Robust Joint Graph Sparse Coding for Unsupervised Spectral Feature Selection

Xiaofeng Zhu, Xuelong Li, *Fellow, IEEE*, Shichao Zhang, *Senior Member, IEEE*,
Chunhua Ju, and Xindong Wu, *Fellow, IEEE*

***Abstract*— In this paper, we propose a new unsupervised spectral feature selection model by embedding a graph regularizer into the framework of joint sparse regression for preserving the local structures of data. To do this, we first extract the bases of training data by previous dictionary learning methods and, then, map original data into the basis space to generate their new representations, by proposing a novel joint graph sparse coding (JGSC) model. In JGSC, we first formulate its objective function by simultaneously taking subspace learning and joint sparse regression into account, then, design a new optimization solution to solve the resulting objective function, and further prove the convergence of the proposed solution. Furthermore, we extend JGSC to a robust JGSC (RJGSC) via replacing the least square loss function with a robust loss function, for achieving the same goals and also avoiding the impact of outliers. Finally, experimental results on real data sets showed that both JGSC and RJGSC outperformed the state-of-the-art algorithms in terms of *k*-nearest neighbor classification performance.**

***Index Terms*— Dimensionality reduction, manifold learning, regression, sparse coding.**

I. Introduction

HIGH-DIMENSIONAL data can be found in all kinds of real applications, such as text mining, image retrieval, and visual recognition [1]–[4]. Although contemporary computers can solve some problems of high-dimensional data, e.g., the issue of time consuming, learning high-dimensional data often suffer from a number of issues, such as the curse of dimensionality and the impact of noise and redundancy. Moreover, it has been shown that the intrinsic dimensionality of high-dimensional data is typically small [5]–[9]. Thus, there have been great interests for searching for such low-dimensional intrinsic dimensionality on high-dimensional data.

Dimensionality reduction techniques (such as feature selection and subspace learning) have been widely used to solve this problem by reducing the dimensions of features [10]–[13]. Feature selection methods, such as statistical t-test and sparse linear regression, find informative feature subsets from original feature set [8], [14], [15], while subspace learning methods, such as Fisher's linear discriminant analysis, canonical correlation analysis [16], [17], and locality preserving projection (LPP), transform original features into a low-dimensional space [8], [15]. In regards to the interpretability of the results, feature selection methods are preferable over subspace learning methods.

Spectral feature selection methods have become effective solutions for dealing with high-dimensional data, because they consider both manifold learning and feature selection for reducing the dimensions of data [14], [18]–[21]. Its rationale is to preserve the local structures of high-dimensional data via manifold learning (i.e., preserving the similarity of high-dimensional data in a low-dimensional space) and to remove redundant features by a sparse regression (i.e., a concatenation of a least square loss function and a sparse regularizer). In this paper, we mathematically formulate the objective function of spectral feature selection methods as

$$\mathcal{F}(\mathbf{S}) = \mathcal{G}(\mathbf{S}) + \mathcal{R}(\mathbf{S}) \tag{1}$$

where $\mathcal{G}(\mathbf{S})$ and $\mathcal{R}(\mathbf{S})$, respectively, are graph-based loss function and regularizer on coefficient matrix $\mathbf{S}$. $\mathcal{G}(\mathbf{S})$ is designed to conduct manifold learning for preserving the local structure of the data, while $\mathcal{R}(\mathbf{S})$ is designed for satisfying the conditions, such as avoiding the issue of overfitting and outputting the sparsity. In multicluster feature selection (MCFS) [18] [see Fig. 1(a) and Table I], $\mathcal{G}(\mathbf{S})$ is a least square loss function, while $\mathcal{R}(\mathbf{S})$ is an ℓ_1-norm regularizer for generating sparse scores of the features. More specifically, the graph-based loss function of MCFS includes two stages, i.e., an eignevalue problem on original data to output the graph representation of original data, and a least square regression between the graph presentation and the class labels, respectively. Minimize the feature redundancy for spectral feature selection (MRSF) [21] [see Fig. 1(b) and Table I] used

Manuscript received May 2, 2015; revised January 18, 2016; accepted January 22, 2016. The work of X. Zhu and S. Zhang was supported in part by the China 973 Program under Grant 2013CB329404, in part by the National Natural Science Foundation of China under Grant 61450001, Grant 61263035, and Grant 61573270, in part by the Guangxi Natural Science Foundation under Grant 2012GXNSFGA060004 and Grant 2015GXNSFCB139011, in part by the Guangxi 100 Plan, in part by the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, and in part by the Guangxi Bagui Scholar Teams for Innovation and Research Project. *(Corresponding authors: Shichao Zhang and Xindong Wu.)*

X. Zhu is with the Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China (e-mail: seanzhuxf@gmail.com).

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China (e-mail: xuelong_li@opt.ac.cn).

S. Zhang and C. Ju are with the School of Computer Science and Information Technology, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: zhangsc_gxnu@163.com; juchunhua@hotmail.com).

X. Wu is with the Department of Computer Science, University of Vermont, Burlington, VT 05405 USA (e-mail: xwu@uvm.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2521602

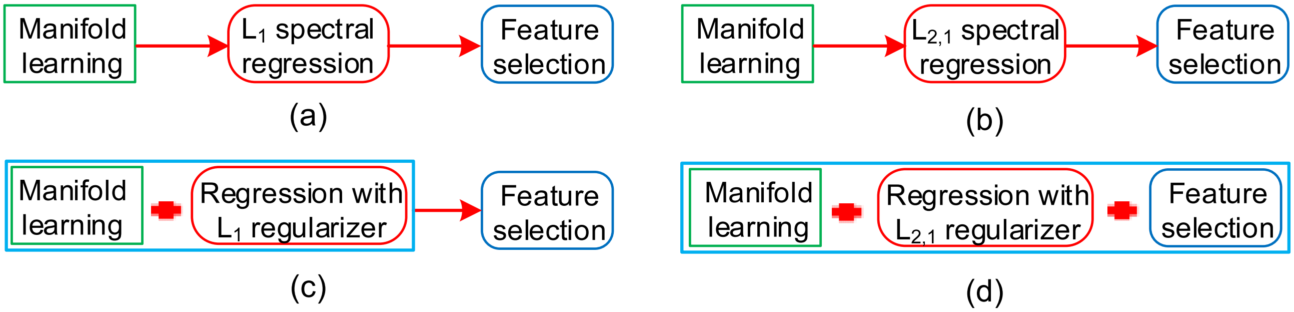


Fig. 1. Illustration of spectral feature selection methods. (a) MCFS. (b) MRSF. (c) GSC. (d) JGSC.

TABLE I

MATHEMATICAL COMPARISON OF SPECTRAL FEATURE SELECTION METHODS DEFINED BY THE FORMULATION $\mathcal{F}(\mathbf{S}) = \mathcal{G}(\mathbf{S}) + \mathcal{R}(\mathbf{S})$

	$\mathcal{G}(\mathbf{S})$	$\mathcal{R}(\mathbf{S})$
MCFS	$(1)\mathbf{Ly} = \lambda\mathbf{Dy}$; (2) $\|\mathbf{y} - \mathbf{xs}\|_F^2$	$\|\mathbf{s}\|_1$
MRSF	$(1)\mathbf{Ly} = \lambda\mathbf{Dy}$; (2) $\|\mathbf{y} - \mathbf{xs}\|_F^2$	$\|\mathbf{s}\|_{2,1}$
GSC	$\|\mathbf{X} - \mathbf{BS}\|_F^2 + \alpha tr(\mathbf{SLS}^T)$	$\|\mathbf{S}\|_1$
JGSC	$\|\mathbf{X} - \mathbf{BS}\|_{2,1} + \alpha tr(\mathbf{SLS}^T)$	$\|\mathbf{S}\|_{2,1}$
RJGSC	$\|\mathbf{X} - \mathbf{BS}\|_F^2 + \alpha tr(\mathbf{SLS}^T)$	$\|\mathbf{S}\|_{2,1}$

an $\ell_{2,1}$-norm regularizer to replace the ℓ_1-norm regularizer of MCFS. The reason is that: 1) the ℓ_1-norm regularizer outputs the sparsity in elements, thus prohibiting for feature selection on multiclass data and 2) the $\ell_{2,1}$-norm regularizer outputs the sparsity through the whole row (corresponding to one feature of the data) and has been widely used in real applications [21], [22]. These two spectral feature selection methods sequentially conduct manifold learning and sparse regression, and easily lead to a suboptimum of feature selection [22]. Thus, integrating manifold learning and sparse regression into a unified framework should be interesting for conducting spectral feature selection.

Graph sparse coding (GSC) [see Fig. 1(c) and Table I] simultaneously conducts manifold learning and a sparse regression to improve the performance in both classification tasks [19] and clustering tasks [14]. That is, $\mathcal{G}(\mathbf{S})$ of GSC includes a least square loss function and a graph-based regularizer, while $\mathcal{R}(\mathbf{S})$ of GSC is an ℓ_1-norm regularizer. In GSC, the least square loss function is used for achieving minimal regression error, and the graph-based regularizer is designed for preserving the local structure of the data. The GSC generates the element sparsity, thus prohibiting for conducting feature selection.

In this paper, we extend [23] to devise a joint GSC (JGSC) method [see Fig. 1(d) and Table I] for conducting classification on high-dimensional data, by simultaneously performing manifold learning and joint sparse regression. The proposed JGSC consists of three key steps.

1) *Basis Extraction:* The bases are derived from the training data via an existing dictionary learning method, such as [24] and [25].
2) *Data Reconstruction:* All the data are mapped into the resulting basis space to generate their new representations by the proposed JGSC model.
3) *Feature Selection:* Due to introducing the $\ell_{2,1}$-norm regularizer, the derived representations of the data contain many zero rows.

