbol{y}\|)$.

3) *The vector* $\boldsymbol{x}^* \in \mathbb{R}^n$ *is said to be a (limiting) subgradient of* f *at* $\boldsymbol{x} \in \operatorname{dom} f$*, written* $\boldsymbol{x}^* \in \partial f(\boldsymbol{x})$*, if there exists* $\{\boldsymbol{x}_n\}_n$*,* $\{\boldsymbol{x}_n^*\}_n$ *such that* $\boldsymbol{x}_n \to \boldsymbol{x}$*,* $f(\boldsymbol{x}_n) \to f(x)$*, and* $\boldsymbol{x}_n^* \in \hat{\partial} f(\boldsymbol{x}_n)$ *with* $\boldsymbol{x}_n^* \to \boldsymbol{x}^*$.

4) *For each* $\boldsymbol{x} \in \operatorname{dom} f$*,* $\boldsymbol{x}$ *is called (limiting)-critical if* $\mathbf{0} \in \partial f(\boldsymbol{x})$.

Remark 3 (Closedness of ∂f) *Let* $(\boldsymbol{x}_k, \boldsymbol{x}_k^*) \in \operatorname{Graph} \partial f$ *be a sequence that converges to* $(\boldsymbol{x}, \boldsymbol{x}^*)$*. By the definition of* $\partial f(\boldsymbol{x})$*, if* $f(\boldsymbol{x}_k)$ *converges to* $f(\boldsymbol{x})$ *then* $(\boldsymbol{x}, \boldsymbol{x}^*) \in \operatorname{Graph} \partial f$.

Remark 4 (Rockafellar and Wets (2009), Example 6.7) *Let* $\mathcal{S}$ *be a closed nonempty subset of* $\mathbb{R}^n$*, then*

$$\partial \delta_{\mathcal{S}}(\bar{\boldsymbol{x}}) = N_{\mathcal{S}}(\bar{\boldsymbol{x}}), \ \bar{\boldsymbol{x}} \in \mathcal{S}. \tag{2}$$

Furthermore, for a smooth mapping $G : \mathbb{R}^n \to \mathbb{R}^m$*, i.e.,* $G(\boldsymbol{x}) := (g_1(\boldsymbol{x}), \cdots, g_m(\boldsymbol{x}))^\intercal$*, define* $\mathcal{S} = G^{-1}(\mathbf{0}) \subset \mathbb{R}^n$*. For* $G(\cdot)$*, its Jacobian at* $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ *is the matrix* $\boldsymbol{J}G(\boldsymbol{x}) := [\frac{\partial g_j}{\partial x_i}(\boldsymbol{x})]_{j,i=1}^{m,n} \in \mathbb{R}^{m \times n}$*. If* $\boldsymbol{J}G(\bar{\boldsymbol{x}})$ *has full rank* m *at a point* $\bar{\boldsymbol{x}} \in \mathcal{S}$*, with* $G(\bar{\boldsymbol{x}}) = \mathbf{0}$*, then its normal cone to* $\mathcal{S}$ *can be explicitly written as*

$$N_{\mathcal{S}}(\bar{\boldsymbol{x}}) = \{\boldsymbol{J}G(\bar{\boldsymbol{x}})^\intercal \boldsymbol{y} \mid \boldsymbol{y} \in \mathbb{R}^m\}.$$

3. A New Unsupervised Feature Selection Model

Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$ be the data matrix with each column $\boldsymbol{x}_i \in \mathbb{R}^{d \times 1}$ being the i-th data point. Let d and n be the number of features and the number of samples, respectively. Suppose these n samples are sampled from c classes. Denote $F = [\boldsymbol{f}_1, \cdots, \boldsymbol{f}_n]^\intercal \in \{0,1\}^{n \times c}$, where $\boldsymbol{f}_i \in \{0,1\}^{c \times 1}$ is the cluster indicator vector for $\boldsymbol{x}_i$. That is, the j-th element of $\boldsymbol{f}_i$ is 1, if $\boldsymbol{x}_i$ is assigned to the j-th cluster, otherwise 0. Following the notation in Yang et al. (2011b), the scaled cluster indicator matrix Y is defined as

$$Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n]^\intercal = F(F^\intercal F)^{-\frac{1}{2}},$$

where $\boldsymbol{y}_i$ is the scaled cluster indicator of $\boldsymbol{x}_i$. It turns out that

$$Y^\intercal Y = (F^\intercal F)^{-\frac{1}{2}} F^\intercal F (F^\intercal F)^{-\frac{1}{2}} = I_c,$$

where $I_c \in \mathbb{R}^{c \times c}$ is an identity matrix.

At first, we use the clustering techniques to learn the scaled cluster indicators of data points, which can be regarded as pseudo class labels. Given a set of data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ and some notion of similarity $S_{ij} \geq 0$ between all pairs of data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the intuitive goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. Spectral clustering is widely used in that it can effectively generate the pseudo labels from the graphs.

In our method, we construct a k-nearest neighbors graph and choose the Gaussian kernel as the weight (see Cai et al., 2005). Specially, we define the affinity graph S as follows:

$$S_{ij} = \begin{cases} \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}), & \boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in \mathcal{N}_k(\boldsymbol{x}_i); \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_k(\boldsymbol{x})$ is the set of k-nearest neighbors of $\boldsymbol{x}$. The corresponding degree matrix can be constructed to D with $D_{ii} = \sum_j S_{ij}$, and Laplacian matrix L of the normalized graph (see Von Luxburg, 2007) is calculated with $L = D^{-\frac{1}{2}}(D - S)D^{-\frac{1}{2}}$, which is symmetric and positive semi-definite. Therefore, the local geometrical structure of data points can be obtained by:

$$\min_{Y \in \mathbb{R}^{n\times c}} \operatorname{Tr}(Y^\intercal L Y) \quad \text{s.t.} \quad Y = F(F^\intercal F)^{-\frac{1}{2}}. \tag{3}$$

This is a discrete optimization problem as the entries of the feasible solution are only allowed to take two particular values, and of course it is a NP-hard problem. A well-known method is to discard the discreteness condition and relax the problem by allowing the entries of the matrix Y to take arbitrary real values. Then, the relaxed problem becomes:

$$\min_{Y \in \mathbb{R}^{n\times c}} \operatorname{Tr}(Y^\intercal L Y) \quad \text{s.t.} \quad Y^\intercal Y = I_c. \tag{4}$$

The next stage is to construct a sparse transformation W on the data matrix X by employing the scaled cluster indicator matrix Y, joined with two regularization terms. We can formulate it as:

$$\min_{W \in \mathbb{R}^{d\times c}} \|Y - X^\intercal W\|_{2,1} + \beta\|W\|_{2,1} + \gamma\|W\|_F^2, \tag{5}$$

where W is a linear and low dimensional transformation matrix, and β and γ are the regularization parameters. In the objective function of the problem (5), the first term represents the linear transformation model to measure the association between the features and the pseudo class labels. The second term constructs the sparsity on the rows of the transformation matrix W, which is beneficial for selecting discriminative features. The third term is to avoid overfitting.

By integrating the spectral clustering (4) and sparse regression (5) in a joint objective function, the model we proposed can be obtained as follows:

$$\begin{aligned} &\min_{W,Y} \operatorname{Tr}(Y^\intercal L Y) + \alpha\|Y - X^\intercal W\|_{2,1} + \beta\|W\|_{2,1} + \gamma\|W\|_F^2 \\ &\text{s.t.} \quad Y^\intercal Y = I_c,\ Y \geq O, \end{aligned} \tag{6}$$

where α is a tuning parameter.

4. Algorithm Description of Our Inexact ALM Method

In this section, we develop an inexact augmented Lagrangian method to solve problem (6), which is a nonconvex optimization problem with a nonsmooth objective function. By introducing auxiliary variables $U, V, \widehat{Y}$, and F, problem (6) can be reformulated into the following equivalent form:

$$
\begin{aligned}
&\min_{W,U,V,Y,F,\widehat{Y}} \operatorname{Tr}(Y^\intercal LY) + \alpha\|U\|_{2,1} + \beta\|V\|_{2,1} + \gamma\|W\|_F^2 + \delta_{\mathcal{S}_1}(\widehat{Y}) + \delta_{\mathcal{S}_2}(F) \\
&\quad \text{s.t.} \quad \begin{cases} Y = F, \\ U = Y - X^\intercal W, \\ V = W, \\ Y = \widehat{Y}, \end{cases}
\end{aligned} \tag{7}
$$

where $\mathcal{S}_1 = \{\, \widehat{Y} \in \mathbb{R}^{n\times c} \mid \widehat{Y}^\intercal \widehat{Y} = I_c \,\}$, and $\mathcal{S}_2 = \{\, F \in \mathbb{R}^{n\times c} \mid O \le F \le E \,\}$.

Let $\Lambda := (\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4) \in \mathbb{R}^{n\times c} \times \mathbb{R}^{d\times c} \times \mathbb{R}^{n\times c} \times \mathbb{R}^{n\times c}$. The augmented Lagrangian function corresponding to problem (7) is defined as follows

$$
\begin{aligned}
L(W, U, V, Y, F, \widehat{Y}, \Lambda; \rho) :=& \operatorname{Tr}(Y^\intercal LY) + \alpha\|U\|_{2,1} + \beta\|V\|_{2,1} + \gamma\|W\|_F^2 + \delta_{\mathcal{S}_1}(\widehat{Y}) \\
&+ \delta_{\mathcal{S}_2}(F) + \langle \Lambda_1, Y - X^\intercal W - U\rangle + \langle \Lambda_2, V - W\rangle \\
&+ \langle \Lambda_3, Y - F\rangle + \langle \Lambda_4, \widehat{Y} - Y\rangle + \frac{\rho}{2}(\|\widehat{Y} - Y\|_F^2 + \|V - W\|_F^2 \\
&+ \|Y - F\|_F^2 + \|Y - X^\intercal W - U\|_F^2),
\end{aligned} \tag{8}
$$

where ρ is a positive penalty parameter.

The ALM method is employed to alternately update the variables $(W, U, V, Y, F, \widehat{Y})$, the multiplier Λ, and the penalty parameter ρ, in order to satisfy the accuracy condition given in (11). The proposed inexact ALM method for solving problem (6) is detailed as follows:

Algorithm 1 Inexact ALM Method for (6)

Input. Data matrix $X \in \mathbb{R}^{d\times n}$. Given predefined parameters $\{\epsilon_k\}_{k\in\mathbb{N}}$, ρ^1, τ, r, $\overline{\Lambda}_{t,min}$, $\overline{\Lambda}_{t,max}$ $(t = 1, 2, 3, 4)$, and $\overline{\Lambda}^1 := (\overline{\Lambda}_1^1, \overline{\Lambda}_2^1, \overline{\Lambda}_3^1, \overline{\Lambda}_4^1)$ that satisfy the condition in Remark 5, for $k = 1, 2, \ldots,$

Output. Sort all the d features according to $\|W_{i:}\|_2$ $(i \in [d])$ and select the top q ranked features.

Step 1: Compute the subproblem

$$
(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k) \approx \operatorname*{arg\,min}_{W,U,V,Y,F,\widehat{Y}} L(W, U, V, Y, F, \widehat{Y}, \overline{\Lambda}^k; \rho^k) \tag{9}
$$

such that

$$
O \le F^k \le E, \quad (\widehat{Y}^k)^\intercal \widehat{Y}^k = I_c, \tag{10}
$$

and there exists $\Theta^k \in \partial L(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k, \overline{\Lambda}^k; \rho^k)$ satisfying

$$
\|\Theta^k\|_\infty \le \epsilon_k. \tag{11}
$$

Step 2: Update the multiplier as:

$$
\begin{aligned}
\Lambda_1^{k+1} &= \overline{\Lambda}_1^k + \rho^k(Y^k - X^\intercal W^k - U^k), \\
\Lambda_2^{k+1} &= \overline{\Lambda}_2^k + \rho^k(V^k - W^k), \\
\Lambda_3^{k+1} &= \overline{\Lambda}_3^k + \rho^k(Y^k - F^k), \\
\Lambda_4^{k+1} &= \overline{\Lambda}_4^k + \rho^k(\widehat{Y}^k - Y^k), \text{‘}
\end{aligned}
$$

where $\overline{\Lambda}_t^{k+1} = \Pi_\Omega \Lambda_t^{k+1}$ and $\Omega = \{\Lambda_t : \overline{\Lambda}_{t,min} \leq \Lambda_t \leq \overline{\Lambda}_{t,max}\}$ $(t = 1, 2, 3, 4)$.
Step 3: Update the penalty parameter:

$$\rho^{k+1} = \begin{cases} \rho^k, & \text{if } \|R_t^k\|_\infty \leq \tau \|R_t^{k-1}\|_\infty \ (t = 1, 2, 3, 4); \\ r\rho^k, & \text{otherwise}, \end{cases} \tag{12}$$

where $R_1^k = Y^k - X^\intercal W^k - U^k$, $R_2^k = V^k - W^k$, $R_3^k = Y^k - F^k$, $R_4^k = \widehat{Y}^k - Y^k$.

Remark 5 *Set the parameters in Algorithm 1 as follows: $\tau \in [0, 1)$, $\rho^1 > 0$, $r > 1$, and the sequence of positive tolerance parameters $\{\epsilon_k\}_{k\in\mathbb{N}}$ is chosen such that $\lim_{k\to+\infty} \epsilon_k = 0$. The parameters $\overline{\Lambda}_1^1, \overline{\Lambda}_2^1, \overline{\Lambda}_3^1, \overline{\Lambda}_4^1, \overline{\Lambda}_{t,min}, \overline{\Lambda}_{t,max}$ are finite-valued matrices satisfying*

$$-\infty < (\overline{\Lambda}_{t,min})_{ij} < (\overline{\Lambda}_{t,max})_{ij} < +\infty \ \forall i, j, \quad t = 1, 2, 3, 4.$$

In Algorithm 1, the may challenge lies in solving (9)–(11). Specifically, given the current iterates $(W^{k-1}, U^{k-1}, V^{k-1}, Y^{k-1}, F^{k-1}, \widehat{Y}^{k-1})$, , the key task is to determine how to generate the next iterate $(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k)$. To address this issue, we propose a PAM method to solve (9)–(11), and we show that a solution exists for (9)–(11) and can be efficiently computed as $\epsilon_k \downarrow 0$. This guarantees that Step 1 in Algorithm 1 is well defined. The following two subsections present the details of the PAM method and its convergence analysis.

4.1 PAM for Augmented Lagrangian Subproblems

In this subsection, we present further details on the implementation of Algorithm 1 and develop a PAM method to solve the augmented Lagrangian subproblems to an arbitrarily prescribed accuracy.

It is evident that the constraint in (11) represents an ϵ^k-perturbation of the critical point condition:

$$0 \in \partial L(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k, \overline{\Lambda}^k; \rho^k). \tag{13}$$

In fact, the proposed PAM method for solving (13) can be interpreted as a regularized proximal six-block Gauss-Seidel method (Attouch et al., 2013). At the k-th outer iteration, problem (13) can be solved to arbitrary accuracy using the following alternating minimizing procedure:

(a) Update $W^{k,j}$:

$$W^{k,j} \in \arg\min \{L(W, U^{k,j-1}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}, \overline{\Lambda}^k; \rho^k) + \frac{C_1^{k,j-1}}{2} \|W - W^{k,j-1}\|_F^2\}, \tag{14}$$

(b) Update $U^{k,j}$:

$$U^{k,j} \in \arg\min \{L(W^{k,j}, U, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}, \overline{\Lambda}^k; \rho^k) + \frac{C_2^{k,j-1}}{2} \|U - U^{k,j-1}\|_F^2\}, \tag{15}$$

(c) Update $V^{k,j}$:

$$V^{k,j} \in \arg\min \{L(W^{k,j}, U^{k,j}, V, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}, \overline{\Lambda}^{k}; \rho^k) + \frac{C_3^{k,j-1}}{2}\|V - V^{k,j-1}\|_F^2\}, \tag{16}$$

(d) Update $Y^{k,j}$:

$$Y^{k,j} \in \arg\min \{L(W^{k,j}, U^{k,j}, V^{k,j}, Y, F^{k,j-1}, \widehat{Y}^{k,j-1}, \overline{\Lambda}^{k}; \rho^k) + \frac{C_4^{k,j-1}}{2}\|Y - Y^{k,j-1}\|_F^2\}, \tag{17}$$

(e) Update $F^{k,j}$:

$$F^{k,j} \in \arg\min \{L(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F, \widehat{Y}^{k,j-1}, \overline{\Lambda}^{k}; \rho^k) + \frac{C_5^{k,j-1}}{2}\|F - F^{k,j-1}\|_F^2\}, \tag{18}$$

(f) Update $\widehat{Y}^{k,j}$:

$$\widehat{Y}^{k,j} \in \arg\min \{L(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}, \overline{\Lambda}^{k}; \rho^k) + \frac{C_6^{k,j-1}}{2}\|\widehat{Y} - \widehat{Y}^{k,j-1}\|_F^2\}, \tag{19}$$

where the proximal parameters $\{C_i^{k,j}\}_{k,j}$ need to satisfy

$$0 < \underline{C} \leq C_i^{k,j} < \overline{C} < \infty, \ k, j \in \mathbb{N}, \ i = 1, 2, 3, 4, 5, 6,$$

for some predetermined positive constants $\underline{C}$ and $\overline{C}$.

