Jointly Sparse Locality Regression for Image Feature Extraction

Dongmei Mo, Zhihui Lai, Xizhao Wang, *Fellow, IEEE*, and Waikeung Wong

***Abstract*—This paper proposes a novel method called Jointly Sparse Locality Regression (JSLR) for feature extraction and selection. JSLR utilizes joint $L_{2,1}$-norm minimization on regularization term, and also introduces the locality to characterize the local geometric structure of the data. There are three main contributions in JSLR for face recognition. Firstly, it eliminates the drawback in ridge regression and Linear Discriminant Analysis (LDA) that when the number of the classes is too small, not enough projections can be obtained for feature extraction. Secondly, by using the local geometric structure as the regularization term, JSLR is able to preserve local information and find an embedding subspace which can detect the most essential data manifold structure. Moreover, since the $L_{2,1}$-norm based loss function is robust to outliers in data points, JSLR provides the joint sparsity for robust feature selection. The theoretical connections of the proposed method and the previous regression methods are explored and the convergence of the proposed algorithm is also proved. Experimental evaluation on several well-known data sets shows the merits of the proposed method on feature selection and classification.**

***Index Terms*—Regression, face recognition, feature extraction, local structure, joint sparsity.**

I. Introduction

SINCE the data used in computer vision or pattern recognition is very high dimensional, it is of great importance to select the key features from large quantities of variables. Besides, the redundancy of the data would affect the performance of some algorithms in practical applications [1], and thus most of the algorithms cannot obtain a good performance in high-dimensional case [2]. Therefore, feature extraction and selection are of great importance in processing the high-dimensional data set [3].

Manuscript received May 4, 2019; revised September 9, 2019 and December 11, 2019; accepted December 13, 2019. Date of publication December 23, 2019; date of current version October 23, 2020. This work was supported in part by The Hong Kong Polytechnic University (Project Code: RHR1) and in part by General Research Fund of the Research Grants Council of Hong Kong (Project Code: 15202217). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jingdong Wang. *(Corresponding author: Waikeung Wong.)*

D. Mo and W. Wong are with the Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hong Kong SAR of China (e-mail: dongmei.mo@connect.polyu.hk; calvin.wong@polyu.edu.hk).

Z. Lai is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China (e-mail: lai_zhi_hui@163.com).

X. Wang is with the College of Computer Science and Software Engineering and Guangdong Key Lab of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

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

Digital Object Identifier 10.1109/TMM.2019.2961508

Up to now, one of the classical methods is Principle Component Analysis (PCA) [4], which is a simple and effective unsupervised method as it solves the eigen decomposition problem to obtain the optimal vectors for dimensionality reduction. Linear Discriminant Analysis (LDA) [5] is a representative supervised method in feature extraction and dimensionality reduction, which uses the label information to improve the performance in classification. By maximizing the ratio of the between-class scatter to the within-class scatter of the training dataset, LDA can obtain an optimal set of discriminative vectors [6]. However, a drawback of LDA is that it is unsuitable for small sample size problem in face recognition. An effective model called PCA+LDA [7], which joints the two major techniques to obtain the discriminant vectors [8], has been proposed to deal with the problem. The other two methods called Particle Swarm Optimization (PSO) [9] and Backtracking Search Algorithm (BSA) [10] can also significantly reduce the number of features so as to reduce the computational complexity and at the same time guarantee the same level of performance.

However, PCA and LDA cannot provide the sparse projections for feature extraction since the learned projections are the linear combination of the data [11]. Recently, sparse regression showed the outstanding performance in feature selection and extraction [12]–[15]. By adding sparsity penalty for feature selection, the accuracy and robustness of these methods might be improved. Thus many studies focused on the sparse learning for variable selection. Zou *et al.* proposed an effective model called Sparse Principle Component Analysis (SPCA) [4] to generate modified principle components with sparse loadings by using the lasso or elastic net constraint [16], [17]. Some other sparse PCA algorithms, such as the SCoTLASS algorithm [18], the DSPCA algorithm [19] were proposed. All of these methods focused on sparse learning without using the class label information. Besides, Feng *et al.* proposed the unsupervised learning method based on maximum information and minimum redundancy (MIMR) [20] for hyperspectral image analysis, and Li *et al.* proposed another unsupervised feature selection method by nonnegative spectral analysis and redundancy control [21].

Some other researchers developed the supervised methods using the label information to perform sparse learning for feature extraction and selection. One of the effective methods is Sparse Discriminant Analysis (SDA) [22], which extends linear discriminant analysis to sparse case by imposing the sparsity constraint. Moreover, to overcome the data piling problem of LDA in the high dimensional and low sample size (HDLSS) case, Qiao *et al.* proposed sparse LDA to obtain sparse linear

discriminant vectors by taking the relationship between Fishers LDA and a generalized eigenvalue problem into consideration [23]. Besides, some semi-supervised methods were also proposed. A semi-supervised method, which used partially labeled data samples, was designed to achieve batch feature selection [24]. Another method called Hessian sparse feature selection based on $L_{2,1/2}$-matrix norm (HFSL) was proposed for semi-supervised sparse feature selection [25]. For the multimodal case, Ding *et al.* proposed a method using multimodal information to jointly learn face representation [26]. Furthermore, sparse regularization learning were also used in classification designation for different pattern recognition tasks [27]–[29].

It is a well-known fact that not all data are distributed on a linear subspace. They may lie on the nonlinear low-dimensional manifold embedding on the high-dimensional ambient space. Therefore, many manifold learning algorithms were proposed. The representative methods include Neighborhood Preserving Embedding (NPE) [30], Isometric Projection (IsoP) [31] and Locality Preserving Projection(LPP) [32], [33], etc. These algorithms aimed to preserve the local geometric structure of the data manifold. By introducing the locality for sparse subspace learning, Cai *et al.* also proposed a new method called Unified Sparse Subspace Learning (USSL) [34]. USSL utilized the elastic net for regression to simultaneously select the most important variables and take the local geometric structure into consideration. Besides, by combining the global pairwise sample similarity with local geometric structure, a new method called GLSPFS [35] was proposed by Liu *et al.* for feature selection.

In recent years, a great deal of attention has been paid to the regression methods with different norms for image recognition, feature extraction and variable selection [36]. For example, nuclear norm regression methods were proposed in [37], [38] for face recognition. The L_1-norm based sparse regularized learning methods [39]–[41] have been used for face reconstruction and recognition [42]. A feature selection algorithm framework called Feature-weighting as Regularized Energy-based Learning (FREL) was proposed by Li *et al.* [43]. Based on least square regularization, Yang *et al.* [44] proposed the discriminative projection method. And the traditional RDA was further developed as Parameterless Reconstructive Discriminant Analysis (PRDA) [45] for feature extraction. In [46], the L_1-norm minimization was employed to design a specific loss function, by which the abundant user tagged Web images are treated as noisy samples and will not be emphasized so as to perform robust semantic video indexing. Other methods, such as [47]–[53] were also proposed to deal with different feature selection problems. The methods in terms of jointly sparse subspace learning attracted great attention in the field of feature selection. Since the $L_{2,1}$-norm based regression loss function is robust to outlier in data set, it can improve the robustness in learning steps. Therefore, some algorithms with joint $L_{2,1}$-norm regularization were proposed to guarantee the joint sparsity for feature extraction. The model called Robust Feature Selection (RFS) [54] via joint $L_{2,1}$-norms minimization showed the good performance for feature selection with joint sparsity. Yang *et al.* proposed another model called Unsupervised Discriminative Feature Selection (UDFS) [55] for sparse subspace learning. Experimental results showed that UDFS outperforms the existing unsupervised feature extraction methods and its main advantage is that UDFS not only uses discriminative information but also uses local structure of datas distribution for feature selection [56]. The $L_{2,1}$-norm regularization is also used in [57] to discover the common features shared across all the clustering tasks so as to obtain a discriminative low dimensional space for clustering. Except for the jointly sparse feature selection, the $L_{2,1}$-norm was also widely used to deal with the joint-sparse recovery problems [58], [59] in computer vision.