This shows that the zero rows (i.e., zero-valued features) of the new representations of the data are unimportant. To achieve efficiency and effectiveness, we delete those rows to obtain a reduced data set. To avoid the impact of the outliers, we also extend the proposed JGSC to a robust JGSC (RJGSC) by replacing the least square loss function of JGSC with a robust loss function.

The contribution of this paper is twofold. First, we identify limitations in previous spectral feature selection methods, such as the issue of the ℓ_1-norm regularizer prohibiting for conducting feature selection on the multiclass data, and the issue that separately conducting manifold learning and sparse regression results in a suboptimum solution. To address these issues, we integrate manifold learning and feature selection into a unified framework, which easily enables the objective function to achieve a global optimum. Moreover, we extend our solution to the RJGSC method to avoid the impact of the outliers. In contrast, both the MCFS and the MRSF separately conduct manifold learning and sparse regression to result in a suboptimum solution. Although the GSC simultaneously conducts manifold learning and an ℓ_1-norm sparse regression, but prohibiting for feature selection.

Second, unlike previous feature selections [18], [21], [26] conduct feature selection on original feature space, our models conduct feature selection on the basis space of the data, which has been shown to be higher level and more abstract representation than the low-level representation, such as raw pixel intensity values [13], [27].

II. RELATED WORK

In this section, we give a brief review of the previous feature selection methods and sparse learning methods.

A. Feature Selection

Given a data set with a large number of features, if some of them are irrelevant, feature selection removes the irrelevant features and, then, outputs relevant features. Feature selection is popular in many real applications [16], [28], such as information retrieval, image analysis, intrusion detection, and bioinformatics. According to design strategies, previous feature selection methods can be broadly categorized into three groups: 1) filter model; 2) wrapper model; and 3) embedded model [28]–[30].

A filter model is usually designed to first analyze the general characteristics of the data and further evaluate the features without involving a learning algorithm. In the existing filter model, feature selection is decided by the predefined criteria, such as mutual information [31] and variable ranking [32]. For example, Laplacian score (LScore) method ranks the features by evaluating the power of locality preservation of each feature [33]. In real applications, the filter model is (relatively) robust against the issue of overfitting, but may fail to select the most useful features.

A wrapper model wraps the selection process to identify relevant features while requiring a predetermined learning algorithm [34]. For example, Maldonado and Weber [35] proposed to find a subset of all the features by maximizing the performance of an Support Vector Machine (SVM) classifier. In real applications, the wrapper model can, in principle, find the most useful features, so it often outperforms filter model. However, the wrapper model is with high computation cost and prone to the issue of overfitting.

An embedded model performs feature selection during the process of model construction [18], [22], [36], [37]. Thus, it usually regards feature selection as a part of the training process, in which useful features are obtained by optimizing the objective function. Recently, embedded models receive increasing interests due to its superior performance. For example, Weston *et al.* [38] added an ℓ_0-norm regularizer into the proposed objective function to achieve sparse solution for performing feature selection, while Liu *et al.* [39] employed an $\ell_{2,1}$-norm regularizer to achieve the similar objective. The embedded model is similar to the wrapper model, but is with less computation cost and less prone to the issue of overfitting. It is noteworthy that the methods (including MCFS, MRSF, GSC, JGSC, and RJGSC) are embedded models.

B. Sparse Learning

The objective function of the traditional sparse learning can be represented as a concatenation of a loss function and a regularizer. Loss function is used to achieve a minimal regression (or reconstruction) error. Existing loss functions include least square loss function, logistic loss function, and squared hinge loss function. The regularizer is often used to generate the sparse results in sparse learning.

Sparse learning distinguishes important elements from unimportant ones by assigning the codes of unimportant elements as zeros and the important ones as nonzeros. This enables sparse learning to reduce the impact of noise and increase the efficiency of learning models [1], [25]; thus, it has been used for various real applications, such as signal classification [4], [8], [40], face recognition [13], [41], and medical image analysis [42].

Studies [40], [43]–[45] showed that different regularizers encourage various sparsity patterns in sparse learning. According to the way to generate sparsity patterns, we categorize existing sparse learning models into two categories: 1) separable sparse learning [41], [43], [46] and 2) joint sparse learning [29], [47]. Separable sparse learning encodes each sample individually, while joint sparse learning simultaneously encodes all samples.

Separable sparse learning employs different regularizers to output different sparse patterns. For example, an ℓ_1-norm regularizer [43], [44], [48] leads to the element sparsity and an $\ell_{2,1}$-norm regularizer [49] results in the group sparsity, while a mixed-norm regularizer combining an ℓ_1-norm with an $\ell_{2,1}$-norm [50] produces the mixed sparsity. In particular, the ℓ_1-norm regularizer makes each element as a singleton to generate the elementwise sparsity, while the group sparsity forces a group condition within one column as a singleton to generate the sparsity in the whole group. Obviously, the $\ell_{2,1}$-norm regularizer takes the natural group structure in one sample into account. The mixed sparsity [49]–[51] was explained as first generating the group sparsity for each sample and then generating the element sparsity in the dense (i.e., nonsparse) groups.

The regularizers, such as the $\ell_{2,1}$-norm regularizer [27], [41], [52], the $\ell_{2,1}^2$-norm regularizer [47], and the $\ell_{1,\infty}$-norm regularizer [53], are often used in joint sparse learning. Different from separable sparse learning, joint sparse learning simultaneously encodes all samples by considering their correlations. For example, the row sparsity (via an $\ell_{2,1}$-norm regularizer) enables all samples to be encoded at the same time, and the sparsity goes through the whole row. The block sparsity [40], [54] via an F-norm regularizer considers the group structure over multiple rows, so it generates the sparsity through the whole block, i.e., multiple rows. The mixed joint sparsity is the combination of the block sparsity and the row sparsity [54].

III. Approach

In this section, we first discuss the generation of the bases of the data and, then, give details of the proposed JGSC. Furthermore, we describe the process of conducting feature selection on the new representations. Finally, we extend the proposed JGSC to the RJGSC.

A. Notations

In this paper, we denote matrices as boldface uppercase letters, vectors as boldface lowercase letters, and scale as normal italic letter, respectively. Given a matrix $\mathbf{X}$, its ith row, jth column, and the element in the ith row and the jth column, respectively, are denoted by $\mathbf{x}^i$, $\mathbf{x}_j$, and $x_{i,j}$. We denote the Frobenius norm and $\ell_{2,1}$-norm of a matrix as $\|\mathbf{X}\|_F^2 = (\sum_i \|\mathbf{x}^i\|_2^2)^{1/2} = (\sum_j \|\mathbf{x}_j\|_2^2)^{1/2}$ and $\|\mathbf{X}\|_{2,1} = \sum_i \|\mathbf{x}^i\|_2 = \sum_i (\sum_j x_{i,j}^2)^{1/2}$, respectively. We also denote the transpose operator, the trace operator, and the inverse of a matrix $\mathbf{X}$ as $\mathbf{X}^T$, tr($\mathbf{X}$), and $\mathbf{X}^{-1}$, respectively.

B. Basis Extraction

In this section, we briefly review the process of learning the bases of the data. In this paper, we used an online dictionary learning method [25] to learn the bases.

Given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ (where each column represents a data point), we want to learn m bases (or dictionaries)[1] $\mathbf{B} \in \mathbb{R}^{d\times m}$ to generate the new representations $\hat{\mathbf{S}} \in \mathbb{R}^{m\times n}$ of $\mathbf{X}$. The objective function is defined as

$$\min_{\{\mathbf{B},\hat{\mathbf{S}}\}} \|\mathbf{X} - \mathbf{B}\hat{\mathbf{S}}\|_F^2 + \lambda \sum_{i=1}^{n} \|\hat{\mathbf{s}}_i\|_1, \quad \text{s.t.} \sum_{i=1}^{n}\sum_{j=1}^{d} b_{i,j}^2 \le 1 \tag{2}$$

where $\sum_{i=1}^{n}\sum_{j=1}^{d} b_{i,j}^2 \le 1$ is to prevent $\mathbf{B}$ from having arbitrarily large values, which would lead to very small values of $\hat{\mathbf{S}}$. We use the package SPAMS [25] to learn the bases $\mathbf{B}$. It should be noteworthy that (2) also outputs the new representation $\hat{\mathbf{S}}$ of $\mathbf{X}$, but, in this paper, we only use $\mathbf{B}$ for the following sections.

C. Data Reconstruction

In this section, we focus on the details of the proposed JGSC model for satisfying the following constraints, i.e., minimizing the reconstruction error, preserving the local structures of the data, and generating the row sparsity.

Given a set of n data points $\mathbf{X}$ and the learnt bases $\mathbf{B}$, the reconstruction process between $\mathbf{B}$ and $\mathbf{X}$ can be achieved by the following least square loss function:

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 \tag{3}$$

where $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$ is the new representation of $\mathbf{X}$.

Given a loss function in (3), a regularizer is often used to avoid the issue of overfitting as well as to meet predefined criteria, such as sparsity. In this paper, to conduct feature selection, we should choose a regularizer to discriminate the important features from the unimportant features. Motivated by the characteristics of sparse learning, we expect that the important features are represented by nonzero values and the unimportant features by zeros. Then, we discard the unimportant features (i.e., the features with zero values) and keep the important features for performing feature selection. However, as mentioned before, the ℓ_1-norm regularizer leads to the element sparsity and cannot satisfy the requirement of feature selection. Instead, the $\ell_{2,1}$-norm regularizer has been designed to measure the distance in feature dimensions via the ℓ_2-norm regularizer, while performing summation over different data points via the ℓ_1-norm [56], [57]. Thus, the $\ell_{2,1}$-norm regularizer leads to the row sparsity as well as to consider the correlations of all the features. The $\ell_{2,1}$-norm regularizer, i.e., the second goal in the proposed objective function, is defined as

$$\|\mathbf{S}\|_{2,1} = \sum_{j=1}^{m} \|\mathbf{s}^j\|_2 \tag{4}$$

where $\mathbf{s}^j$ is the jth row of matrix $\mathbf{S}$, which indicates the effect of the jth feature to all the data points.