By direct calculation, the subproblems in (14)-(19) have closed-form solutions given by:

(a) For (14): If $d \leq n$,

$$W^{k,j} = (aI_d + \rho^k XX^\intercal)^{-1}Z$$

and if $n < d$, we can apply the Woodbury matrix identity (Higham, 2002) to obtain

$$W^{k,j} = \left(\frac{1}{a}I_d - \frac{\rho^k}{a^2}X(I_n + \frac{\rho^k}{a}X^\intercal X)^{-1}X^\intercal\right)Z,$$

where $a = 2\gamma + \rho^k + C_1^{k,j-1}$ and

$$Z = X\overline{\Lambda}_1^k + \overline{\Lambda}_2^k + \rho^k XY^{k,j-1} - \rho^k XU^{k,j-1} + \rho^k V^{k,j-1} + C_1^{k,j-1}W^{k,j-1}.$$

(b) For (15): $U^{k,j} = (U_{i:}^{k,j})_{i\in[n]}$, where $U_{i:}^{k,j}$ is the row vector of $U^{k,j}$.

Set

$$N = Y^{k,j-1} - X^\intercal W^{k,j} + \frac{\overline{\Lambda}_1^k}{\rho^k}$$

and denote $N_{i:}$ as the i-th row vector of N. Then,

$$U_{i:}^{k,j} = \max\left\{0, 1 - \frac{\alpha}{\|\rho^k N_{i:} + C_2^{k,j-1}U_{i:}^{k,j-1}\|_2}\right\}\left(\frac{\rho^k}{\rho^k + C_2^{k,j-1}}N_{i:} + \frac{C_2^{k,j-1}}{\rho^k + C_2^{k,j-1}}U_{i:}^{k,j-1}\right).$$

(c) For (16): $V^{k,j} = (V^{k,j}_{i:})_{i\in[d]}$, where $V^{k,j}_{i:}$ is the row vector of $V^{k,j}$.
Set

$$M = W^{k,j} - \frac{\overline{\Lambda}_2^k}{\rho^k}$$

and its row vector is denoted by $M_{i:}$. Then,

$$V^{k,j}_{i:} = \max\left\{0, 1 - \frac{\beta}{\|\rho^k M_{i:} + C_3^{k,j-1}V^{k,j-1}_{i:}\|_2}\right\}\left(\frac{\rho^k}{\rho^k + C_3^{k,j-1}}M_{i:} + \frac{C_3^{k,j-1}}{\rho^k + C_3^{k,j-1}}V^{k,j-1}_{i:}\right).$$

(d) For (17):

$$Y^{k,j} = [2L + (3\rho^k + C_4^{k,j-1})I]^{-1}P,$$

where

$$P = \overline{\Lambda}_4^k - \overline{\Lambda}_3^k - \overline{\Lambda}_1^k + \rho^k X^\intercal W^{k,j} + \rho^k U^{k,j} + \rho^k F^{k,j-1} + \rho^k \widehat{Y}^{k,j-1} + C_4^{k,j-1}Y^{k,j-1}.$$

(e) For (18):
$F^{k,j} = (F^{k,j}_{st})_{s\in[n],t\in[c]}$ and $F^{k,j}_{st} = \Pi_{[0,1]}A_{st}$, where

$$A = (A_{st})_{s\in[n],t\in[c]} = \frac{\rho^k(Y^{k,j} + \frac{\overline{\Lambda}_3^k}{\rho^k}) + C_5^{k,j-1}F^{k,j-1}}{\rho^k + C_5^{k,j-1}}.$$

(f) For (19):
$\widehat{Y}^{k,j} = UI_{n\times c}V^\intercal$, where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{c\times c}$ are orthogonal matrices obtained from the SVD of the matrix

$$\frac{\rho^k(Y^{k,j} - \frac{\overline{\Lambda}_4^k}{\rho^k}) + C_6^{k,j-1}\widehat{Y}^{k,j-1}}{\rho^k + C_6^{k,j-1}} = U\sum V^\intercal$$

with $\sum \in \mathbb{R}^{n\times c}$ being a diagonal matrix.

The inner iteration terminates if there exists $\Theta^{k,j} \in \partial L(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j}, \overline{\Lambda}^k; \rho^k)$ such that

$$\|\Theta^{k,j}\|_\infty \leq \epsilon_k, \quad O \leq F^{k,j} \leq E, \quad (\widehat{Y}^{k,j})^\intercal\widehat{Y}^{k,j} = I_c,$$

where $\Theta^{k,j} := (\Theta_1^{k,j}, \Theta_2^{k,j}, \Theta_3^{k,j}, \Theta_4^{k,j}, \Theta_5^{k,j}, \Theta_6^{k,j}) \in \mathbb{R}^{d\times c}\times\mathbb{R}^{n\times c}\times\mathbb{R}^{d\times c}\times\mathbb{R}^{n\times c}\times\mathbb{R}^{n\times c}\times\mathbb{R}^{n\times c}$ is concretely expressed as

$$\begin{cases} \Theta_1^{k,j} := \rho^k X(Y^{k,j-1} - Y^{k,j}) + \rho^k X(U^{k,j} - U^{k,j-1}) + \rho^k(V^{k,j-1} - V^{k,j}) \\ \qquad + C_1^{k,j-1}(W^{k,j-1} - W^{k,j}), \\ \Theta_2^{k,j} := \rho^k(Y^{k,j-1} - Y^{k,j}) + C_2^{k,j-1}(U^{k,j-1} - U^{k,j}), \\ \Theta_3^{k,j} := C_3^{k,j-1}(V^{k,j-1} - V^{k,j}), \\ \Theta_4^{k,j} := \rho^k(F^{k,j-1} - F^{k,j}) + \rho^k(\widehat{Y}^{k,j-1} - \widehat{Y}^{k,j}) + C_4^{k,j-1}(Y^{k,j-1} - Y^{k,j}), \\ \Theta_5^{k,j} := C_5^{k,j-1}(F^{k,j-1} - F^{k,j}), \\ \Theta_6^{k,j} := C_6^{k,j-1}(\widehat{Y}^{k,j-1} - \widehat{Y}^{k,j}). \end{cases} \tag{20}$$

The overall algorithmic framework of the proposed PAM method is summarized in Algorithm 2, and its convergence analysis is presented in the next subsection.

```
Algorithm 2 PAM Method for (9)–(11)
  Input:
  Let $(W^{1,0},U^{1,0},V^{1,0},Y^{1,0},F^{1,0},\widehat{Y}^{1,0})$ be any initialization;
  For $k \geq 2$, set $(W^{k,0}, U^{k,0}, V^{k,0}, Y^{k,0}, F^{k,0}, \widehat{Y}^{k,0}) = (W^{k-1}, U^{k-1}, V^{k-1}, Y^{k-1}, F^{k-1}, \widehat{Y}^{k-1})$;

  Output: $(W^k,U^k,V^k,Y^k,F^k,\widehat{Y}^k)$;
  Step 1: Reiterate on $j$ until $\|\Theta^{k,j}\|_\infty \leq \epsilon_k$, where $\Theta^{k,j}$ is defined by (20);
     1. Compute $W^{k,j}$ by (14);
     2. Compute $U^{k,j}$ by (15);
     3. Compute $V^{k,j}$ by (16);
     4. Compute $Y^{k,j}$ by (17);
     5. Compute $F^{k,j}$ by (18);
     6. Compute $\widehat{Y}^{k,j}$ by (19);
  Step 2: Set
     $$(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k) := (W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j})$$
  and $\Theta^k := \Theta^{k.j}$.
```

4.2 Convergence Analysis for Algorithm 2

To simplify the notation, we introduce $T := (W, U, V, Y, F, \widehat{Y})$ and denote the Lagrangian function at the k-th outer iteration as $L_k(T) := L(W, U, V, Y, F, \widehat{Y}, \overline{\Lambda}^k; \rho^k)$. In this part, we aim to establish the convergence for Algorithm 2. Specifically, we show that the solution set of (9)–(11) is nonempty, thereby confirming that Algorithm 1 is well defined when utilizing Algorithm 2 to solve the subproblems in Step 1.

We first claim that for each $k \in \mathbb{N}$, $\Theta^{k,j} := (\Theta_1^{k,j}, \Theta_2^{k,j}, \Theta_3^{k,j}, \Theta_4^{k,j}, \Theta_5^{k,j}, \Theta_6^{k,j})$ defined by (20) satisfies

$$\Theta^{k,j} \in \partial L_k(T^{k,j}), \quad \forall\, j \in \mathbb{N}.$$

To establish this, we begin by observing that the augmented Lagrangian $L_k(T)$ admits the decomposition

$$L_k(T) = f_1(W) + f_2(Y) + f_3(U) + f_4(V) + f_5(F) + f_6(\widehat{Y}) + g_k(T), \tag{21}$$

where the individual components are defined as

$$\begin{aligned} f_1(W) :=& \gamma\|W\|_F^2,\ \ f_2(Y) := \operatorname{Tr}(Y^\intercal L Y),\ \ f_3(U) := \alpha\|U\|_{2,1}, \\ f_4(V) :=& \beta\|V\|_{2,1},\ \ f_5(F) := \delta_{\mathcal{S}_2}(F),\ \ f_6(\widehat{Y}) := \delta_{\mathcal{S}_1}(\widehat{Y}), \end{aligned}$$

and

$$\begin{aligned} g_k(T) :=& \langle \overline{\Lambda}_1^k, Y - X^\intercal W - U\rangle + \langle \overline{\Lambda}_2^k, V - W\rangle + \langle \overline{\Lambda}_3^k, Y - F\rangle + \langle \overline{\Lambda}_4^k, \widehat{Y} - Y\rangle \\ &+ \frac{\rho^k}{2}(\|Y - X^\intercal W - U\|_F^2 + \|\widehat{Y} - Y\|_F^2 + \|V - W\|_F^2 + \|Y - F\|_F^2). \end{aligned}$$

A direct calculation shows that $\Theta^{k,j} := (\Theta_1^{k,j}, \Theta_2^{k,j}, \Theta_3^{k,j}, \Theta_4^{k,j}, \Theta_5^{k,j}, \Theta_6^{k,j})$ can be represented in terms of the partial derivatives of $g := g_k$ as follows

$$\begin{cases} \Theta_1^{k,j} = -\nabla_W g(W^{k,j}, U^{k,j-1}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) - C_1^{k,j-1}(W^{k,j} - W^{k,j-1}) + \nabla_W g(T^{k,j}), \\ \Theta_2^{k,j} = -\nabla_U g(W^{k,j}, U^{k,j}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) - C_2^{k,j-1}(U^{k,j} - U^{k,j-1}) + \nabla_U g(T^{k,j}), \\ \Theta_3^{k,j} = -\nabla_V g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) - C_3^{k,j-1}(V^{k,j} - V^{k,j-1}) + \nabla_V g(T^{k,j}), \\ \Theta_4^{k,j} = -\nabla_Y g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j-1}, \widehat{Y}^{k,j-1}) - C_4^{k,j-1}(Y^{k,j} - Y^{k,j-1}) + \nabla_Y g(T^{k,j}), \\ \Theta_5^{k,j} = -\nabla_F g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j-1}) - C_5^{k,j-1}(F^{k,j} - F^{k,j-1}) + \nabla_F g(T^{k,j}), \\ \Theta_6^{k,j} = -\nabla_{\widehat{Y}} g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j}) - C_6^{k,j-1}(\widehat{Y}^{k,j} - \widehat{Y}^{k,j-1}) + \nabla_{\widehat{Y}} g(T^{k,j}). \end{cases} \tag{22}$$

Moreover, based on $(W^{k,j-1}, U^{k,j-1}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1})$ and invoking 8.8(c) from Rockafellar and Wets (2009), the first-order optimality conditions for the subproblems in (14)–(19) can be written as:

$$\begin{cases} \nabla_W g(W^{k,j}, U^{k,j-1}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) + \nabla f_1(W^{k,j}) + C_1^{k,j-1}(W^{k,j} - W^{k,j-1}) = O, \\ \xi^{k,j} + \nabla_U g(W^{k,j}, U^{k,j}, V^{k,j-1}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) + C_2^{k,j-1}(U^{k,j} - U^{k,j-1}) = O, \\ \zeta^{k,j} + \nabla_V g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j-1}, F^{k,j-1}, \widehat{Y}^{k,j-1}) + C_3^{k,j-1}(V^{k,j} - V^{k,j-1}) = O, \\ \nabla f_2(Y^{k,j}) + \nabla_Y g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j-1}, \widehat{Y}^{k,j-1}) + C_4^{k,j-1}(Y^{k,j} - Y^{k,j-1}) = O, \\ \vartheta^{k,j} + \nabla_F g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j-1}) + C_5^{k,j-1}(F^{k,j} - F^{k,j-1}) = O, \\ \varsigma^{k,j} + \nabla_{\widehat{Y}} g(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j}) + C_6^{k,j-1}(\widehat{Y}^{k,j} - \widehat{Y}^{k,j-1}) = O, \end{cases} \tag{23}$$

where $\xi^{k,j} \in \partial f_3(U^{k,j})$, $\zeta^{k,j} \in \partial f_4(V^{k,j})$, $\vartheta^{k,j} \in \partial f_5(F^{k,j})$, and $\varsigma^{k,j} \in \partial f_6(\widehat{Y}^{k,j})$. By combining (22) and (23), we obtain

$$\begin{cases} \Theta_1^{k,j} = \nabla f_1(W^{k,j}) + \nabla_W g(T^{k,j}), \\ \Theta_2^{k,j} = \xi^{k,j} + \nabla_U g(T^{k,j}), \\ \Theta_3^{k,j} = \zeta^{k,j} + \nabla_V g(T^{k,j}), \\ \Theta_4^{k,j} = \nabla f_2(Y^{k,j}) + \nabla_Y g(T^{k,j}), \\ \Theta_5^{k,j} = \vartheta^{k,j} + \nabla_F g(T^{k,j}), \\ \Theta_6^{k,j} = \varsigma^{k,j} + \nabla_{\widehat{Y}} g(T^{k,j}). \end{cases}$$

Therefore, Proposition 2.1 in Attouch et al. (2010), we conclude that for each $k \in \mathbb{N}$,

$$\Theta^{k,j} \in \partial L_k(W^{k,j}, U^{k,j}, V^{k,j}, Y^{k,j}, F^{k,j}, \widehat{Y}^{k,j}), \quad \forall\, j \in \mathbb{N},$$

completing the proof of the claim.

The following theorem establishes the convergence of Algorithm 2, thereby guaranteeing that Step 1 of Algorithm 1 is well defined. The proof is based on the general convergence result in Attouch et al. (2013, Theorem 6.2).

Theorem 6 *In Algorithm 1, set parameters $r > 1$ and $\rho^1 > 0$. Then, for each $k \in \mathbb{N}$, the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ generated by Algorithm 2 converges, and*

$$\|\Theta^{k,j}\|_\infty \to 0 \quad \text{as } j \to \infty.$$

Proof It is known that the sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are semi-algebraic, and so are their corresponding indicator functions (Bolte et al., 2014). In addition, both the quadratic functions $\boldsymbol{x}^\intercal L\boldsymbol{x}$ and the norm function $\|\boldsymbol{x}\|_p$ (where p is rational) are semi-algebraic. Using the fact that composition of semi-algebraic functions is semi-algebraic, it follows that L_k is a semi-algebraic function. Moreover, the semi-algebraic functions are known to satisfy the Kurdyka-Lojasiewicz (KL) property (Bolte et al., 2014, Appendic), implying that L_k is a KL function. From the expression of L_k in (21) , the following properties can be observed: (i) Each f_i $(i = 1, 2, 3, 4, 5, 6)$ is a proper lower semicontinuous function; (ii) g_k is a C^1-function with locally Lipschitz continuous gradient. Therefore, from Attouch et al. (2013, Theorem 6.2), it follows that to establish the convergence of the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ generated by Algorithm 2, it suffices to demonstrate that for each $k \in \mathbb{N}$, the function L_k is bounded below and the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ is bounded.

Clearly, the five terms in L_k, namely f_1, f_3, f_4, f_5 and f_6, are bounded below with a lower bound of 0 and coercive. The remaining residual terms are given by

$$\begin{aligned} f_2(Y) + g_k(W,U,V,Y,F,\widehat{Y}) =& \mathrm{Tr}(Y^\intercal LY) + \langle\overline{\Lambda}_1^k, Y - X^\intercal W - U\rangle + \langle\overline{\Lambda}_2^k, V - W\rangle + \langle\overline{\Lambda}_3^k, Y - F\rangle \\ &+ \langle\overline{\Lambda}_4^k, \widehat{Y} - Y\rangle + \frac{\rho^k}{2}(\|\widehat{Y} - Y\|_F^2 + \|V - W\|_F^2 + \|Y - F\|_F^2 \\ &+ \|Y - X^\intercal W - U\|_F^2). \end{aligned}$$

We can rewrite this expression as

$$f_2(Y) + g_k(W,U,V,Y,F,\widehat{Y}) = g_{1,k}(W,U,Y) + g_{2,k}(W,V,Y,F,\widehat{Y}),$$

where

$$g_{1,k}(W,U,Y) = \mathrm{Tr}(Y^\intercal LY) + \langle\overline{\Lambda}_1^k, Y - X^\intercal W - U\rangle + \frac{\rho^k}{2}\|Y - X^\intercal W - U\|_F^2$$

and

$$\begin{aligned} g_{2,k}(W,V,Y,F,\widehat{Y}) =& \langle\overline{\Lambda}_2^k, V - W\rangle + \langle\overline{\Lambda}_3^k, Y - F\rangle + \langle\overline{\Lambda}_4^k, \widehat{Y} - Y\rangle + \frac{\rho^k}{2}(\|\widehat{Y} - Y\|_F^2 + \|V - W\|_F^2 \\ &+ \|Y - F\|_F^2). \end{aligned}$$

By direct calculations, we have

$$g_{1,k}(W,U,Y) = \mathrm{Tr}(Y^\intercal LY) + \frac{\rho^k}{2}\left\|Y - X^\intercal W - U + \frac{\overline{\Lambda}_1^k}{\rho^k}\right\|_F^2 - \frac{\rho^k}{2}\left\|\frac{\overline{\Lambda}_1^k}{\rho^k}\right\|_F^2$$

and

$$\begin{aligned} g_{2,k}(W,V,Y,F,\widehat{Y}) =& \frac{\rho^k}{2}\Big[\Big\|V - W + \frac{\overline{\Lambda}_2^k}{\rho^k}\Big\|_F^2 + \Big\|Y - F + \frac{\overline{\Lambda}_3^k}{\rho^k}\Big\|_F^2 + \Big\|\widehat{Y} - Y + \frac{\overline{\Lambda}_4^k}{\rho^k}\Big\|_F^2 - \Big(\Big\|\frac{\overline{\Lambda}_2^k}{\rho^k}\Big\|_F^2 \\ &+ \Big\|\frac{\overline{\Lambda}_3^k}{\rho^k}\Big\|_F^2 + \Big\|\frac{\overline{\Lambda}_4^k}{\rho^k}\Big\|_F^2\Big)\Big]. \end{aligned}$$

Thus, both $g_{1,k}(W, U, Y)$ and $g_{2,k}(W, V, Y, F, \widehat{Y})$ are bounded below. Consequently, the function L_k defined by (21) is bounded below and coercive for each $k \in \mathbb{N}$.