Although a lot of methods have been developed to improve the performance of regression methods, there still exist some problems to be solved. For example, when the number of the class is too small, not enough projections can be obtained by the classical regression methods and/or their extensions to achieve higher classification accuracy. Also, most existing regression method do not simultaneously consider the geometric structure of the data as well as the sparsity of the projection matrix. In this paper, we propose a novel model called Jointly Sparse Locality Regression (JSLR) for feature extraction and selection. JSLR can not only avoid the limitation in the existing regression methods but also guarantee the sparsity by using $L_{2,1}$-norm regularization on the projection matrix. What is more, JSLR incorporates the local structure of the data in regression form, by which the optimization problem can be easily optimized so as to obtain better performance of feature extraction with less computational time.

The main contributions of this paper are described as below:

1) The number of the projections in LDA-based methods or regression-based methods is limited by the rank of the so-called between-class scatter matrix or the number of the classes. The proposed method can break out the limitation to obtain more projections for feature extraction by designing a novel regression model.
2) Theoretical connections between the proposed method and the previous regression methods are discovered. Moreover, the convergence of the proposed algorithm is also proved.
3) The experimental results of the proposed model with or without $L_{2,1}$-norm regularization indicate that adding $L_{2,1}$-norm penalty on the projection matrix can obtain joint sparsity for feature extraction so as to achieve high recognition rate.

The rest of this paper is organized as follows: In Section II, we discuss the related works and the extension based on ridge regression will be shown in Section III. In Section IV, we propose our objective function and the local optimal solution. Section V focuses on theoretical analysis (the convergence and the computational complexity). The proposed model will be evaluated by several well-known databases in Section VI. In Section VII, we draw a conclusion for this paper.

II. Related Works

In this section, the notations used in this paper will be briefly described and the related works will be reviewed.

A. Notations

Scalars are denoted as lowercase or uppercase italic letters, i.e. $i, j, d, p, n, c, \alpha_1, \alpha_2$ etc. while vectors are represented as bold lowercase italic letters, i.e. $\boldsymbol{x}, \boldsymbol{y}$, etc. Matrices are defined as bold uppercase italic letters, i.e. $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W}$ etc.

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in R^{d \times n}$ then $\boldsymbol{X}$ denotes a $d \times n$ matrix as the original data set, where n is the number of total training samples and d denotes the features dimension for each sample. Let $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_c] \in R^{n \times c}$ to be a $n \times c$ matrix as the label matrix falling into c classes.

B. Regressions

Ridge Regression [60] is a regularized least square method for multivariate learning. It aims to solve the multicollinearity problem of covariates in samples.

The optimization problem of the simplest regression is

$$\boldsymbol{P}^0 = \arg\min_{\boldsymbol{P}} ||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{P}||_F^2 \tag{1}$$

where $\boldsymbol{X}$ denotes the training set of n training data. The matrix $\boldsymbol{P} \in R^{d \times c}$ aims to lead the linear dependency between the training data and the corresponding labels. By setting the derivatives of (1) with respect to $\boldsymbol{P}$ equaling to 0, we have the optimal solution

$$\boldsymbol{P}^0 = \left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{Y} \tag{2}$$

However this optimal solution is only suitable for the case when $\boldsymbol{X}\boldsymbol{X}^T$ is a full-rank matrix. Because of the small-sample size problem, the matrix $\boldsymbol{X}\boldsymbol{X}^T$ may be not a full-rank one. Therefore, to solve the singular problem in computing the inverse of $\boldsymbol{X}\boldsymbol{X}^T$ the L_2-norm regularized term was added to (1), and then we have the classical ridge regression optimization problem:

$$\boldsymbol{P}^* = \arg\min_{\boldsymbol{P}} ||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{P}||_F^2 + \alpha||\boldsymbol{P}||_F^2 \tag{3}$$

By setting the derivatives of (3) with respect to $\boldsymbol{P}$ equaling to 0, we have the optimal solution for (3) as

$$\boldsymbol{P}^* = \left(\boldsymbol{X}\boldsymbol{X}^T + \alpha\boldsymbol{I}\right)^{-1}\boldsymbol{X}\boldsymbol{Y} \tag{4}$$

For further analysis in the following sections, we need to represent the optimal solution of (3). Based on the SVD of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, the optimal solution can be represented as

$$\boldsymbol{P}^* = \boldsymbol{U}\frac{\boldsymbol{D}}{\boldsymbol{D}^2 + \alpha\boldsymbol{I}}\boldsymbol{V}^T\boldsymbol{Y} \tag{5}$$

From (2) and (5), we can know that the optimal projection matrix $\boldsymbol{P}^0$ and $\boldsymbol{P}^*$ have the size, i.e. $d \times c$. That is, we can obtain only c projective vectors for feature extraction.

C. The Review of LPP

LPP [32], [33] computes the best linear approximations to the eigenfunctions of the manifold's Laplace Beltrami operator. It aims to preserve local information and to find an embedding subspace which detects the most essential data manifold structure [61], [62]. The objective function of LPP is to minimize

$$\begin{aligned}\frac{1}{2}\sum_{ij}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2\overline{\boldsymbol{w}}_{ij} &= \frac{1}{2}\sum_{ij}\left\|\boldsymbol{B}^T\boldsymbol{x}_i - \boldsymbol{B}^T\boldsymbol{x}_j\right\|^2\overline{\boldsymbol{w}}_{ij} \\ &= tr\left(\boldsymbol{B}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}\right) = tr\left(\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{B}\right)\end{aligned} \tag{6}$$

where matrix $\boldsymbol{B}$ denotes the transformation matrix, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ denote the low-dimensional vectors of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in subspace $\boldsymbol{B}$, respectively. $\bar{\boldsymbol{W}}$ is supposed to be the similarity matrix of all pairwise data points, $\boldsymbol{L} = \overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}$ is Laplacian matrix. $\bar{\boldsymbol{D}}$ is a diagonal matrix and its element $\boldsymbol{d}_{ii}$ is column or row sum of matrix $\bar{\boldsymbol{W}}$ (because $\bar{\boldsymbol{W}}$ is symmetric), i.e. $\overline{\boldsymbol{d}}_{\boldsymbol{ii}} = \sum_i \overline{\boldsymbol{w}}_{\boldsymbol{ij}}$.

The similarity matrix $\overline{\boldsymbol{w}}_{\boldsymbol{ij}}$ is defined as:

$$\bar{\boldsymbol{w}}_{ij} = \begin{cases} exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/t\right), & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 < \varepsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where parameter $t \in R$, ε denotes the radius of the local neighborhood and it can be a sufficiently small positive value ($\varepsilon > 0$). In Eq. (7), the similarity matrix $\overline{\boldsymbol{w}}_{\boldsymbol{ij}}$ might be sensitive to the value of the parameter t. To solve this problem, recently a parameter-free method was proposed in [63].

By considering the similarity matrix $\bar{\boldsymbol{W}}$, the relationship between each data pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in original space can be preserved by reconstructing the relationship between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ in the low dimensional space $\boldsymbol{B}$ with $\sum_{ij}(\boldsymbol{y_i} - \boldsymbol{y_j})^2\overline{\boldsymbol{w}}_{\boldsymbol{ij}}$ where $\boldsymbol{y_i} = \boldsymbol{B}^T\boldsymbol{x_i}$ and $\boldsymbol{y_j} = \boldsymbol{B}^T\boldsymbol{x_j}$. The optimal projections can be obtained by solving the following generalized eigen-function:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T b = \lambda\mathbf{X}\bar{\mathbf{D}}\mathbf{X}^T. \tag{8}$$

Suppose $\lambda_i (i = 1, 2, \ldots, d)$ are eigenvalues of problem 8, we can sort the eigenvalues in ascending order, then matrix $\boldsymbol{B} = [\boldsymbol{b}^1, \boldsymbol{b}^2, \ldots, \boldsymbol{b}^k]$ combined of k eigenvectors corresponding to the first k smallest eigenvalues is the final projection matrix of LPP.

III. The Extension Based on Ridge Regression

In this section, we firstly review the definition of $L_{2,1}$-norm and its property. Then we analyze the advantages and disadvantages of ridge regression. Meanwhile, we also propose a simple extension based on ridge regression.

A. The Definition of $L_{2,1}$-Norm and Its Property

Some well-known models such as PCA, multilinear PCA (MPCA) [64], etc. use L_2-norm as the measurement to compute the optimal projections in computer vision and face recognition. However, a large amount of experimental results have shown that in sparse feature selection, L_1-norm outperforms L_2-norm because of its generalization and the robustness for classification [16], [17], [61]. By combining the advantages of both L_1-norm and part property of L_2-norm, researchers obtain joint $L_{2,1}$-norm minimization on both loss functions and regularization term for robust sparse learning for feature extraction [53]. Therefore, we use the $L_{2,1}$-norm instead of L_2-norm as a

new measurement for model design to overcome the problem of L_2-norm being sensitive to outliers in a certain sense [54].