From a machine learning point of view, a well-defined regularization term can produce a generalized solution to the objective function and, thus, result in a better performance for the final goal [20]. In this paper, following the idea in [46] and [58], we devise a regularization term with the assumption that, if some data points, e.g., $\mathbf{x}_i$ and $\mathbf{x}_j$, are involved in regressing the response variables and are also related to each other, their corresponding weight coefficients (i.e., $\mathbf{s}_i$ and $\mathbf{s}_j$) should have the same or similar relation, since the ith data point $\mathbf{x}_i$ in $\mathbf{X}$ corresponds to the ith column $\mathbf{s}_i$ in $\mathbf{S}$ in our regression framework. To do this, we penalize a loss function with the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (i.e., w_{ij}) on $\|\mathbf{s}_i - \mathbf{s}_j\|_2^2$. In particular, we impose the relation between columns in $\mathbf{X}$ to be reflected in the relation between the respective rows in $\mathbf{S}$ by defining the following embedding function:

$$\frac{1}{2}\sum_{i,j}^{d} w_{ij}\|\mathbf{s}_i - \mathbf{s}_j\|_2^2 = \operatorname{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^T) \tag{5}$$

where w_{ij} denotes an element in the feature similarity matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n\times n}$, which encodes the relation between features in the data points. $\mathbf{L} \in \mathbb{R}^{n\times n}$ (where $\mathbf{L}_{ii} = \sum_{j=1}^{n} w_{ij}$) is a diagonal Laplacian matrix. To obtain the similarity matrix $\mathbf{W}$, we first use a radial basis kernel function to measure the similarity between two samples and, then, construct a data adjacency graph by regarding each data point as a node and using k-nearest neighbors (kNNs) to compute the edge weights.

D. Joint Graph Sparse Coding

By integrating the above three goals into a unified framework, the final objective function of JGSC is defined as follows and its pseudocode is listed in Algorithm 1:

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 + \alpha \operatorname{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^T) + \lambda\|\mathbf{S}\|_{2,1} \tag{6}$$

where $\alpha \ge 0$ and $\lambda \ge 0$ are the tuning parameters.

In (6), the first two terms are designed to simultaneously achieve minimal regression error (via the first term) and the preservation of a local structure of the data (via the second term). The last two terms are designed to generate the row sparsity (via the $\ell_{2,1}$-norm regularizer), by taking the correlations among all the features (via the $\ell_{2,1}$-norm regularizer) and the possible correlations among all data points (via the second term) into account.

Different from the methods (e.g., MCFS and MRSF) that preserve the local structure of the data by two sequential steps, i.e., manifold learning to output the graph-based representation of original data and a least square regression, respectively, our JGSC integrates these two steps into a unified framework. Besides, the MCFS utilizes the ℓ_1-norm regularizer, while both our JGSC and the MRSF utilize the $\ell_{2,1}$-norm regularizer. GSC also uses the same method as ours to preserve the local structure of the data, but uses the ℓ_1-norm regularizer prohibiting for feature selection. Note that Yang *et al.* [55] focused on the $\ell_{2,1}$-norm regularizer for conducting feature selection, but not considering to preserve the neighborhood among samples.

[1] Actually, Yang *et al.* [55] had indicated that different numbers of the basis (i.e., different values of m) will affect the quality of $\mathbf{S}$. However, in our experiments, for simplicity, we only set the number of basis equivalent to the dimensions of features, since such setting enables our methods to output significant performance.

```
Algorithm 1 Proposed JGSC Algorithm
  Input: $\mathbf{X} \in \mathbb{R}^{d \times n}$
  Learn the bases $\mathbf{B}$ from $\mathbf{X}$; See Sec. III-B.
  Generate $\mathbf{S}$ of $\mathbf{X}$ by Algorithm 3; See Sec. III-C
```

```
Algorithm 2 Proposed RJGSC Algorithm
  Input: $\mathbf{X} \in \mathbb{R}^{d \times n}$
  Learn the bases $\mathbf{B}$ from $\mathbf{X}$; See Sec. III-B.
  Generate $\mathbf{S}$ of $\mathbf{X}$ by Algorithm 3;
```

E. Features Selection

After conducting the proposed JGSC model, we obtain a new representation $\mathbf{S}$ of the data $\mathbf{X}$. Due to the $\ell_{2,1}$-norm regularizer, many rows in $\mathbf{S}$ shrink to zeros. This indicates that the corresponding features (i.e., these zero rows) are not important. To achieve efficiency and effectiveness, we may remove them to perform feature selection. More specifically, we first rank the rows in $\mathbf{S}$ in the descending order according to the ℓ_2-norm value of each individual row $\|\mathbf{s}^j\|_2$, $j = 1, \ldots, m$ and, then, select top-ranked rows as the results of feature selection.

F. Extension

It is noteworthy that the previous self-taught dimensionality reduction (STDR) method [56] employed the same regularizers (i.e., the graph Laplacian and the $\ell_{2,1}$-norm regularizer) in (6) but utilized a robust loss function. Moreover, the STDR [56] made the conclusion that the robust loss function could avoid the impact of outliers in the high-dimensional and small-sized data, which means that the dimensions of the data are high and the number of the data is small. In this paper, we extend our JGSC to RJGSC for avoiding the impact of the outliers. In particular, the RJGSC replaces the least square loss function of our JGSC in (6) with the robust loss function to avoid the impact of the outliers and also achieve the same goals of the JGSC. The objective function of the RJGSC is defined as follows and its pseudocode is listed in Algorithm 2:

$$\begin{cases} \min\limits_{\mathbf{B},\hat{\mathbf{S}}} \|\mathbf{X} - \mathbf{B}\hat{\mathbf{S}}\|_F^2 + \lambda \sum\limits_{i=1}^{n} \left\|\hat{\mathbf{s}}_i\right\|_1, \quad \text{s.t.} \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{d} b_{i,j}^2 \leq 1 \\ \min\limits_{\mathbf{S}} \; \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_{2,1} + \alpha \text{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^T) + \lambda\|\mathbf{S}\|_{2,1}. \end{cases} \tag{7}$$

We listed the difference between the proposed RJGSC and the STDR as follows. First, the STDR learns the bases of training data from external data, while our RJGSC obtains the bases from training data. That is, the STDR conducts self-taught learning, while the RJGSC conducts unsupervised learning. Second, the STDR was only designed to focus on learning limited high-dimensional training data, while the JGSC does not make such assumption. Last but not least, the STDR employed MATLAB function lyap with at least cubic time complexity to solve the Sylvester equation, i.e., the optimization of (8), while our RJGSC uses an analytical solution with at most quadratic time complexity.

```
Algorithm 3 TOSC Algorithm
  Input: $\mathbf{X} \in \mathbb{R}^{d \times n}, \mathbf{B} \in \mathbb{R}^{d \times m}, \mathbf{L} \in \mathbb{R}^{n \times n}, \alpha$ and $\lambda$;
  Initialize $t = 0$;
  $\mathbf{C}_0$ as a random $m \times m$ matrix;
  repeat
    Update $\mathbf{S}(t+1)$ in Eq.8 by Algorithm 4;
    Update $\mathbf{C}(t+1)$ via Eq.9;
    $t = t$+1;
  until the objective function in Eq.6 converges
```

```
Algorithm 4 AS3 Algorithm
  Input: $\mathbf{X}$, $\mathbf{B}$, $\mathbf{L}$, $\mathbf{C}$, $\alpha$ and $\lambda$;
  Conduct SVD on $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{C})$ and $\alpha\mathbf{L}$;
  Obtain $\tilde{\mathbf{S}}$ and $\mathbf{E}$ according to Eq.13;
  Obtain $\tilde{\mathbf{s}}_{i,j}$ according to Eq.15;
  Obtain $\mathbf{S}$ according to Eq.16;
```

IV. Optimization

Since the objective function in (6) is convex and nonsmooth, so it admits a global solution. In this section, we design the iteratively reweighed framework [59] to optimize the objective function in (6) via iteratively calculating the gradient of $\|\mathbf{S}\|_{2,1}$. Then, we prove that the proposed algorithm makes the objective function in (6) converge to its global optimum.

A. Proposed Optimization Algorithm

By setting the derivative of the objective function in (6) with respect to $\mathbf{S}$ as zero, we obtain

$$(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{C})\mathbf{S} + \mathbf{S}(\alpha\mathbf{L}) = \mathbf{B}^T\mathbf{X} \tag{8}$$

where $\mathbf{C}$ is a diagonal matrix with its ith diagonal element calculated as

$$c_{i,i} = \frac{1}{2\|\mathbf{s}^i\|_2}. \tag{9}$$

By observing (8), we know that $\mathbf{C}$ depends on the value of $\mathbf{S}$ and $\mathbf{S}$ also depends on the value of $\mathbf{C}$. Hence, it is impractical to compute $\mathbf{S}$ (or $\mathbf{C}$) directly. In this paper, we design a novel iterative algorithm to optimize (8) by alternatively computing $\mathbf{S}$ and $\mathbf{C}$ [i.e., an iterative algorithm to optimize $\mathbf{S}$ and $\mathbf{C}$ (TOSC)]. We first summarize the details in Algorithm 3 and, then, prove that in each iteration, the updated $\mathbf{S}$ and $\mathbf{C}$ make that the value of the objective function in (6) decreases. As shown in Algorithm 3, in each iteration, given a fixed $\mathbf{C}$, the value of $\mathbf{S}$ is first calculated using (8). Then, $\mathbf{C}$ is updated using (9). The iteration process is repeated until there is no change to the value of the objective function.