Next, we prove the boundedness of the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ for each $k \in \mathbb{N}$ by contradiction. Assume, for contradiction, that there exists $k_0 \in \mathbb{N}$ such that the sequence $\{T^{k_0,j}\}_{j\in\mathbb{N}}$ is unbounded, i.e., $\lim_{j\to\infty} \|T^{k_0,j}\| = \infty$. Then,due to the coerciveness of the function L_{k_0}, it follows that the sequence $\{L_{k_0}(T^{k_0,j})\}_{j\in\mathbb{N}}$ must diverge to infinity.

We now define the following terms:

$$\begin{aligned}
\widetilde{L}^1_{k_0,j} &= L_{k_0}(W^{k_0,j+1}, U^{k_0,j}, V^{k_0,j}, Y^{k_0,j}, F^{k_0,j}, \widehat{Y}^{k_0,j}),\\
\widetilde{L}^2_{k_0,j} &= L_{k_0}(W^{k_0,j+1}, U^{k_0,j+1}, V^{k_0,j}, Y^{k_0,j}, F^{k_0,j}, \widehat{Y}^{k_0,j}),\\
\widetilde{L}^3_{k_0,j} &= L_{k_0}(W^{k_0,j+1}, U^{k_0,j+1}, V^{k_0,j+1}, Y^{k_0,j}, F^{k_0,j}, \widehat{Y}^{k_0,j}),\\
\widetilde{L}^4_{k_0,j} &= L_{k_0}(W^{k_0,j+1}, U^{k_0,j+1}, V^{k_0,j+1}, Y^{k_0,j+1}, F^{k_0,j}, \widehat{Y}^{k_0,j}),\\
\widetilde{L}^5_{k_0,j} &= L_{k_0}(W^{k_0,j+1}, U^{k_0,j+1}, V^{k_0,j+1}, Y^{k_0,j+1}, F^{k_0,j+1}, \widehat{Y}^{k_0,j}).
\end{aligned}$$

Applying the update rules (14)–(19), we obtain the following inequalities

$$\begin{aligned}
&\widetilde{L}^1_{k_0,j} + \frac{C_1^{k_0,j}}{2}\|W^{k_0,j+1} - W^{k_0,j}\|_F^2 \le L_{k_0}(T^{k_0,j});\\
&\widetilde{L}^2_{k_0,j} + \frac{C_2^{k_0,j}}{2}\|U^{k_0,j+1} - U^{k_0,j}\|_F^2 \le \widetilde{L}^1_{k_0,j};\\
&\widetilde{L}^3_{k_0,j} + \frac{C_3^{k_0,j}}{2}\|V^{k_0,j+1} - V^{k_0,j}\|_F^2 \le \widetilde{L}^2_{k_0,j};\\
&\widetilde{L}^4_{k_0,j} + \frac{C_4^{k_0,j}}{2}\|Y^{k_0,j+1} - Y^{k_0,j}\|_F^2 \le \widetilde{L}^3_{k_0,j};\\
&\widetilde{L}^5_{k_0,j} + \frac{C_5^{k_0,j}}{2}\|F^{k_0,j+1} - F^{k_0,j}\|_F^2 \le \widetilde{L}^4_{k_0,j};\\
&L_{k_0}(T^{k_0,j+1}) + \frac{C_6^{k_0,j}}{2}\|\widehat{Y}^{k_0,j+1} - \widehat{Y}^{k_0,j}\|_F^2 \le \widetilde{L}^5_{k_0,j}.
\end{aligned}$$

Summing all the inequalities above yields:

$$L_{k_0}(T^{k_0,j+1}) + \frac{C}{2}\|T^{k_0,j+1} - T^{k_0,j}\|_F^2 \le L_{k_0}(T^{k_0,j}),\ j \in \mathbb{N}.$$

This implies that $\{L_{k_0}(T^{k_0,j})\}_{j\in\mathbb{N}}$ is a nonincreasing sequence. Since it was previously assumed to diverge to infinity, we reach a contradiction. Hence, the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ must be bounded for each $k \in \mathbb{N}$.

From the above analysis, and by invoking Attouch et al. (2013, Theorem 6.2), we conclude that for each $k \in \mathbb{N}$, the sequence $\{T^{k,j}\}_{j\in\mathbb{N}}$ has finite length, i.e., $\sum_{j=1}^{\infty} \|T^{k,j+1} - T^{k,j}\|_F < \infty$, and converges to a critical point of L_k. Finally, based on the expression of $\Theta^{k,j}$ in (20), it follows that for each $k \in \mathbb{N}$, we have $\|\Theta^{k,j}\|_\infty \to 0$ as $j \to \infty$. This completes the proof. ■

5. Convergence Analysis of Our Inexact ALM Method

In this section, the convergence of the inexact ALM method presented in Algorithm 1 is discussed. We begin by reformulating problem (7) using vector notation. Let $\boldsymbol{x} \in \mathbb{R}^{2dc+4nc}$ denote the column vector obtained by concatenating the columns of the matrices W, U, V, Y, F, and $\widehat{Y}$, i.e.,

$$\boldsymbol{x} := \mathrm{Vec}([W|U|V|Y|F|\widehat{Y}]). \tag{24}$$

Define the set $\Gamma := \{\boldsymbol{x} \in \mathbb{R}^{2dc+4nc} \mid p(\boldsymbol{x}) = \mathbf{0}\}$, where $p(\boldsymbol{x}) \in \mathbb{R}^{\frac{c(c+1)}{2}}$ is the vectorized form of the lower triangular part of the symmetric matrix $\widehat{Y}^\intercal \widehat{Y} - I_c$. Using these notations, problem (7) can be rewritten as:

$$\min_{\boldsymbol{x} \in \Gamma} f(\boldsymbol{x}) \quad \text{s.t. } h(\boldsymbol{x}) = \mathbf{0}, \tag{25}$$

where $h(\boldsymbol{x}) \in \mathbb{R}^{3nc+dc}$ is defined by

$$h(\boldsymbol{x}) := \mathrm{Vec}([Y - X^\intercal W - U | V - W | Y - F | \widehat{Y} - Y]).$$

The objective function $f(\boldsymbol{x})$ is given by:

$$f(\boldsymbol{x}) := \sum_{j=1}^{c} Y_{:j}^\intercal L Y_{:j} + \gamma \|W_{:j}\|_2^2 + \delta_{\mathcal{S}'}(F_{:j}) + \sum_{i=1}^{n} \alpha \|U_{i:}\|_2 + \sum_{i=1}^{d} \beta \|V_{i:}\|_2,$$

where $Y_{:j}, W_{:j}$, and $F_{:j}$ denote the column vectors of Y, W and F, respectively, and $U_{i:}$ and $V_{i:}$ denote the row vectors of U and V, respectively. Additionally, $\mathcal{S}' = \{\boldsymbol{y} \in \mathbb{R}^n \mid \mathbf{0} \le \boldsymbol{y} \le \mathbf{1}\}$.

Let $\boldsymbol{\lambda} := \mathrm{Vec}([\Lambda_1|\Lambda_2|\Lambda_3|\Lambda_4])$, $m_1 := 3nc + dc$, and $m_2 := \frac{c(c+1)}{2}$. For notational convenience, let h_i for $i \in [m_1]$ and p_i for $i \in [m_2]$ denote the i-th components of h and p, respectively. The corresponding augmented Lagrangian function associated with problem (25) is then defined as

$$L(\boldsymbol{x}, \boldsymbol{\lambda}; \rho) := f(\boldsymbol{x}) + \sum_{i=1}^{m_1} \lambda_i h_i(\boldsymbol{x}) + \frac{\rho}{2} \sum_{i=1}^{m_1} (h_i(\boldsymbol{x}))^2,$$

where $\boldsymbol{x} \in \Gamma$. To facilitate the subsequent convergence analysis, we introduce the definition of a stationary point for problem (7), which is formulated based on the following function:

$$\mathcal{F}(W, U, V, Y, F, \widehat{Y}) := \mathrm{Tr}(Y^\intercal L Y) + \alpha \|U\|_{2,1} + \beta \|V\|_{2,1} + \gamma \|W\|_F^2 + \delta_{\mathcal{S}_2}(F).$$

Definition 7 *We say that $T^* := (W^*, U^*, V^*, Y^*, F^*, \widehat{Y}^*)$ is a stationary point of problem (7) if there exist $Z^* \in \partial \mathcal{F}(T^*)$, $\Lambda^* := (\Lambda_1^*, \Lambda_2^*, \Lambda_3^*, \Lambda_4^*) \in \mathbb{R}^{n \times c} \times \mathbb{R}^{d \times c} \times \mathbb{R}^{n \times c} \times \mathbb{R}^{n \times c}$, and $\Upsilon^* \in \mathbb{R}^{c \times c}$ such that the following conditions are satisfied*

$$\begin{cases} Z^* + \nabla\big(\langle \Lambda_1^*, Y^* - X^\intercal W^* - U \rangle + \langle \Lambda_2^*, V^* - W^* \rangle + \langle \Lambda_3^*, Y^* - F^* \rangle \\ \qquad + \langle \Lambda_4^*, \widehat{Y}^* - Y^* \rangle + \langle \Upsilon^*, \widehat{Y}^{*\intercal} \widehat{Y}^* - I_c \rangle\big) = O, \\ Y^* = F^*, U^* = Y^* - X^\intercal W^*, V^* = W^*, Y^* = \widehat{Y}^*, \widehat{Y}^{*\intercal} \widehat{Y}^* = I_c. \end{cases}$$

Clearly, $(W^*, U^*, V^*, Y^*, F^*, \widehat{Y}^*)$ is a stationary point of the problem (7) if and only if the vector $\boldsymbol{x}^* := \mathrm{Vec}([W^*|U^*|V^*|Y^*|F^*|\widehat{Y}^*])$ is a stationary point of the problem (25). That is, there exist $\boldsymbol{\theta}^* \in \partial f(\boldsymbol{x}^*)$, $\boldsymbol{\lambda}^* \in \mathbb{R}^{m_1}$, and $\boldsymbol{\upsilon}^* \in \mathbb{R}^{m_2}$ such that the following system holds

$$\begin{cases} \boldsymbol{\theta}^* + \sum_{i=1}^{m_1} \lambda_i^* \nabla h_i(\boldsymbol{x}^*) + \sum_{i=1}^{m_2} \upsilon_i^* \nabla p_i(\boldsymbol{x}^*) = \boldsymbol{0}, \\ h(\boldsymbol{x}^*) = \boldsymbol{0}, \\ p(\boldsymbol{x}^*) = \boldsymbol{0}. \end{cases} \tag{26}$$

Before proceeding, we claim that for any $\bar{\boldsymbol{x}} \in \Gamma$, the gradient vectors of h and p, i.e., $\{\nabla h_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_1} \cup \{\nabla p_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_2}$ are linearly independent. Specifically, we show that

$$\sum_{i=1}^{m_1} \bar{y}_i \nabla h_i(\bar{\boldsymbol{x}}) + \sum_{i=1}^{m_2} \hat{y}_i \nabla p_i(\bar{\boldsymbol{x}}) = \boldsymbol{0}$$

if and only if $\bar{y}_i = \hat{y}_i = 0$ for all i. This linear independence is a key component in establishing the convergence of Algorithm 1.

Lemma 8 *Suppose that $\bar{\boldsymbol{x}} \in \Gamma$. Then, the gradient vectors $\{\nabla h_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_1} \cup \{\nabla p_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_2}$ are linearly independent, where the functions h and p are defined as in (25).*

Proof For convenience, we define the block diagonal matrices $A \in \mathbb{R}^{dc \times nc}$, $B \in \mathbb{R}^{dc \times dc}$ and $C \in \mathbb{R}^{nc \times nc}$ as follows:

$$A = \begin{bmatrix} -X & & & \\ & -X & & \\ & & \ddots & \\ & & & -X \end{bmatrix}, \quad B = \begin{bmatrix} I_d & & & \\ & I_d & & \\ & & \ddots & \\ & & & I_d \end{bmatrix}, \quad C = \begin{bmatrix} I_n & & & \\ & I_n & & \\ & & \ddots & \\ & & & I_n \end{bmatrix}.$$

Based on the structure of $\boldsymbol{x}$ defined in (24), the Jacobian matrices $\boldsymbol{J}h(\boldsymbol{x})$ and $\boldsymbol{J}p(\boldsymbol{x})$ can be expressed as

$$\boldsymbol{J}h(\boldsymbol{x}) = \left[\begin{array}{c|c|c|c} A & -B & O_{dc\times nc} & O_{dc\times nc} \\ \hline -C & O_{nc\times dc} & O_{nc\times nc} & O_{nc\times nc} \\ \hline O_{dc\times nc} & B & O_{dc\times nc} & O_{dc\times nc} \\ \hline C & O_{nc\times dc} & C & -C \\ \hline O_{nc\times nc} & O_{nc\times dc} & -C & O_{nc\times nc} \\ \hline O_{nc\times nc} & O_{nc\times dc} & O_{nc\times nc} & C \end{array}\right]^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{J}p(\boldsymbol{x}) = \left[\begin{array}{c} O_{dc\times m_2} \\ \hline O_{nc\times m_2} \\ \hline O_{dc\times m_2} \\ \hline O_{nc\times m_2} \\ \hline O_{nc\times m_2} \\ \hline G(\boldsymbol{x}) \end{array}\right]^{\mathsf{T}},$$

where $G(\boldsymbol{x})$ is defined in (27) and $\{\widehat{Y}_{:i}\}_{i=1}^{c}$ denote the column vectors of $\widehat{Y}$.

Since $\boldsymbol{x} \in \Gamma$, the column vectors $\{\widehat{Y}_{:i}\}_{i=1}^{c}$ are orthogonal to each other. Therefore, the columns of $G(\boldsymbol{x})$ are mutually orthogonal. Note that the first $3nc + 2dc$ columns of $\boldsymbol{J}p(\boldsymbol{x})$ constitute a zero matrix. Consequently, based on the structures of $\boldsymbol{J}h(\boldsymbol{x})$ and $\boldsymbol{J}p(\boldsymbol{x})$, it follows that the gradient vectors $\{\nabla h_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_1} \cup \{\nabla p_i(\bar{\boldsymbol{x}})\}_{i=1}^{m_2}$ are linearly independent for any $\boldsymbol{x} \in \Gamma$. ■

Let $\{T^k\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 1. We now demonstrate that this sequence is bounded, which ensures the existence of at least one convergent subsequence of $\{T^k\}_{k\in\mathbb{N}}$.

$$G(\boldsymbol{x}) = \begin{bmatrix} 2\widehat{Y}_{:1} & \widehat{Y}_{:2} & \widehat{Y}_{:3} & \cdots & \widehat{Y}_{:c} & O_{n\times1} & O_{n\times1} & \cdots & O_{n\times1} & & O_{n\times1} & O_{n\times1} & O_{n\times1} \\ O_{n\times1} & \widehat{Y}_{:1} & O_{n\times1} & \cdots & O_{n\times1} & 2\widehat{Y}_{:2} & \widehat{Y}_{:3} & \cdots & \widehat{Y}_{:c} & & \vdots & \vdots & \vdots \\ O_{n\times1} & O_{n\times1} & \widehat{Y}_{:1} & \cdots & O_{n\times1} & O_{n\times1} & \widehat{Y}_{:2} & \cdots & O_{n\times1} & \cdots & O_{n\times1} & O_{n\times1} & \vdots \\ \vdots & \vdots & \ddots & \ddots & O_{n\times1} & \vdots & \ddots & \ddots & \vdots & & 2\widehat{Y}_{:c-1} & \widehat{Y}_{:c} & O_{n\times1} \\ O_{n\times1} & O_{n\times1} & \cdots & O_{n\times1} & \widehat{Y}_{:1} & O_{n\times1} & \cdots & O_{n\times1} & \widehat{Y}_{:2} & & O_{n\times1} & \widehat{Y}_{:c-1} & 2\widehat{Y}_{:c} \end{bmatrix} \tag{27}$$

Lemma 9 *Suppose that the parameters r and ρ^1 in Algorithm 1 are chosen such that $r > 1$ and $\rho^1 > 0$. Then, the sequence $\{T^k\}_{k\in\mathbb{N}}$ is bounded, and consequently, it contains at least one convergent sequence.*

