Local and Global Structure Preservation for Robust Unsupervised Spectral Feature Selection

Xiaofeng Zhu, Shichao Zhang, *Senior Member, IEEE*, Rongyao Hu, Yonghua Zhu, and Jingkuan Song

Abstract—This paper proposes a new unsupervised spectral feature selection method to preserve both the local and global structure of the features as well as the samples. Specifically, our method uses the self-expressiveness of the features to represent each feature by other features for preserving the local structure of features, and a low-rank constraint on the weight matrix to preserve the global structure among samples as well as features. Our method also proposes to learn the graph matrix measuring the similarity of samples for preserving the local structure among samples. Furthermore, we propose a new optimization algorithm to the resulting objective function, which iteratively updates the graph matrix and the intrinsic space so that collaboratively improving each of them. Experimental analysis on 12 benchmark datasets showed that the proposed method outperformed the state-of-the-art feature selection methods in terms of classification performance.

Index Terms—Feature selection, graph matrix, dimensionality reduction, subspace learning

1 Introduction

The goal of feature selection is designed to reduce the dimensionality of the data and keep useful information as much as possible [1], [2], [3], [4], [5]. The most popular solution for feature selection is to select the features with high-scores based on predefined metric. This makes feature selection certainly remove irrelevant (or uninformative) features, reduce the dimensionality, speed up the execution time, decrease the storage cost, improve the performance of learning models, and so on [6], [7], [8], [9], [10], [11]. However, it is still a challenging issue to improve the effectiveness of feature selection. To address this, this paper designs a robust unsupervised spectral feature selection method to preserve the local and global structure among training samples and their corresponding features.

It is difficult to obtain enough labelled information in many real applications [12], [13], [14], [15], so unsupervised dimensionality reduction is a very practical technique. Therefore, unsupervised spectral feature selection (USFS) method has been developed to incorporate the feature selection and subspace learning into a framework, so as to generate interpretable and robust feature selection models. The USFS has thus attracted extensive research interests [16], [17], [18] and has been successfully applied in the domains of data mining and machine learning. For example, Cai et al. [16] and Zhao et al. [1] advocated a two-phase USFS method to measure the importance of features in a dataset. They first conduct an eigenvalue decomposition on original data to obtain a graph representation, and then a least square regression between the derived graph representation and the original data is performed with an ℓ_1-norm regularizer [16] and an $\ell_{2,1}$-norm regularizer [1], respectively. Recently, Du et al. [12] proposed to learn both the adaptive global structure and local structure among samples in a feature selection model, in which an $\ell_{2,1}$-norm regularizer was employed to select important features.

A common characteristic among previous USFS methods is the construction of the graph matrix on original data. While these feature selection methods have displayed pretty promising in unsupervised spectral feature selection, there are still some limitations that should be addressed for real applications. First, because there are usually noise and redundancy in original data, the constructed graph matrix may be of low-quality to degrade the effectiveness of feature selection models. Second, in some USFS methods, the construction of the graph matrix gives a consideration of preserving either the local structure or the global structure among the samples. This paper advocates to preserve both the local and global structure among training samples because these two kinds of geometry structure have been demonstrated to strengthen the performance of USFS methods due to providing complimentary information to each other [19], [20]. Third, although the correlation among features has been shown its importance in constructing robust feature selection models [8], [21], existing USFS methods did not consider that. Lastly, the learning of the graph matrix and the feature selection are carried out in two separated processes. This can easily lead to a suboptimal result, even though each of these two processes could achieve their individual optimization.

- *X. Zhu, S. Zhang, and R. Hu are with the Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China. E-mail: seanzhuxf@gmail.com, zhangsc@mailbox.gxnu.edu.cn, hu_No1@126.com.*
- *Y. Zhu is with the School of Computer, Electronics, and Information, Guangxi University, Nanning 530004, China. E-mail: yhzhu66@qq.com.*
- *J. Song is with the University of Electronic Science and Technology of China, Chengdu Shi, Sichuan Sheng 610051, China. E-mail: jingkuan.song@gmail.com.*

Manuscript received 8 May 2017; revised 26 Sept. 2017; accepted 9 Oct. 2017. Date of publication 25 Oct. 2017; date of current version 2 Feb. 2018. (Corresponding author: Shichao Zhang.) Recommended for acceptance by W. Zhang. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2017.2763618

To deal with the above four limitations, in this paper we propose a robust USFS method. We list its main contributions as follows.

First, different from previous USFS methods, the proposed method efficiently utilizes the feature-level representation property and the low-rank constraint on the weight matrix, respectively, to consider both the local correlation and the global correlation of features for feature selection. As a result, these two kinds of correlations can provide complementary information to each other.

Second, the proposed method uses the low-rank constraint to preserve the global structure of training samples, and to learn the graph matrix for preserving the local structure of training samples. Accordingly, these two kinds of structure preservations can provide complementary information to each other so that the effectiveness of feature selection is well improved. It is noteworthy that previous USFS methods (e.g., [22], [23]) only preserve one of them.

Third, in the proposed robust USFS method, the correlation among features and the correlation among training samples are both identified from the intrinsic low-dimensional space of original data. This delivers the profit of avoiding the adversely impact of noise and redundancy in original data. To the best of our knowledge, there is no literature focused on simultaneously learning the feature correlations and the sample correlations. And there is only a few literature such as [22], [24], [25] focused on learning the graph matrix to preserve the local structure of training samples from the intrinsic space, while a number of USFS methods were designed to learn the graph matrix from original data such as [16], [18], [26].

Lastly, the proposed method jointly and iteratively performs the graph matrix construction and the feature selection in the intrinsic low-dimensional space. Moreover, our method uses the low-rank constraint to avoid the influence of noise and redundancy, which is not well solved in previous methods. For example, [1], [16] separately carried out them and often resulted in suboptimal results. [10], [18], [27] jointly optimized the graph matrix construction and the feature selection, and learned the graph matrix from original data. [12], [22] jointly and iteratively optimized them and learned the graph matrix from the intrinsic low-dimensional space, but ignoring the influence of noise and redundancy.

2 Related Work

In this section, we first review previous feature selection methods, and then analyze previous USFS methods in detail.

2.1 Feature Selection

In the domain of dimensionality reduction, feature selection methods try to find a subset of the original features, while subspace learning methods transform the high-dimensional data to their low-dimensional space, including linear transformation (such as Principal Component Analysis (PCA) [28], Fisher's Linear Discriminant Analysis [29], Canonical Correlation Analysis (CCA) [30], and Locality Preserving Projection (LPP)) and nonlinear transformation (such as kernel PCA [31], kernel LDA [32], and kernel CCA [30]). Usually, feature selection outputs interpretable result and subspace learning leads to robust models [10], [33].

Since many real applications prefer the interpretable ability of dimensionality reduction, this paper focuses on the study of feature selection. Existing feature selection methods can be partitioned into different categories according to various perspectives. For example, existing feature selection methods can be partitioned into three subgroups via the learning models, such as filter methods (i.e., which selects the important features independent on the learning model [34]), wrapper methods (which searches the important features guided by accuracy [35]), and embedded methods (where features are selected to be removed based on the prediction errors during the process of the construction of the models [12], [18], [22]). Feature selection methods can also be parted into unsupervised feature selection methods [12], [22], [23], [36], [37], supervised feature selection methods [38], [39] and semi-supervised feature selection methods [40], [41], [42], [43], according to the label information.

In this paper, we focus on the study of USFS since 1) it is an embedded method which has been shown to outperform either filter methods or wrapper methods [18], [22], [43]; 2) the label information is difficult to be obtained due to all kinds of reasons, such as limited sources and costs [23], [36], [39]; 3) spectral feature selection has been demonstrated to output robust and interpretable result [38], [39].

2.2 Unsupervised Spectral Feature Selection

USFS methods usually belong to embedded methods and include two key components, i.e., the graph matrix learning to conduct subspace learning and a sparsity-inducing regularizer (such as an ℓ_1-norm regularizer and an $\ell_{2,1}$-norm regularizer) to conduct feature selection. According to the modes of conducting these two key components, previous USFS methods can be classified into three categorizes, i.e., Sequential USFS methods, joint USFS methods, iteratively joint USFS methods.

Sequential USFS methods first conduct subspace learning to obtain the graph representation of the data, and then conduct a sparse feature selection between the resulting graph representation and the original data by sparsity-inducing regularizers. For example, the Multi-Cluster Feature Selection (MCFS) method [16] uses an ℓ_1-norm regularizer, while both the Minimize the feature Redundancy for spectral Feature Selection (MRFS) method [45] and the joint Feature Selection and Subspace Learning (FSFL) method [44] use the $\ell_{2,1}$-norm regularizer. Both MRFS and FSFL outperformed MCFS due to considering the global correlation among the features via the group sparsity, i.e., the $\ell_{2,1}$-norm regularizer. The difference between MRFS and FSFL is that MRFS preserves the pairwise sample similarity, i.e., the global correlation among the samples, via a kernel matrix of the feature matrix, while FSFL constructs a sparse kNN graph to preserve the local structure among the samples, i.e., the local correlation among the samples.