The $L_{2,1}$-norm of a matrix is defined as

$$\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum\nolimits_{j=1}^{m} \boldsymbol{m}_{ij}^2} = \sum_{i=1}^{n} \left\|\boldsymbol{m}^{\boldsymbol{i}}\right\|_2 \tag{9}$$

where the i-th row and the j-th column of a matrix $\boldsymbol{M} = (\boldsymbol{m}_i j)$ are denoted as $\boldsymbol{m}^i$ and $\boldsymbol{m}_j$.

The common advantage of $L_{2,1}$-norm and L_1-norm based loss function is that they are more robust to outliers. However, the major difference between $L_{2,1}$-norm and L_1-norm is that $L_{2,1}$-norm regularization is suitable for selecting meaningful or more powerful discriminant features from the data points with joint sparsity. The $L_{2,1}$-norm based regularized methods can eliminate those useless interferences via making the elements in some rows of the projection matrix become zero such that the important features of the data points are emphasized and the insignificant features are ignored (filtered out) when conducting feature selection or extraction. Another advantage of $L_{2,1}$-norm is that the $L_{2,1}$-norm based methods are fast convergent and thus the computational cost is lower (this can be verified from computational cost of the $L_{2,1}$-norm based methods compared with the L_1-norm based methods in Table XI in Experiment section) [54], [56].

In all, employing the $L_{2,1}$-norm instead of L_1-norm as the regularization can obtain the joint sparsity to improve the performance and at the same time reduce the computational cost for efficient feature extraction and selection on image recognition tasks [54].

B. A Key Drawback in Traditional Regression

In (1) and (3), there exists a problem that when the number of the classes is too small, the traditional models cannot obtain enough projections for achieving good performance in pattern recognition. Thus, it is possible that learning more projection may improve the performance in feature extraction and classification [4]. In order to obtain more projections in the regression model, a tractable approach is to modify the representation $||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{P}||_F^2$ to be $||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T||_F^2$, which means that the matrix $(\boldsymbol{B}\boldsymbol{A}^T) \in R^{d\times c}$ takes the place of the matrix $\boldsymbol{P} \in R^{d\times c}$ in the model. Thus we have the following optimization problem:

$$(\boldsymbol{A}^*, \boldsymbol{B}^*) = \arg\min_{\boldsymbol{A},\boldsymbol{B}} ||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T||_F^2, \text{ s.t. } \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \tag{10}$$

where $\boldsymbol{A}$ is a $c \times k$ matrix and the size of matrix $\boldsymbol{B}$ is $d \times k$ where the notation k is any positive integer and c denotes the number of classes. In other words, the optimal solution $\boldsymbol{B}$ with size $d \times k$ is able to break out the limitation of class number of the training data since the size of $\boldsymbol{B}$ is not related to the class number and the variable k in $\boldsymbol{B}$ is not related to the class number and the variable k in $\boldsymbol{B}$ can be set as value that is larger than c, while $\boldsymbol{P}$ with size $d \times c$ indicates that it can obtain at most c projections for feature selection.

From (10), we have

$$\begin{aligned} &||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T||_F^2 \\ &\quad = Tr(\boldsymbol{Y}^T\boldsymbol{Y}) - Tr(2\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}). \end{aligned} \tag{11}$$

By setting the derivatives of (11) with respect to $\boldsymbol{B}$ equaling to 0, the problem (11) is minimized at

$$\boldsymbol{B}^* = \left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}, \tag{12}$$

where $\boldsymbol{B}^*$ represents the optimal solution of (10).

Denote the SVD of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, where $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, we have

$$\boldsymbol{B}^* = \boldsymbol{U}\frac{1}{\boldsymbol{D}}\boldsymbol{V}^T\boldsymbol{Y}\boldsymbol{A}. \tag{13}$$

For (1) and 10), we have following propositions:

Proposition 1: Suppose $\boldsymbol{X}\boldsymbol{X}^T$ is the full-rank matrix. Let $\boldsymbol{P}^0$ be the optimal solution to (1) and $\boldsymbol{B}^*$ be the optimal solution to 10), if $k = c$ (i.e. the number of projection is equal to the number of class), then span($\boldsymbol{B}^*$) = span($\boldsymbol{P}^0$).

If $k > c$, the optimal solution $\boldsymbol{B}^*$ with size $d \times k$ in (10) can obtain k projections instead of c projections as obtained by $\boldsymbol{P}^0$ in (1), which breaks out the small-class problem. In Proposition 1, the reason why the small-class problem is addressed by (10) is that the $\boldsymbol{P}^0$ in (1) has c projections while the optimal solution $\boldsymbol{B}^*$ for (10) can learn k projections to perform feature extraction and classification, where k can be set as any integer. In other words, the number of the learned projections from (10) is not limited by the number of class and thus the small-class problem is addressed.

Similarly, for (3) and 10), we have following proposition:

Proposition 2: Let $\boldsymbol{P}^*$ be the optimal solution to (3) and $\boldsymbol{B}^*$ be the optimal solution to 10), if $k = c$, (i.e. the number of projection is equal to the number of class), then span($\boldsymbol{B}^*$) = span($\boldsymbol{P}^*$). Furthermore if $\alpha \to 0$, the metric matrices derived by $\boldsymbol{B}^*$ and $\boldsymbol{P}^*$ for classification are equivalent to each other.

If $k > c$, the optimal solution $\boldsymbol{B}^*$ with size $d \times k$ in (10) can obtain k projections instead of c projections as obtained by $\boldsymbol{P}^*$ in (3), which breaks out the small-class problem.

Proposition 2 indicates that when $\alpha \to 0$ (or using $\alpha = \varepsilon$, whereε is a very small number), the performance using $\boldsymbol{B}^*$ and $\boldsymbol{P}^*$ for feature extraction and classification will achieve the same results. If $k > c$, (10) can obtain more than c projections to perform feature selection or extraction, which provides the theoretical guarantee for the performance of (10). From Propositions 1 and 2, we can draw the following conclusion:

Corollary 1: If $\alpha \to 0$ and the matrix $\boldsymbol{X}\boldsymbol{X}^T$ is nonsingular, (1) and (3) have the same solution space.

C. Other Drawbacks of Ridge Regression

Adding L_2-norm term for regression is of great importance to deal with the singular problem in (1). Moreover, it shows that no matter the matrix $\boldsymbol{X}\boldsymbol{X}^T$ is singular or nonsingular, the classical ridge regression in (3) is able to obtain the optimal solution and (1) is only a special case of (3) with the regularization parameter $\alpha = 0$. However, there are still some obvious disadvantages in

(3) since the optimal solution $\boldsymbol{P}^*$ is not sparse, and thus it loses the feature selection function. Furthermore, the optimal solution $\boldsymbol{P}^*$ in classical ridge regression model only contains the global information of the dataset and it ignores the local geometric structure. Thus, it is necessary to develop a new algorithm to deal with the above problems so as to enhance the effectiveness in feature extraction and pattern recognition. In the next section, we will propose a new model by jointing $L_{2,1}$-norm regularization and locality regression to deal with the above problems.

IV. Jointly Sparse Locality Regression Analysis

In this section, the motivations and discussion are firstly present and then the proposed objective optimization problem as well as the local optimal solution will be presented.

A. The Motivations and Discussion

Based on the discussion in Section III-C and III-D, we can conclude the drawbacks of most existing regression methods into three aspects. First, due to the limitation of small-class problem, most regression methods cannot obtain enough projections to discover an effective projection matrix for discriminant feature extraction and classification. Second, the local structure of the data plays an important role in reconstructing the relationship between different data pairs in the low dimensional space. However, most existing regression methods do not take the local structure into consideration when performing feature selection or extraction. Third, there is no specific regression methods that are designed as regression form incorporating the local structures of the data as well as the sparsity of projections for feature selection and extraction.