Proof It follows from (10) that the sequence $\{F^k\}_{k\in\mathbb{N}}$ and $\{\widehat{Y}^k\}_{k\in\mathbb{N}}$ are bounded. The first four partial subdifferentials of L in (11) guarantee the existence of $\xi^k \in \alpha\partial\|U\|_{2,1}$, $\zeta^k \in \beta\partial\|V\|_{2,1}$, and $M^k = (M_1^k, M_2^k, M_3^k, M_4^k) \in \mathbb{R}^{d\times c}\times\mathbb{R}^{n\times c}\times\mathbb{R}^{d\times c}\times\mathbb{R}^{n\times c}$ such that

$$\begin{cases} M_1^k = 2\gamma W^k - X\overline{\Lambda}_1^k - \overline{\Lambda}_2^k - \rho^k X(Y^k - X^\intercal W^k - U^k) - \rho^k(V^k - W^k), \\ M_2^k = \xi^k - \overline{\Lambda}_1^k - \rho^k(Y^k - X^\intercal W^k - U^k), \\ M_3^k = \zeta^k + \overline{\Lambda}_2^k + \rho^k(V^k - W^k), \\ M_4^k = 2LY^k + \overline{\Lambda}_1^k + \overline{\Lambda}_3^k - \overline{\Lambda}_4^k + \rho^k(Y^k - X^\intercal W^k - U^k) + \rho^k(Y^k - F^k) - \rho^k(\widehat{Y}^k - Y^k), \end{cases} \tag{28}$$

where $\|M^k\|_\infty \le \epsilon^k$. By adding M_2^k and M_4^k, we obtain

$$M_2^k + M_4^k = \xi^k + (2L + 2\rho^k I)Y^k + \overline{\Lambda}_3^k - \overline{\Lambda}_4^k - \rho^k F^k - \rho^k\widehat{Y}^k,$$

which implies

$$Y^k = [2(L + \rho^k I)]^{-1}(M_2^k + M_4^k - \xi^k - \overline{\Lambda}_3^k + \overline{\Lambda}_4^k + \rho^k F^k + \rho^k\widehat{Y}^k). \tag{29}$$

Let $D\mathrm{diag}(\sigma_1, \cdots, \sigma_n)D^\intercal$ be the SVD of the symmetric and positive semi-definite matrix L. Then, from (29), we obtain

$$\begin{aligned} Y^k =& D\mathrm{diag}\left(\frac{1}{2(\sigma_1+\rho^k)}, \frac{1}{2(\sigma_2+\rho^k)}, \cdots, \frac{1}{2(\sigma_n+\rho^k)}\right) D^\intercal(M_2^k + M_4^k - \xi^k - \overline{\Lambda}_3^k + \overline{\Lambda}_4^k) \\ &+ D\mathrm{diag}\left(\frac{\rho^k}{2(\sigma_1+\rho^k)}, \frac{\rho^k}{2(\sigma_2+\rho^k)}, \cdots, \frac{\rho^k}{2(\sigma_n+\rho^k)}\right) D^\intercal(F^k + \widehat{Y}^k). \end{aligned} \tag{30}$$

Since $\{\rho^k\}_{k\in\mathbb{N}}$ is non-decreasing and $2(L + \rho^1 I) \succ 0$, it follows that $2(L + \rho^k I) \succ 0$ for all $k \in \mathbb{N}$, implying $2(\sigma_i + \rho^k) > 0$ for all $i = 1, 2, \cdots, n$. Therefore, we have

$$\begin{cases} 0 < \frac{1}{2(\sigma_i+\rho^k)} \le \frac{1}{2(\sigma_i+\rho^1)} < +\infty, \quad i = 1, 2, \cdots, n; \\ 0 < \frac{\rho^k}{2(\sigma_i+\rho^k)} \le \frac{1}{2}, \quad i = 1, 2, \cdots, n. \end{cases} \tag{31}$$

Note that the sequence $\{\xi^k\}_{k\in\mathbb{N}}$, $\{M_2^k\}_{k\in\mathbb{N}}$, $\{M_4^k\}_{k\in\mathbb{N}}$, $\{\overline{\Lambda}_3^k\}_{k\in\mathbb{N}}$, and $\{\overline{\Lambda}_4^k\}_{k\in\mathbb{N}}$ are all bounded. Then, from (30) and (31), we conclude that the sequence $\{Y^k\}_{k\in\mathbb{N}}$ is bounded.

Similarly, from the expressions of M_3^k and M_2^k in (28), it follows that the sequences $\{\rho^k(V^k - W^k)\}_{k\in\mathbb{N}}$ and $\{\rho^k(Y^k - X^\intercal W^k - U^k)\}_{k\in\mathbb{N}}$ are bounded. Therefore, from the expression for M_1^k in (28), we deduce that the sequence $\{W^k\}_{k\in\mathbb{N}}$ is also bounded. Since $\rho^k \geq \rho^1$, this implies that the sequences $\{V^k - W^k\}_{k\in\mathbb{N}}$ and $\{Y^k - X^\intercal W^k - U^k\}_{k\in\mathbb{N}}$ are bounded as well. Hence, the sequences $\{V^k\}_{k\in\mathbb{N}}$ and $\{U^k\}_{k\in\mathbb{N}}$ are bounded. In conclusion, the sequence $\{(W^k, U^k, V^k, Y^k, F^k, \widehat{Y}^k)\}_{k\in\mathbb{N}}$ is bounded. The proof is complete. ■

As noted in Remark 4, the vectorized representation of the normal cone $\partial\delta_{\mathcal{S}_1}(T) = N_{\mathcal{S}_1}(T)$ for $T \in \mathcal{S}_1$ can be written as

$$N_\Gamma(\boldsymbol{x}) = \{\boldsymbol{J}p(\boldsymbol{x})^\intercal \boldsymbol{\upsilon} | \boldsymbol{\upsilon} \in \mathbb{R}^{m_2}\} = \Big\{ \sum_{i=1}^{m_2} \upsilon_i \nabla p_i(\boldsymbol{x}) \big| \boldsymbol{\upsilon} \in \mathbb{R}^{m_2} \Big\}$$

for $\boldsymbol{x} \in \Gamma$, since the linear independence of the gradient vectors $\{\nabla p_i(\boldsymbol{x})\}_{i=1}^{m_2}$ (see Lemma 8) ensures $\operatorname{rank}(\boldsymbol{J}p(\boldsymbol{x})) = m_2$. Therefore, based on the well-definedness of (11) and its vectorized formulation, we can obtain a sequence of solutions $\boldsymbol{x}^k$ such that, for each $k \in \mathbb{N}$, there exist vectors $\boldsymbol{\theta}^k \in \partial f(\boldsymbol{x}^k)$ and $\boldsymbol{\upsilon}^k \in \mathbb{R}^{m_2}$ satisfying

$$\|\boldsymbol{\theta}^k + \sum_{i=1}^{m_1} (\overline{\lambda}_i^k + \rho^k h_i(\boldsymbol{x}^k)) \nabla h_i(\boldsymbol{x}^k) + \sum_{i=1}^{m_2} \upsilon_i^k \nabla p_i(\boldsymbol{x}^k)\|_\infty \leq \epsilon^k, \tag{32}$$

where $\overline{\boldsymbol{\lambda}}^k = \operatorname{Vec}([\overline{\Lambda}_1^k | \overline{\Lambda}_2^k | \overline{\Lambda}_3^k | \overline{\Lambda}_4^k])$. In what follows, leveraging Lemma 8 and Lemma 9, we demonstrate that any accumulation point $\boldsymbol{x}^*$ of the sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$, i.e., the sequence obtained by vectorizing $\{T^k\}_{k\in\mathbb{N}}$, is a stationary point of problem (25).

Theorem 10 *Let $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 1, and let $\boldsymbol{x}^*$ be an accumulation point, i.e., there exists a subsequence $\{\boldsymbol{x}^k\}_{k\in\mathcal{K}}$ with $\mathcal{K} \subseteq \mathbb{N}$ such that $\lim_{k\in\mathcal{K}} \boldsymbol{x}^k = \boldsymbol{x}^*$. Then $\boldsymbol{x}^*$ is a stationary point of problem (25).*

Proof We first demonstrate that $\boldsymbol{x}^*$ satisfies the feasibility conditions of problem (25), i.e., $h(\boldsymbol{x}^*) = 0$ and $p(\boldsymbol{x}^*) = 0$. From (10), we know that $p(\boldsymbol{x}^k) = 0$ for all $k \in \mathbb{N}$. The continuity of function p then implies $p(\boldsymbol{x}^*) = 0$, so $\boldsymbol{x}^* \in \Gamma$. To show $h(\boldsymbol{x}^*) = 0$, we consider two cases depending on whether the penalty sequence $\{\rho^k\}_{k\in\mathbb{N}}$ is bounded.

Case I. Suppose the sequence $\{\rho^k\}_{k\in\mathbb{N}}$ is bounded. By the update rule of penalty parameter in (12), ρ^k eventually stabilizes after some k_0, implying that $\|h(\boldsymbol{x}^{k+1})\|_\infty \leq \tau \|h(\boldsymbol{x}^k)\|_\infty$ for all $k \geq k_0$, where $\tau \in [0, 1)$. By standard continuity arguments, we conclude that $h(\boldsymbol{x}^*) = 0$.

Case II. Now assume that the sequence $\{\rho^k\}_{k\in\mathbb{N}}$ is unbounded. From (32), for each $k \in \mathcal{K}$, there exists a vector $\boldsymbol{d}^k$ satisfying $\|\boldsymbol{d}^k\|_\infty \leq \epsilon^k$, where $\epsilon^k \downarrow 0$, such that

$$\boldsymbol{\theta}^k + \sum_{i=1}^{m_1} (\overline{\lambda}_i^k + \rho^k h_i(\boldsymbol{x}^k)) \nabla h_i(\boldsymbol{x}^k) + \sum_{i=1}^{m_2} \upsilon_i^k \nabla p_i(\boldsymbol{x}^k) = \boldsymbol{d}^k \tag{33}$$

for some $\boldsymbol{\theta}^k \in \partial f(\boldsymbol{x}^k)$. Dividing both sides of (33) by ρ^k, we obtain

$$\sum_{i=1}^{m_1} \Big(\frac{\overline{\lambda}_i^k}{\rho^k} + h_i(\boldsymbol{x}^k) \Big) \nabla h_i(\boldsymbol{x}^k) + \sum_{i=1}^{m_2} \hat{\upsilon}_i^k \nabla p_i(\boldsymbol{x}^k) = \frac{\boldsymbol{d}^k - \boldsymbol{\theta}^k}{\rho^k}, \tag{34}$$

where $\hat{\boldsymbol{\upsilon}}^k = \frac{\boldsymbol{\upsilon}^k}{\rho^k}$. Define

$$H(\boldsymbol{x})^\intercal := [\boldsymbol{J}h(\boldsymbol{x})^\intercal \ \boldsymbol{J}p(\boldsymbol{x})^\intercal]$$

and

$$\boldsymbol{\eta}^k := \Big(\frac{\overline{\lambda}_1^k}{\rho^k} + h_1(\boldsymbol{x}^k), \cdots, \frac{\overline{\lambda}_{m_1}^k}{\rho^k} + h_{m_1}(\boldsymbol{x}^k), \hat{\upsilon}_1^k, \cdots, \hat{\upsilon}_{m_2}^k\Big)^\intercal.$$

Then, (34) can be rewritten as

$$H(\boldsymbol{x}^k)^\intercal \boldsymbol{\eta}^k = \frac{\boldsymbol{d}^k - \boldsymbol{\theta}^k}{\rho^k}.$$

Since the gradients ∇h_i and ∇p_i are continuous, it follows that $H(\boldsymbol{x}^k) \to H(\boldsymbol{x}^*)$ as $k \to \infty$ with $k \in \mathcal{K}$. Given that $\boldsymbol{x}^* \in \Gamma$ and by Lemma 8, the gradient vectors $\{\nabla h_i(\boldsymbol{x}^*)\}_{i=1}^{m_1} \cup \{\nabla p_i(\boldsymbol{x}^*)\}_{i=1}^{m_2}$ are linearly independent, so $H(\boldsymbol{x}^*)$ has full rank. Thus, $H(\boldsymbol{x}^k)H(\boldsymbol{x}^k)^\intercal \to H(\boldsymbol{x}^*)H(\boldsymbol{x}^*)^\intercal \succ 0$. By the fact that eigenvalues of a symmetric matrix vary continuously with its matrix values Horn and Johnson (2012), for sufficiently large $k \in \mathcal{K}$, $H(\boldsymbol{x}^k)H(\boldsymbol{x}^k)^\intercal$ is nonsingular, and we obtain

$$\boldsymbol{\eta}^k = [H(\boldsymbol{x}^k)H(\boldsymbol{x}^k)^\intercal]^{-1} H(\boldsymbol{x}^k) \frac{\boldsymbol{d}^k - \boldsymbol{\theta}^k}{\rho^k}.$$

Since the function f is convex, the set $\cup_{\boldsymbol{x} \in \mathcal{X}} \partial f(\boldsymbol{x})$ is bounded whenever $\mathcal{X}$ is bounded. A proof of this result can be found in Bertsekas (1997, Proposition B.24(b)). By letting $\mathcal{X} = \{\boldsymbol{x}^k\}_{k \in \mathcal{K}}$ and invoking Lemma 9, which establishes the boundedness of $\{\boldsymbol{x}^k\}_{k \in \mathcal{K}}$, we conclude that the sequence $\{\boldsymbol{\theta}^k\}_{k \in \mathcal{K}}$ is also bounded. Moreover, since $\|\boldsymbol{d}^k\|_\infty \le \epsilon^k \downarrow 0$ and the penalty sequence $\{\rho^k\}_{k \in \mathbb{N}}$ is unbounded, it follows that for $k \in \mathcal{K}$,

$$\boldsymbol{\eta}^k \to 0, \quad \text{as } k \to \infty.$$

Finally, noting the boundedness of the multipliers $\{\overline{\boldsymbol{\lambda}}^k\}_{k \in \mathcal{K}}$, we conclude that $h_i(\boldsymbol{x}^*) = 0$ for all i. Hence, $h(\boldsymbol{x}^*) = 0$, as desired.

Next, we show that $\boldsymbol{x}^*$ is a stationary point. Since $\{\boldsymbol{\theta}^k\}_{k \in \mathcal{K}}$ is bounded, there exists a subsequence $\mathcal{K}_1 \subseteq \mathcal{K}$ such that $\lim_{k \in \mathcal{K}_1} \boldsymbol{\theta}^k = \boldsymbol{\theta}^*$. Also, since $\lim_{k \in \mathcal{K}_1} \boldsymbol{x}^k = \boldsymbol{x}^*$ and $\boldsymbol{\theta}^k \in \partial f(\boldsymbol{x}^k)$, by the closedness property of the subdifferential we obtain

$$\boldsymbol{\theta}^* \in \partial f(\boldsymbol{x}^*).$$

By the update rule $\lambda_i^{k+1} = \overline{\lambda}_i^k + \rho^k h_i(\boldsymbol{x}^k)$ for all i, (32) implies that for all $k \in \mathcal{K}_1$,

$$\boldsymbol{\theta}^k + \sum_{i=1}^{m_1} \lambda_i^{k+1} \nabla h_i(\boldsymbol{x}^k) + \sum_{i=1}^{m_2} \upsilon_i^k \nabla p_i(\boldsymbol{x}^k) = \boldsymbol{d}^k, \tag{35}$$

where $\boldsymbol{\theta}^k \in \partial f(\boldsymbol{x}^k)$ and $\boldsymbol{d}^k$ satisfies $\|\boldsymbol{d}^k\|_\infty \le \epsilon^k \downarrow 0$. Define

$$\boldsymbol{\omega}^k := (\lambda_1^{k+1}, \cdots, \lambda_{m_1}^{k+1}, \upsilon_1^k, \cdots, \upsilon_{m_2}^k)^\intercal. \tag{36}$$

Then, (35) becomes

$$H(\boldsymbol{x}^k)^\intercal \boldsymbol{\omega}^k = \boldsymbol{d}^k - \boldsymbol{\theta}^k.$$

As before, for sufficiently large $k \in \mathcal{K}_1$, the matrix $H(\boldsymbol{x}^k)H(\boldsymbol{x}^k)^\intercal$ is nonsingular. Consequently,

$$\boldsymbol{\omega}^k = [H(\boldsymbol{x}^k)H(\boldsymbol{x}^k)^\intercal]^{-1}H(\boldsymbol{x}^k)(\boldsymbol{d}^k - \boldsymbol{\theta}^k).$$

Taking the limit as $k \to \infty$ with $k \in \mathcal{K}_1$, we obtain

$$\boldsymbol{\omega}^k \to \boldsymbol{\omega}^* = -[H(\boldsymbol{x}^*)H(\boldsymbol{x}^*)^\intercal]^{-1}H(\boldsymbol{x}^*)\boldsymbol{\theta}^*.$$

Passing to the limit in (35) as $k \to \infty$ with $k \in \mathcal{K}_1$, we arrive at

$$\boldsymbol{\theta}^* + \sum_{i=1}^{m_1} \lambda_i^* \nabla h_i(\boldsymbol{x}^*) + \sum_{i=1}^{m_2} \upsilon_i^* \nabla p_i(\boldsymbol{x}^*) = \mathbf{0},$$

where λ_i^* and υ_i^* are the respective limits of λ_i^{k+1} and υ_i^k. The existence of $\boldsymbol{\omega}^*$ guarantees the existence of these multipliers. Therefore, (26) holds, and $\boldsymbol{x}^*$ is a stationary point of problem (25). ■

By combining Lemma 9 and Theorem 10, we immediately obtain the following convergence result in matrix form.