Joint USFS methods jointly conduct subspace learning and sparse feature selection in a framework. Their difference is the method of conducting subspace learning, i.e., the method of the construction of the graph matrix. For example, the Joint Embedding Learning and Sparse Regression (JELSR) method [18] uses the kNN-based graph Laplacian regularizer to preserve the local structure of the samples, the Robust Spectral learning framework for unsupervised

TABLE 1
The Summarization Between Previous USFS Methods and Our Proposed Method

Methods	Feature correlation	Sample correlation	Noise & redundancy	Dynamic graph matrix	Joint learning	Learning space
MCFS [16]	×	Local	×	×	×	Original space
FSFL [44]	Global	Local	×	×	×	Original space
MRFS [45]	Global	Global	×	×	×	Original space
JELSR [18]	Global	Local	×	×	√	Original space
RSFS [17]	Global	Global	×	×	√	Original space
NDFS [36]	Global	Local	×	×	√	Original space
FSASL [12]	Global	Local	×	√	√	Intrinsic space
SOGFS [22]	Global	Local	×	√	√	Intrinsic space
Proposed	Local & Global	Local & Global	√	√	√	Intrinsic space

This table has five blocks and the last four blocks describe the characteristic of sequential USFS methods, joint USFS methods, iteratively joint USFS methods, and our proposed method, respectively.

Feature Selection (RSFS) method [17] utilizes the local kernel regression to capture the nonlinear geometrical information of the samples, and the Nonnegative Discriminative Feature Selection (NDFS) method [36] learns a pseudo cluster labels and then uses it to learn the graph matrix.

Iteratively joint USFS methods claim that the selected features highly depend on the learned graph matrix, so they iteratively update the graph matrix and the selected features until the algorithm converges. For example, the unsupervised Feature Selection with Adaptive Structure Learning (FSASL) method [12] iteratively obtains the adaptive graph matrix and the adaptive features until both of them stop changing. Different from FSASL, the Structured Optimal Graph Feature Selection (SOGFS) method [22] adds one more constraints (i.e., the consistency of the graph matrix) to iteratively and jointly perform feature selection and the graph matrix learning.

Finally, the difference between previous USFS methods and our proposed method is listed in Table 1.

3 Approach

3.1 Notations

In this paper, we denote matrices, vectors, and scalars, respectively, as boldface uppercase letters, boldface lowercase letters, and normal italic letters. We summarize other notations used in this paper in Table 2.

3.2 Local Feature Correlation

Let the feature matrix $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^n] = [\mathbf{x}_1, \ldots, \mathbf{x}_d] \in \mathbb{R}^{n \times d}$ represent n d-dimensional samples. Motivated by the widely used sample-level self-expressiveness property sparsely representing each sample by other samples [46], we utilize the feature-level self-expressiveness property to represent each feature by all features with the following formulation:

$$\mathbf{x}_i \approx \sum\nolimits_{j=1}^{d} \mathbf{x}_j z_{j,i}, \ i = 1, \ldots, d, \tag{1}$$

where the element $z_{j,i}$ of the weight matrix $\mathbf{Z} \in \mathbb{R}^{d \times d}$ is the weight between the ith feature $\mathbf{x}_i$ and the jth feature $\mathbf{x}_j$. The assumption of Eq. (1) is that 1) the important features should be used to represent other features and should not be represented by the uninformative features, and the uninformative features should be represented by the important features and should be removed out the representation of all the features. With this assumption, Eq. (1) outputs small (or even zero) weight and large weight, respectively, to uninformative features and important features in the right side of Eq. (1). Obviously, Eq. (1) meets the assumption of feature selection, i.e., features are related or redundant on high-dimensional data [8].

By regarding the prediction of each feature as a task and constraining the sparsity across tasks with an $\ell_{2,1}$-norm regularizer, we change Eq. (1) to its matrix form and thus have the following least square objective function

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \gamma \|\mathbf{Z}\|_{2,1}, \tag{2}$$

where γ is a tuning parameter. The $\ell_{2,1}$-norm regularizer on $\mathbf{Z}$ (i.e., $\|\mathbf{Z}\|_{2,1}$) penalizes $\mathbf{Z}$ by encouraging the row sparsity, i.e., elements of some rows of $\mathbf{Z}$ are all zeros, to un-select the corresponding features in $\mathbf{X}$.

Eqs. (1) and (2) indicate that each feature (e.g., $\mathbf{x}_i$ in left-hand side of Eq. (1)) is represented by a linear combination of a subset of all features in right-hand side of Eq. (1), and the corresponding weight vector is the ith column $\mathbf{z}_i$ of $\mathbf{Z}$ in Eq. (2). Obviously, the larger the values in the $\mathbf{z}_i$, the more the corresponding features involve in the representation of the feature $\mathbf{x}_i$. In particular, if there is a zero-row in $\mathbf{z}^j$ (where $\mathbf{z}^j = [z_{j,1}, \ldots, z_{j,d}]$), then the corresponding feature (i.e., $\mathbf{x}_j$ in right-hand side of Eq. (1)) will not participate in the representation of features. That is, the features participating in the representation of all features should be important, while those not participating in the representation process should be discarded by means of feature selection, i.e., $\|\mathbf{Z}\|_{2,1}$.

TABLE 2
The Used Notations in This Paper

$\mathbf{X}$	the feature matrix of the training data
$\mathbf{x}$	a vector of $\mathbf{X}$
$\mathbf{x}^i$	the ith row of $\mathbf{X}$
$\mathbf{x}_j$	the jth column of $\mathbf{X}$
$x_{i,j}$	the element in the ith row and the jth column of $\mathbf{X}$
$\|\|\mathbf{X}\|\|_F$	the Frobenius norm of $\mathbf{X}$, i.e., $\|\|\mathbf{X}\|\|_F = \sqrt{\sum_{i,j} \mathbf{x}_{i,j}^2}$
$\|\|\mathbf{X}\|\|_{2,1}$	the $\ell_{2,1}$-norm of $\mathbf{X}$, i.e., $\|\|\mathbf{X}\|\|_{2,1} = \sum_i \sqrt{\sum_j x_{i,j}^2}$
$rank(\mathbf{X})$	the rank of $\mathbf{X}$
$\mathbf{X}^T$	the transpose of $\mathbf{X}$
$tr(\mathbf{X})$	the trace of $\mathbf{X}$
$\mathbf{X}^{-1}$	the inverse of $\mathbf{X}$

3.3 Global Feature Correlation & Global Structure Preservation

Eq. (2) finds the sparse representation of each feature individually (namely, *local feature correlation*) but no global constraint on its solution, and thus may be inaccurate at capturing the global structure of the features to largely depress the performance of feature selection on the grossly corrupted features. Since the corrupted data (including noisy/redundant features and outliers) have been indicated to largely increase its rank in real applications, the low-rank constraint has been used to help correct corruption via a low-rank constraint to output robust feature selection models [47], [48]. Specifically, given a low-rank assumption on $\mathbf{Z}$, i.e., $\mathbf{Z} = \mathbf{AB}$, where $\mathbf{A} \in \mathbb{R}^{d\times r}$, $\mathbf{B} \in \mathbb{R}^{r\times d}$, and $r \leq min(n, d)$, Eq. (2) is changed to

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{XAB}\|_F^2 + \gamma\|\mathbf{AB}\|_{2,1}, \tag{3}$$

In Eq. (3), the reduced matrix $\mathbf{XA} \in \mathbb{R}^{n\times r}$, which is then multiplied by $\mathbf{B}$ to represent the feature matrix $\mathbf{X}$ (i.e., the first $\mathbf{X}$ in $\mathbf{X} - \mathbf{XAB}$), has less than r latent factors. Geometrically, $\mathbf{A}$ (or $\mathbf{B}$) has the effect of transforming $\mathbf{X}$ (or $\mathbf{XA}$) to a new space, i.e., conducting subspace learning by considering the correlation among d (or r) features (i.e., all features as a group), namely, *global feature correlation*. Therefore, the low-rank constraint on $\mathbf{A}$ (or $\mathbf{B}$) has the effect of subspace learning by considering the global feature correlation. In particular, Eq. (13) further indicates that such subspace learning actually conducts LDA via considering the global feature correlation to preserve the global structure of the samples.

Unlike Eq. (2) using the feature-level self-expressiveness property to only consider the local feature correlation, Eq. (3) simultaneously uses the feature-level self-expressiveness property and a low-rank constraint (i.e., replacing $\mathbf{Z}$ by $\mathbf{A}$ and $\mathbf{B}$) to consider the local feature correlation and the global feature correlation, respectively, and thus resulting in local self-expressiveness and global self-expressiveness.

3.4 Local Structure Preservation

Previous literatures have shown that both the global structure and the local structure of the samples may provide complementary information to strengthen the performance of dimensionality reduction [49], [50], [51], so this paper proposes to preserve the global structure of the samples via a low-rank constraint in Eq. (3) and also to preserve the local structure of the samples via learning a graph matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ on a low-dimensional space. Intuitively, given the feature matrix $\mathbf{X}$ and its weight matrix $\mathbf{W}$, we follow the literature [7] to have the following objective function:

$$\min_{\mathbf{W}} \sum_{i,j}^{n} \|\mathbf{x}^i\mathbf{W} - \mathbf{x}^j\mathbf{W}\|_2^2 s_{i,j}, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is also a transformation matrix transferring the high-dimensional data $\mathbf{X}$ to a new space spanned by $\mathbf{XW}$, and the element $s_{i,j}$ of the graph matrix $\mathbf{S}$ denotes the similarity between the ith sample $\mathbf{x}^i$ and the jth sample $\mathbf{x}^j$. Moreover, if the ith sample $\mathbf{x}^i$ is one of the k-nearest neighbors of the jth sample $\mathbf{x}^j$, then the value of the heat kernel (i.e., $f(\mathbf{x}^i, \mathbf{x}^j) = exp(-\frac{\|\mathbf{x}^i-\mathbf{x}^j\|_2^2}{2\sigma^2})$ where σ is a tuning parameter) is regarded as the value of $s_{i,j}$; otherwise $s_{i,j} = 0$.