Currently, deep learning technique is a research hotspot and it has been applied to the tasks of face recognition and object classification [65]. In spite of the high recognition rate of deep learning methods, behind is large-scale computing and long-term training. What is more, when the amount of data is not large enough, using deep learning methods for classification tends to obtain low performance because of the overfitting. In addition, most feature extraction methods based on deep learning [66], [67] do not consider the local structures of the data when doing convolutional operations. Even though they can obtain more abstract interpretation of the data, the relationship among different images is still missing. Therefore, developing efficient traditional feature extraction methods is still necessary for face recognition.

In conclusion, it is desirable to design a method that can solve the drawbacks of the existing regression methods and improve the performance of feature extraction to obtain high recognition rate with less computing time compared to the time-consuming and complicated deep learning methods.

B. The Objective Function of JSLR

To deal with the problems presented in Section III-C, Jointly Sparse Locality Regression Analysis (JSLR) is proposed to obtain a subset of jointly sparse projections for feature extraction and selection from the original data set. We also introduce the locality preserving regularized term to the model so as to characterize the local geometric structure of the data. Thus, we present the objective function with joint $L_{2,1}$-norm penalty and locality regularization. Let $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k]$ and $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k]$ be the variables of the following regression problem:

$$\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}} = \arg\min_{\boldsymbol{A},\boldsymbol{B}} \left(\sum_{i=1}^{n} ||\boldsymbol{y}_i - \boldsymbol{x}_i{}^T B A^T||_2^2 + \alpha_1 ||\boldsymbol{B}||_{2,1} + \alpha_2 \sum_{i=1}^{n}\sum_{j=1}^{n} ||\boldsymbol{B}^T\boldsymbol{x}_i - \boldsymbol{B}^T\boldsymbol{x}_j||_2^2 \bar{\boldsymbol{w}}_{ij} \right)$$
$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \tag{14}$$

or in the matrix form

$$\left(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}\right) = \arg\min_{\boldsymbol{A},\boldsymbol{B}} \left(||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T||_F^2 + \alpha_1 ||\boldsymbol{B}||_{2,1} + \alpha_2 \sum_{i=1}^{n}\sum_{j=1}^{n} ||\boldsymbol{B}^T\bar{\boldsymbol{x}}_i - \boldsymbol{B}^T\bar{\boldsymbol{x}}_j||_2^2 \bar{\boldsymbol{w}}_{ij} \right),$$
$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \tag{15}$$

where α_1 and α_2 are the regularization parameters. Since (14) and (15) have two variables and two kinds of norms in the model, they are not easy to be solved directly. Therefore, an alternatively iterative approach will be developed to solve the optimization problem in next section.

C. The Solutions of JSLR

From the definition of the $L_{2,1}$-norm on the projection matrix $\boldsymbol{B}$, we have the diagonal matrix $\boldsymbol{D_B}$ denoted as [54]

$$(\boldsymbol{D_B})_{ii} = \frac{1}{2||\boldsymbol{b}^i||_2}, \tag{16}$$

where $\boldsymbol{b}^i$ represents the i-th row of matrix $\boldsymbol{B}$.

Then from [56], we have the following equation:

$$||\boldsymbol{B}||_{2,1} = Tr(\boldsymbol{B}^T\boldsymbol{D_B}\boldsymbol{B}). \tag{17}$$

With the above preparation, we have

$$\begin{aligned} &||\boldsymbol{Y} - \boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T||_F^2 + \alpha_1||\boldsymbol{B}||_{2,1} \\ &\quad + \alpha_2 \sum_{i=1}^{n}\sum_{j=1}^{n} ||\boldsymbol{B}^T\boldsymbol{x}_i - \boldsymbol{B}^T\boldsymbol{x}_j||_2^2 \bar{\boldsymbol{w}}_{ij} \\ &= Tr\left(\boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}\right. \\ &\quad \left. + \alpha_1\left(\boldsymbol{B}^T\boldsymbol{D_B}\boldsymbol{B}\right) + \alpha_2\boldsymbol{B}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}\right). \end{aligned} \tag{18}$$

Since the optimization problem has two variables, we need to fix one to compute the other. For fixed $\boldsymbol{A}$, by setting the derivatives of (18) with respect to $\boldsymbol{B}$ equaling to 0, (18) is minimized by

$$\bar{\boldsymbol{B}} = \left(\boldsymbol{X}\boldsymbol{X}^T + \alpha_1\boldsymbol{D}_B + \alpha_2\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A} \tag{19}$$

Hence, when $\boldsymbol{A}$ is fixed, the objective function of 14 or 15 is minimized at the local optimal solution $\boldsymbol{B}$. When fixing $\boldsymbol{B}$, $Tr(\boldsymbol{Y}^T\boldsymbol{Y} + \boldsymbol{B}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B} + \alpha_1(\boldsymbol{B}^T\boldsymbol{D_B}\boldsymbol{B}) + \alpha_2\boldsymbol{B}^T\boldsymbol{X}(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}})\boldsymbol{X}^T\boldsymbol{B})$ becomes a constant and thus it can be

ignored. In such case, the following maximization problem gives the optimal solution to (18):

$$\max_{\boldsymbol{A}} Tr\left(\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}\right) \text{ s.t. } \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}. \tag{20}$$

Let $\bar{\boldsymbol{A}}$ be the optimization of (20). From the Theorem 4 in [4], we have

$$\bar{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{V}^T, \tag{21}$$

where $\boldsymbol{U}$, $\boldsymbol{V}$ is the SVD decomposition value of $\boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{B}$.

In addition, we can also have the following conclusion from above formulation:

Theorem 1: Let $\bar{\boldsymbol{B}}$ be the local optimal solution of the optimization problem (14) or (15). If $\alpha_1 \to 0$ and $\alpha_2 \to 0$, the linear subspace spanned by the optimal solution of (14) or (15) approximates to the linear subspace spanned by $\boldsymbol{P}^0$ and $\boldsymbol{P}^*$, namely, span($\bar{\boldsymbol{B}}$) = span($\boldsymbol{P}^0$) and span($\bar{\boldsymbol{B}}$) = span($\boldsymbol{P}^*$).

Proof: The proof is in the Appendix.

For (19), when $\alpha_1 = 0$ and $\alpha_2 = 0$, then $\overline{\boldsymbol{B}} = (\mathbf{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A} = \boldsymbol{B}^* = \boldsymbol{P}^0\boldsymbol{A}$, where $\boldsymbol{P}^0$ and $\boldsymbol{B}^*$ is the optimal solution corresponding to (1) and (10). As Proposition 1 and Proposition 2 have presented the relationship between (10) and (1), (10) and (3) respectively, it is easy for us to have the following conclusions: ■

Corollary 2: With the same assumptions and notations as in Theorem 1, when $\alpha_2 = 0, \alpha_1 \to 0$, $\boldsymbol{P}^0$ and $\bar{\boldsymbol{B}}$ have the same linear subspace, namely, span($\boldsymbol{P}^0$) = span($\bar{\boldsymbol{B}}$).

Corollary 3: With the same assumptions and notations as in Theorem 1, when $\alpha_1 = 0, \alpha_2 \to 0$, $\boldsymbol{P}^0$ and $\bar{\boldsymbol{B}}$ have the same linear subspace, namely, span($\boldsymbol{P}^0$) = span($\bar{\boldsymbol{B}}$).

In summary, from Theorem 1, Corollary 2 and Corollary 3, we can know that either (14) or (15) provides a basic theoretical guarantee for the effectiveness of the proposed regression model. Namely, when the parameters of the proposed model are set suitably, the optimal solution space of the ridge regression can be derived from (15). This means that the optimal projection of JSLR can approximate to the subspace spanned by the traditional regression models. Besides, by utilizing the advantages of $L_{2,1}$-norm regularization and locality preserving property, the proposed model is able to compute the jointly sparse projections and preserve the local geometric structure of the data for feature extraction. The detail of the iterative algorithm was illustrated in Algorithm 1.

Algorithm 1: JSLR Algorithm

Input: The training data $\boldsymbol{X} \in R^{d\times n}$,
the training data label $\boldsymbol{Y} \in R^{n\times c}$,
matrices $\bar{\boldsymbol{D}} \in R^{n\times n}$, $\bar{\boldsymbol{W}} \in R^{n\times n}$,
the objective dimension k ($k = 1, 2, \ldots, n$),
maximum number of the iteration: $\max Step$.

Step 1: Compute matrices $\bar{\boldsymbol{D}}$, $\bar{\boldsymbol{W}}$, and initialize matrix $\boldsymbol{D}_B$, $step = 0$, $converged = false$.