Theorem 11 *Let the parameters satisfy $r > 1$ and $\rho^1 > 0$ in Algorithm 1, and let $\{T^k\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 1. Then, the accumulation point set of $\{T^k\}_{k\in\mathbb{N}}$ is nonempty, and every accumulation point is a stationary point of the original problem (7).*

6. Experiment Study

In this section, we conduct numerical experiments to demonstrate the effectiveness of Algorithm 1. All experiments are performed using MATLAB (2020a) on a laptop of 16G of memory and Inter Core i7 2.3Ghz CPU . We compare our method against several state-of-the-art unsupervised feature selection methods on ten real-world datasets. These datasets include one speech signal dataset (Isolet[*]), two microarray datasets (lung[¶], 9_Tumors[†]), and seven image datasets (UMIST[‡], JAFFE(Lyons et al. (1999)),ORL[¶],COIL20[¶],YALEB[§], CMU-PIE(Sim et al. (2003)), warpPIE10P[¶]). Table 1 summarizes the details of these ten benchmark datasets used in the experiments. In addition to verifying the effectiveness of our method on the above datasets, we also provide analyses on stability, robustness, feature recovery capabilities, and parameter sensitivity. Moreover, we report the computational complexity of the iterative algorithms.

Methods to Compare. We compare the performance of Algorithm 1 with the following unsupervised feature selection methods:

- **Baseline**: All of the original features are adopted.
- **MaxVar** (Krzanowski, 1987): Features corresponding to the maximum variance are selected to obtain the expressive features.

[*] https://jundongl.github.io/scikit-feature/datasets.html

[†] https://github.com/primekangkang/Genedata

[‡] https://cs.nyu.edu/ roweis/data.html

[§] http://www.cad.zju.edu.cn/home/dengcai/Data/data.html

Table 1: Dataset Description

Dataset	Size	# of Features	# of Classes
Isolet	1560	617	26
UMIST	575	644	20
JAFFE	213	676	10
ORL	400	1024	40
COIL20	1440	1024	20
YALEB	2414	1024	38
CMU-PIE	832	1024	68
warpPIE10P	210	2420	10
lung	203	3312	5
9_Tumors	60	5726	9

- **LS** (He et al., 2005): Laplacian Score, in which features are selected with the most consistency with Gaussian Laplacian matrix.
- **SPEC** (Zhao and Liu, 2007): According to spectrum of the graph to select features.
- **MCFS** (Cai et al., 2010): Multi-cluster feature selection, it uses the l_1-norm to regularize the feature selection process as a spectral information regression problem.
- **NDFS** (Li et al., 2012): Nonnegative discriminative feature selection, which addressed feature discriminability and correlation simultaneously.
- **UDFS** (Yang et al., 2011a): Unsupervised discriminative feature selection incorporated discriminative analysis as well as $l_{2,1}$-norm minimization, which is formalized as a unified framework.
- **UDPFS** (Wang et al., 2022): Unsupervised discriminative projection for feature selection to select discriminative features by conducting fuzziness learning and sparse learning simultaneously.

Evaluation Measures. Similar to previous work, and based on the attained clustering results and the ground truth information, we evaluate the performance of the unsupervised feature selection methods using two widely utilized evaluation metrics, i.e., clustering ACCuracy (ACC) and Normalized Mutual Information (NMI) (Yang et al., 2011a). The higher the ACC and NMI are, the better the clustering performance is.

Given one sample $\boldsymbol{x}_i \in \{\boldsymbol{x}_i\}_{i=1}^n$, denote y_i be the ground truth label and l_i be the predicted clustering label. The ACC is defined as

$$\text{ACC} = \frac{1}{n}\sum_{i=1}^{n} \delta(y_i, map(l_i)),$$

where $\delta(a,b) = 1$ if $a = b$; otherwise $\delta(a,b) = 0$, and $map(l_i)$ is the permutation mapping function that maps each cluster label l_i to the equivalent label from the data set.

Given two random variables P and Q, P denotes the true labels and Q represents clustering results. The NMI of P and Q is defined as:

$$\mathrm{NMI}(P, Q) = \frac{I(P;Q)}{\sqrt{H(P)H(Q)}},$$

where $I(P;Q)$ is the mutual information between P and Q, $H(P)$ and $H(Q)$ are the entropies of P and Q, respectively.

Experiment Setting. In our experiments, the parameters of Algorithm 1 are set as follows:

$$\tau = 0.99, \quad r = 1.01, \quad \rho^1 = c/2, \quad \overline{\Lambda}_1^1 = \overline{\Lambda}_3^1 = \overline{\Lambda}_4^1 = O_{n\times c}, \quad \overline{\Lambda}_2^1 = O_{d\times c},$$

and

$$\overline{\Lambda}_{N,min} = -100E, \;\; \overline{\Lambda}_{N,max} = 100E \quad (N = 1, 2, 3, 4), \quad \epsilon^k = 0.995^k \;\; (k \in \mathbb{N}).$$

The parameters in Algorithm 2 are set as $\underline{C} = C_i^{k,j} = \overline{C} = 0.5$. The iteration is terminated if the iteration number exceeds 20.

Among the compared methods, several hyper-parameters need to be specified in advance. We fix number of neighboring parameter $k = 5$ for LS, SPEC, MCFS, UDFS, NDFS, and our proposed method. Since the UDPFS method involves some finely tuned parameters in its original implementation, we follow the grid-search strategy specified in the original paper to tune its parameters. For all other methods, we perform grid search over a common range of values $\{10^{-6}, 10^{-5}, 10^{-4}, \cdots, 10^4, 10^5, 10^6\}$, and report the best clustering results from the optimal parameters for all the methods. Since the optimal number of selected features is unknown, we set different number of selected features for all datasets, tuning the selected feature number from $\{50, 100, 150, 200, 250, 300\}$. After completing the feature selection process, we use K-means algorithm to cluster the data into c groups. Since the initial center points have a significant impact on the performance of K-means algorithm, we conduct K-means algorithm 20 times repeatedly with random initialization to report the mean and standard deviation values of ACC and NMI.

In the following subsections, we systematically evaluate the proposed method in terms of clustering performance, stability, robustness, feature recovery capability, parameter sensitivity, and computational complexity.

6.1 Clustering Performance Analysis

The experiments results of different methods on the datasets are summarized in Tables 2 and 3. The best results are highlighted in red. It can be observed that our method performs the best on the vast majority of datasets. Even on a few datasets where it does not achieve the optimal results, its performance is still quite good, especially on the 9_Tumors dataset, where it achieves the second-best performance.

Considering the averaging of all numerical results, it is evident that the performance of our method surpasses that of other state-of-the-art methods. The superior performance of our method can be attributed to the following aspects: Firstly, we employ a technique similar to NDFS to establish the model, i.e., simultaneously learning the pseudo class label indicators and the feature selection matrix. However, unlike NDFS, we utilize $l_{2,1}$-norm to characterize the linear loss function between features and pseudo labels, while also considering the prevention of overfitting. Secondly, unlike the commonly used processing methods, we apply a convergent algorithm that can simultaneously optimize all variables in the feature selection model. In the previous section, we have proven the convergence property of our algorithm. Since the iterative sequence of our algorithm converges to stationary points, it achieves better results than other methods.

In addition, we conduct a sensitivity analysis on the number of clusters c. For datasets with more than 10 clusters including "Isolet,UMIST,ORL,COIL20,YALEB, CMU-PIE", we set the perturbation to ± 5 clusters; for datasets with fewer than 10 clusters including "JAFFE, warpPIE10p, lung, 9_Tumors", we set the perturbation to ± 1 cluster. Under these perturbation conditions, we study the feature selection performance of the proposed method. The corresponding results are shown in the last two columns of Tables 2 and 3. "Ours$_-$" represents the results corresponding to the number of clusters minus 1 or minus 5 for the respective dataset, while "Ours$_+$" represents the results corresponding to the number of clusters plus 1 or plus 5. When the performance under cluster perturbation is better than other methods, we highlight the results in green.

As can be seen from the results, the accuracy of cluster estimation has minimal impact on experimental performance. For instance, in the "UMIST, ORL, COIL20, CMU-PIE, warpPIE10P, 9_Tumors, YALEB" datasets, the results with inaccurate cluster estimations were better than those with accurate cluster numbers. This indicates that the proposed method does not heavily rely on knowing the exact number of clusters in advance. Moreover, the performance results showed minimal fluctuations when the number of clusters for small datasets changed from cluster number -1 to cluster number $+1$, and for large datasets, from cluster number -5 to cluster number $+5$. Overall, these results demonstrate that our feature selection method is highly robust and stable with respect to cluster number estimation, maintaining good performance even when the number of clusters is inaccurate.

Table 2: Clustering results (ACC±STD%) of different feature selection methods on the real-world datasets.

Dataset	All features	LS	Maxvar	MCFS	NDFS	SPEC	UDFS	UDPFS	Ours	Ours$_-$	Ours$_+$
Isolet	60.9±2.1	58.7±1.5	56.9±2.3	64.5±4.3	61.6±4.4	56.5±3.0	57.8± 3.1	49.7±2.3	65.8±3.9	64.7±3.9	63.7±3.8
UMIST	41.8±2.7	45.9±2.9	45.8±2.8	46.3±3.6	47.8±2.6	47.9±3.0	45.7± 2.5	45.9±2.8	54.0±3.7	55.2±3.7	52.4±3.0
JAFFE	72.5±9.2	74.0±7.6	67.3±5.8	78.8±9.1	74.7±6.9	76.9±7.2	75.8± 8.5	75.7±7.8	80.4±6.6	78.9±8.0	79.9±9.9
ORL	49.7±3.2	49.9±2.4	50.8±1.4	55.7±3.7	50.5±3.0	51.4± 2.2	53.3±4.1	52.0±3.7	52.9±3.4	53.3±3.1	53.7±2.7
COIL20	62.7±3.1	62.2±1.9	61.4±1.6	63.0±3.7	58.7± 4.1	65.5±3.8	60.2±4.2	57.5±3.6	61.6±3.8	62.3±4.4	63.0±3.5
YALEB	10.0±0.6	11.4±0.6	9.6±0.3	17.1±0.6	21.9± 0.6	8.6±0.2	14.5±0.7	11.4±0.3	26.3±1.1	26.5±1.0	26.9±1.0
CMU-PIE	25.1±1.1	25.1±1.1	30.8±1.2	26.5±1.0	34.6± 1.5	25.1±1.1	29.3±1.1	24.5±0.7	35.1±1.6	34.8±1.6	35.2±1.5
warpPIE10P	25.7±1.4	30.2±0.4	28.5±2.6	33.2±2.7	40.0± 3.3	27.1±0.7	44.6±4.3	46.8±3.1	50.1±5.5	52.6±4.3	54.2±5.0
lung	65.0±3.6	74.9±0.2	68.0±9.4	77.6±11.0	63.3±6.9	64.1±7.9	72.3±10.9	78.2±8.1	82.4±7.9	83.5±5.1	81.6±3.6
9_Tumors	40.8±3.7	42.3±2.6	41.2±2.6	42.4±3.6	44.0±3.7	35.8±2.4	43.0± 4.3	46.8±4.1	44.1±4.1	43.3±3.6	45.0±4.3
Mean	45.4±3.1	47.5±2.1	46.0±3.0	50.5±4.3	49.7±3.7	45.9±3.2	49.7±4.4	48.9±3.7	55.3±4.2	55.5±3.9	55.6±3.8

In our proposed method, the processes of spectral clustering and feature selection are conducted simultaneously, thus influencing each other. Considering the spectral clustering process itself, we observe that all features in a high-dimensional dataset X, including noise and irrelevant features, can impact the pseudo class labels Y. Therefore, when building

Table 3: Clustering results (NMI±STD%) of different feature selection methods on the real-world datasets.

Dataset	All features	LS	Maxvar	MCFS	NDFS	SPEC	UDFS	UDPFS	Ours	Ours_{-}	Ours_{+}
Isolet	75.7±0.8	73.2±0.9	74.8±1.3	77.7±1.7	77.1± 2.2	72.4±1.1	74.7±1.8	68.0±1.1	80.5±1.3	78.1±1.5	79.0±1.2
UMIST	62.3±2.3	63.9±1.8	63.5±1.5	66.7±1.9	66.5±1.8	65.2±2.0	65.9±1.8	66.2±1.7	70.5±1.2	70.2±2.0	69.6±1.5
JAFFE	80.0±5.7	79.4±7.0	70.3±4.2	83.4±5.0	83.1±3.4	82.8±3.8	84.8±3.7	84.8±3.6	89.0±5.2	88.6±4.0	88.6±4.5
ORL	70.0±1.7	71.1± 1.3	70.7± 2.1	76.8±1.8	73.2±1.9	71.4± 1.3	74.7±1.6	74.2±1.5	74.9±1.7	74.3±1.4	74.4±1.1
COIL20	77.1±1.3	72.5±1.1	71.9±0.7	76.5±1.7	74.0± 1.6	75.3±1.6	75.4±1.3	73.7±1.3	76.3±2.3	76.5±2.4	76.7±1.7
YALEB	14.2±0.7	18.4±1.0	13.1±0.4	30.6±0.7	37.6±0.7	12.7±0.2	23.9±0.8	20.2±0.5	42.7±0.6	42.4±0.5	43.1±0.7
CMU-PIE	52.2±0.7	52.2±0.7	58.9±0.6	53.7±0.7	62.1± 0.5	52.2±0.7	56.8±0.9	52.3±0.8	62.6±0.6	62.9±0.8	63.1±0.6
warpPIE10P	25.3±3.0	29.6±2.9	24.7±2.8	35.9±3.4	40.7±3.1	25.3±3.0	48.2±2.8	51.0±4.7	53.0±3.7	56.1±3.1	60.7±3.7
lung	51.6±1.9	53.1± 0.5	57.8± 3.9	67.5±7.0	53.0±3.5	52.5± 5.6	61.3±5.8	64.9±3.1	69.0±4.4	68.5±5.4	67.6±5.8
9_Tumors	39.5±3.1	41.0±2.3	40.2±2.5	41.1±2.7	44.7±4.5	34.5±2.4	44.1±4.3	48.5±4.7	44.8±3.2	44.9±3.0	46.7±3.2
Mean	54.8±2.1	55.4±2.0	54.6±2.0	61.0±2.7	61.2±2.3	54.4±2.2	61.0±2.5	60.4±2.3	66.3±2.4	66.3±2.4	67.0±2.4

the model, we establish a linear regression model between the pseudo class labels Y and the high-dimensional data X through the feature matrix W, and introduce Frobenius norm regularization to prevent overfitting. This regularization method effectively prevents the model from excessively relying on noise or irrelevant features when the number of features is large, thereby affecting the effectiveness of spectral clustering and ultimately leading to inaccurate feature selection. To illustrate the necessity of incorporating the overfitting prevention term, we conduct an ablation study. By setting the regularization parameter γ to 0 and using the same experimental setup as before, we employ a grid search strategy to test the clustering performance of the proposed method on various datasets without the overfitting prevention term. The result can be shown in Table 4.

Experimental results show that without the Frobenius norm regularization, the clustering performance of the model significantly decreases. This indicates that regularization plays a crucial role in the feature selection process, effectively enhancing the model's robustness and accuracy. Through regularization, the model can reduce sensitivity to noise and irrelevant features while enhancing focus on useful features. This improves the clustering performance and feature selection accuracy, enabling the model to maintain good performance when dealing with high-dimensional and complex datasets. Therefore, incorporating the overfitting prevention term is essential for improving the effectiveness of our method.

Table 4: The clustering results of the proposed feature selection method on the real-world datasets without regularization terms to prevent overfitting.

Dataset	Isolet	UMIST	JAFFE	ORL	COIL20	YALEB	CMU-PIE	warpPIE10p	lung	9_Tumors	**Mean**
ACC±STD%	65.8±3.9	52.8±2.7	79.1±8.7	51.9±3.1	61.5±4.8	24.4±1.0	34.0±1.3	43.1±4.1	65.4±8.1	41.2±4.4	51.9±4.2
NMI±STD%	80.4±1.3	70.2± 1.8	89.0± 5.2	74.6±1.6	76.2± 2.0	40.3±0.8	62.1±0.7	45.0±3.0	55.8±4.0	43.6± 4.0	63.7±2.4

6.2 Stability Analysis of Unsupervised Feature Selection Methods

In this subsection, we evaluate the stability (Nogueira et al., 2018) of all unsupervised feature selection methods. Specifically, we investigate whether the feature selection method proposed in this paper, as well as the comparison methods, can consistently select the same subset of features across different subsets of the original data. We fix the number of selected features to 300. To ensure the accuracy of the evaluation, we employ the following procedure to achieve and validate this goal.

First, we adopt the k-fold cross-validation technique. Specifically, we divide the data into k subsets and independently run the feature selection method on each subset. After independently running the feature selection method on each subset, we compare the features selected in each run. The stability of feature selection will be evaluated using a consistency metric. In this paper, we use the Jaccard similarity coefficient (Leskovec et al., 2020) to measure the similarity of feature selection across different cross-validation folds. The Jaccard similarity coefficient is used to calculate the similarity between two sets and is given by the formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the feature subsets selected in two different runs. Based on this, we construct a symmetric Jaccard similarity matrix $J \in \mathbb{R}^{3\times 3}$ with diagonal elements set to 0, and compute the average value of the non-zero elements in J.

We compare the method proposed in this paper with other unsupervised feature selection methods to evaluate their performance. This comparative analysis will help assess the effectiveness and stability of the proposed method. In the experiment, k is set to 3, and all the parameters follow the previous settings. We repeat the above process 5 times and record the average results. Finally, we report the best results of all the algorithms using different parameters. The detailed results can be found in the Table 5. The best results are highlighted in red. It can be observed that our method demonstrates higher feature selection consistency across all datasets compared to other comparison methods, demonstrating greater stability in feature selection.