Although Eq. (4) has been widely used in previous USFS methods [16], [18], [45], it learns a fixed graph matrix $\mathbf{S}$ from original high-dimensional data $\mathbf{X}$ before learning $\mathbf{W}$. That is, the graph matrix learning is independent on the low-dimensional space learning. In this way, if original data are corrupted by noise and redundancy (it always true in real applications), then an incorrect graph matrix may be outputted. Moreover, Eq. (4) needs tune two parameters (i.e., k and σ), which is time-consuming. In particular, the quality of $\mathbf{S}$ has been reported very sensitive to the tuning of σ [50]. This motivates us to learn the graph matrix from the 'clean' data (i.e., a low-dimensional space with noise and redundancy as less as possible) and to reduce the number of the tuning parameters. However, the truth is that neither the graph matrix nor the low-dimensional space are known in advance. To address this, we couple the graph matrix learning with the low-dimensional space learning together to iteratively optimize them so that achieving their individually optimal result. As a result, we may learn the graph matrix by following the distribution of the samples, rather than using the heat kernel function to learn a fixed graph matrix, which also needs tune the parameter σ. We thus devise the following objective function:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W}} \ & \sum_{i,j}^{n} (\|\mathbf{x}^i\mathbf{W} - \mathbf{x}^j\mathbf{W}\|_2^2 s_{i,j} + \beta\|\mathbf{s}_i\|_2^2), \\ & s.t., \forall i, \mathbf{s}_i^T\mathbf{1} = 1, s_{i,i} = 0, \\ & \quad s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0, \end{aligned} \tag{5}$$

where β is a tuning parameter, $\|\cdot\|_2$ is the ℓ_2-norm of a vector, $\|\mathbf{s}_i\|_2^2$ is used to avoid the trivial solution, $\mathbf{1}$ and $\mathcal{N}(i)$ represent an all-one-element vector and the set of the nearest neighbors of the ith sample, respectively, and the constraint $\mathbf{s}_i^T\mathbf{1} = 1$ is used to obtain shift invariant similarity. As a consequence, Eq. (5) outputs small value (i.e., similarity) of $s_{i,j}$ for distant samples and large value of $s_{i,j}$ for close samples.

Unlike that the USFS methods [16], [18], [45] use Eq. (4) to learn a fixed graph matrix by tuning two parameters, Eq. (5) learns a dynamic graph matrix by only tuning a parameter k since the similarity among the samples is learnt according to the distribution of the samples, i.e., the learnt low-dimensional space spanned by $\mathbf{XW}$. Moreover, the dynamic graph matrix is iteratively learnt according to the optimized low-dimensional space so that learning a graph matrix from 'clean' data. Different from the dynamic graph matrix in [22] representing each sample by all samples, Eq. (5) represents each sample by only k nearest neighbor samples. Obviously, our proposed method easily avoids the influence of outliers which are usually far away its k nearest neighbors.

3.5 Objective Function

Although the graph matrix $\mathbf{S}$ is learnt from the low-dimensional space spanned by $\mathbf{XW}$ in Eq. (5), neither the graph matrix $\mathbf{S}$ nor the low-dimensional space are known. As a result, Eq. (5) may output unreliable models. This paper combines the constraints in Eq. (3) with Eq. (5) to address this issue. Specifically, by regarding the weight matrix $\mathbf{W}$ in Eq. (5) as the low-rank weight matrices $\mathbf{AB}$, i.e., $\mathbf{W} = \mathbf{AB}$, we combine Eq. (3) with Eq. (5) to yield our final objective function as follows:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{B},\mathbf{S}}\ & \sum_{i,j}^{n}\|\mathbf{x}^i\mathbf{AB}-\mathbf{x}^j\mathbf{AB}\|_2^2 s_{i,j}+\alpha\|\mathbf{X}-\mathbf{XAB}\|_F^2\\ &+\beta\sum_{i}^{n}\|\mathbf{s}_i\|_2^2+\gamma\|\mathbf{AB}\|_{2,1}\\ & s.t.,\forall i,\mathbf{s}_i^T\mathbf{1}=1,s_{i,i}=0,\\ &\quad s_{i,j}\geq 0 \text{ if } j\in\mathcal{N}(i),\text{otherwise } 0,\end{aligned}\tag{6}$$

where α,β and γ are tuning parameters. Eq. (6) iteratively updates the graph matrix $\mathbf{S}$ and the low-rank transformation matrix $\mathbf{AB}$ until all of variables achieve their individually optimal result. In this way, Eq. (6) uses the second term (i.e., the local self-expressiveness of features) and the low-rank constraint on both $\mathbf{A}$ and $\mathbf{B}$ (i.e., the global self-expressiveness of features) to consider the local feature correlation and the global feature correlation, respectively, and uses the iteratively updated graph matrix and the low-rank constraint to preserve the local structure of the samples and the global structure of the samples, respectively.

As a consequence, given the optimal $\mathbf{A}$ and $\mathbf{B}$, we calculate the ℓ_2-norm values of $(\mathbf{AB})^i, i=1,\ldots,d$, and then sort them in descending order. We finally select top r features corresponding to the top r ranked ℓ_2-norm values as the final result of our proposed feature selection method.

3.6 Optimization

Eq. (6) is not jointly convex to all the variables (i.e., $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$), but is convex for each variable while fixing the others. In this paper, we employ the alternative optimization strategy to optimize Eq. (6), i.e., iteratively optimizing each variable while fixing the others until the algorithm converges. We list the resulting pseudo in Algorithm 1.

3.6.1 *Update* $\mathbf{B}$ *and* $\mathbf{A}$ *by Fixing* $\mathbf{S}$

The optimizations of Eq. (6) on the variables $\mathbf{A}$ and $\mathbf{B}$ are convex but non-smooth due to the $\ell_{2,1}$-norm regularizer on $\mathbf{AB}$. In this paper, we employ the framework of Iteratively Reweighted Least Square (IRLS) [52] to optimize Eq. (6) via iteratively optimizing $\mathbf{A}$ and $\mathbf{B}$ until the predefined stopping criteria is satisfied.

With the fixed $\mathbf{S}$, Eq. (6) is changed to

$$\begin{aligned}\min_{\mathbf{A},\mathbf{B}}\ & \sum_{i,j}\|\mathbf{x}^i\mathbf{AB}-\mathbf{x}^j\mathbf{AB}\|_2^2 s_{i,j}\\ &+\alpha\|\mathbf{X}-\mathbf{XAB}\|_F^2+\gamma\|\mathbf{AB}\|_{2,1}.\end{aligned}\tag{7}$$

By following the IRLS framework, we rewrite Eq. (7) as

$$\begin{aligned}\min_{\mathbf{A},\mathbf{B}}\ & tr(\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{LXAB})+\alpha\|\mathbf{X}-\mathbf{XAB}\|_F^2\\ &+\gamma tr(\mathbf{B}^T\mathbf{A}^T\mathbf{PAB}),\end{aligned}\tag{8}$$

where $\mathbf{L}=\mathbf{Q}-\mathbf{S}\in\mathbb{R}^{n\times n}$ is a Laplacian matrix and $\mathbf{Q}$ is a diagonal matrix with its ith element $q_{i,i}=\sum_{j=1}^{n}s_{i,j}$, and the ith element of the diagonal matrix $\mathbf{P}\in\mathbb{R}^{d\times d}$ is defined as

$$p_{ii}=\frac{1}{2\|(\mathbf{AB})^i\|_2^2}, i=1,\ldots,d\tag{9}$$

where $(\mathbf{AB})^i$ is the ith row of $\mathbf{AB}$. By fixing $\mathbf{A}$, we set the derivative of Eq. (8) with respect to $\mathbf{B}$ to zero and solve the resulting equation to obtain

$$\mathbf{B}^*=(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{X},\tag{10}$$

where $\mathbf{S}_t=\mathbf{X}^T\mathbf{LX}+\alpha\mathbf{X}^T\mathbf{X}+\gamma\mathbf{P}$.

Then, we rewrite the Eq. (8) to the following expression,

$$\begin{aligned}\min_{\mathbf{A},\mathbf{B}}\ & tr(\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{LXAB})\\ &+\alpha tr(\mathbf{X}^T\mathbf{X}-\mathbf{X}^T\mathbf{XAB}-\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{X}\\ &+\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{XAB})+\gamma tr(\mathbf{B}^T\mathbf{A}^T\mathbf{PAB}).\end{aligned}\tag{11}$$

By substituting Eq. (10) back into Eq. (11), and Eq. (11) is changed to

$$\begin{aligned}&\max_{\mathbf{A}}\ tr(\mathbf{X}^T\mathbf{XA}(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{X}\\ &\qquad+\mathbf{X}^T\mathbf{XA}(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{X})\\ &\Leftrightarrow\\ &\max_{\mathbf{A}}\ tr(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{XX}^T\mathbf{XA}.\end{aligned}\tag{12}$$

Further, we obtain

$$\max_{\mathbf{A}}\ tr(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}_b\mathbf{A},\tag{13}$$

where $\mathbf{S}_b=\mathbf{X}^T\mathbf{XX}^T\mathbf{X}$. $\mathbf{S}_t$ and $\mathbf{S}_b$, respectively, are similar to the total-class scatter matrix and the between-class scatter matrix defined in the LDA method [53]. Therefore, the solution of Eq. (13) can be solved via eigenvalue decomposition, i.e., the global optimal solution of Eq. (13) is the top r eigenvectors of $\mathbf{S}_t^{-1}\mathbf{S}_b$ corresponding to r nonzero eigenvalues. Moreover, similar to the between-class scatter matrix, $\mathbf{S}_b$ in this work can be regarded as the between-sample correlation matrix, which preserves the global structure of the samples.

In this way, we can yield $\mathbf{A}$ by solving Eq. (13) and then yield $\mathbf{B}$ by Eq. (10). Moreover, we iteratively update $\mathbf{A}$ and $\mathbf{B}$ until the resulting objective function value is stable. The detail of optimizing $\mathbf{A}$ and $\mathbf{B}$ is listed in Algorithm 2.