In (9), given a fixed $\mathbf{S}$, it is easy to solve $\mathbf{C}$. However, it is the Sylvester equation for solving $\mathbf{S}$ with a fixed $\mathbf{C}$. Usually, to optimizing the Sylvester equation via MATLAB function lyap or software, LAPACK needs at least cubic time complexity. In this paper, we propose an analytical solution listed in Algorithm 4 with the time complexity $min(n^2d, d^3)$ (where d is the dimensions). In this section, we explain the detail of Algorithm 4.

Since both $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{C})$ and $\alpha\mathbf{L}$ are positive semidefinite, we perform singular value decomposition on them to obtain

$$\begin{aligned} \mathbf{B}^T\mathbf{B} + \lambda\mathbf{C} &= \mathbf{U}\Sigma_1\mathbf{U}^T \\ \alpha\mathbf{L} &= \mathbf{V}\Sigma_2\mathbf{V}^T \end{aligned} \tag{10}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices. Then, (8) can be expressed as

$$\mathbf{U}\Sigma_1\mathbf{U}^T\mathbf{S} + \mathbf{S}\mathbf{V}\Sigma_2\mathbf{V}^T = \mathbf{B}^T\mathbf{X}. \tag{11}$$

By multiplying $\mathbf{U}^T$ and $\mathbf{V}$ from the left and the right on both sides of (11), respectively, we obtain

$$\Sigma_1\mathbf{U}^T\mathbf{S}\mathbf{V} + \mathbf{U}^T\mathbf{S}\mathbf{V}\Sigma_2 = \mathbf{U}^T\mathbf{B}^T\mathbf{X}\mathbf{V}. \tag{12}$$

By denoting

$$\begin{aligned} \tilde{\mathbf{S}} &= \mathbf{U}^T\mathbf{S}\mathbf{V} \\ \mathbf{E} &= \mathbf{U}^T\mathbf{B}^T\mathbf{X}\mathbf{V}. \end{aligned} \tag{13}$$

Then, (12) becomes

$$\Sigma_1\tilde{\mathbf{S}} + \tilde{\mathbf{S}}\Sigma_2 = \mathbf{E}. \tag{14}$$

Note that $\Sigma_1 = \mathrm{diag}(\sigma_1^{(1)}, \ldots, \sigma_1^{(d)})$ and $\Sigma_2 = \mathrm{diag}$ $(\sigma_2^{(1)}, \ldots, \sigma_2^{(m)})$ are positive definite, and $\sigma_1^{(i)} > 0$, $i = 1, \ldots, d$. Thus, each element in $\tilde{\mathbf{S}}$ can be obtained by

$$\tilde{s}_{i,j} = \frac{e_{i,j}}{\sigma_1^i + \sigma_2^j}. \tag{15}$$

After obtaining the $\tilde{\mathbf{S}}$, we can obtain the optimum $\mathbf{S}$ as

$$\mathbf{S} = \mathbf{U}\tilde{\mathbf{S}}\mathbf{V}^T. \tag{16}$$

B. Convergence

We prove that the proposed Algorithm 3 makes the value of the objective function in (6) monotonically decrease. We first give a lemma from [22] and [54] as follows.

Lemma 1: For any nonzero row vectors $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{u}' \in \mathbb{R}^n$, the following holds:

$$\left(\frac{\|\mathbf{u}'\|_2^2}{2\|\mathbf{u}\|_2} - \|\mathbf{u}'\|_2\right) - \left(\frac{\|\mathbf{u}\|_2^2}{2\|\mathbf{u}\|_2} - \|\mathbf{u}\|_2\right) \geq 0. \tag{17}$$

Theorem 1: In each iteration, Algorithm 3 monotonically decreases the objective function value in (6).

Proof: In Algorithm 3, we denote the tth iteration of (6) without the last term $\lambda\|\mathbf{S}\|_{2,1}$ as $\mathcal{L}(t) = \|\mathbf{X} - \mathbf{B}\mathbf{S}(t)\|_F^2 + \alpha\mathrm{tr}(\mathbf{S}(t)\mathbf{L}(\mathbf{S}(t))^T)$, where $\mathcal{L}(t)$, $\mathbf{C}(t)$, and $\mathbf{S}(t)$, respectively, are the optimum value in the tth iteration for $\mathcal{L}$, $\mathbf{C}$, and $\mathbf{S}$. According to the iteratively reweighed framework [60], optimizing the nonsmooth convex $\|\mathbf{S}\|_{2,1}$ can be transferred to iteratively update $\mathbf{C}$ and $\mathbf{S}$ in $\mathrm{tr}(\mathbf{S}^T\mathbf{C}\mathbf{S})$, that is

$$\begin{aligned} &\mathcal{L}(t+1) + \lambda\mathrm{tr}((\mathbf{S}(t+1))^T\mathbf{C}(t)\mathbf{S}(t+1)) \\ &\qquad \leq \mathcal{L}(t) + \lambda\mathrm{tr}((\mathbf{S}(t))^T\mathbf{C}(t)\mathbf{S}(t)). \end{aligned} \tag{18}$$

Changing the trace form into its summation form, we get

$$\mathcal{L}(t+1) + \lambda\sum_{i=1}^{m}\frac{\|\mathbf{s}(t+1)^i\|_2^2}{2\ \mathbf{s}(t)^i\|_2} \leq \mathcal{L}(t) + \lambda\sum_{i=1}^{m}\frac{\|\mathbf{s}(t)^i\|_2^2}{2\|\mathbf{s}(t)^i\|_2}$$

where $\mathbf{s}(t+1)^i$ is the ith row of the matrix $\mathbf{S}(t+1)$. By simple modification, we can have

$$\begin{aligned} &\mathcal{L}(t+1) + \lambda\sum_{i=1}^{m}\left(\frac{\|\mathbf{s}(t+1)^i\|_2^2}{2\|\mathbf{s}(t)^i\|_2} - \|\mathbf{s}(t+1)^i\|_2 + \|\mathbf{s}(t+1)^i\|_2\right) \\ &\quad \leq \mathcal{L}(t) + \lambda\sum_{i=1}^{m}\left(\frac{\|\mathbf{s}(t)^i\|_2^2}{2\|\mathbf{s}(t)^i\|_2} - \|\mathbf{s}(t)^i\|_2 + \|\mathbf{s}(t)^i\|_2\right). \end{aligned}$$

After reorganizing terms, we finally have

$$\begin{aligned} &\mathcal{L}(t+1) + \lambda\sum_{i=1}^{m}\|\mathbf{s}(t+1)^i\|_2 \\ &\quad + \lambda\sum_{i=1}^{m}\left(\left(\frac{\|\mathbf{s}(t+1)^i\|_2^2}{2\|\mathbf{s}(t)^i\|_2} - \|\mathbf{s}(t+1)^i\|_2\right)\right. \\ &\qquad\qquad \left. - \left(\frac{\|\mathbf{s}(t)^i\|_2^2}{2\|\mathbf{s}(t)^i\|_2} - \|\mathbf{s}(t)^i\|_2\right)\right) \\ &\quad \leq \mathcal{L}(t) + \lambda\sum_{i=1}^{m}\|\mathbf{s}(t)^i\|_2. \end{aligned}$$

According to Lemma 1, the following inequality holds:

$$\mathcal{L}(t+1) + \lambda\sum_{i=1}^{m}\|\mathbf{s}(t+1)^i\|_2 \leq \mathcal{L}(t) + \lambda\sum_{i=1}^{m}\|\mathbf{s}(t)^i\|_2.$$

■

We can follow this section to solve (7) and prove its convergence. In particular, we first set the derivative of the objective function in (7) with respect to $\mathbf{S}$ to zero and, then, obtain

$$(\mathbf{B}^T\tilde{\mathbf{D}}\mathbf{B} + \lambda\tilde{\mathbf{C}})\mathbf{S} + \mathbf{S}(\alpha\mathbf{L}) = \mathbf{B}^T\tilde{\mathbf{D}}\mathbf{X} \tag{19}$$

where $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{D}}$, respectively, are the diagonal matrices with their ith diagonal elements as $\tilde{c}_{i,i} = (1/2\|\mathbf{s}^i\|_2)$ and $\tilde{d}_{i,i} = (1/2\|(\mathbf{X} - \mathbf{B}\mathbf{S})^i\|_2)$. $(\mathbf{X} - \mathbf{B}\mathbf{S})^i$ stands for the ith row of matrix $(\mathbf{X} - \mathbf{B}\mathbf{S})$. We then solve (7) with the similar principles in the TOSC algorithm and the AS3 algorithm. Finally, we can prove the convergence of (7) based on Theorem 1 and Lemma 1 by the following inequality $\alpha\mathrm{tr}(\mathbf{S}(t+1)\mathbf{L}\mathbf{S}(t+1)^T) + \mathrm{tr}((\mathbf{X} - \mathbf{B}\mathbf{S}(t+1))^T\tilde{\mathbf{D}}(t)(\mathbf{X} - \mathbf{B}\mathbf{S}$ $(t+1))) + \lambda\mathrm{tr}(\mathbf{S}(t+1)^T\tilde{\mathbf{C}}(t)\mathbf{S}(t+1)) \leq \alpha\mathrm{tr}(\mathbf{S}(t)\mathbf{L}\mathbf{S}(t)^T) + \mathrm{tr}((\mathbf{X} - \mathbf{B}\mathbf{S}(t))^T\tilde{\mathbf{D}}(t)(\mathbf{X} - \mathbf{B}\mathbf{S}(t))) + \lambda\mathrm{tr}(\mathbf{S}(t)^T\tilde{\mathbf{C}}(t)\mathbf{S}(t))$.

V. EXPERIMENTAL ANALYSIS

A. Experimental Settings

We compared the proposed methods with the state-of-the-art dimensionality reduction methods in real applications, including text mining (PCMAC and BASEHOCK), face recognition (AR10P and PIE10P), and bioinformatics (TOX and SMK-CAN).[2] The details of the data sets were presented in Table II.

[2] All data sets were downloaded from the urls http://featureselection.asu.edu/datasets.php and http://www.zjucadcg.cn/dengcai/Data/data.html.