Table 5: Stability analysis of different feature selection methods on the real-world datasets.

Dataset	LS	Maxvar	MCFS	NDFS	SPEC	UDFS	UDPFS	Ours
Isolet	85.2%	91.0%	33.4%	84.8%	87.1%	76.3%	89.5%	91.9%
UMIST	60.6%	86.1%	32.2%	66.3%	82.0%	63.7%	47.4%	88.1%
JAFFE	67.0%	80.0%	38.1%	81.8%	84.8%	63.0%	53.6%	91.7%
ORL	56.0%	60.9%	22.0%	85.2%	79.8%	53.4%	38.3%	92.5%
COIL20	74.1%	87.3%	23.6%	79.4%	90.1%	42.4%	46.1%	92.7%
YALEB	83.8%	85.2%	17.7%	67.7%	85.3%	67.4%	76.3%	87.6%
CMU-PIE	74.3%	72.3%	20.0%	80.5%	87.5%	51.3%	62.6%	90.1%
warpPIE10P	47.8%	53.2%	17.9%	21.9%	59.3%	26.6%	35.1%	61.1%
lung	31.4%	62.9%	22.6%	83.1%	61.8%	32.9%	22.8%	94.1%
9_Tumors	6.5%	63.0%	52.3%	76.3%	13.6%	59.4%	5.0%	86.3%
Mean	58.7%	74.2%	28.0%	72.7%	73.1%	53.6%	47.7%	87.6%

6.3 Stability Analysis of Iterative Algorithms

We will now demonstrate that our algorithm is more stable than other iterative algorithms including: UDPFS (Wang et al., 2022), NDFS (Li et al., 2012) and UDFS (Yang et al., 2011a). Following the notation in Li et al. (2012); Yang et al. (2011a); Wang et al. (2022), we denote the feature selection matrix as W in these methods and define

$$\eta = \frac{\|W_{k+1} - W_k\|_F}{\|W_k - W_{k-1}\|_F},$$

where W_k represents the k-th iterative point. To thoroughly assess the stability of our algorithm, we randomly initialize cluster indicator matrix Y and feature selection matrix W 20 times. Under the parameter setting of the optimal results obtained by corresponding method, we record the average results of η. The experimental results are presented in Figure 1.

It can be observed that the value of η for other three methods fluctuates irregularly, whereas our method stabilizes after fewer iterations and consistently remains below 1. Further analysis shows that, with an increasing number of iterations k, the norm $\|W_{k+1} - W_k\|_F$ gradually decreases in our method. This indicates that our method ensures the sequence $\{W_k\}_{k\in\mathbb{N}}$ changes regularly according to the iterative rules, with the "distance" between the adjacent points W_{k+1} and W_k progressively reducing. This observation aligns with the convergence theory we previously proved. In constrast, since the values of η of UDPFS, NDFS and UDFS exhibit irregular fluctuations, indicating that their iterative sequences $\{W_k\}_{k\in\mathbb{N}}$ "jump" unpredictably and lack a clear convergence trend. Therefore, our algorithm is more stable during the iterative process.

6.4 Feature Recovery Capabilities

In this subsection, we investigate the feature recovery capabilities of the proposed feature selection method using synthetic datasets with predefined relevant and irrelevant features. While the experiments in Section 6.2 focused on the stability of selected features, our goal here is to directly evaluate the method's ability to recover the true underlying features. To this end, we construct synthetic datasets where the ground-truth relevant features are known a prior and are mixed with additional noisy features. This controlled setup enables a quantitative assessment of how accurately the method identifies the true features and to what extent it mistakenly selects irrelevant ones.

To facilitate the following description, we define each synthetic dataset $\hat{X}$ to have the size $n_{sample} \times n_{feature}$, where $n_{feature} = n_{true} + n_{noise}$. Here, n_{true} denotes the number of true features, n_{noise} denotes the number of noise features, and $\mathbf{1}_{n_{true}}$ represents a row vector of ones with length n_{true}. We generate synthetic datasets of various sizes with $n_{sample} \in \{200, 1000\}$, $n_{true} \in \{200, 300, 400, 500\}$, and $n_{noise} \in \{200, 400, 800, 1200\}$, resulting in 32 different combinations summarized in Table 6. All Gaussian clusters are generated with equal sample size.

Synthetic data with independent noise. We first construct synthetic Gaussian mixture datasets following the idea in (Dy and Brodley, 2004), where relevant features are sampled from Gaussian clusters and irrelevant features are independently drawn from Gaussian noise. Specifically, for each cluster k, all relevant features are independently sampled from a Gaussian distribution with the mean corresponding to the k-th cluster center and a standard deviation of 1. We use 4 clusters for $n_{sample} = 200$ and 5 clusters for $n_{sample} = 1000$, with cluster centers set to $[\pm 2, \pm 4] \times \mathbf{1}_{n_{true}}$ and $[\pm 2, \pm 4, 6] \times \mathbf{1}_{n_{true}}$, respectively. This yields the true feature matrix $X_{true} \in \mathbb{R}^{n_{sample} \times n_{true}}$. The irrelevant features $X_{noise} \in \mathbb{R}^{n_{sample} \times n_{noise}}$ are independently generated from the standard Gaussian distribution $\mathcal{N}(0, 1)$, without any dependency on the relevant features. Concatenating them gives the data $X_{fea} = [X_{true}, X_{noise}]$. To eliminate any positional bias, the columns of X_{fea} are randomly shuffled to produce the final dataset $\hat{X}$.

We apply the proposed method to the synthetic dataset $\hat{X}$ with the number of selected features fixed at n_{true}, and evaluate its feature recovery capability by computing the proportion of correctly identified true features. Since the Frobenius norm regularization plays a key role in the selection process, we fix the regularization parameter $\gamma = 100$, and set $\alpha = \beta = 1e-6$. We evaluate all 32 combinations of n_{sample}, n_{true}, n_{noise}, and normalize each sample to have unit ℓ_2 norm, and repeat every experiment 20 times with different random seeds. The averaged results are reported in Table 6. The results show that as the number of noise features increases, the feature selection accuracy of the proposed method remains unaffected. The method consistently achieves perfect recovery by successfully identifying 100% of the true features. This excellent performance is attributed to the simplicity of the synthetic setting, where irrelevant features are completely independent of the relevant ones, yielding a clear separation between the relevant and irrelevant dimensions and enabling perfect recovery.

Table 6: The proportion of relevant features selected by the proposed method across synthetic datasets of different sizes under independent noise feature interference.

n_{sample}	n_{true}	n_{noise}			
		200	400	800	1200
200	200	100%	100%	100%	100%
	300	100%	100%	100%	100%
	400	100%	100%	100%	100%
	500	100%	100%	100%	100%
1000	200	100%	100%	100%	100%
	300	100%	100%	100%	100%
	400	100%	100%	100%	100%
	500	100%	100%	100%	100%

Table 7: The proportion of relevant features selected by the proposed method across synthetic datasets of different sizes under correlated noise feature interference.

n_{sample}	n_{true}	n_{noise}			
		200	400	800	1200
200	200	92.0%	85.3%	72.8%	64.5%
	300	94.5%	89.7%	81.0%	73.6%
	400	95.7%	91.9%	85.2%	79.4%
	500	96.5%	93.3%	87.5%	82.5%
1000	200	92.2%	85.9%	73.5%	64.5%
	300	94.7%	89.9%	80.9%	74.1%
	400	96.0%	92.1%	85.3%	79.1%
	500	96.6%	93.6%	88.0%	83.2%

Synthetic data with correlated noise. To make the task more challenging, we further introduce a fraction of irrelevant features that are linearly correlated with the relevant features, thereby introducing redundancy. The relevant features are generated in the same way as described above. For the irrelevant features, we first sample Gaussian noise features from $\mathcal{N}(0, 1)$ and then randomly select 10% of them to create perturbed copies of the relevant features. Specifically, each selected noise feature $\hat{\boldsymbol{y}} \in \mathbb{R}^{n_{sample}}$ is constructed as $\hat{\boldsymbol{y}} = \boldsymbol{y} + \epsilon$, where $\boldsymbol{y}$ is a relevant feature and $\epsilon \sim \mathcal{N}(0, 1)$ represents independent Gaussian disturbance. As a result, while most irrelevant features remain independent, a small subset exhibit strong correlations with the relevant ones. The correlated noise features are concatenated with the relevant features to form $X_{fea} = [X_{true}, X_{noise}]$. As in the previous case, we randomly shuffle the columns of X_{fea} to obtain the final dataset $\hat{X}$ used in this experiment.

We use the same experimental settings as before, repeat each experiment 20 times, and report the averaged results. As shown in Table 7, recovery performance improves as the number of relevant features increases but degrades as the number of irrelevant features grows. For instance, when $n_{sample} = 200$, $n_{true} = 200$, and $n_{noise} = 200$, among which 10% (i.e., 20 features) are perturbed copies of relevant features, the total feature dimension is 400. The number of selected features is fixed to n_{true}. Under this configuration, the proposed method selects 200 features, among which 184 correspond to the ground-truth relevant ones, yielding a recovery rate of 92%. When the number of noise features increases to 1200, the recovery rate decreases to 64.5%, where 129 of the 200 selected features being truly relevant and 71 being noise. This outcome can be attributed to the structure of the feature set: among the total 1400 features, only 200 are truly relevant, while the remaining 1200 are noise, including 120 perturbed copies of the relevant features. These perturbed noise features are highly correlated with the true features and constitute a substantial portion of the candidate set (200 features). Notably, only 71 noise features are selected, which is smaller than the total number of perturbed copies (120), demonstrating that the proposed method is capable of filtering out redundant correlated features. Unlike the independent noise case, the recovery rate is no longer 100%. This degradation arises because correlated noise obscures the distinction between relevant and irrelevant features. Occasionally, the method selects perturbed copies rather than the exact relevant features, leading to a reduced success rate.

6.5 Robustness Analysis

In this subsection, we summarize the main results of our robustness analysis. To evaluate the robustness of our model under different noise levels, we add Gaussian noise with a mean of 0 and a variance of σ^2 to the dataset. Due to the varying ranges of element values in the dataset, directly adding the uniform noise levels may lead to uneven effects across different datasets. In particular, this might result in excessive noise in some datasets, masking key features, while in others, the noise might be too minimal to effectively disturb the data, thereby affecting the algorithm's performance and the accuracy of the results. To address this issue, we employ a method that adjusts the noise level based on the data range. This ensures that the noise has a relatively consistent impact on data points of different scales within their respective ranges, thus preventing excessive or insufficient noise effects on certain datasets. Specifically, we first calculate the minimum and maximum values of each dataset

X to determine the data range $R := \max(X) - \min(X)$. Next, we choose a proportion factor ϱ and set the noise standard deviation σ as a certain proportion of the data range R, i.e., $\sigma = \varrho \times R$. Then, we generate a Gaussian noise matrix $N \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ and add it to the original dataset, i.e., $X' = X + N$. In order to make a fair comparison, we conduct the experiments under the parameter setting of the optimal results obtained by each method for the chosen dataset. Meanwhile, in order to avoid the influence of the randomness of noise in a single experiment, we uniformly do ten experiments for each noise level, and then average the results as the final result.

Figure 2 and Figure 3 show the robustness of the iterative methods here considered on the all datasets with different levels of noise. As shown in Figures, on COIL20, the competing methods show relatively flat trends in ACC and NMI, whereas our method not only follows the same trend but consistently outperforms them at every noise level. On seven other datasets (Isolet, UMIST, JAFFE, YaleB, lung, CMU-PIE, and warpPIE10p), competing methods suffer substantial declines in ACC and NMI as noise increases, while our method maintains high performance throughout. Visually, only on ORL and 9_Tumors does our method exhibit a pronounced step-wise decline in performance. Notably, on 9_Tumors, the only significant drop occurs when noise increases from 5% to 10%, after which performance remains stable-even up to 25% noise. More importantly, although our noise-free ACC and NMI on these two datasets are lower than those of some competing methods, once noise is introduced, our method consistently outperforms all alternatives.

To assess robustness more precisely, we introduce the performance-retention ratio:

$$R(\varrho) = \frac{A(\varrho)}{A(0)},$$

where $A(\varrho)$ denotes the ACC or AMI at noise level $\varrho \in \{5\%, 10\%, 15\%, 20\%, 25\%\}$ added in datasets and $A(0)$ is the corresponding noise-free value (refer to Table 2 and Table 3).

We compare all methods based on their performance-retention ratios $R(\varrho)$ at the highest noise level ($\varrho = 25\%$). As shown in Table 8, our method consistently achieves the highest performance-retention ratio across all datasets, with the best results being highlighted in red. On average, our method retains over 80% of its performance, and remaining above this threshold on the vast majority of datasets and thereby clearly demonstrating strong robustness under heavy corruption. In contrast, competing methods achieve an average retention below 72%, and exceed 80% on only a limited number of datasets.

In conclusion, our method exhibits better robustness, both visually and quantitatively. The experimental results show that as the noise level increases, our proposed method can maintain stable performance across most datasets, indicating its strong resistance to data perturbations. We test our method on multiple datasets, covering different fields and application scenarios. Regardless of the noise level, our method consistently outperforms other methods on all datasets. In particular, the newly introduced metric, known as performance-retention ratio, provides clear evidence that our method offers greater noise resistance compared to competing methods.

6.6 Parameter Sensitivity Analysis

Like many other feature selection algorithms, our proposed method requires several parameters α, β, γ to be set in advance. To analyze the sensitivity of these parameters, we

Table 8: Comparison of performance-retention ratios across methods.

Dataset	Ours		NDFS		UDFS		UDPFS	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Isolet	80.1%	85.5%	66.6%	74.7%	77.9%	82.9%	75.9%	82.9%
UMIST	89.1%	89.6%	62.8%	63.6%	80.3%	81.8%	82.8%	83.4%
JAFFE	90.5%	90.4%	59.2%	54.3%	77.3%	76.1%	78.6%	78.8%
ORL	56.7%	73.4%	41.0%	57.1%	54.0%	66.9%	55.8%	68.1%
COIL20	97.4%	95.9%	95.9%	93.8%	92.0%	91.8%	95.8%	94.3%
YALEB	83.7%	84.3%	54.8%	50.8%	65.5%	58.2%	76.3%	66.8%
CMU-PIE	73.5%	83.5%	47.4%	67.5%	62.5%	78.9%	67.3%	80.1%
warpPIE10p	82.4%	82.1%	57.8%	38.3%	58.1%	43.8%	58.8%	43.3%
lung	85.6%	84.1%	69.5%	20.4%	66.0%	40.9%	67.4%	60.7%
9_Tumors	68.7%	64.1%	64.8%	58.8%	66.3%	58.5%	60.4%	54.8%
Mean	80.8%	83.3%	62.0%	57.9%	70.0%	68.0%	71.9%	71.3%

conduct extensive experiments by varying their values across a wide range, from 10^{-6} to 10^6. Specifically, we plot α, β and γ along the X, Y and Z axes, respectively, and record the corresponding clustering performance. We observe that the parameters α and β have more effect on the performance than the parameter γ on the given datasets. Therefore, in the subsequent analysis, we focus on discussing the parameters α and β and conduct the parameter sensitivity study in terms of α, β, while fixing γ to some values. Both α and β are tuned from $\{10^{-6}, 10^{-5}, \cdots, 10^5, 10^6\}$. To facilitate representation, we use the logarithmic values of α, β and γ for the X, Y and Z axes, respectively. The results across all datasets are presented in Figure 4, where the color bar reflects the range of clustering performance under this grid search strategy.

Overall, the results demonstrate that our method is generally robust to the choice of parameters across most datasets. With such a large search space spanning 12 orders of magnitude (from 10^{-6} to 10^6), it is expected that certain datasets may exhibit performance fluctuations at specific parameter values. Notably, we observe that there are two datasets, specifically YALEB ((k) and (l)) and lung ((q) and (r)), exhibit relatively high sensitivity to parameter values under this wide range, with fluctuations in both ACC and NMI exceeding 16. We attribute this sensitivity primarily to the large span of the parameter grid. To verify this, we conduct further experiments on these two datasets in which the regularization parameters α and β are restricted to the range $\{1, 2, 3, \cdots, 9, 10\}$, i.e., parameters are chosen at intervals of 1. The corresponding sensitivity results are shown in Figure 5. It can be seen that under this more moderate parameter range, the performance of both datasets becomes much more stable, while still achieving competitive clustering performance. This confirms that the observed sensitivity is largely due to the wide parameter span rather than an inherent issue with our method.

Additionally, we examine the parameter sensitivity of representative comparison methods, specifically NDFS and UDFS, on these two datasets (YALEB and lung). As shown in Figure 6 and Figure 7, these methods also exhibit noticeable sensitivity to parameter choices on these datasets. In particular, UDFS and NDFS show a large performance fluctuation range on lung, while NDFS also shows a large fluctuation range on YALEB. Although UDFS appears relatively stable on the YALEB dataset, this is primarily attributable to its overall poor performance on this dataset. Due to its low effectiveness across all parameter settings, the absolute performance variation remains limited (ACC ranges only from 9.0 to 14.5, and

NMI from 11.8 to 23.9), making the sensitivity seem less pronounced. These observations suggest that the degree of sensitivity is more closely related to the characteristics of these two datasets rather than the specific method employed.