3.6.2 *Update* $\mathbf{S}$ *by Fixing* $\mathbf{B}$ *and* $\mathbf{A}$

Given the fixed $\mathbf{A}$ and $\mathbf{B}$, Eq. (6) becomes

$$\begin{aligned}\min_{\mathbf{S}}\ & \sum_{i,j}^{n}||\mathbf{x}^i\mathbf{AB}-\mathbf{x}^j\mathbf{AB}||_2^2 s_{i,j}+\beta\sum_{i,j}^{n}s_{i,j}^2\\ & s.t.,\forall i,\mathbf{s}_i^T\mathbf{1}=1,s_{i,i}=0,\\ &\quad s_{i,j}\geq 0 \text{ if } j\in\mathcal{N}(i),\text{otherwise } 0,\end{aligned}\tag{14}$$

We first yield k nearest neighbors of all samples via calculating their euclidean distance, and then set the value of $s_{i,j}$ as 0 if the jth sample does not belong to one of k nearest neighbors of the ith sample, otherwise, the values $s_{i,j}$ is obtained by Eq. (15).

Since the optimization of $\mathbf{S}$ is equal to independently optimize each vector $\mathbf{s}_i, i=1,\ldots,n$, we further change Eq. (14) to individually optimize $\mathbf{s}_i, i=1,\ldots,n$, as follows:

$$\min_{\mathbf{s}_i^T\mathbf{1}=1,s_{i,i}=0,s_{i,j}\geq 0}\sum_{j}^{n}(\|\mathbf{x}^i\mathbf{AB}-\mathbf{x}^j\mathbf{AB}\|_2^2 s_{i,j}+\beta s_{i,j}^2).\tag{15}$$

By denoting $\mathbf{F}\in\mathbb{R}^{n\times n}$ where $f_{i,j}=||\mathbf{x}^i\mathbf{AB}-\mathbf{x}^j\mathbf{AB}||_2^2$, we rewrite Eq. (15) as follows:

$$\min_{\mathbf{s}_i^T\mathbf{1}=1,s_{i,i}=0,s_{i,j}\geq 0}\left\|\mathbf{s}_i+\frac{1}{2\beta}\mathbf{f}_i\right\|_2^2,\tag{16}$$

We further obtain the Lagrangian function of Eq. (16) as,

$$\min_{\mathbf{s}_i,\boldsymbol{\tau},\eta} \left\|\mathbf{s}_i + \frac{1}{2\beta}\mathbf{f}_i\right\|_2^2 - \boldsymbol{\tau}(\mathbf{s}_i^T\mathbf{1} - 1) - \eta^T\mathbf{s}_i, \tag{17}$$

where $\boldsymbol{\tau}$ and η are the Lagrangian multipliers. According to the Karush–Kuhn–Tucker (KKT) conditions [54], we yield the closed-form solution of $s_{i,j}, j = 1, \ldots, n$ as

$$s_{i,j} = (-\frac{1}{2\beta}\mathbf{f}_{i,j} + \tau)_+. \tag{18}$$

Algorithm 1. The Pseudo Code of Solving Eq. (6)

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, α, γ, and r;
Output: $\mathbf{A} \in \mathbb{R}^{d\times r}$, $\mathbf{B} \in \mathbb{R}^{r\times d}$, and $\mathbf{S} \in \mathbb{R}^{n\times n}$;
1. Calculate k nearest neighbors of all samples;
2. Initialize $\mathbf{S}$ by Eq. (5) where $\mathbf{W}$ is an identity matrix;
3. **repeat:**
   3.1. Update $\mathbf{A}$ and $\mathbf{B}$ via Algorithm (2);
   3.2. Update $\mathbf{S}$ by Eq. (16);
   3.3. Calculate $\mathbf{L} = \mathbf{Q} - \frac{\mathbf{S}^T+\mathbf{S}}{2}$;
   until converge

Algorithm 2. The Pseudo Code of Solving $\mathbf{A}$ and $\mathbf{B}$

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{L} \in \mathbb{R}^{n\times n}$, α, k, and r;
Output: $\mathbf{A} \in \mathbb{R}^{d\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times d}$;
1. Initialize $\mathbf{P} = \mathbf{I} \in \mathbb{R}^{d\times d}$;
2. **repeat:**
   2.1. Calculate $\mathbf{A}$ by Eq. (13);
   2.2. Calculate $\mathbf{B}$ by Eq. (10);
   2.3. Calculate $\mathbf{P}$ by Eq. (9);
   until converge

3.7 Convergence Analysis, Complexity, and Parameters' Determination

In this section, we first prove the convergence of both Algorithms 2 and 1, and then analyze the complexity of Algorithm 1. Finally, we discuss parameters' determination of Algorithm 1.

3.7.1 Convergence Analysis of Algorithm 2

We first list the following Lemma:

Lemma 1. *The inequality*

$$\sqrt{u} - \frac{u}{2\sqrt{v}} \leq \sqrt{v} - \frac{v}{2\sqrt{v}}, \tag{19}$$

is always hold for all positive real numbers of u *and* v *[6].*

We then prove the convergence of Algorithm 2 by the following Theorem 1:

Theorem 1. *The objective function value of Eq. (7) monotonically decreases until Algorithm 2 converges.*

Proof. While fixing $\mathbf{S}$, we denote the tth iteration of a matrix $\mathbf{A}$ as $\mathbf{A}^{(t)}$ and $\mathbf{A}^{(t+1)}$, respectively. According to Algorithm 2, we have

$$\begin{aligned} &< \mathbf{A}^{(t+1)}, \mathbf{B}^{(t+1)} > \\ &= \arg\ \min_{\mathbf{A},\mathbf{B}}\ tr(\mathbf{B}^{(t)^T}\mathbf{A}^{(t)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}) \\ &\quad + \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}\right\|_F^2 + \gamma tr(\mathbf{B}^{(t)^T}\mathbf{A}^{(t)^T}\mathbf{P}^{(t)}\mathbf{A}^{(t)}\mathbf{B}^{(t)}), \end{aligned} \tag{20}$$

which indicates that

$$\begin{aligned} &tr(\mathbf{B}^{(t+1)^T}\mathbf{A}^{(t+1)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}\right\|_F^2 \\ &+ \gamma tr(\mathbf{B}^{(t+1)^T}\mathbf{A}^{(t+1)^T}\mathbf{P}^{(t)}\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}) \\ &\leq tr(\mathbf{B}^{(t)^T}\mathbf{A}^{(t)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}\right\|_F^2 + \gamma tr(\mathbf{B}^{(t)^T}\mathbf{A}^{(t)^T}\mathbf{P}^{(t)}\mathbf{A}^{(t)}\mathbf{B}^{(t)}). \end{aligned} \tag{21}$$

□

By denoting $\mathbf{W} = \mathbf{A}\mathbf{B}$, we obtain $\mathbf{W}^{(t)}=\mathbf{A}^{(t)}\mathbf{B}^{(t)}$, $\mathbf{W}^{(t+1)}=\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}$. According to Eq. (9), Eq (21) can further be rewritten as follows:

$$\begin{aligned} &tr(\mathbf{W}^{(t+1)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t+1)}\right\|_F^2 + \gamma\sum_{i=1}^{d}\frac{\left\|\mathbf{w}^{i(t+1)}\right\|_2^2}{\left\|\mathbf{w}^{i(t)}\right\|_2} \\ &\leq tr(\mathbf{W}^{(t)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t)}\right\|_F^2 + \gamma tr\sum_{i=1}^{d}\frac{\left\|\mathbf{w}^{i(t)}\right\|_2^2}{\left\|\mathbf{w}^{i(t)}\right\|_2}, \end{aligned} \tag{22}$$

where $\mathbf{w}^{i(t)}$ and $\mathbf{w}^{i(t+1)}$ denote ith row of $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(t+1)}$, respectively. According to Lemma 1, we have

$$\left\|\mathbf{w}^{i(t+1)}\right\|_2 - \frac{\left\|\mathbf{w}^{i(t+1)}\right\|_2^2}{\left\|\mathbf{w}^{i(t)}\right\|_2} \leq \left\|\mathbf{w}^{i(t)}\right\|_2 - \frac{\left\|\mathbf{w}^{i(t)}\right\|_2^2}{\left\|\mathbf{w}^{i(t)}\right\|_2}. \tag{23}$$

By plugging Eq. (23) into Eq. (22), we have

$$\begin{aligned} &tr(\mathbf{W}^{(t+1)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t+1)}\right\|_F^2 + \gamma\sum_{i=1}^{d}\left\|\mathbf{w}^{i(t+1)}\right\|_2 \\ &\leq tr(\mathbf{W}^{(t)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t)}\right\|_F^2 + \gamma\sum_{i=1}^{d}\left\|\mathbf{w}^{i(t)}\right\|_2. \end{aligned} \tag{24}$$

We finally have

$$\begin{aligned} &tr(\mathbf{W}^{(t+1)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t+1)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t+1)}\right\|_F^2 + \gamma\left\|\mathbf{W}^{(t+1)}\right\|_2 \\ &\leq tr(\mathbf{W}^{(t)^T}\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}^{(t)}) \\ &+ \alpha\left\|\mathbf{X} - \mathbf{X}\mathbf{W}^{(t)}\right\|_F^2 + \gamma\left\|\mathbf{W}^{(t)}\right\|_2. \end{aligned} \tag{25}$$

According to Eq. (25), we can know that Algorithm 2 is going to be convergent.