Step 2: While $!converged$ and $step <= \max Step$
- Compute $\boldsymbol{B}$ using
$\boldsymbol{B} = (\boldsymbol{X}\boldsymbol{X}^T + \alpha_1\boldsymbol{D}_{\boldsymbol{B}} + \alpha_2\boldsymbol{X}(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}})\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}$
- Compute $\boldsymbol{D}_B$ using $(\boldsymbol{D}_B)_{ii} = \frac{1}{2||\boldsymbol{b}^i||_2}$
- Compute $\boldsymbol{A}$ using $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^T$
- Update $converged = true$ when $\boldsymbol{B}$ is approximately changeless.

Step 3: Standardize the matrix $\boldsymbol{B}$ to a final normalized matrix
and return it for feature selection.

Output: Low-dimensional discriminative subspace $\boldsymbol{B} \in R^{d\times k}, k = 1, 2, \ldots, n$.

D. Comparison and Discussion

In this section, we compare our algorithm JSLR with other methods, such as PCA, SPCA, LDA, LPP and so on. Both PCA and SPCA are outstanding in data processing and dimensionality reduction. PCA projects the original d-dimensional data onto k ($<< d$)-dimensional linear subspace with the combination of all the original variables. SPCA aims to produce modified sparse principal components by lasso (or elastic net) technique. But it just focuses on the global structure of the original data and ignore the local structure. Different from SPCA, JSLR can efficiently preserve the local geometric structure of the data set.

Some other subspace learning algorithms, LPP, NPE, etc. are able to preserve local structure of the original data. However, they cannot provide the jointly sparse property for the learned subspace. Compared with them, JSLR achieves this goal by adding $L_{2,1}$-norm regularization to make the elements in some rows of the projection to be 0 for efficient feature extraction and selection.

Ridge regression is frequently used in face recognition. However, when the class number of training sample is too small, ridge regression cannot obtain more projections than the number of the classes for feature extraction. The same problem exists in LDA. In contrast, the number of the projections of JSLR is not limited by the number of the classes in training data. In spite of given a small number of classes in training sample set, JSLR can obtain any number of projections for feature selection and the number of the projections is freely set by the users.

In summary, the advantages of JSLR against PCA, SPCA, LDA, ridge regression and LPP are that JSLR can obtain joint sparsity and preserve the local structure for pattern recognition. Another major difference between JSLR and other classical methods is that the number of the training sample classes in JSLR is allowed to be very small but it still can learn more projections than the number of classes. These advantages make JSLR achieve high recognition rate.

V. Theoretical Analysis

In this section, we present the theoretical analysis including convergence analysis and computational complexity analysis.

A. The Convergence

To verify the convergences of the proposed iterative algorithm, we begin with the following Lemmas:

Lemma 1: [54] For any two non-zero constants a and b, we have the following inequality:

$$\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}}. \tag{22}$$

Lemma 2: [54] Denoted $\boldsymbol{V}$ as any nonzero matrix, $\boldsymbol{V} \in R$, the following inequality holds:

$$\sum_i ||\boldsymbol{v}_t^i||_2 - \sum_i \frac{||\boldsymbol{v}_t^i||_2^2}{2||\boldsymbol{v}_{t-1}^i||_2} \leq \sum_i ||\boldsymbol{v}_{t-1}^i||_2 - \sum_i \frac{||\boldsymbol{v}_{t-1}^i||_2^2}{2||\boldsymbol{v}_{t-1}^i||_2}, \tag{23}$$

where $\boldsymbol{v}_t^i$, $\boldsymbol{v}_{t-1}^i$ denote the i-th row of matrix $\boldsymbol{V}_t$ and $\boldsymbol{V}_{t-1}$.

Proof: Let $||\boldsymbol{v}_t^i||_2^2$ and $||\boldsymbol{v}_{t-1}^i||_2^2$ be the substitute of a and b in (22), the following inequality is valid for any i.

$$||\boldsymbol{v}_t^i||_2 - \frac{||\boldsymbol{v}_t^i||_2^2}{2||\boldsymbol{v}_{t-1}^i||_2} \leq ||\boldsymbol{v}_{t-1}^i||_2 - \frac{||\boldsymbol{v}_{t-1}^i||_2^2}{2||\boldsymbol{v}_{t-1}^i||_2}, \tag{24}$$

Thus, (23) as the sum form of (24) also holds (22). With the above Lemma 1 and Lemma 2, we have the following theorem: ■

Theorem 2: Given all the parameters in the objective function except $\boldsymbol{A}$ and $\boldsymbol{B}$, the iterative approach shown in Algorithm 1 will monotonically decrease the objective function value of (14) or (15) in each iteration and provides a local optimal solution of the problem.

Proof: For simplicity, we denote the objective function of (18) as $F(\boldsymbol{B}, \boldsymbol{A}) = F(\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{D}_{\boldsymbol{B}})$. Suppose for the $(t-1)$-th iteration, both $\boldsymbol{A}_{t-1}$ and $\boldsymbol{B}_{t-1}$ can be obtained. Then we have the following inequality from (19):

$$F(\boldsymbol{B}_t, \boldsymbol{A}_{t-1}, (\boldsymbol{D}_{\boldsymbol{B}})_{t-1}) \leq F(\boldsymbol{B}_{t-1}, \boldsymbol{A}_{t-1}, (\boldsymbol{D}_{\boldsymbol{B}})_{t-1}). \tag{25}$$

For $\boldsymbol{A}_t$, as its optimal value comes from SVD and this will further decrease the value of the objective function, it goes

$$F(\boldsymbol{B}_t, \boldsymbol{A}_t, (\boldsymbol{D}_{\boldsymbol{B}})_{t-1}) \leq F(\boldsymbol{B}_{t-1}, \boldsymbol{A}_{t-1}, (\boldsymbol{D}_{\boldsymbol{B}})_{t-1}). \tag{26}$$

In (18), since $\boldsymbol{Y}^T\boldsymbol{Y}$ is a constant, it can be ignored and we need to minimize

$$Tr\left(-2\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A} + \boldsymbol{B}^T(\boldsymbol{X}\boldsymbol{X}^T + \alpha_1\boldsymbol{D}_{\boldsymbol{B}} + \alpha_2\boldsymbol{X}\left(\bar{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T)\boldsymbol{B}\right)$$

As we have obtained the optimal $\boldsymbol{B}_t$ and $\boldsymbol{A}_t$, then the following inequality holds:

$$\begin{aligned} &Tr\left(-2{\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_t + {\boldsymbol{B}_t}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \alpha_1(\boldsymbol{D}_{\boldsymbol{B}})_{t-1} + \alpha_2\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\right)\boldsymbol{B}_t\right) \\ &\leq Tr\left(-2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_{t-1} + {\boldsymbol{B}_{t-1}}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \alpha_1(\boldsymbol{D}_{\boldsymbol{B}})_{t-1} + \alpha_2\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\right)\boldsymbol{B}_{t-1}\right) \end{aligned} \tag{27}$$

That is

$$\begin{aligned} &Tr\left(-2{\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_t + {\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}_t + \alpha_2{\boldsymbol{B}_t}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_t\right) + \alpha_1\sum_i \frac{||\boldsymbol{b}_t^i||_2^2}{2||\boldsymbol{b}_{t-1}^i||_2} \\ &\leq Tr\left(-2\mathbf{B}_{t-1}{}^T\mathbf{X}\mathbf{Y}\mathbf{A}_{t-1} + \mathbf{B}_{t-1}{}^T\mathbf{X}\mathbf{X}^T\mathbf{B}_{t-1} + \alpha_2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_{t-1}\right) + \alpha_1\sum_i \frac{||\boldsymbol{b}_{t-1}^i||_2^2}{2||\boldsymbol{b}_{t-1}^i||_2} \end{aligned} \tag{28}$$

TABLE I
THE COMPUTATIONAL COMPLEXITIES

Iteration variable	computational complexities
$\boldsymbol{B}$	$O(d^3)$
$\boldsymbol{D}_{\boldsymbol{B}}$	$O(d^2)$
$\boldsymbol{A}$	$O(d^3)$