TABLE II

DETAILS ON THE USED DATA SETS IN THE EXPERIMENTS

Dataset	#(Instances)	#(Features)	#(class)
PCMAC	1943	3289	2
BASEHOCK	1993	4862	2
AR10P	130	2400	10
PIE10P	210	2420	10
TOX-171	171	5748	4
SMK-CAN-187	187	19993	2

B. Experiment Setup

We compared the JGSC and the RJGSC with the following algorithms.

1) *Original:* All original features are used to conduct kNN classification. We want to know whether the dimensionality reduction algorithms can improve the classification performance.
2) *LPP:* It does not take [33] least square regression into account, compared with our JGSC.
3) *JFS:* Joint feature selection (JFS) achieves the minimal reconstructor error but not considers the local structures of target data. We changed the supervised feature selection algorithm robust feature selection in [60] to generate the JFS in our experiments.
4) *LScore:* It belongs [33] to a filter model of feature selection. The score of a feature is evaluated by its locality preserving power. The features have high score if data points in the same topic are close to each other.
5) *MCFS:* It selects [18] features by sequentially conducting manifold learning and spectral regression.
6) *MRSF:* It sequentially conducts manifold learning and joint sparse regression.
7) *GSC:* It first employs [14], [19] the ℓ_1-norm regularizer for achieving the element sparsity and, then, performs feature selection on the sparse results via the score rule of MCFS.
8) *UDFS:* Unsupervised discriminative feature selection (UDFS) [26] conducts feature selection by combining discriminative analysis with the local structures of target data.
9) *RJGSC:* It is the variation of the STDR [56] algorithm. RJGSC and STDR have the same objective function. However, RJGSC uses the training data to learn the bases of the training data, while STDR uses external data to learn the bases.

In our experiments, we separately used all feature selection methods (except original) to conduct feature selection and, then, used the kNN classifier via setting the number of k as 5 to conduct classification tasks. In all algorithms, the number of left dimensionality was kept as $\{10\%, 20\%, \ldots, 80\%\}$ of those of the original ones for each data set. For original, we directly used libSVM toolbox to conduct kNN classification.

We used a tenfold cross-validation technique for all methods. In addition, we first randomly partitioned the whole data set into ten subsets. We then selected one subset for testing and used the remaining nine subsets for training [61]. We repeated the whole process ten times to avoid the possible bias during data set partitioning for cross validation. The final result was computed by averaging results from all experiments. For the model selection of our method, i.e., the selection of α and λ in (6) and (7), we set the parameter spaces with four levels for each parameter in (6) and (7) and, then, apply them to conduct a fivefold inner cross validation. The parameters resulted in the best performance were used in testing. Moreover, for fair comparison, we set the values of parameters for the competing algorithms according to the instructions in the corresponding papers. We also conducted fivefold inner cross validation to conduct model selection for each competing method. In particular, for eigenvalue-based methods, such as LPP and LScore, we determined their optimal features based on their respective eigenvalues according to [33]. For JFS, MCFS, MRFS, GSC, and UDFS, respectively, we optimized their parameters by cross validating the values in the ranges of $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, $\{10^{-4}, 10^{-2}, \ldots, 10^{4}\}$, $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, $\{10^{-5}, 10^{-3}, \ldots, 10^{5}\}$, and $\{10^{-5}, 10^{-4}, \ldots, 10^{2}\}$.

We used average classification accuracy (ACA) as evaluation metric to compare all the methods. Given a data point $\mathbf{x}_i$, and let y_i and $\hat{y}_i$ be the derived labels via a classification algorithm and the true label, respectively, the ACA is defined as follows:

$$\text{ACA} = \frac{\sum_{i=1}^{n} \delta(y_i, \hat{y}_i)}{n} \tag{20}$$

where n is the sample size, $\delta(x, y) = 1$ if $x = y$, and $\delta(x, y) = 0$, otherwise. The larger the value of the ACA is, the better the method is.

In the following sections, we first tested the difference between the JGSC and the RJGSC on synthetic data sets, and evaluated their convergence rate on real data sets. Then, we verified the parameters' sensitivity on λ and α in (6) and (7), respectively. Furthermore, we compared both the JGSC and the RJGSC with all the competing methods on small data sets. Finally, we analyze the experimental results by comparing our methods with the competing methods on two large data sets, in terms of ACA.

C. Simulation Study

In this section, we justify the validity of the proposed methods (i.e., the JGSC and the RJGSC) on synthetic data and also compare them with the comparison methods. For the simulation study, we generated the data using a linear regression model of $\mathbf{X} = \mathbf{BS} + \mathbf{E}$, where $\mathbf{X} \in \mathbb{R}^{d\times n}$, $\mathbf{B} \in \mathbb{R}^{d\times m}$, $\mathbf{E} \in \mathbb{R}^{d\times n}$, and $\mathbf{S} \in \mathbb{R}^{m\times n}$ are the original data, the bases, the noise matrix, and the new presentations of the original data, respectively. We also generated the label matrix $\mathbf{Y} \in \mathbb{R}^{c\times n}$ (where c is the number of classes) represented by 0–1 encoding method. In particular, for each class, we generated n_i samples by setting the first d_0 rows relevant to the classes and the remaining $d - d_0$ rows irrelevant for discrimination. The samples of each class were generated from multivariate normal distribution. We then used the package SPAMS [25] to obtain the bases. We constructed $\mathbf{S}$ by setting the first d_0 rows with the values drawn from $\mathcal{N}(0, 1)$ and

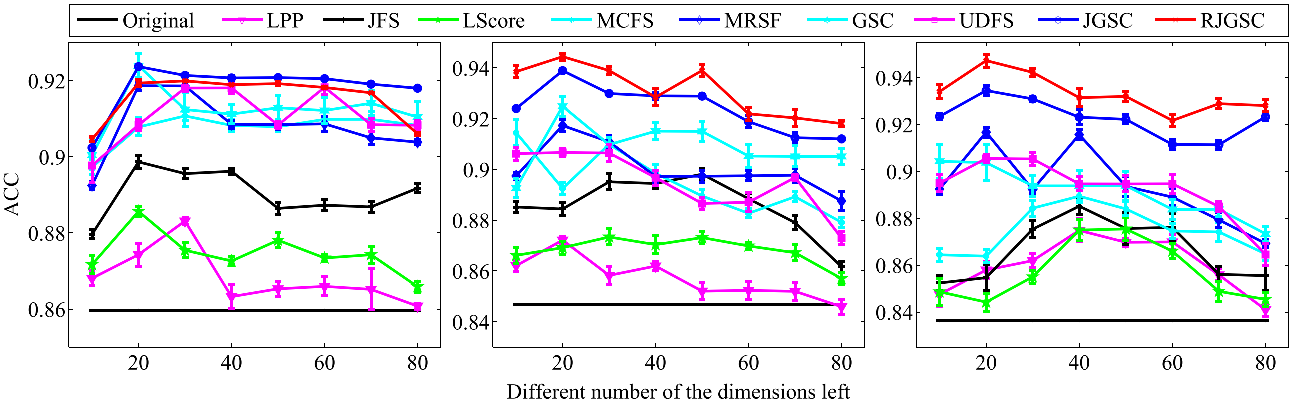
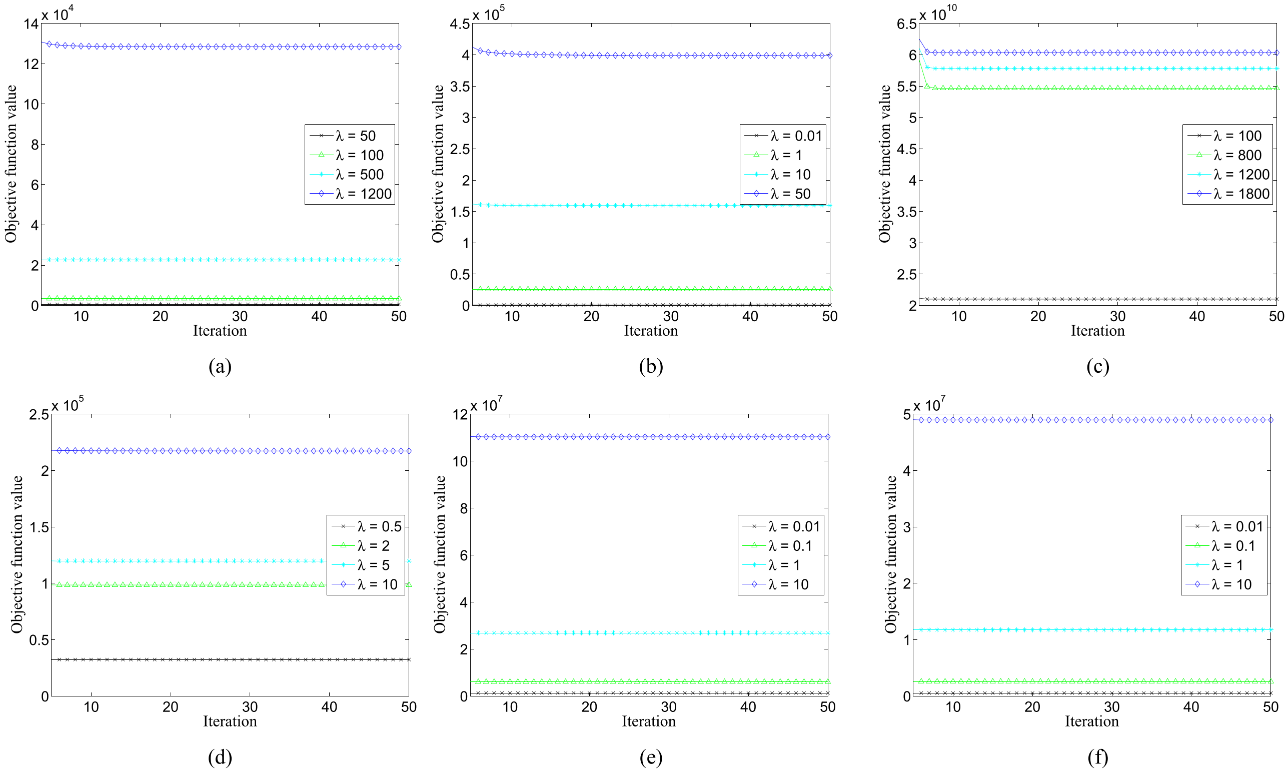


Fig. 2. ACA on different synthetic data sets, i.e., Toy1 (left), Toy2 (middle), and Toy3 (right), respectively. Note that the range shown at the curves represents the standard deviation. (Best viewed in color.)