3.7.2 Convergence Analysis of Algorithm 1

We prove the convergence of Algorithm 1 by the following Theorem 2:

Theorem 2. *The objective function value of Eq. (6) monotonically decreases until Algorithm 1 converges.*

Proof. After the tth iteration, we have obtained the optimal $\mathbf{A}^{(t)}$, $\mathbf{B}^{(t)}$ and $\mathbf{S}^{(t)}$. In the $(t+1)$th iteration, we need to optimize $\mathbf{S}^{(t+1)}$ by fixing $\mathbf{A}^{(t)}$ and $\mathbf{B}^{(t)}$. □

According to Eq. (18), we know that $s_{i,j}^{(t+1)}$ has a closed-form solution, i.e., global solution, for all $i, j = 1, \ldots, n$. Thus we have the following inequality:

$$\begin{aligned}
&\textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t)}\mathbf{B}^{(t)} - \mathbf{x}^j\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_2^2 s_{i,j}^{(t+1)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t+1)}||_2^2 + \gamma||\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_{2,1} \\
&\leq \textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t)}\mathbf{B}^{(t)} - \mathbf{x}^j\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_2^2 s_{i,j}^{(t)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t)}||_2^2 + \gamma||\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_{2,1}.
\end{aligned} \tag{26}$$

When fixing $\mathbf{S}^{(t+1)}$ to update $\mathbf{A}^{(t+1)}$ and $\mathbf{B}^{(t+1)}$, we have the following inequality according to Theorem 1

$$\begin{aligned}
&\textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)} - \mathbf{x}^j\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_2^2 s_{i,j}^{(t+1)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t+1)}||_2^2 + \gamma||\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_{2,1} \\
&\leq \textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t)}\mathbf{B}^{(t)} - \mathbf{x}^j\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_2^2 s_{i,j}^{(t+1)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t+1)}||_2^2 + \gamma||\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_{2,1}.
\end{aligned} \tag{27}$$

By integrating Eq. (26) with Eq. (27), we obtain

$$\begin{aligned}
&\textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)} - \mathbf{x}^j\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_2^2 s_{i,j}^{(t+1)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t+1)}||_2^2 + \gamma||\mathbf{A}^{(t+1)}\mathbf{B}^{(t+1)}||_{2,1} \\
&\leq \textstyle\sum_{i,j}^{n} ||\mathbf{x}^i\mathbf{A}^{(t)}\mathbf{B}^{(t)} - \mathbf{x}^j\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_2^2 s_{i,j}^{(t)} \\
&+ \alpha||\mathbf{X} - \mathbf{X}\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_F^2 \\
&+ \beta\textstyle\sum_{i}^{n} ||\mathbf{s}_i^{(t)}||_2^2 + \gamma||\mathbf{A}^{(t)}\mathbf{B}^{(t)}||_{2,1}.
\end{aligned} \tag{28}$$

From Eq. (28), we know that the objective function value of Eq. (6) decreases after each iteration of Algorithm 1. Hence, Theorem 2 has been proved.

3.7.3 Complexity Analysis

In each iteration, the time cost of Algorithm 1 focuses on the computation cost of $\mathbf{X}^T\mathbf{L}\mathbf{X} + \alpha\mathbf{X}^T\mathbf{X} + \gamma\mathbf{P}$, $(\mathbf{A}^T\mathbf{S}_t\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{X}$ in Eq. (10), and $\mathbf{f}_{i,j}$ in Eq. (18), and their corresponding complexity are $max\{O(nd^2), O(n^2d)\}$, $O(r^3)$, and $O(nd^2)$, where n, d, and r, respectively, are the number of the samples, the features, and the rank of the feature matrix $\mathbf{X}$. In our experiments, our method usually converges within 30 iterations, so the time complexity of Algorithm 1 is $max\{O(nd^2), O(n^2d)\}$ $(n, d \gg r)$.

3.7.4 Parameters' Determination

The parameter β determines the number of nearest neighbors of samples in the graph representation. Specifically, $\beta = 0$ means that only one element in s_i does not equal to zero, i.e., the number of nearest neighbors k is 1. $\beta \to \infty$ means that all elements in s_i are non-zero, i.e., the number of nearest neighbors k is n (the number of samples).

In this paper, we assume that there are k nearest neighbors for each sample. By denoting $\hat{\mathbf{f}}_i = \{\hat{f}_{i,1}, \ldots, \hat{f}_{i,n}\}$ as a descend order of $\mathbf{f}_i$, $i = 1, \ldots, n$, we know that Eq. (18) indicates the following constraint, i.e., $s_{i,k+1} = 0$ and $s_{i,k} > 0$. That is

$$\begin{cases} -\frac{1}{2\beta}\hat{f}_{i,k+1} + \tau \leq 0 \\ -\frac{1}{2\beta}\hat{f}_{i,k} + \tau > 0 \end{cases}. \tag{29}$$

Based on the constraint $\mathbf{s}_i^T\mathbf{1} = 1$, then we have

$$\begin{aligned}
&\textstyle\sum_{j=1}^{k}\Big(-\frac{1}{2\beta}\hat{f}_{i,k} + \tau\Big) = 1 \\
&\Rightarrow \tau = \frac{1}{k} + \frac{1}{2k\beta}\textstyle\sum_{j=1}^{k}\hat{f}_{i,k}.
\end{aligned} \tag{30}$$

By combining Eq. (29) with Eq. (30), we can obtain with the following inequality with respect to β,

$$\frac{k}{2}\hat{f}_{i,k} - \frac{1}{2}\sum_{j=1}^{k}\hat{f}_{i,j} < \beta \leq \frac{k}{2}\hat{f}_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k}\hat{f}_{i,j}. \tag{31}$$

Finally, we yield a closed-form solution s_i which has k non-zero elements, so we ultimately make β set as

$$\beta = \frac{k}{2}\hat{f}_{i,k+1} - \frac{1}{2}\sum_{v=1}^{k}\hat{f}_{i,v}, \tag{32}$$

where k is the number of nearest neighbors of ith samples and can be tuned by cross-validation methods.

For each $i, i = 1, \ldots, n$, we totally have Eq. (32) so that we have n different values on β. Hence, in our implementation, we follow the literature [20] to set the final value of β as the average of n different values on β, i.e.,

$$\beta = \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{k}{2}\hat{f}_{i,k+1} - \frac{1}{2}\sum_{v=1}^{k}\hat{f}_{i,v}\Big). \tag{33}$$

After fixing β, Eq. (6) needs to tune the parameters k, α, and γ. In this paper, we empirically determine the value of k since Eq. (6) can automatically adjust the value of $s_{i,j}$ (via assign small value to the neighbors far from the sample) if we set large value to k. α is used to balance the magnitude between $\sum_{i,j}^{n}\|\mathbf{x}^i\mathbf{A}\mathbf{B} - \mathbf{x}^j\mathbf{A}\mathbf{B}\|_2^2 s_{i,j}$ and $\|\mathbf{X} - \mathbf{X}\mathbf{A}\mathbf{B}\|_F^2$, and β is used for controlling the sparsity of $\mathbf{A}\mathbf{B}$. In this paper, we employ a cross-validation method to estimate them.

4 Experiments

In this section, we evaluate our proposed method by comparing with eight comparison methods on twelve data sets in terms of classification performance.

4.1 Datasets

We downloaded the data sets (such as HillValley, Ecoli, Cane, and Isolet) and the data sets (such as Yale-32, Colon,

WarpAR, Pixraw, Coil, DBWorld and Orl), respectively, from UCI Machine Learning Repository[1] and the website of Feature Selection Data sets.[2] We also downloaded the data set Lung from [55].

These data sets come from all kinds of applications, such as text data (such as Cane, DBWorld, and Isolet), biological data (such as Colon, Ecoli, and Lung), and image data (such as HillValley, Yale-32, WarpAR, Coil, Orl, and Pixraw). Moreover, three of them (such as Colon, DBWorld, and HillValley) are binary data sets and the others are multi-class data sets. The number of features is from 100 to 10,000, and the number of samples varies from 62 to 1,559. In particular, the number of features of seven data sets is larger than the number of samples, such as Colon, DBWorld, Lung, Pixraw, WarpAR, Yale-32, and Orl. This makes the construction of feature selection very challengeable.

4.2 Comparison Methods

Multi-Cluster Feature Selection [16] first solves an eigenvalue problem to construct the graph representation, and then utilizes the least square regression to connect the derived graph representation and the original data to rank the features.

Minimize the feature Redundancy for spectral Feature Selection [1] uses the $\ell_{2,1}$-norm regularizer to replace the ℓ_1-norm regularizer in MCFS to rank the features via considering the correlations among the features.

Nonnegative Discriminative Feature Selection [36] jointly learns the local geometric structure of the data and the sparse linear regression with an $\ell_{2,1}$-norm regularizer.

Joint Embedding Learning and Sparse Regression [37] simultaneously takes into account a Laplacian regularizer and the weight matrix to rank the scores of the features.

Trace Ratio formulation unsupervised Feature Selection (TRFS) [23] extends the criterion of trace ratio to unsupervised feature selection framework, via combining the k-means method with an $\ell_{2,1}$-norm regularizer into the proposed feature selection model.

Unsupervised Feature Selection with Adaptive Structure Learning [12] first utilizes the adaptive structure of the data to construct both the global learning and the local learning, and then integrates them with an $\ell_{2,1}$-norm regularizer to select the significant features.

Structured Optimal Graph Feature Selection [22] learns the global structure among the samples from the low-dimensional feature space to select important features.

Regularized Self-Representation (RSR) [8] uses the feature-level self-representation property to represent each feature by the important features, and then employs the $\ell_{2,1}$-norm regularizer to conduct group sparsity on the coefficient matrix, such that filtering the redundant and irrelative features.