Finally, to further validate the claim that only a few datasets exhibit parameter sensitivity, we extend the sensitivity analysis to six additional datasets[¶]: GLIOMA, nci9, pixraw10P, lymphoma, warpAR10P, and Yale. As shown in Figure 8, our method maintains stable performance across the wide parameter range on these datasets, which supports our conclusion that our method is generally insensitive to parameter choices for most datasets.

6.7 Computational Complexity Analysis

We now analyze and compare the computational complexity of our proposed method with that of UDPFS, NDFS, and UDFS, all of which rely on iterative optimization procedures to compute the feature selection matrix W.

UDPFS: The overall computational complexity of UDPFS is primarily concentrated in two components: updating the feature selection matrix W^k and performing fuzzy K-Means clustering. Specifically, the update of W^k involves the eigen-decomposition step with complexity $\mathcal{O}(d^3)$, and the computation of the with-in class scatter matrix S_w, which costs $\mathcal{O}(dn^2 + d^2n + n^2c)$. The fuzzy K-Means clustering step requires $\mathcal{O}(d^2n + dnc)$. As a result, the total computatioanl complexity of UDPFS is $\mathcal{O}(d^3 + dn^2 + d^2n + n^2c + dnc)$.

NDFS: The computational complexity of NDFS mainly arises from two components: updating the scaled cluster indicator matrix F^k and the feature selection matrix W^k. Specifically, updating W^k involves the matrix inversion and several matrix multiplications, resulting in a computational complexity of $\mathcal{O}(d^2n + d^3 + dnc)$. The update of F^k requires multiple matrix multiplications, leading to a complexity of $\mathcal{O}(dc + d^2n + dn^2 + n^2c + nc^2)$. Therefore, the overall computational complexity of NDFS is $\mathcal{O}(d^3+d^2n+dn^2+n^2c+nc^2+dnc)$.

UDFS: The computational complexity of UDFS is dominated by two stages: computing the auxiliary matrix M and updating the feature selection matrix W^k. Computing M involves an unsupervised local discriminative analysis step costing $\mathcal{O}(dn^2+n^2 \log n)$, followed by matrix multiplications of cost $\mathcal{O}(dn^2 + d^2n)$. Updating W^k is driven by an eigen-decomposition of a $d \times d$ matrix, with complexity $\mathcal{O}(d^3)$. Therefore, the total computational complexity of UDFS is $\mathcal{O}(dn^2 + d^2n + n^2 \log n + d^3)$.

Ours: The overall computational complexity of our framework is dominated by the inner PAM loop. In particular, it depends on the update rules for the variables $W^{k,j}$, $U^{k,j}$, $V^{k,j}$, $Y^{k,j}$, $F^{k,j}$, $\hat{Y}^{k,j}$, as detailed in (14)-(19). Fortunately, each subproblem admits a closed-form solution, allowing for an explicit and tractable complexity analysis.

Specifically, the main computational cost in updating $W^{k,j}$ comes from inverting the matrix $aI_d + \rho^k XX^\intercal$ and performing the matrix multiplications involved in computing Z. When $d < n$, the complexity is $\mathcal{O}(d^2n + d^3 + dnc)$; whereas when $n < d$, it becomes $\mathcal{O}(dn^2+n^3+dnc)$. Updating $U^{k,j}$ requires computing N, with complexity $\mathcal{O}(dnc)$. Updating $V^{k,j}$ involves $\mathcal{O}(dc)$; updating $Y^{k,j}$ entails inverting a symmetric positive definite matrix $2L + (3\rho^k + C_4^{k,j-1})I$ and computing matrix P, leading to $\mathcal{O}(n^2c + \frac{1}{3}n^3 + dnc)$; updating $F^{k,j}$ costs $\mathcal{O}(nc)$; and updating $\hat{Y}^{k,j}$ involves an SVD with complexity $\mathcal{O}(nc^2)$. Therefore,

[¶] https://jundongl.github.io/scikit-feature/datasets.html the total computational complexity per iteration is

$$\mathcal{O}(d^2n + d^3 + n^3 + n^2c + dnc + nc^2) \quad \text{if } d < n,$$

or

$$\mathcal{O}(dn^2 + n^3 + n^2c + dnc + nc^2) \quad \text{if } d > n.$$

Overall, when the number of features exceeds the number of samples ($d > n$), as is common in Biological Data, Text Data, and Face Image Data[‖], our method exhibits significant advantages in computational complexity. On datasets where $d < n$, such as Isolet, COIL20 and YALEB used in our experiments, although our algorithm has a higher theoretical cost, it consistently achieves better feature selection performance than above methods (see Table 2 and Table 3) and demonstrates enhanced stability (see Figure 1 and Table 5) as well as robustness against noise (see Figure 2, Figure 3 and Table 8)

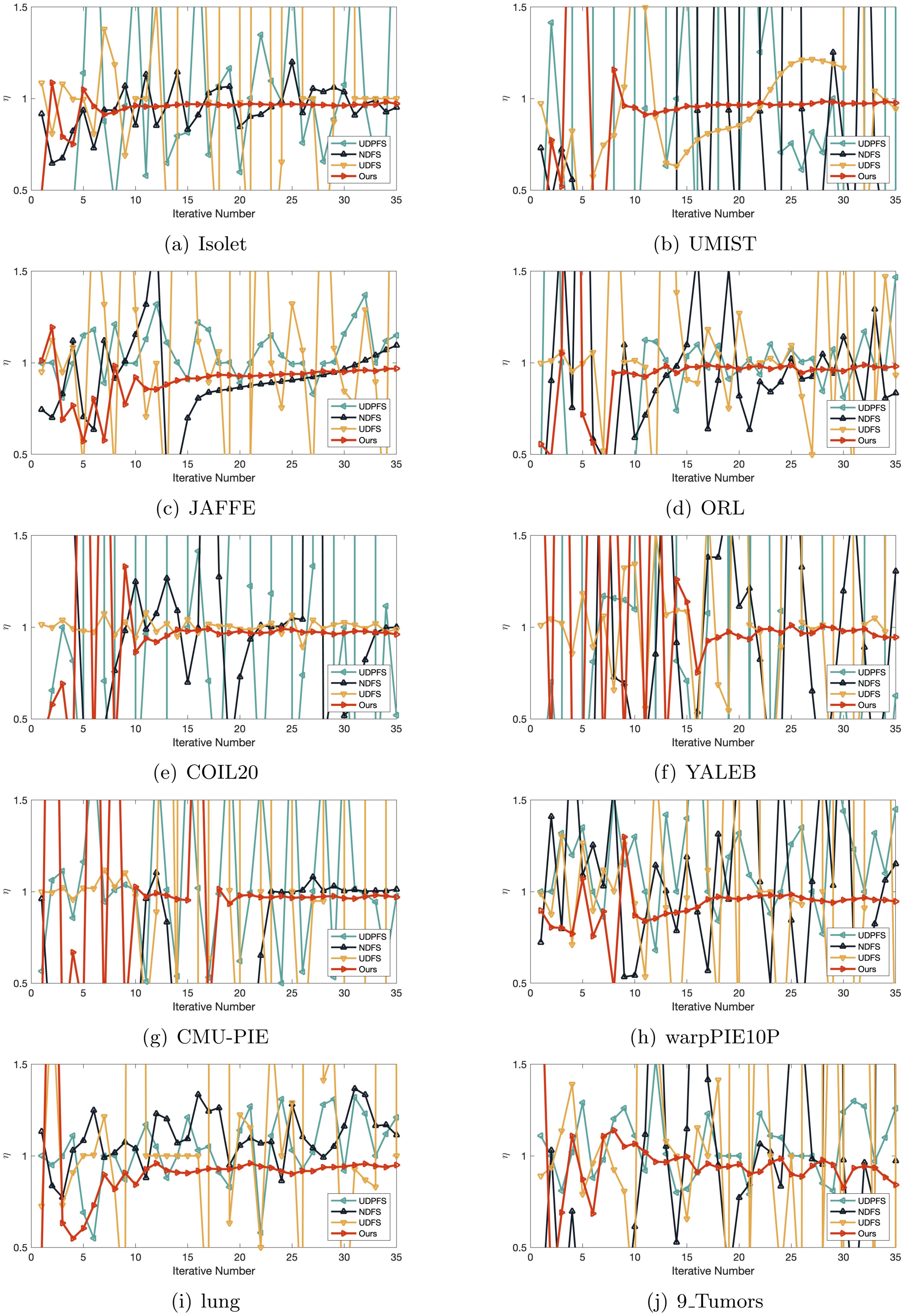
7. conclusion

In this paper, we proposed an ideal feature selection model based on an $l_{2,1}$-norm regularized regression formulation with non-negative orthogonality constraints. This formulation effectively identifies the most representative features from high-dimensional data. To solve this challenging model, we developed an inexact augmented Lagrangian multiplier (IALM) method. Specifically, we employed a proximal alternating minimization (PAM) strategy to handle the resulting subproblems, where each subproblem admits a closed-form solution. A key theoretical contribution lies in establishing the convergence of the sequence generated by the proposed algorithm, a guarantee that is often absent in state-of-the-art unsupervised feature selection methods. Quantitative and qualitative experimental results consistently demonstrated the superior effectiveness of our proposed method.

Acknowledgments

The work of Liping Zhang is supported by the National Natural Science Foundation of China (Grant No. 12571323, 12271291). Defeng Sun' research is supported by the Hong Kong RGC Senior Research Fellow Scheme [No. SRFS22235S02] and the GRF Grant 15307822. Yan Li' research is supported by Beijing Natural Science Foundation (Grant No. 1264061).

[‖] https://jundongl.github.io/scikit-feature/datasets



(a) Isolet

(b) UMIST

(c) JAFFE

(d) ORL

(e) COIL20

(f) YALEB

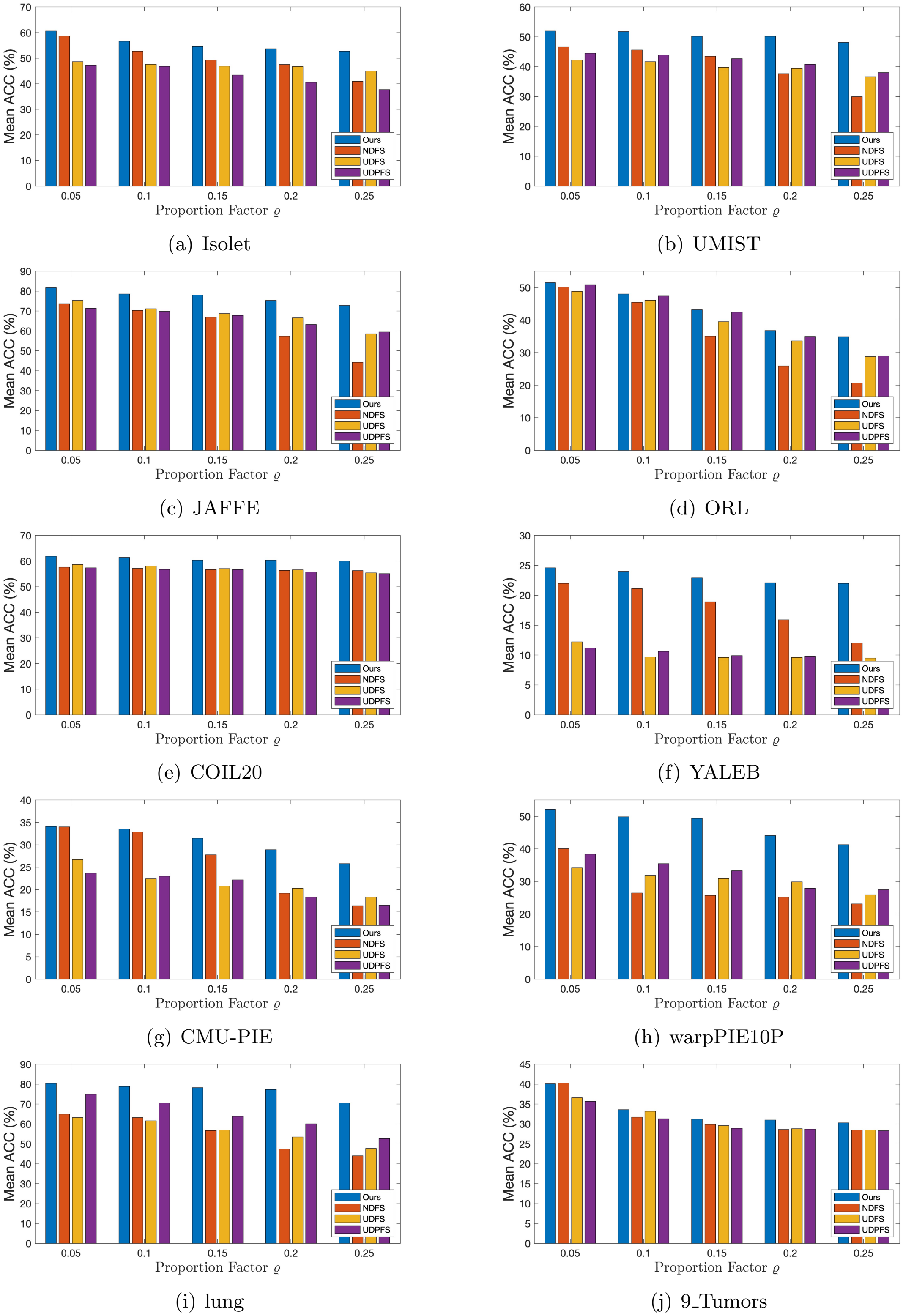
(g) CMU-PIE

(h) warpPIE10P

(i) lung

(j) 9_Tumors

Figure 1: Stability comparison curves between our proposed method and other iterative methods across all datasets are provided.



(a) Isolet

(b) UMIST

(c) JAFFE

(d) ORL

(e) COIL20

(f) YALEB

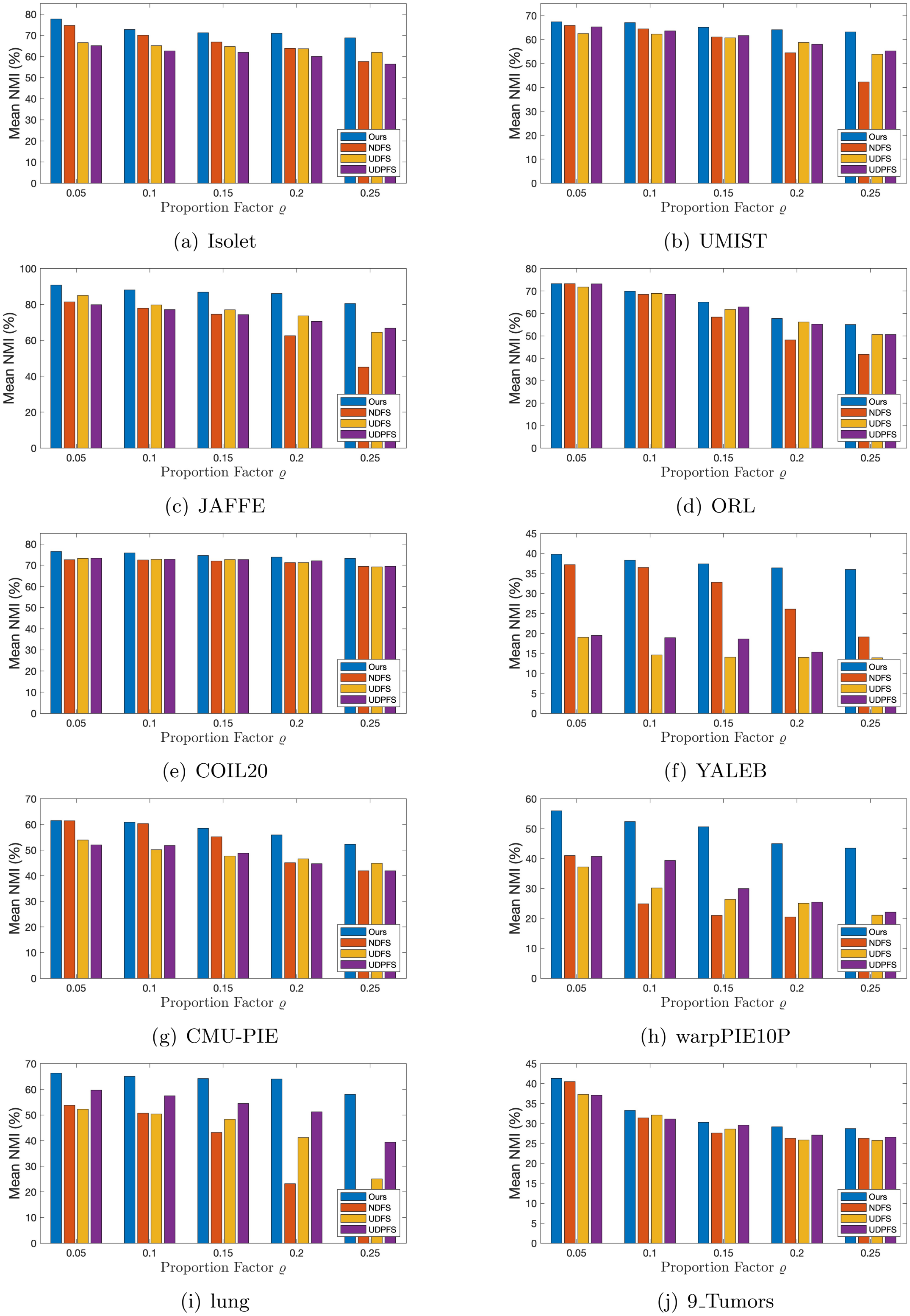
(g) CMU-PIE

(h) warpPIE10P

(i) lung

(j) 9_Tumors

Figure 2: Robustness comparison of ACC between our proposed method and other iterative methods under all dataset perturbations.