(a) (b) (c)

(d) (e) (f)

Fig. 3. Convergence rate of JGSC for solving the proposed objective function with fixed α. (a) PCMAC. (b) BASEHOCK. (c) TOX-171. (d) SMK-CAN-187. (e) AR10P. (f) PIE10P.

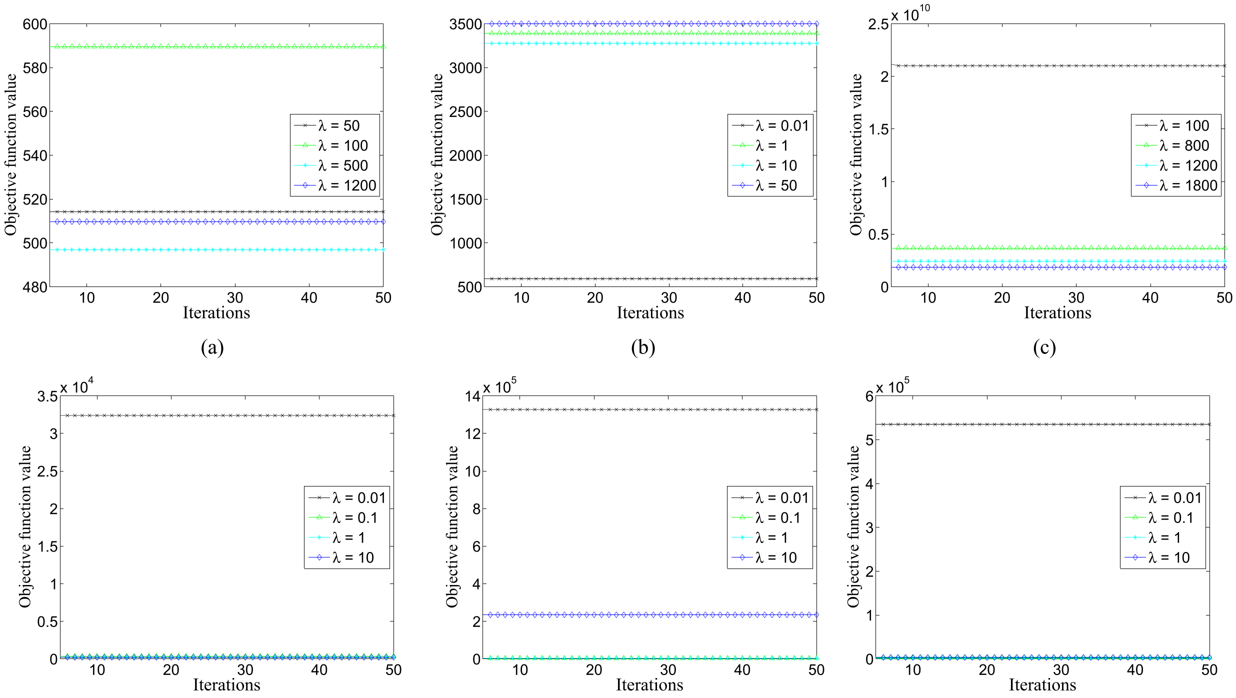
the rest $(d - d_0)$ rows zero. We then obtained the noise $\mathbf{E}$ from $\mathcal{N}(0, \nu\Sigma(0.1))$, where $\Sigma(0.1)$ was a covariance matrix with the diagonal elements of 1 and the off-diagonal elements of 0.1. After obtaining $\mathbf{S}$, $\mathbf{B}$, and $\mathbf{E}$ as described above, we obtained the noise observations $\mathbf{X}$ via the linear regression model and then centered and standardized it. We generated data sets of Toy1 by setting $n_1 = 50$, $n_2 = 40$, $n_3 = 50$, $d = 200$, $d_0 = 80$, and $\nu = 10^{-3}$, Toy2 by setting $n_1 = 50$, $n_2 = 50$, $n_3 = 50$, $d = 300$, $d_0 = 100$, and $\nu = 10^{1}$, Toy3 by setting $n_1 = 50$, $n_2 = 40$, $n_3 = 110$, $d = 300$, $d_0 = 80$, and $\nu = 10^{3}$. In the three synthetic data sets, Toy1 contains the least outliers, while Toy3 has the most outliers.

Fig. 2 reported the results on three synthetic data sets. Obviously, the proposed methods obtain the best performance, compared with the competing methods. By comparing our JGSC with our RJGSC, the RJGSC beats the JGSC on two data sets, i.e., Toy2 and Toy3. This occurred because the RJGSC is more robust than JGSC for dealing with the data sets with noise or outliers.

D. Convergence Rate

In this section, we tested the convergence rate of both the JGSC and the RJGSC. Because of lack of space, we reported some results in Figs. 3 and 4 on the objective function values with fixed values of α and varied λ.[3]

[3] In this paper, for better view, we plotted the figures of convergence rate from the fifth iteration as the gap among the first five iterations is very huge.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Convergence rate of RJGSC for solving its proposed objective function with fixed α. (a) PCMAC. (b) BASEHOCK. (c) TOX-171. (d) SMK-CAN-187. (e) AR10P. (f) PIE10P.

From Figs. 3 and 4, the objective function values of two algorithms rapidly decreased at the first few iterations and, then, became stable after about 30 iterations (or even less than 20 iterations in many cases) on all data sets. This confirms the fast convergence rate of our proposed optimization method on the JGSC and the RJGSC. Similar cases could also be found in other cases with other settings on α and λ.

E. Parameters' Sensitivity

In this section, we studied the parameters' sensitivity of both the JGSC and the RJGSC with respect to λ and α in (6) and (7), respectively. Because of the limited space, we only reported the results of the case with 50% dimensionality left from the original dimensions, because other cases have similar results. We reported the results in Figs. 5 and 6.

In Figs. 5 and 6, both the JGSC and the RJGSC were sensitive to the parameter setting. Moreover, the maximal improvement between the best results and the worst results on average varied from 17.76% to 85.38% and 16.34% to 83.97%, respectively, on all data sets. Therefore, it is necessary to assign different parameters' settings for obtaining stable classification performance.

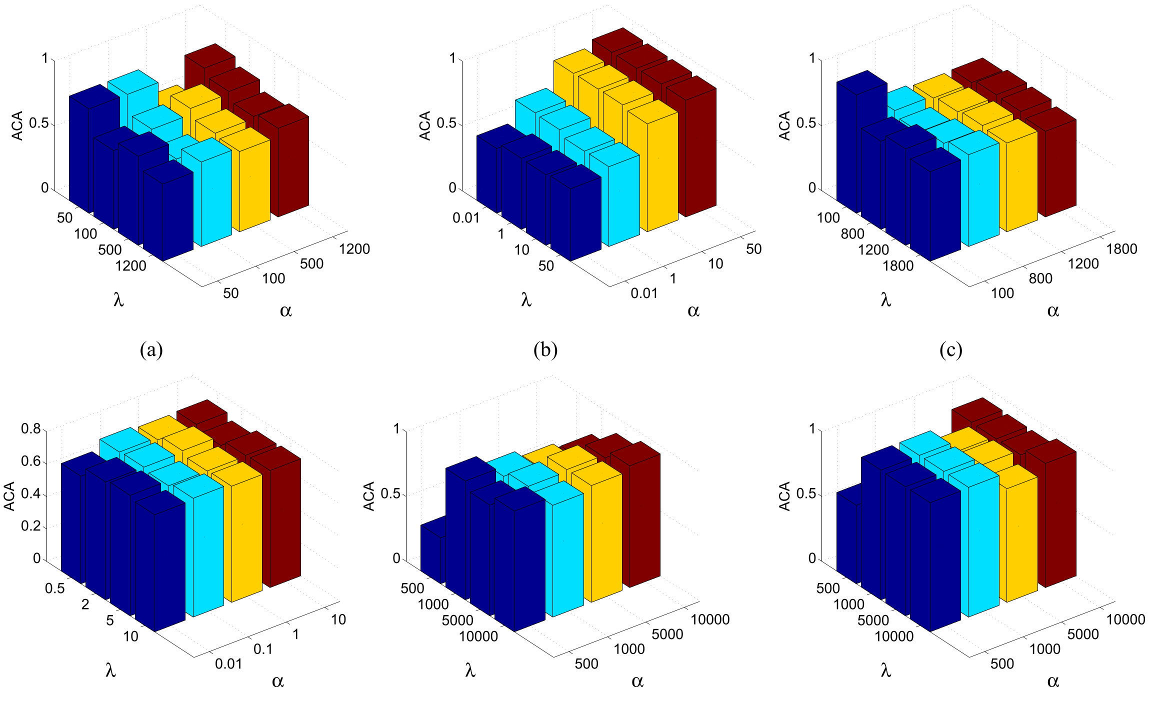
F. Classification Results by All Algorithms

The classification performance of all methods was shown in Fig. 7, where the horizontal axis represented the number of the dimensions left after performing feature selection (varying from 10% to 80%) and the vertical axis described the ACA result.

Both the JGSC and the RJGSC outperformed the competing methods, since our methods overcame the drawbacks of the previous spectral feature selection methods. However, the competing methods only solve part of the problems. Moreover, both the JGSC and the RJGSC outperformed the UDFS, which also considered two constraints (i.e., discriminative ability and manifold learning with an $\ell_{2,1}$-norm regularizer) for feature selection.