The comparison methods include two sequential USFS methods (such MCFS and MRFS), three joint USFS methods (such as NDFS, JELSR, TRFS), two iteratively joint USFS methods (such as FSASL and SOGFS), and a newly unsupervised feature selection method (i.e., RSR). We also regarded the method using all features to conduct classification tasks as Baseline.

1. http://archive.ics.uci.edu/ml/.
2. http://featureselection.asu.edu/data sets.php.

In the comparison methods, first, RSR only utilizes the feature-level self-representation property to consider the local feature correlation for feature selection. Second, two sequential USFS methods only consider the local structure of the samples. Moreover, they learn the local structure from the original feature space, which may contain noise and redundancy. Furthermore, the sequential steps may result in suboptimal feature selection result. Last, these five joint USFS methods (i.e., NDFS, JELSR, TRFS, FSASL and SOGFS) jointly learn the geometry (either local or global) structure and conduct the selection of the features to avoid the suboptimal issue of sequential USFS methods. By contrast, our proposed method preserves both the global structure and the local structure among the samples, by considering both the global feature correlation and the local feature correlation among the features. Moreover, both the feature correlation and the sample correlation are learnt from the 'clean' data, i.e., the intrinsic low-dimensional space of the original high-dimensional data.

4.3 Experimental Setting

In our experiments, we first used all feature selection methods to selection features, and then ran the SVM classifier on the selected features to conduct classification tasks. For the method Baseline, we directly ran SVM to obtain the classification result.

We used 10-fold cross-validation to compare all methods. Specifically, we first randomly partitioned the whole data set into 10 subsets. We then selected one subset for testing and used the remaining 9 subsets for training. We repeated the whole process 10 times to avoid the possible bias during data set partitioning for cross-validation. The final result was computed by averaging results from all experiments. We conduct 5-fold cross-validation on the training data to conduct model selection. That is, we separated the training data into five parts, where one of parts is used to validate the model built by the left four parts. In the validation step, we used the grid search method to search the best parameters' combination by the given ranges of the parameters. We selected the parameters' combination with the best classification performance in the validation step to test the testing data. In particular, we empirically set the value of k as 15 and other parameters' range as $\{10^{-3}, \ldots, 10^{3}\}$ for all methods to make fair comparison, where all the methods obtained their best performance.

We evaluated our method with all the comparison methods via the evaluation metric Average Classification Accuracy (ACA). We also investigated the robustness of our proposed method in terms of three aspects, such as the effect of low-rank constraint, the influence of parameters' setting, and the convergence of our proposed Algorithm 1.

4.4 Experimental Result on Classification Accuracy

Fig. 1 reported the ACA result of all methods, where the horizontal axis represented the number of the left dimensions after conducting feature selection.

Obviously, our proposed method achieved the best performance, followed by SOGFS, FSASL, RSR, NDFS, JELSR, TRFS, MRFS, MCFS, and Baseline. For example, our proposed method improved by 10.1 and 24.7 percent,

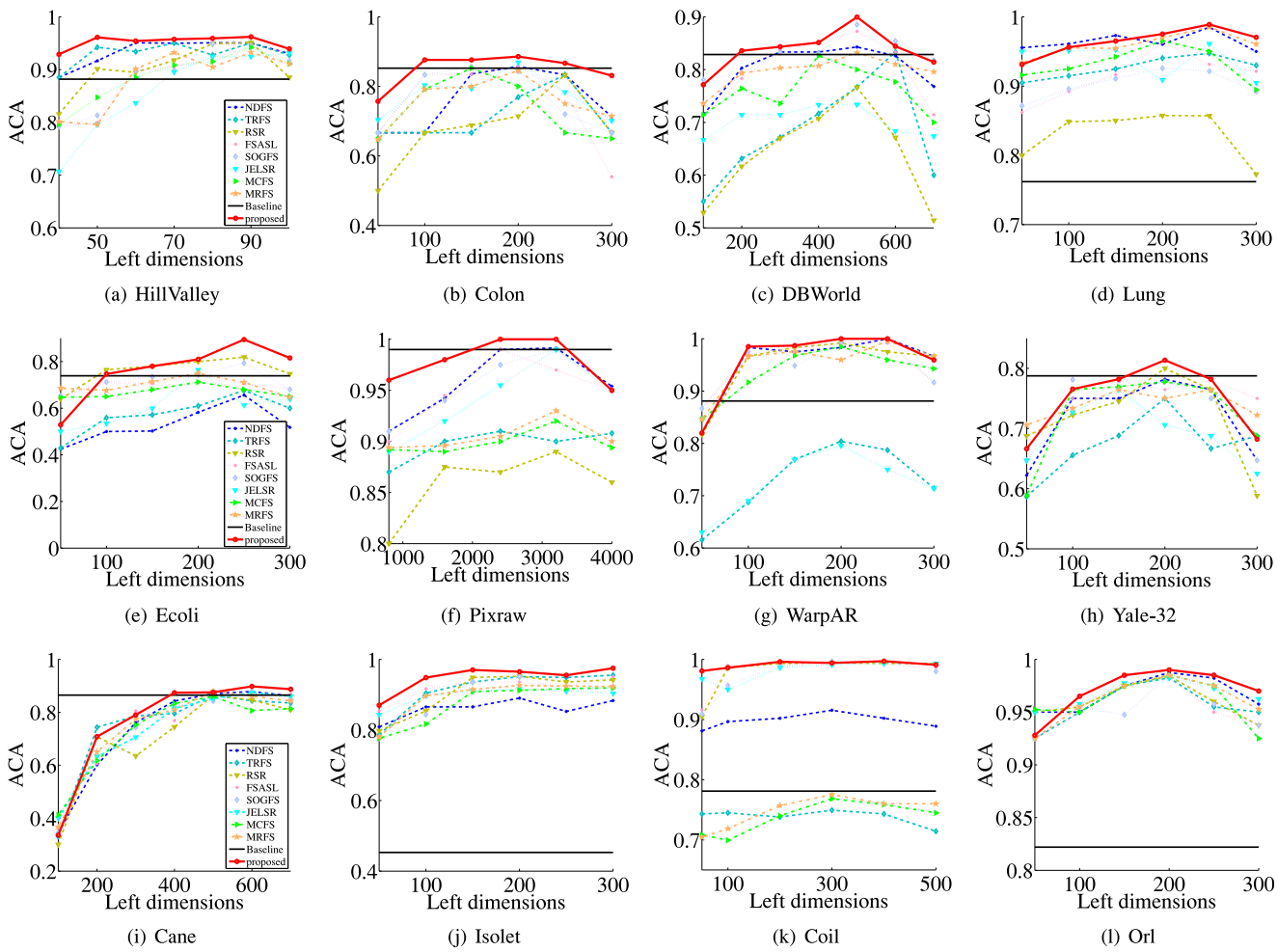


Fig. 1. ACA result of all methods on all data sets at different number of selected features.

respectively, compared to SOGFS (the best comparison method) in data set Ecoli and MCFS in data set Coil (the worst feature selection method). Besides, we had the following observation.

First, the classification performance of all feature selection methods first increased and then began to decrease with the increase of the selected features. For example, the ACA results were about 65 and 82 percent, respectively, while keeping the left features as 50 and 200, and then went down to 65 percent while keeping the left features as 300, at the data set Yale-32. This indicated that it is necessary to conduct feature selection for dealing with high-dimensional data since high-dimensional data contain noise or redundancy.

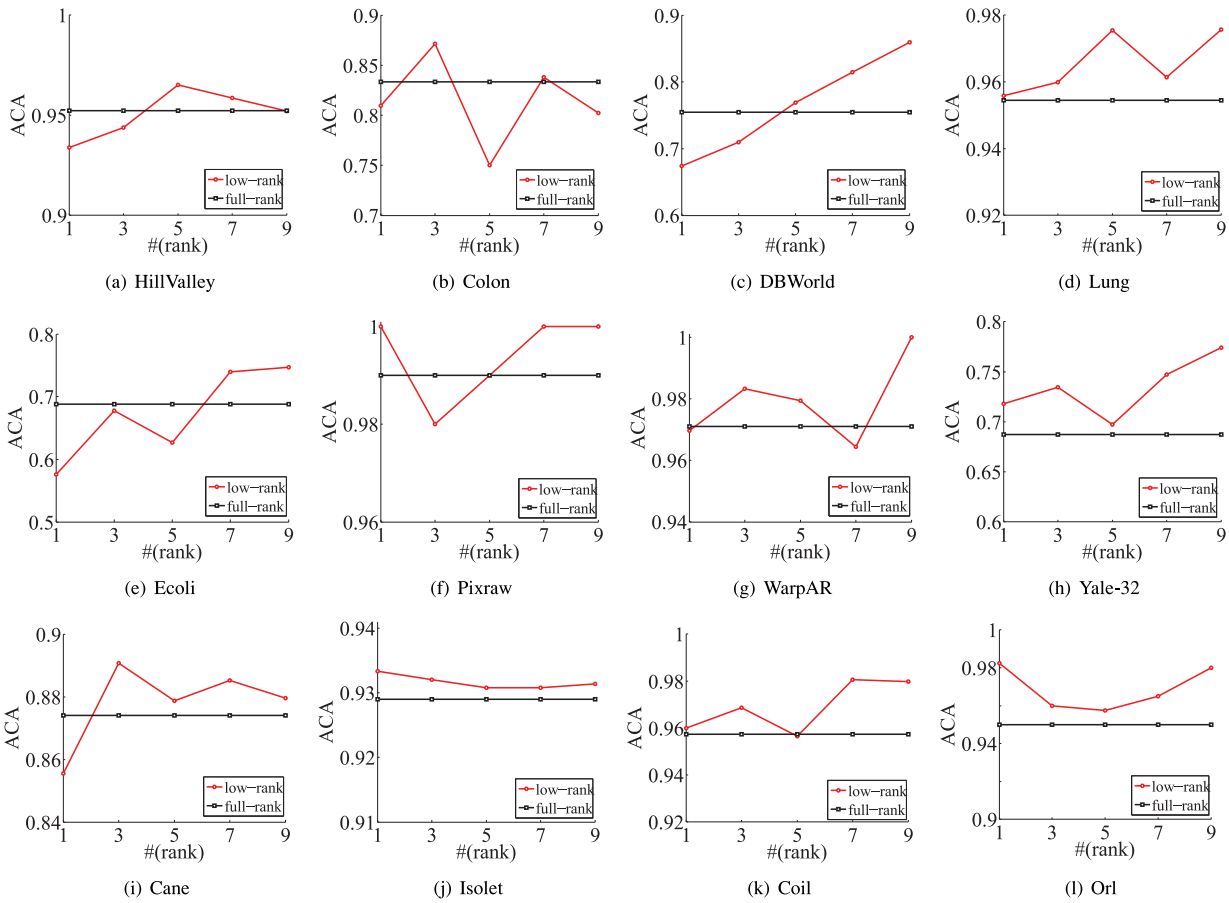
Second, most of feature selection methods outperformed Baseline, which used all features to conduct classification. For example, our proposed method and MCFS (the worst feature selection method), respectively, improved on average by 17.9 and 7.4 percent, compared to Baseline. This verified the necessary of conducting feature selection for dealing with high-dimensional data again.