(a) Isolet

(b) UMIST

(c) JAFFE

(d) ORL

(e) COIL20

(f) YALEB

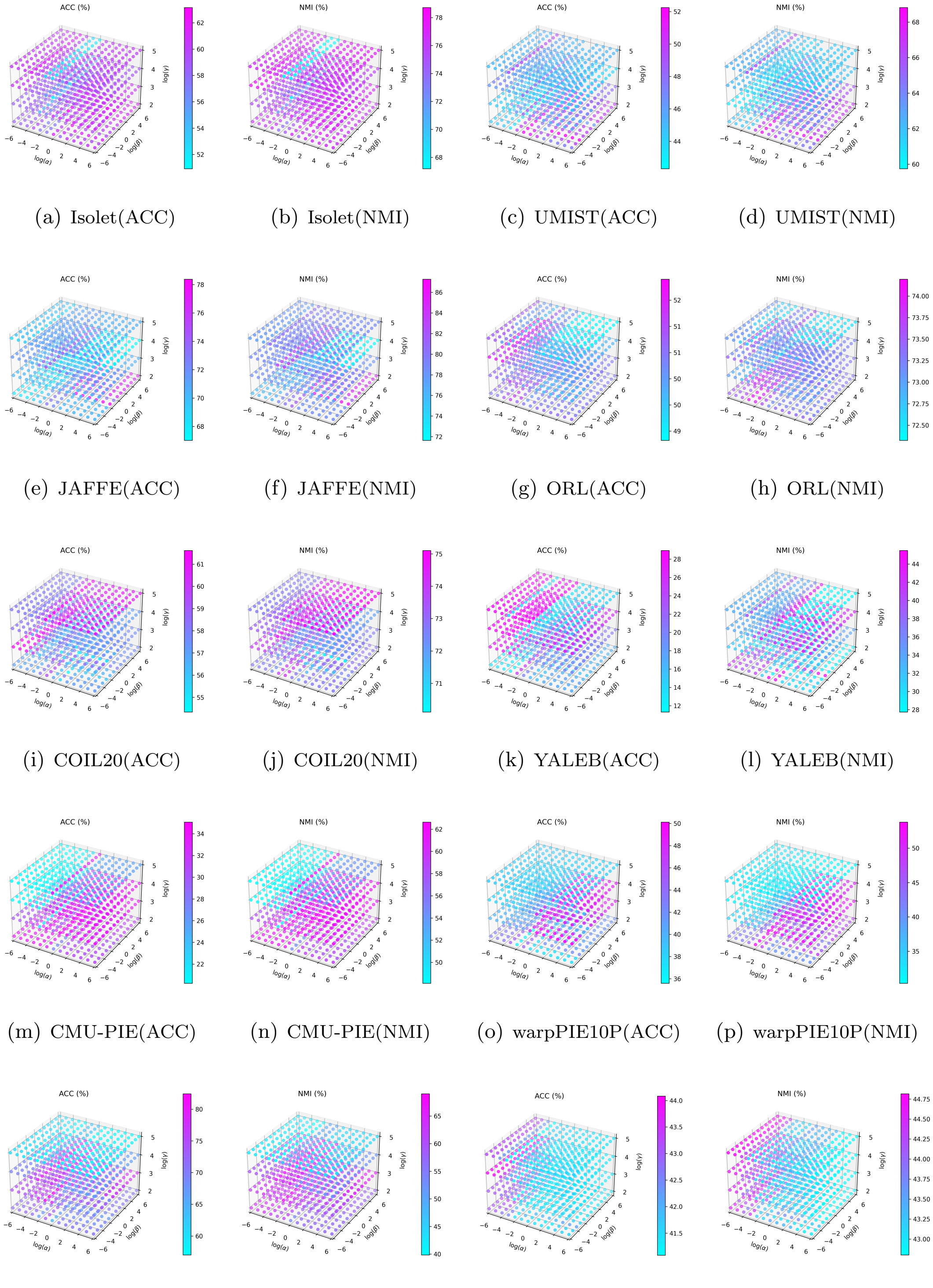
(g) CMU-PIE

(h) warpPIE10P

(i) lung

(j) 9_Tumors

Figure 3: Robustness comparison of NMI between our proposed method and other iterative methods under all dataset perturbations.



(a) Isolet(ACC) (b) Isolet(NMI) (c) UMIST(ACC) (d) UMIST(NMI)

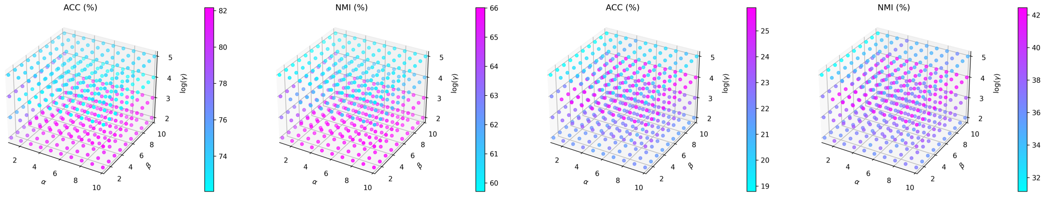
(e) JAFFE(ACC) (f) JAFFE(NMI) (g) ORL(ACC) (h) ORL(NMI)

(i) COIL20(ACC) (j) COIL20(NMI) (k) YALEB(ACC) (l) YALEB(NMI)

(m) CMU-PIE(ACC) (n) CMU-PIE(NMI) (o) warpPIE10P(ACC) (p) warpPIE10P(NMI)

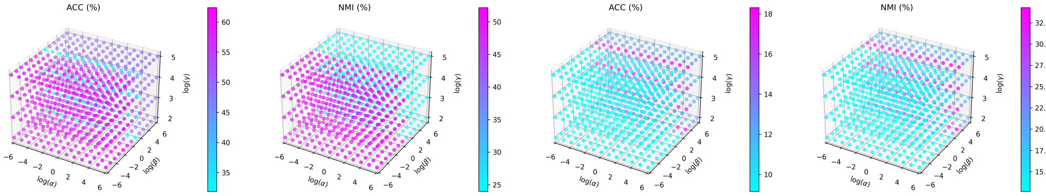
(q) lung(ACC) (r) lung(NMI) (s) 9_Tumors(ACC) (t) 9_Tumors(NMI)

Figure 4: Performance evaluation across all datasets with different values of α, β, and γ using a grid search strategy.



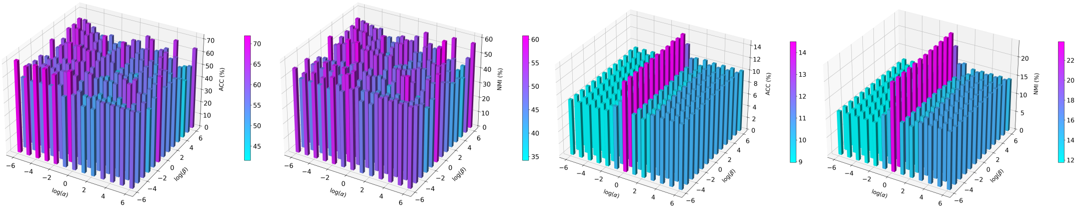
(a) lung(ACC) (b) lung(NMI) (c) YALEB(ACC) (d) YALEB(NMI)

Figure 5: Performance evaluation on the lung and YALEB datasets using a finer grid search over a narrower range of α, β, and γ.



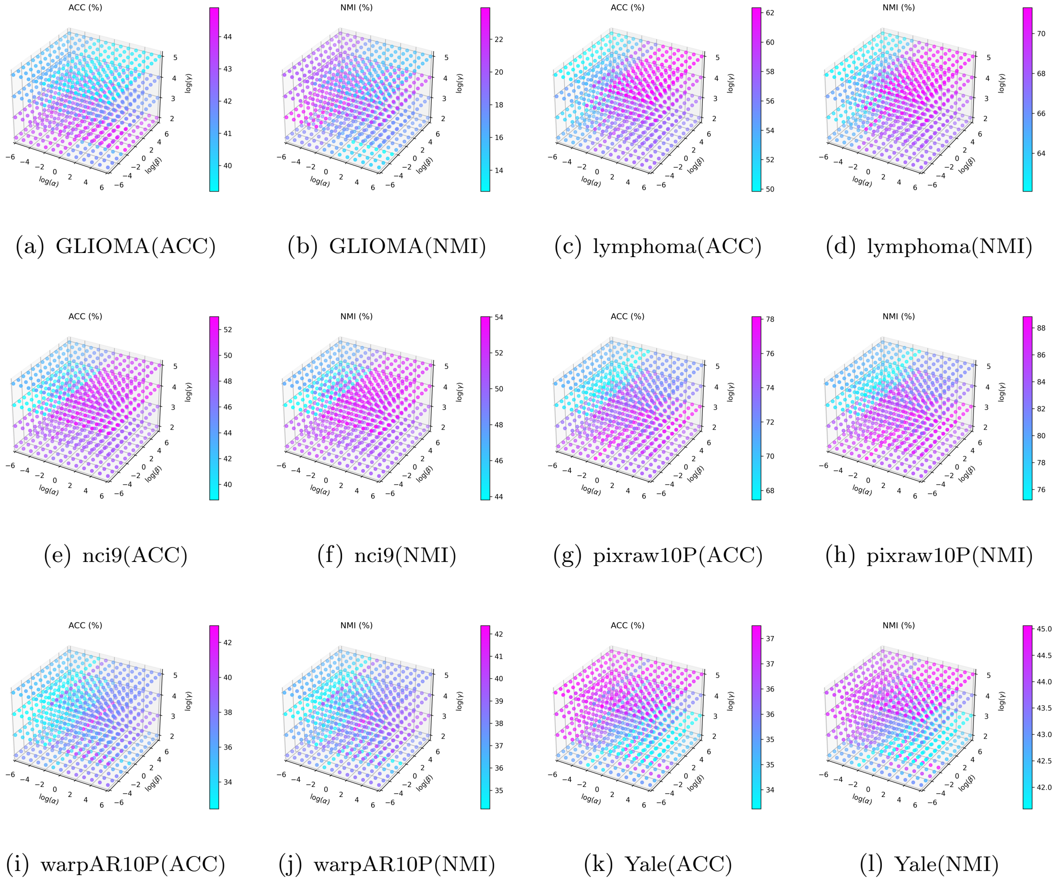
(a) lung(ACC) (b) lung(NMI) (c) YALEB(ACC) (d) YALEB(NMI)

Figure 6: Performance of NDFS on the lung and YALEB with different combinations of α, β, and γ using grid search.



(a) lung(ACC) (b) lung(NMI) (c) YALEB(ACC) (d) YALEB(NMI)

Figure 7: Performance of UDFS on the lung and YALEB under different parameter settings using a grid search strategy.



(a) GLIOMA(ACC)

(b) GLIOMA(NMI)

(c) lymphoma(ACC)

(d) lymphoma(NMI)

(e) nci9(ACC)

(f) nci9(NMI)

(g) pixraw10P(ACC)

(h) pixraw10P(NMI)

(i) warpAR10P(ACC)

(j) warpAR10P(NMI)

(k) Yale(ACC)

(l) Yale(NMI)

Figure 8: Performance of the proposed method on six additional datasets under a wide range of parameter settings using a grid search strategy.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, 2009.

Roberto Andreani, Ernesto G Birgin, José Mario Martínez, and María Laura Schuverdt. On augmented lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18(4):1286–1309, 2008.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1): 459–494, 2014.

Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 333–342, 2010.

David Charte, Francisco Charte, and Francisco Herrera. Reducing data complexity using autoencoders with class-informed loss functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9549–9560, 2022.

Hong Chen, Feiping Nie, Rong Wang, and Xuelong Li. Fast unsupervised feature selection with bipartite graph and $l_{2,0}$-norm constraint. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4781–4793, 2023.

Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang. Proximal gradient method for nonsmooth optimization over the stiefel manifold. *SIAM Journal on Optimization*, 30(1):210–239, 2020a.

Zheng Chen, Meng Pang, Zixin Zhao, Shuainan Li, Rui Miao, Yifan Zhang, Xiaoyue Feng, Xin Feng, Yexian Zhang, Meiyu Duan, et al. Feature selection may improve deep neural networks for the bioinformatics problems. *Bioinformatics*, 36(5):1542–1552, 2020b.

Manoranjan Dash, Hua Liu, and Jun Yao. Dimensionality reduction of unsupervised data. In *Proceedings 9th IEEE International Conference on Tools with Artificial Intelligence*, pages 532–539. IEEE, 1997.

Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 126–135, 2006.

Richard O Duda and Peter E Hart. Stork. dg, pattern classification. *John Willey and Sons, New York*, 2001.

Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

Jie Gui, Dacheng Tao, Zhenan Sun, Yong Luo, Xinge You, and Yuan Yan Tang. Group sparse multiview patch alignment framework with view consistency for image classification. *IEEE Transactions on Image Processing*, 23(7):3126–3137, 2014.

Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Advances in Neural Information Processing Systems*, 18, 2005.

Nicholas J Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge university press, 2012.

Chenping Hou, Feiping Nie, Xuelong Li, Dongyun Yi, and Yi Wu. Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, 44(6):793–804, 2013.

Bo Jiang, Xiang Meng, Zaiwen Wen, and Xiaojun Chen. An exact penalty approach for optimization with nonnegative orthogonality constraints. *Mathematical Programming*, 198 (1):855–897, 2023.

Josef Kittler. Feature selection and extraction. *Handbook of Pattern Recognition and Image Processing*, 1986.

A Klaser, C Schmid, and CL Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition*, volume 42, pages 3169–3176, 2011.

Wojtek J Krzanowski. Selection of variables to preserve multivariate data structure, using principal components. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 36(1):22–33, 1987.

Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58:431–449, 2014.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Data Sets*. Cambridge university press, 2020.

Jiaxiang Li, Shiqian Ma, and Tejes Srivastava. A Riemannian alternating direction method of multipliers. *Mathematics of Operations Research*, 2024. doi: 10.1287/moor.2023.0068.

Jundong Li, Jiliang Tang, and Huan Liu. Reconstruction-based unsupervised feature selection: An embedded approach. In *IJCAI*, pages 2159–2165, 2017.

Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, and Hanqing Lu. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 2012.

Zhengxin Li, Feiping Nie, Jintang Bian, Danyang Wu, and Xuelong Li. Sparse pca via $l_{2,p}$-norm regularization for unsupervised feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. doi: 10.1109/TPAMI.2021.3121329.

Chunfeng Lian, Mingxia Liu, Jun Zhang, and Dinggang Shen. Hierarchical fully convolutional network for joint atrophy localization and alzheimer's disease diagnosis using structural mri. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):880–893, 2018.

Huan Liu, Hiroshi Motoda, and Lei Yu. A selective sampling approach to active feature selection. *Artificial Intelligence*, 159(1-2):49–74, 2004.

Michael J Lyons, Julien Budynek, and Shigeru Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12): 1357–1362, 1999.

Mahdokht Masaeli, Yan Yan, Ying Cui, Glenn Fung, and Jennifer G Dy. Convex principal feature selection. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 619–628. SIAM, 2010.

Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 306–313. IEEE, 2002.

Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676, 2008.

Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. *Advances in Neural Information Processing Systems*, 23, 2010.

Feiping Nie, Wei Zhu, and Xuelong Li. Unsupervised feature selection with structured graph optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Feiping Nie, Wei Zhu, and Xuelong Li. Structured graph optimization for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):1210–1222, 2019.

Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the stability of feature selection algorithms. *Journal of Machine Learning Research*, 18(174):1–54, 2018.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

Mingjie Qian and Chengxiang Zhai. Robust unsupervised feature selection. In *Twenty-third International Joint Conference on Artificial Intelligence*, 2013.

Yitian Qian, Shaohua Pan, and Lianghai Xiao. Error bound and exact penalty method for optimization problems with nonnegative orthogonal constraint. *IMA Journal of Numerical Analysis*, 44(1):120–156, 2024.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

Giorgio Roffo, Simone Melzi, Umberto Castellani, Alessandro Vinciarelli, and Marco Cristani. Infinite feature selection: a graph-based feature filtering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4396–4410, 2020.

Hao-Jun Michael Shi, Shenyinying Tu, Yangyang Xu, and Wotao Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016.

Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12): 1615–1618, 2003.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

Rong Wang, Jintang Bian, Feiping Nie, and Xuelong Li. Unsupervised discriminative projection for feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):942–953, 2022.

Suhang Wang, Jiliang Tang, and Huan Liu. Embedded unsupervised feature selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. $l_{2,1}$-norm regularized discriminative feature selection for unsupervised. In *22th International Joint Conference on Artificial Intelligence*, 2011a.

Yi Yang, Heng Tao Shen, Feiping Nie, Rongrong Ji, and Xiaofang Zhou. Nonnegative spectral clustering with discriminative regularization. In *25th AAAI Conference on Artificial Intelligence*, 2011b.

Jiho Yoo and Seungjin Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 140–147. Springer, 2008.

Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, and Thuc Duy Le. Multi-source causal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2240–2256, 2019.

Kai Zhang, Sheng Zhang, Jun Liu, Jun Wang, and Jie Zhang. Greedy orthogonal pivoting algorithm for non-negative matrix factorization. In *International Conference on Machine Learning*, pages 7493–7501. PMLR, 2019.

Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference on Machine learning*, pages 1151–1157, 2007.

Pengfei Zhu, Wangmeng Zuo, Lei Zhang, Qinghua Hu, and Simon CK Shiu. Unsupervised feature selection by regularized self-representation. *Pattern Recognition*, 48(2):438–446, 2015.

Pengfei Zhu, Qinghua Hu, Changqing Zhang, and Wangmeng Zuo. Coupled dictionary learning for unsupervised feature selection. In *30th AAAI Conference on Artificial Intelligence*, 2016.