Then the above inequality indicates

$$\begin{aligned} &Tr\left(-2{\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_t + {\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}_t + \alpha_2{\boldsymbol{B}_t}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_t\right) + \alpha_1\sum_i ||\boldsymbol{b}_t^i||_2 - \alpha_1\left(\sum_i ||\boldsymbol{b}_t^i||_2 - \sum_i \frac{||\boldsymbol{b}_t^i||_2^2}{2||\boldsymbol{b}_{t-1}^i||_2}\right) \\ &\leq Tr\left(-2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_{t-1} + {\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}_{t-1} + \alpha_2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_{t-1}\right) + \alpha_1\sum_i ||\boldsymbol{b}_{t-1}^i||_2 - \alpha_1\left(\sum_i ||\boldsymbol{b}_{t-1}^i||_2 - \sum_i \frac{||\boldsymbol{b}_{t-1}^i||_2^2}{2||\boldsymbol{b}_{t-1}^i||_2}\right) \end{aligned} \tag{29}$$

According to Lemma 2, we further have

$$\begin{aligned} &Tr\left(-2{\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_t + {\boldsymbol{B}_t}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}_t + \alpha_2{\boldsymbol{B}_t}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_t\right) + \alpha_1\sum_i ||\boldsymbol{b}_t^i||_2 \\ &\leq Tr\left(-2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}_{t-1} + {\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}_{t-1} + \alpha_2{\boldsymbol{B}_{t-1}}^T\boldsymbol{X}\left(\overline{\boldsymbol{D}} - \overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\boldsymbol{B}_{t-1}\right) + \alpha_1\sum_i ||\boldsymbol{b}_{t-1}^i||_2. \end{aligned} \tag{30}$$

That is

$$\begin{aligned} F(\boldsymbol{B}_t, \boldsymbol{A}_t) &= F(\boldsymbol{B}_t, \boldsymbol{A}_t, (\boldsymbol{D}_{\boldsymbol{B}})_t) \\ &\leq F(\boldsymbol{B}_{t-1}, \boldsymbol{A}_{t-1}) = F(\boldsymbol{B}_{t-1}, \boldsymbol{A}_{t-1}, (\boldsymbol{D}_{\boldsymbol{B}})_{t-1}). \end{aligned} \tag{31}$$

From (31), we can conclude that the objective function value of (14) or (15) is monotonically decreased via the updating rule presented in Algorithm 1. Therefore, the proposed iterative algorithm finally converges to the local optimal solution. ■

B. Computational Complexity Analysis

For simplicity, we assume the dimension of training samples is d. Our proposed algorithm aims to compute the matrix $\boldsymbol{A}$ and $\boldsymbol{B}$. Computing $\boldsymbol{B}$ in in (19) needs O(d^3) while computing $\boldsymbol{D}_{\boldsymbol{B}}$ in (16) needs O(d^2). Since SVD of $\boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{B}$ also needs O(d^3), then the computational complexity of $\boldsymbol{A}$ is also O(d^3). It is easy to know that the main complexity of the algorithm is O(Td^3), where T denotes the number of iterations for convergence. Table I lists the computational complexities of each variable.

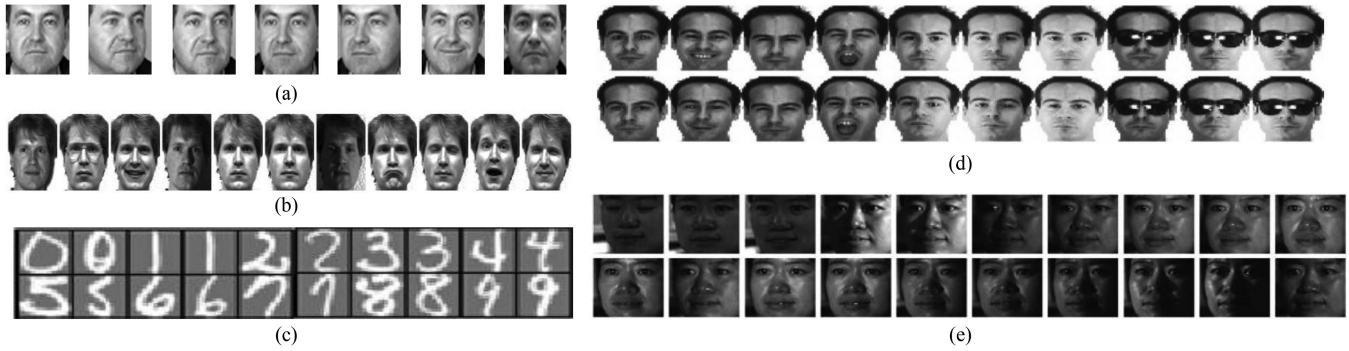


Fig. 1. Examples from FERET, Yale, USPS, AR, and CMU PIE data sets. (a) FERET. (b) Yale. (c) USPS. (d) AR. (e) CMU PIE.

VI. Experiments

In this section, to evaluate the proposed JSLR algorithm for feature selection, we conducted a set of experiments from three aspects: experiments on small-scale face databases, experiments on large-scale databases and experiments based on deep learning. In experiments, several classical as well as state-of-the-art methods are used as compared methods. They are the classical principle component analysis method PCA, the classical Ridge Regression (RR) [60], the Linear Discriminant Analysis (LDA) [5], the traditional sparse learning method SLDA based on L_1-norm [23], the local structure learning method Locality Preserving Projection (LPP) [32], the regression analysis of locality preserving projections via sparse penalty (SpLPP) [61] which applies sparsity penalty and minimization based on L_1-norm to locality preserving projections, the dictionary learning methods (i.e. label consistent K-SVD (LC-KSVD2) [68] and the Locality Constrained and Label Embedding Dictionary Learning (LCLE-DL) [69]), the most related $L_{2,1}$-norm regularization methods for feature selection and subspace learning (i.e. Unsupervised Discriminative Feature Selection (UDFS) [56] and Robust Feature Selection (RFS) [54]). In addition, the proposed method without $L_{2,1}$-norm regularization named JSLR($\alpha 1 = 0$) (i.e. the second term in the proposed objective function in Eq. (14) is removed) was added as a compared method to all experiments to evaluate the effectiveness of the jointly sparse regularization.

In all experiments we make comparison in the avenue of deep learning (the method is called Deep-NN in this paper). Deep-NN is completed by the following two steps. Firstly, we use the deep convolutional neural network (CNN) as the feature extractor to obtain the deep features of all samples. This process is similar to [70]. Secondly, we use the nearest neighbor classifier (NN) for classification. For the proposed JSLR, we also use the deep features instead of the traditional image features as input and this method is called Deep-JSLR for easy understanding. Note that the deep features of character database are obtained according to the tutorial of MNIST network on official Caffe site (http://caffe.berkeleyvision.org/gathered/examples/mnist.html.)

A. Experiments on Small-Scale Database

In this section, experiments on four databases, including FERET, AR, CMU PIE and Yale database, were conducted to evaluate the performance of the proposed method versus the compared methods under different variations of facial expression and lighting condition.

1) Experiments on FERET Face Database: The FERET face database [71] includes 1,400 images of 200 individuals (each individual has seven images). In the experiment, the facial portion of each original image was automatically cropped based on the location of the eyes, and the cropped images were resized to 40×40 pixels. The sample images of one person are shown in Fig. 1(a).

Experimental Setting: For all the databases, the image set is partitioned into two parts, i.e. the gallery and probe sets. In each database, l (l is no more than the number of class) images of each class are randomly selected to form the gallery set and the remaining images are used as the probe set. PCA was used as pre-processing to reduce the dimension of data. Then the proposed method and the compared methods were used to perform feature extraction, independently. Finally, nearest neighbor classifier was used for classification. The experiments were independently performed 10 times. The average recognition rates and the corresponding dimensions as well as the standard deviations of each method were listed on the Table II. Besides, the comparison results were also shown in the Fig. 2(c)–(f) when 5 images of each individual were randomly selected for training and the remaining images were used for testing. The dimensions of the projection matrices were set as empirical value and marked on the horizontal axis. The variables except parameter α_1 and α_2 in JSLR were randomly initialized in our experiments.

Exploration of the Performance of the Parameters: In order to explore the optimal parameters for JSLR on different data sets, we analyzed the values of the parameters α_1 (Alpha1) and α_2 (Alpha2). For the other compared methods, since in most of cases the best performance lie on the area of $[10^{-3}, 10^3]$, as introduced in the corresponding papers, we fixed their parameters on the area of $[10^{-3}, 10^3]$ and report the best results.