Last but not least, sequential USFS methods (i.e., MCFS and MRFS) were worse than joint USFS methods (i.e., NDFS, TRFS, FSASL, SOGFS and JELSR). For example, the average classification accuracy of our proposed method on average increased by 23.21, 18.83, 4.41, 6.40, 4.33, 15.39, 9.28, 6.53 and 2.36 percent, respectively, than the performance of NDFS, TRFS, RSR, FSASL, SOGFS, JELSR, MCFS and the Baseline, on the data set Ecoli. The reason may be that sequential USFS methods sequentially conduct subspace learning and feature selection to possible result in suboptimal result.

4.5 Effect of Low-Rank Constraint

We investigated the influence of the effect of different number of ranks (i.e., $r \in \{1, 3, 5, 7, 9\}$) in Eq. (6) at different data sets, and reported the ACA result in Fig. 2, where the horizonal axis indicates the number of kept ranks. It is worth noting that the number of real classes in both the binary data sets (such as Colon, HillValley, and DBWorld) and the multi-class data sets (such as Lung and Ecoli) is less than 9, but we still set the rank of their feature matrices as 9 since the real rank of these corresponding data sets is large than 9.

From Fig. 2, we observed that the performance with a low-rank constraint in most of cases outperformed the performance of the cases with full-rank. For example, the average classification accuracy of the proposed method with low-rank constraints increased by 1.12, 4.67, 0.27, 1.17 and 1.9 percent, respectively, compared to the results of our proposed method with the full-rank constraint on the data sets Lung, Yale-32, Isolet, Coil, and Orl. This manifested that it is reasonable to analyze high-dimensional data with a low-rank constraint in feature selection. The reason is that the low-rank constraint conducting subspace learning helped find the low-dimensional space of high-dimensional data via considering the global feature correlation.

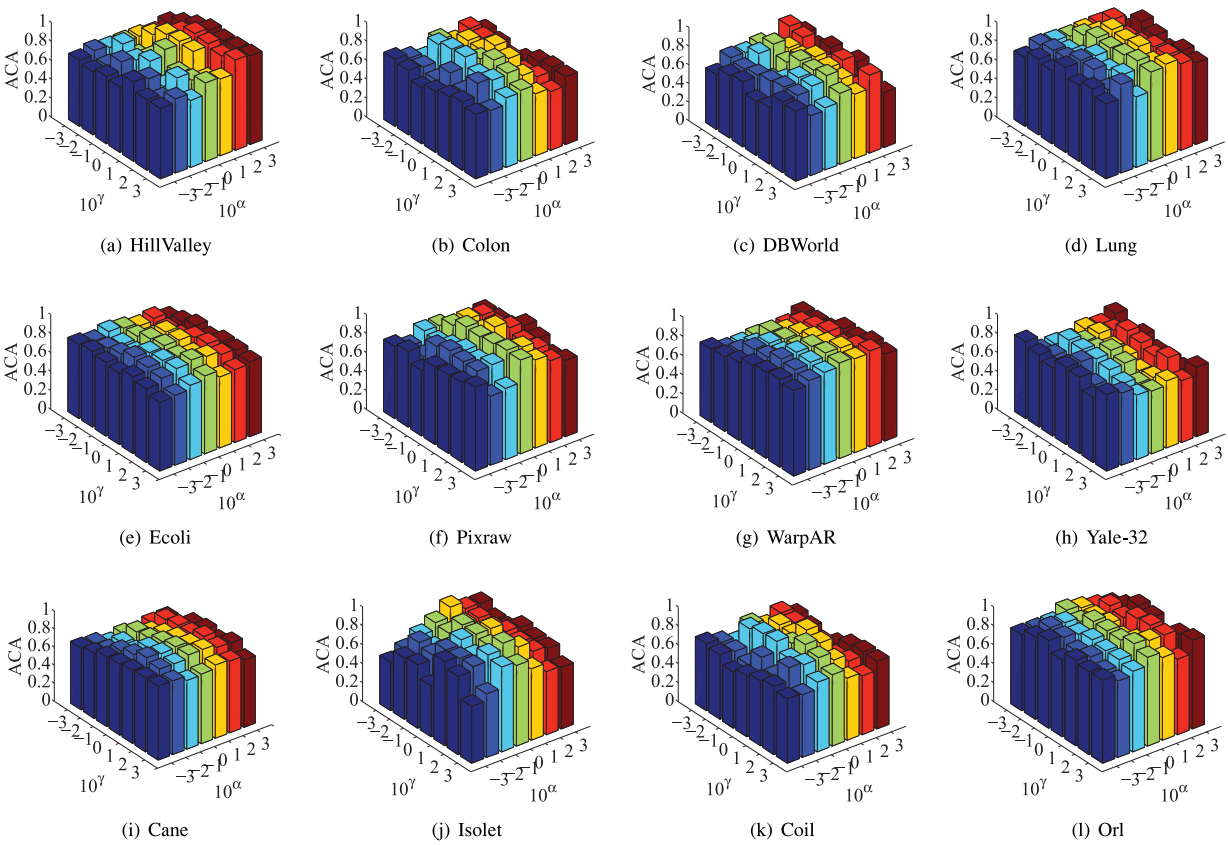


(a) HillValley (b) Colon (c) DBWorld (d) Lung

(e) Ecoli (f) Pixraw (g) WarpAR (h) Yale-32

(i) Cane (j) Isolet (k) Coil (l) Orl

Fig. 2. ACA result of our proposed method at different number of ranks.



(a) HillValley (b) Colon (c) DBWorld (d) Lung

(e) Ecoli (f) Pixraw (g) WarpAR (h) Yale-32

(i) Cane (j) Isolet (k) Coil (l) Orl

Fig. 3. ACA result of our proposed method at different parameters' setting on the variables α and γ.

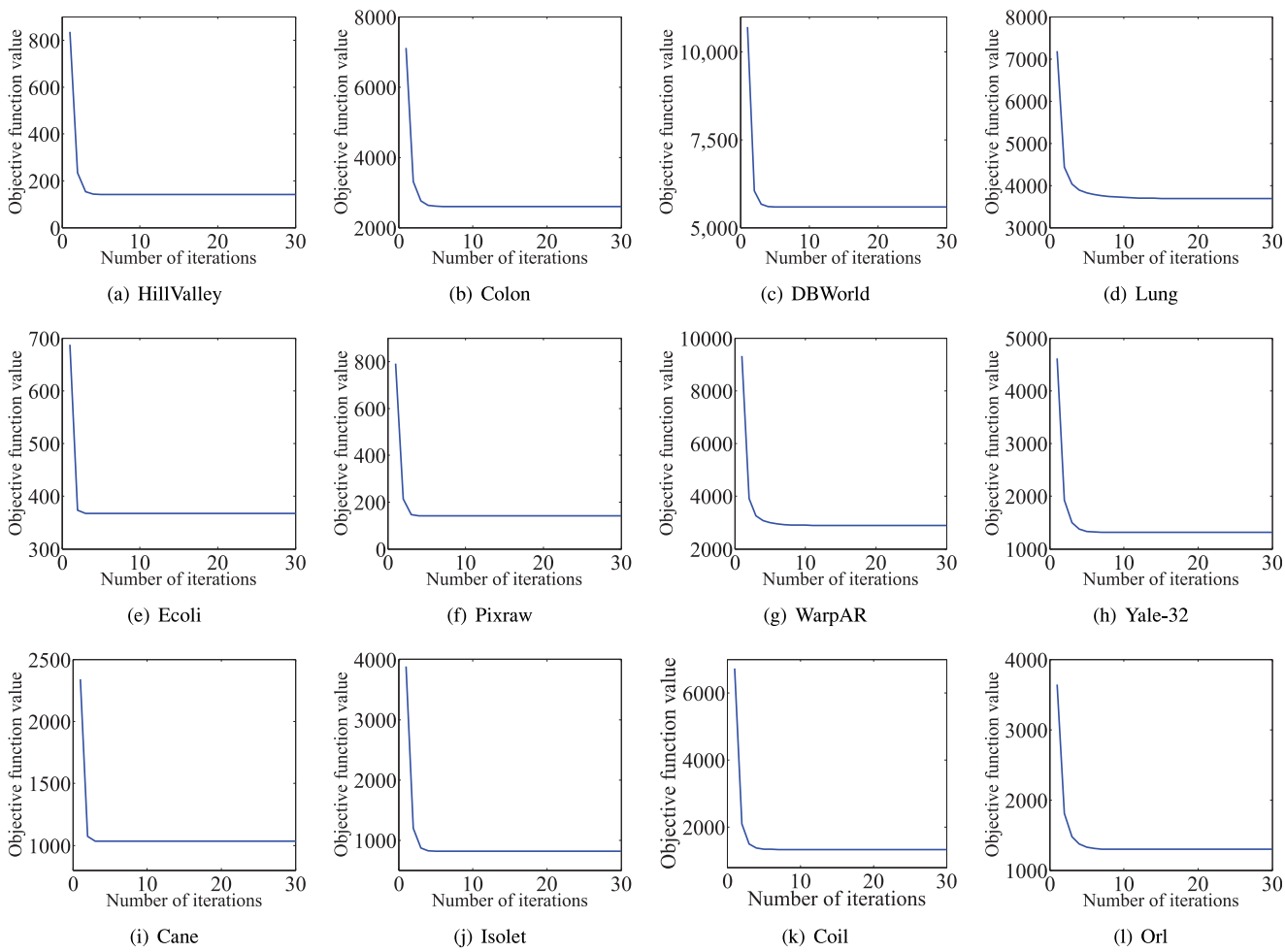


Fig. 4. ACA result of our proposed Algorithm 1 at different iterations on different datasets.