In the simulation study, the RJGSC was found more robust than the JGSC for dealing with the data sets with outliers. However, in our experiments, it is not true on real data sets, which may not contain outliers. Since Cai *et al.* [18] and Yang *et al.* [26] have shown that their methods (e.g., the UDFS and the MCFS) outperformed the popular methods, such as the LPP [33] and the LScore [33], we can make the conclusion that both the JGSC and the RJGSC outperformed the LPP and the LScore.

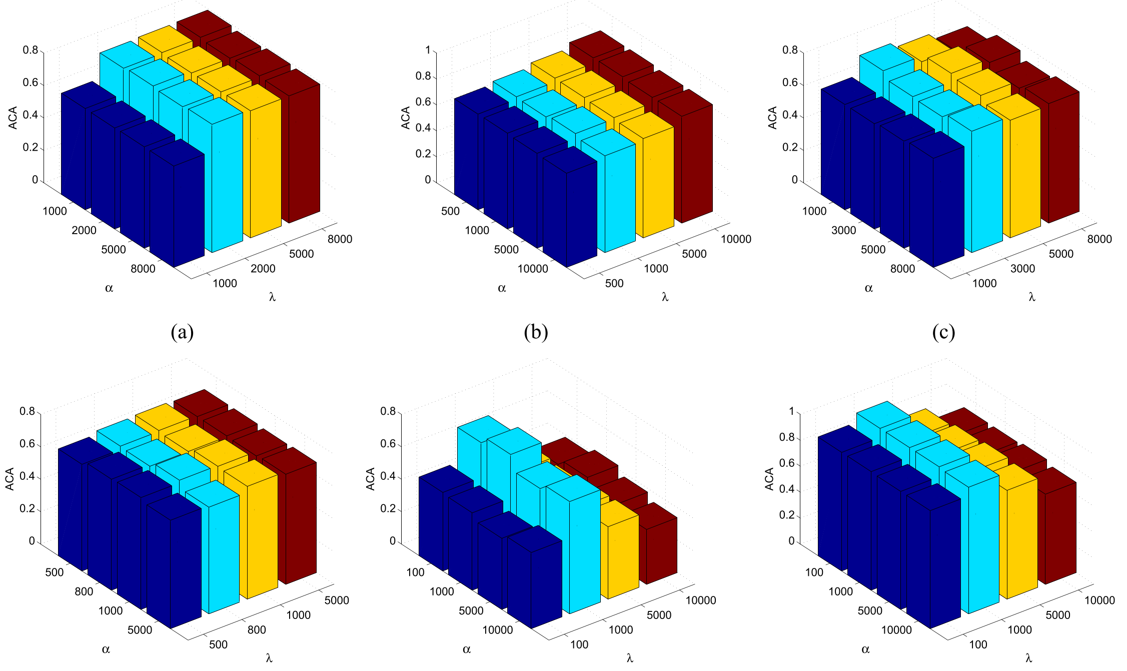
Although both the MCFS and the GSC generate the element sparsity, the classification performance of the MCFS was worse than the GSC, because the GSC simultaneously performs manifold learning and sparse regression, while the MCFS sequentially performs them. This demonstrates the reasonability of simultaneously performing manifold learning and sparse regression. On the other hand, the MRSF has more advantages than the MCFS by replacing the ℓ_1-norm regularizer with the $\ell_{2,1}$-norm regularizer, such as considering the correlations among the spectral features and leading to the row sparsity, so the MRSF outperformed the MCFS in our experiments. Furthermore, the MRSF also outperformed the JFS, since the MRSF took subspace learning (i.e., the



(a) (b) (c)

(d) (e) (f)

Fig. 5. ACA results on different data sets with different parameters for JGSC. (a) PCMAC. (b) BASEHOCK. (c) TOX-171. (d) SMK-CAN-187. (e) AR10P. (f) PIE10P.



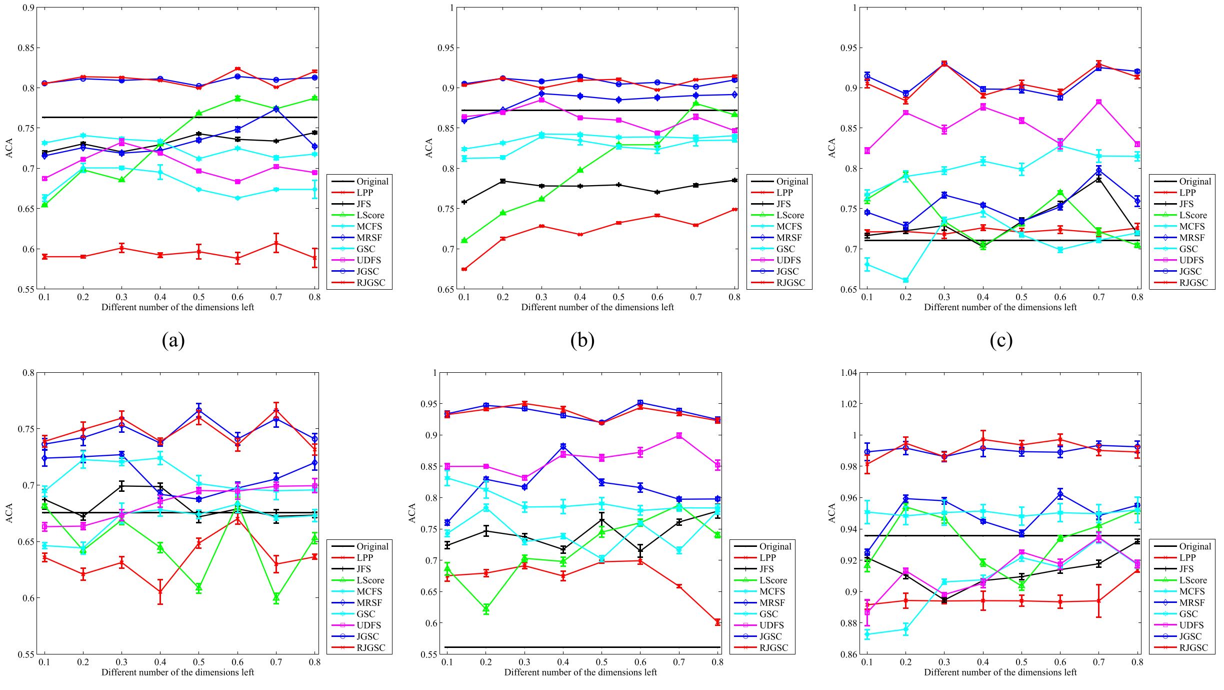
(a) (b) (c)

(d) (e) (f)

Fig. 6. ACA results on different data sets with different parameters for RJGSC. (a) PCMAC. (b) BASEHOCK. (c) TOX-171. (d) SMK-CAN-187. (e) AR10P. (f) PIE10P.

correlations among all the data points) into the process of feature selection.

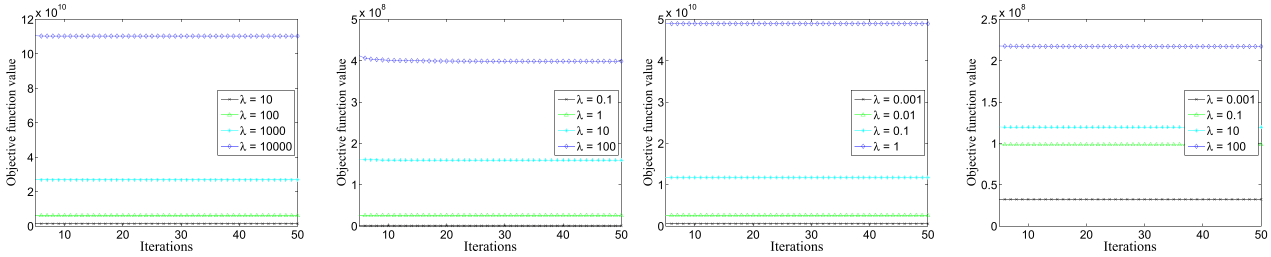
Original outperformed some dimensionality reduction algorithms on some data sets. However, these dimensionality reduction algorithms are more efficient. Thus, it is crucial for conducting dimensionality reduction on high-dimensional data, which is consistent with the conclusion in [18], [26], and [33]. On the other hand, although some algorithms (such as the JFS, the LPP, and the LScore) conducted feature selection by considering one constraint, the JFS was better than either the LPP or the LScore. It showed that conducting feature selection in new space rather than in the original space may be better than those in the original space, because such a new feature space



(a) (b) (c)

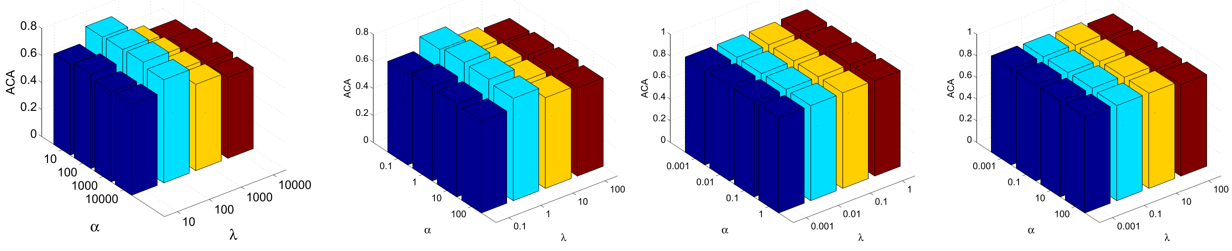
(d) (e) (f)

Fig. 7. ACA on various parameters' settings on different data sets. Note that the range shown at the curves represents the standard deviation. (a) PCMAC. (b) BASEHOCK. (c) TOX-171. (d) SMK-CAN-187. (e) AR10P. (f) PIE10P. (Best viewed in color.)