In this experiment, we analyze the impacts of various parameter values on the performance of JSLR and the average recognition rates of different dimensions from 5 to 200. Table II shows the best average recognition rates based on 10 times running and the corresponding dimensions as well as the standard deviations of each method with l ($l = 4, 5$) images of each individual for training while the remaining images were used for testing.

TABLE II
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF DIFFERENT METHODS ON FERET FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR ($\alpha1$=0)	JSLR	Deep-NN	Deep-JSLR
4	54.55 ±8.54 28*5	54.72 ±8.74 27*5	60.77 ±20.04 30*5	46.80 ±9.61 30*5	61.23 ±19.96 30*5	53.12 ±12.55 12*5	63.82 ±21.07 40*5	55.53 ±23.33 13*5	59.42 ±13.46 30*5	46.72 ±9.55 30*5	73.22 ±18.77 30*5	**74.15 ±18.34 29*5**	99.47 ±0.46 30*5	**100.00 ±0.00 26*5**
5	65.50 ±5.48 30*5	65.90 ±5.65 27*5	76.45 ±7.76 30*5	61.50 ±4.21 30*5	77.18 ±7.61 29*5	66.65 ±5.27 11*5	78.35 ±7.33 40*5	74.92 ±10.55 8*5	67.95 ±6.33 29*5	60.08 ±6.69 30*5	90.70 ±6.67 27*5	**91.35 ±6.42 29*5**	99.55 ±0.33 30*5	**100.00 ±0.00 20*5**

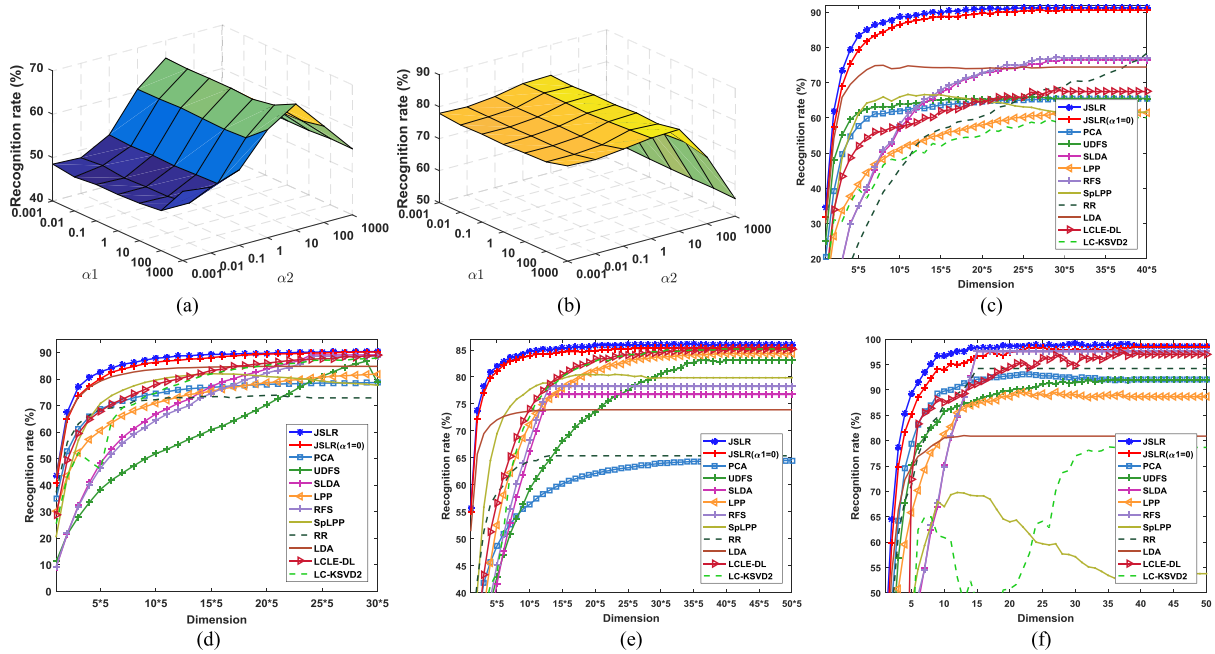


Fig. 2. The recognition rate versus the parameters α_1 and α_1 on the (a) FERET and (b) AR face database, respectively. The recognition rates (%) versus the dimensions of different methods on the (c) FERET, (d) AR, (e) CMU PIE, (f) Yale face databases, respectively.

Fig. 2(a) shows the recognition rates when the two parameters α_1 and α_2 change from 10^{-3} to 10^3. Fig. 2(c) shows the average recognition rates versus various dimensions of different methods.

It is easy to know that the optimal value of the parameter α_1 lies on the area of $[10^{-3}, 10^2]$ while the optimal value of the parameter α_2 lies on the area of $[10^{-2}, 10^3]$. In other words, JSLR is efficient and robust among these areas. By contrast, when the values of the two parameters lie on other areas, it will cause the larger decline of the recognition rates.

As it can be seen from Fig. 2(c), the recognition rates of JSLR as well as JSLR($\alpha1 = 0$) are the highest. The results shown in Table II and Fig. 2(c) indicate that JSLR and JSLR($\alpha1 = 0$) outperform PCA, SLDA, UDFS, RFS, LPP and SpLPP, RR, LDA, LCLE-DL, LC-KSVD2 in feature extraction. Besides, from Table II, we can easily know that Deep-JSLR outperforms Deep-NN.

2) Experiments on AR Face Database: The AR face database [72] contains the pictures of 120 individuals (each individual has 20 images). The face portion of each image was manually cropped (because of missing eye coordinates) and then normalized to 50×40 pixels. The sample images of one person are shown in Fig. 1(d).

In this experiment, we randomly selected l ($l = 4, 5, 6$) images of each individual for training, and the rest of the images in the data set were used for testing. From Fig. 2(b), we can know that the optimal values of parameter α_1 and α_2 were both $[10^{-3}, 10^2]$. Thus, we used this area for JSLR to obtain the comparison results. Table III listed the performance of different methods. Fig. 2(d) showed the average testing recognition rates. It is obvious that JSLR or Deep-JSLR outperforms the other methods.

3) Experiments on CMU PIE Database: The CMU PIE face database [73] contains 68 individuals with 41,368 face images as a whole. We selected a subset (C29) containing 1632 images from 68 individuals (each providing 24 images). All of these face images were automatically aligned based one-eye coordinates and cropped to 32×32 pixels. Fig. 1(e) shows the sample images from this database.

In this experiment, l ($l = 4, 5, 6$) images of each individual were randomly selected for training, and the rest of the images in the data set were used for testing. The optimal areas of α_1 and α_2 were the same with the areas on AR database. Table IV presents the performance of different methods. Fig. 2(e) shows the average testing recognition rates and indicates that JSLR outperforms the other methods again.

TABLE III
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON AR FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR ($\alpha1$=0)	JSLR	Deep-NN	Deep-JSLR
4	76.86 ±4.56 30*5	85.66 ±8.48 29*5	87.97 ±10.95 24*5	80.40 ±9.25 30*5	87.41 ±11.41 24*5	79.03 ±7.76 12*5	68.29 ±5.89 21*5	80.06 ±11.78 23*5	**89.89** **±7.76** **29*5**	84.97 ±6.96 30*5	88.82 ±10.79 30*5	89.08 ±10.64 28*5	83.43 ±12.78 29*5	**90.23** **±9.77** **28*5**
5	78.67 ±5.41 30*5	86.73 ±8.89 29*5	89.07 ±11.27 24*5	81.91 ±9.78 30*5	88.58 ±11.67 24*5	82.01 ±8.34 15*5	73.91 ±7.99 21*5	84.78 ±11.77 21*5	88.87 ±8.28 30*5	87.88 ±7.44 30*5	90.19 ±10.63 30*5	**90.33** **±10.60** **29*5**	89.26 ±9.36 29*5	**94.70** **±7.02** **29*5**
6	80.47 ±4.97 30*5	90.35 ±5.85 28*5	94.24 ±8.11 24*5	86.11 ±7.89 30*5	93.85 ±8.51 24*5	86.76 ±6.15 16*5	79.41 ±6.02 21*5	91.82 ±9.38 22*5	91.50 ±5.92 30*5	92.26 ±5.85 30*5	95.05 ±8.02 30*5	**95.23** **±8.04** **28*5**	93.27 ±3.07 29*5	**97.52** **±1.84** **30*5**
15	92.08 ±18.01 30*5	92.08 ±3.44 28*5	99.22 ±20.38 24*5	98.80 ±0.71 30*5	99.22 ±0.41 24*5	95.03 ±2.54 16*5	96.30 ±1.57 21*5	99.12 ±0.68 21*5	98.02 ±1.16 30*5	97.10 ±1.26 30*5	99.55 ±0.32 27*5	**99.60** **±0.32** **19*5**	97.68 ±2.91 30*5	**99.73** **±0.74** **30*5**
16	92.35 ±18.07 30*5	92.35 ±3.48 28*5	99.27 ±20.41 24*5	98.92 ±0.78 30*5	99.27 ±0.45 24*5	95.19 ±1.72 16*5	96.69 ±1.90 21*5	99.25 ±0.53 21*5	98.08 ±1.17 30*5	97.50 ±1.01 30*5	99.63 ±0.28 18*5	**99.77** **±0.25** **19*5**	98.67 ±1.38 30*5	**99.98** **±0.07** **22*5**