4.6 Parameters' Sensitivity

We tuned the parameters α and γ within the range of $\{10^{(-3)}, 10^{(-2)}, \ldots, 10^{3}\}$ and listed the results in Fig. 3.

As shown in Fig. 3, the proposed method is sensitive to the parameters' setting. That is, different parameter combinations output different classification results. Hence, it is necessary to tune the parameters in our methods. More specifically, α is used to control the magnitude between the local representation term $\sum_{i,j}^{n} \|\mathbf{x}^i\mathbf{AB} - \mathbf{x}^j\mathbf{AB}\|_2^2 s_{i,j}$ and the global representation term $\|\mathbf{X} - \mathbf{XAB}\|_F^2$, while γ in Eq. (6) is used to adjust the sparsity of $\mathbf{AB}$. In Fig. 3, we can find that our method achieves the best performance on the data sets Ecoli and Isolet while setting $\alpha = 1$, and $\gamma = 100$. However, our method produces the best ACA 96.03 percent with $\alpha = 0.01$, and $\gamma = 1$ for the data set Yale-32.

4.7 Convergence

Fig. 4 shows the behavior of the objective values of our proposed optimization algorithm (i.e., Algorithm 1) with respect to the increase of the iterations. In our experiments, we set the stop criteria of both Algorithms 1 and 2 as 10^{-3}, i.e., $\frac{\|obj(t+1)-obj(t)\|_2^2}{obj(t)} \leq 10^{-3}$, where $obj(t)$ represents the tth iteration objective function value of Eq. (6).

From Fig. 4 we can find 1) the proposed Algorithm 1 to optimize the proposed objective function in Eq. (6) monotonically decreases the objective function values until Algorithm 1 achieves converges; 2) the proposed Algorithm 1 needs a few iterations (i.e., less than 20) to reach the convergence, which is very efficient.

It is noteworthy that our proposed Algorithm 2 also achieves convergence within 30 iterations at all data sets. We did not list them due to the limited space.

5 Conclusion

This paper has proposed a novel unsupervised spectral features selection method by iteratively learning the graph matrix and selecting the features. Specifically, we embedded the feature-level self-expressiveness property, a low-rank constraint, the graph matrix learning, and an $\ell_{2,1}$-norm regularizer in a framework, to yield an interpretable and robust low-dimensional space and the graph matrix measuring the similarity in the learnt low-dimensional space. Experimental results on real data sets verified that our proposed method achieved the best classification performance, compared to the state-of-the-art feature selection methods.

In the future work, we will extend our proposed framework to conduct feature selection on the high-dimensional data with incomplete data since incomplete data sets are often found in industrial applications.

ACKNOWLEDGMENTS

This work was supported in part by the China Key Research Program (Grant No: 3722016YFB1000905), the Nation Natural Science Foundation of China (Grants No: 61573270 and 61672177), the China 1000-Plan National Distinguished Professorship, the Guangxi Natural Science Foundation (Grant No: 2015GXNSFCB139011), the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents, the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, the Research Fund of Guangxi Key Lab of MIMS (16-A-01-01 and 16-A-01-02), the Innovation Project of Guangxi Graduate Education (YCSW2017039), and the Guangxi Bagui Teams for Innovation and Research. Rongyao Hu and Yonghua Zhu have equivalent contributions to this work.

REFERENCES

[1] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.

[2] R. Hu, et al., "Graph self-representation method for unsupervised feature selection," *Neurocomput.*, vol. 220, pp. 130–137, 2017.

[3] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, Jan. 2011.

[4] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, and A. Cichocki, "Sparse Bayesian classification of EEG for brain–computer interface," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2256–2267, Nov. 2016.

[5] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2017.2673241.

[6] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[7] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 153–160.

[8] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, 2015.

[9] Y. Zhang, Y. Wang, J. Jin, and X. Wang, "Sparse Bayesian learning for obtaining sparsity of eeg frequency bands based feature vectors in motor imagery classification," *Int. J. Neural Syst.*, vol. 27, no. 02, 2017, Art. no. 1650032.

[10] X. Zhu, Z. Huang, Y. Yang, H. T. Shen, C. Xu, and J. Luo, "Self-taught dimensionality reduction on the high-dimensional small-sized data," *Pattern Recognit.*, vol. 46, no. 1, pp. 215–229, 2013.

[11] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, 2017, Art. no. 43.

[12] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 209–218.

[13] X. Zhu, L. Zhang, and Z. Huang, "A sparse embedding and least variance encoding approach to hashing," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3737–3750, Sep. 2014.

[14] Y. Zhang, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Frequency recognition in SSVEP-based BCI using multiset canonical correlation analysis," *Int. J. Neural Syst.*, vol. 24, no. 04, 2014, Art. no. 1450013.

[15] Z. Zhang, L. Bai, Y. Liang, and E. Hancock, "Joint hypergraph learning and sparse regression for feature selection," *Pattern Recognit.*, vol. 63, pp. 291–309, 2017.

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

[17] L. Shi, L. Du, and Y.-D. Shen, "Robust spectral learning for unsupervised feature selection," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 977–982.

[18] C. Hou, F. Nie, X. Li, and D. Yi, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[19] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.

[20] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2033–2044, Sep. 2017.

[21] X. Zhu, H. Suk, L. Wang, S. Lee, and D. Shen, "A novel relational regularization feature selection method for joint regression and classification in AD diagnosis," *Med. Image Anal.*, vol. 38, pp. 205–214, 2017.

[22] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1302–1308.

[23] D. Wang, F. Nie, and H. Huang, "Unsupervised feature selection via unified trace ratio formulation and K-means clustering (TRACK)," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 306–321.

[24] X. Zhu, H.-I. Suk, S.-W. Lee, and D. Shen, "Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 607–618, Mar. 2016.

[25] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, "Quantization-based hashing: A general framework for scalable image and video retrieval," *Pattern Recognit.*, 2017. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.03.021

[26] X. Zhu, H.-I. Suk, H. Huang, and D. Shen, "Low-rank graph-regularized structured sparse regression for identifying genetic biomarkers," *IEEE Trans. Big Data*, vol. 3, no. 4, pp. 405–414, Oct.-Dec. 2017.

[27] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017.

[28] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.

[29] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural Netw. Signal Process. IX*, vol. 1, no. 1, 1999, Art. no. 1.

[30] X. Zhu, Z. Huang, H. T. Shen, J. Cheng, and C. Xu, "Dimensionality reduction by mixed kernel canonical correlation analysis," *Pattern Recognit.*, vol. 45, no. 8, pp. 3003–3016, 2012.

[31] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.

[32] J. Yang, A. F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, Feb. 2005.

[33] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017.

[34] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[35] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[36] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 1026–1032.

[37] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[38] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[39] P. Smialowski, D. Frishman, and S. Kramer, "Pitfalls of supervised feature selection," *Bioinf.*, vol. 26, no. 3, pp. 440–443, 2010.

[40] K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: Constraint, relevance, and redundancy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1131–1143, May 2014.

[41] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 793–802.

[42] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.
[43] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, 2017.
[44] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 22, no. 1, 2011, Art. no. 1294.
[45] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.
[46] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[47] L. Chen and J. Z. Huang, "Sparse reduced-rank regression for simultaneous dimension reduction and variable selection," *J. Amer. Statistical Assoc.*, vol. 107, no. 500, pp. 1533–1545, 2012.
[48] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, vol. 11, pp. 2057–2078, 2010.
[49] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 705–712.
[50] H. Wang, F. Nie, and H. Huang, "Globally and locally consistent unsupervised projection," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1328–1333.
[51] J. Yang, D. Zhang, J.-Y. Yang, and B. Niu, "Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 650–664, Apr. 2007.
[52] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, 2010.
[53] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[54] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[55] D. Singh, et al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

Xiaofeng Zhu is a faculty member with Guangxi Normal University, China. His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.

Shichao Zhang is a China 1000-Plan distinguished professor with the Guangxi Normal University, China. His research interests include data mining and partitioning. He is a senior member of the IEEE and a member of the ACM.

Rongyao Hu is working toward the master's degree at Guangxi Normal University, China. His current research interests include data mining and pattern recognition.

Yonghua Zhu is working toward the master's degree at Guangxi University, China. His current research interests include data mining and machine learning.

Jingkuan Song is a full professor with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include multimedia data analysis and image retrieval.