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

Fig. 8. Convergence rate of JGSC and RJGSC for solving its proposed objective function with fixed α. (a) News of JGSC. (b) News of RJGSC. (c) Reuters of JGSC. (d) Reuters of RJGSC.



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

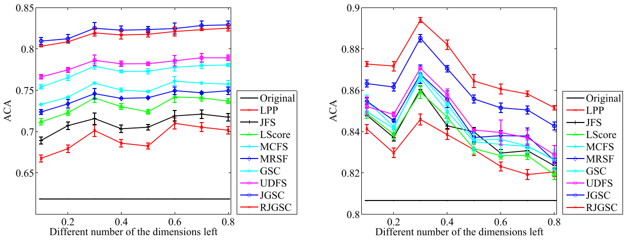
Fig. 9. ACA results on large data sets with different parameters for JGSC and RJGSC. (a) News of JGSC. (b) News of RJGSC. (c) Reuters of JGSC. (d) Reuters of RJGSC.

(i.e., the basis space) is a higher level and more abstract representation [56], [62].

G. Comparison on Large Data Sets

In this section, we evaluate performance on two large text data sets, i.e., News and Reuters, respectively. News data set includes 18 846 samples and 26 214 features, while Reuters data set includes 8293 samples and 18 933 features. We used the same setting as in Section V-B. The experimental results are reported in Figs. 8–10.

Again, the proposed methods achieved the best results, outperforming all the competing methods. The feature



(a) (b)

Fig. 10. ACA on various parameters' settings on two large data sets. The range shown at the curves indicates the standard deviation. (a) News. (b) Reuters. (Best viewed in color.)

selection strategies were also helpful in enhancing classification accuracy, compared with the original method.

VI. Conclusion

In this paper, we have proposed a novel feature selection method to deal with high-dimensional data by taking the graph regularizer and the $\ell_{2,1}$-norm regularizer into the least square regression framework. The experimental results have shown the effectiveness of the proposed method by comparing with the state-of-the-art methods. In the future, we will extend the proposed method into its kernel edition to deal with more complex cases, such as data sets with limited data points.

References

[1] G. Yu, G. Zhang, Z. Zhang, Z. Yu, and L. Deng, "Semi-supervised classification based on subspace sparse representation," *Knowl. Inf. Syst.*, vol. 43, no. 1, pp. 81–101, 2015.

[2] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 780–792, Apr. 2014.

[3] Q. Zhu, L. Shao, X. Li, and L. Wang, "Targeting accurate object extraction from an image: A comprehensive study of natural image matting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 185–207, Feb. 2015.

[4] F. Nie, H. Wang, H. Huang, and C. Ding, "Joint Schatten p-norm and ℓ_p-norm robust matrix completion for missing value recovery," *Knowl. Inf. Syst.*, vol. 42, no. 3, pp. 525–544, 2013.

[5] X. Lu and X. Li, "Multiresolution imaging," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 149–160, Jan. 2014.

[6] X. Lu, Y. Wang, and Y. Yuan, "Sparse coding from a Bayesian perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 929–939, Jun. 2013.

[7] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[8] 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.

[9] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.

[10] R. Hong, M. Wang, Y. Gao, D. Tao, X. Li, and X. Wu, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 669–680, May 2014.

[11] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel, "Greedy column subset selection for large-scale data sets," *Knowl. Inf. Syst.*, vol. 45, no. 1, pp. 1–34, 2015.

[12] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.

[13] P. Nirmala, L. R. Sulochana, and N. Rethnasamy, "Centrality measures-based algorithm to visualize a maximal common induced subgraph in large communication networks," *Knowl. Inf. Syst.*, vol. 46, no. 1, pp. 213–239, 2015.

[14] M. Zheng *et al.*, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[15] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation with partially shared features for event detection using few exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1789–1802, Sep. 2014.

[16] 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.

[17] X. Zhu, H.-I. Suk, S.-W. Lee, and D. Shen, "Canonical feature selection for joint regression and multi-class identification in Alzheimer's disease diagnosis," *Brain Imag. Behavior*, pp. 1–11, Aug. 2015.

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

[19] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely—Laplacian sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3555–3561.

[20] Y. Yang, Y.-T. Zhuang, F. Wu, and Y.-H. Pan, "Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 437–446, Apr. 2008.

[21] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. AAAI*, 2010, pp. 55–60.

[22] X. Zhu, H.-I. Suk, S.-W. Lee, and D. Shen, "Subspace regularized sparse multi-task learning for multi-class neurodegenerative disease identification," *IEEE Trans. Biomed. Eng.*, to be published.

[23] X. Zhu, X. Wu, W. Ding, and S. Zhang, "Feature selection by joint graph sparse coding," in *Proc. SDM*, 2013, pp. 803–811.

[24] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, Jul. 2014.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[26] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proc. 22nd IJCAI*, 2011, pp. 1589–1594.

[27] 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.

[28] S. B. Kotsiantis, "Feature selection for machine learning classification problems: A recent overview," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 1–20, 2011.

[29] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1749–1760, Nov. 2015.

[30] W. Wang, D. Cai, L. Wang, Q. Huang, X. Xu, and X. Li, "Synthesized computational aesthetic evaluation of photos," *Neurocomputing*, vol. 172, pp. 244–252, Jan. 2016.

[31] S. Foitong, P. Rojanavasu, B. Attachoo, and O. Pinngern, "Estimating optimal feature subsets using mutual information feature selector and rough sets," in *Proc. 13th PAKDD*, 2009, pp. 973–980.

[32] R. Caruana and V. R. de Sa, "Benefitting from the variables that variable selection discards," *J. Mach. Learn. Res.*, vol. 3, pp. 1245–1264, Mar. 2003.

[33] X. He and P. Niyogi, "Locality preserving projections," in *Proc. NIPS*, 2003, pp. 197–204.

[34] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, no. 46, no. 1, pp. 389–422, 2002.

[35] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Inf. Sci.*, vol. 179, no. 13, pp. 2208–2217, 2009.

[36] X. Li, L. Mou, and X. Lu, "Scene parsing from an MAP perspective," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1876–1886, Sep. 2015.

[37] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 200–214, Jan. 2016.

[38] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, Jan. 2003.

[39] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proc. 25th Conf. UAI*, 2009, pp. 339–348.

[40] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *J. Mach. Learn. Res.*, vol. 12, pp. 2777–2824, Feb. 2011.

[41] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.

[42] X. Zhu, H.-I. Suk, and D. Shen, "A novel matrix-similarity based loss function for joint regression and classification in AD diagnosis," *NeuroImage*, vol. 100, pp. 91–105, Oct. 2014.

[43] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[44] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 143–152.

[45] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, Jun. 2006.

[46] X. Zhu, Z. Huang, H. Cheng, J. Cui, and H. T. Shen, "Sparse hashing for fast multimedia search," *ACM Trans. Inf. Syst.*, vol. 31, no. 2, 2013, Art. ID 9.

[47] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

[48] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[49] X. Zhu, Z. Huang, J. Cui, and H. T. Shen, "Video-to-shot tag propagation by graph sparse group lasso," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 633–646, Apr. 2013.

[50] J. Peng *et al.*, "Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer," *Ann. Appl. Statist.*, vol. 4, no. 1, pp. 53–77, 2010.

[51] P. Sprechmann, I. Ramírez, G. Sapiro, and Y. C. Eldar, "C-HiLasso: A collaborative hierarchical sparse modeling framework," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4183–4198, Sep. 2011.

[52] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse SVM for feature selection on very high dimensional datasets," in *Proc. 27th ICML*, 2010, pp. 1047–1054.

[53] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $l_{1,\infty}$ regularization," in *Proc. 26th Annu. ICML*, 2009, pp. 857–864.

[54] 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. 2015.

[55] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[56] 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.

[57] X. Zhu, H.-I. Suk, and D. Shen, "Matrix-similarity based loss function and feature selection for Alzheimer's disease diagnosis," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 3089–3096.

[58] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[59] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[60] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. NIPS*, 2010, pp. 1813–1821.

[61] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," *Statist. Sci.*, vol. 27, no. 4, pp. 538–557, 2012.

[62] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th ICML*, 2007, pp. 759–766.

Xiaofeng Zhu received the Ph.D. degree in computer science from The University of Queensland, Brisbane, QLD, Australia.

His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.

Xuelong Li (M'02–SM'07–F'12) is a full professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China.

Shichao Zhang (M'04–SM'04) received the Ph.D. degree in computer science from Deakin University, Geelong, VIC, Australia.

He is currently a China 1000-Plan Distinguished Professor with the Department of Computer Science, Zhejiang Gongshang University, Hangzhou, China. He has authored over 60 international journal papers and 70 international conference papers. His current research interests include data quality and pattern discovery.

Prof. Zhang is a member of the Association for Computing Machinery. As a Chief Investigator, he has won four Australian Large ARC Grants, three China 863 Programs, two China 973 Programs, and five NSFs of China Grants. He served/serves as an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *Knowledge and Information Systems*, and the IEEE INTELLIGENT INFORMATICS BULLETIN. He served as a PC Chair or Conference Chair for six international conferences.

Chunhua Ju received the Ph.D. degree in statistics with a minor in information science from Xiamen University, Xiamen, China, in 2002.

He is currently a Professor of Management Science and Engineering with Zhejiang Gongshang University, Hangzhou, China. He is a Ph.D. Supervisor and the Dean of the College of Computer Science and Information Engineering with Zhejiang Gongshang University. His current research interests include intelligent information processing, data mining, and e-commerce.

Prof. Ju had won the award for the New Century Excellent Talents in University of China.

Xindong Wu (M'95–SM'95–F'11) received the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is currently a Professor of Computer Science with the University of Vermont, Burlington, VT, USA, and a Yangtze River Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His current research interests include data mining, big data analytics, knowledge engineering, and Web systems.

Prof. Wu is a fellow of the American Association for the Advancement of Science.