TABLE IV
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON CMU PIE FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR ($\alpha1$=0)	JSLR	Deep-NN	Deep-JSLR
4	57.11 ±12.66 40*5	77.62 ±8.85 37*5	69.25 ±13.46 13*5	80.01 ±9.16 40*5	71.55 ±12.92 13*5	76.21 ±8.32 20*5	56.33 ±10.50 13*5	36.31 ±8.54 8*5	**81.43** **±6.78** **39*5**	81.05 ±8.15 36*5	80.60 ±11.12 34*5	**81.43** **±10.4** **30*5**	96.32 ±1.10 38*5	**97.54** **±0.92** **37*5**
5	64.52 ±12.27 40*5	83.13 ±4.65 36*5	76.76 ±11.02 13*5	84.35 ±4.85 40*5	78.25 ±10.19 13*5	80.48 ±6.86 19*5	65.39 ±9.26 13*5	73.92 ±10.23 13*5	85.11 ±4.32 37*5	85.22 ±3.64 39*5	85.44 ±5.19 33*5	**86.08** **±4.54** **34*5**	96.73 ±1.25 39*5	**97.96** **±1.17** **34*5**
6	68.84 ±10.20 40*5	85.63 ±4.74 38*5	78.45 ±9.59 13*5	85.62 ±5.04 37*5	79.77 ±7.85 13*5	83.51 ±5.44 19*5	71.36 ±10.07 13*5	78.19 ±10.30 13*5	85.50 ±3.56 40*5	85.28 ±3.80 35*5	85.16 ±5.15 38*5	**85.85** **±4.54** **33*5**	96.99 ±1.10 39*5	**98.14** **±1.07** **38*5**
19	94.24 ±5.91 40*5	94.24 ±5.91 38*5	92.29 ±8.22 13*5	92.09 ±8.03 40*5	92.09 ±8.34 13*5	91.88 ±8.39 19*5	**94.41** **±5.46** **13*5**	91.06 ±9.55 13*5	87.00 ±9.74 40*5	90.09 ±7.52 40*5	92.91 ±7.65 40*5	93.12 ±7.55 40*5	99.32 ±0.77 33*5	**100.00** **±0.00** **14*5**
20	93.16 ±7.17 40*5	93.16 ±7.17 38*5	90.29 ±10.25 13*5	90.18 ±9.95 40*5	90.11 ±10.28 13*5	90.22 ±9.95 19*5	**93.68** **±6.56** **13*5**	89.04 ±11.73 13*5	85.57 ±11.96 40*5	88.54 ±9.17 40*5	91.29 ±9.44 23*5	91.47 ±9.28 30*5	99.34 ±0.76 33*5	**100.00** **±0.00** **14*5**

TABLE V
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON YALE FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR ($\alpha1$=0)	JSLR	Deep-NN	Deep-JSLR
4	91.24 ±3.46 21	90.48 ±4.02 39	96.95 ±2.49 15	85.05 ±5.06 37	96.48 ±2.58 15	68.10 ±9.27 11	91.62 ±3.24 15	66.76 ±10.23 10	97.90 ±1.41 40	78.86 ±41.68 33	97.71 ±1.61 31	**98.95** **±1.05** **27**	99.71 ±0.40 40	**100.00** **±0.00** **12**
5	93.11 ±3.60 22	92.11 ±4.57 36	97.56 ±2.39 15	89.56 ±3.58 22	97.56 ±2.21 15	69.78 ±5.49 12	94.22 ±3.57 15	81.00 ±5.50 13	97.44 ±1.80 37	78.78 ±41.78 35	98.44 ±1.57 39	**99.22** **±1.20** **30**	99.56 ±0.47 40	**100.00** **±0.00** **11**

4) Experiments on Yale Database: The Yale face database [43] contains 165 grayscale images of 15 individuals. Each image was manually cropped (because of no eye coordinates provided) and resized to 50 × 40 pixels. Fig. 1(b) shows the sample images from this database.

In this experiment, l ($l = 4, 5$) images of each individual were randomly selected for training, and the rest of the images in the data set were used for testing. The values of α_1 and α_2 were both from 10^{-3} to 10^2. The performances of the different methods are shown in Table V. Fig. 2(f) shows the average recognition rates. It clearly indicates that JSLR and Deep-JSLR can obtain the best performance when the traditional image features and deep features are used as input.

B. Experiments on Large-Scale Database

In this section, two different databases are used to evaluate the performance of the proposed method based on large-scale data learning.

1) Experiments on USPS Database: The United States Postal Service(USPS) database [55] consists of 1,100 of each handwritten digit (0-9). The images in this database are resized to 16 × 16 pixels. The performance of JSLR on large-sample data set was evaluated on this database. Fig. 1(c) shows the sample images from this database.

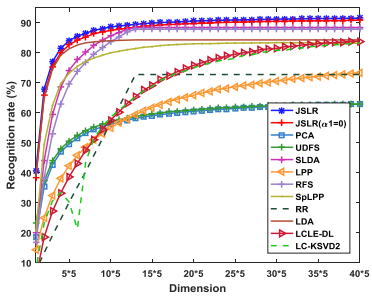
In this experiment, l ($l = 400, 500, 600, 5, 10$) images of each class were randomly selected for training, and the rest of the images in the data set were used for testing. The parameter α_1 and α_2 were both from 10^{-3} to 10^3. Table VI shows the performance of the different methods. From the result, we can know that JSLR or Deep-JSLR can achieve better performance than other compared methods on this database. Particularly, when only $5/1100$ images of each class are used for training, the proposed method can obtain higher recognition rate than not only the compared methods but also the Deep-NN.

2) Experiments on PIE67 × 170 Database: The PIE67 × 170 database is a subset of the CMU PIE face database [73].

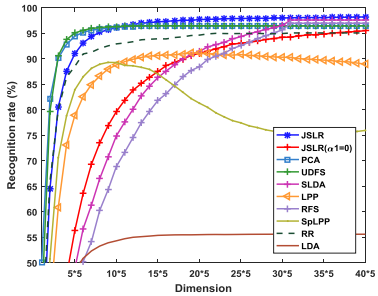
TABLE VI
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON USPS FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR ($\alpha1=0$)	JSLR	Deep-NN	Deep-JSLR
400	90.55 ±0.13 7*5	91.17 ±0.12 10*5	91.37 ±0.35 2*5	89.41 ±0.30 6*5	90.42 ±0.40 2*5	91.02 ±0.19 6*5	90.34 ±0.17 2*5	86.78 ±0.00 1*5	87.21 ±0.56 4*5	90.78 ±0.19 30*5	92.02 ±0.22 30*5	**92.35 ±0.18 20*5**	91.44 ±0.28 1*5	**95.74 ±0.27 15*5**
500	91.02 ±0.15 7*5	91.75 ±0.18 9*5	91.91 ±0.23 2*5	90.14 ±0.21 7*5	90.81 ±0.31 2*5	91.90 ±0.16 6*5	90.87 ±0.23 2*5	87.23 ±0.00 1*5	87.22 ±0.59 4*5	91.60 ±0.27 30*5	92.52 ±0.18 30*5	**92.77 ±0.14 28*5**	92.18 ±0.17 1*5	**96.00 ±0.26 21*5**
600	91.49 ±0.17 7*5	92.20 ±0.24 8*5	92.28 ±0.35 2*5	90.58 ±0.17 6*5	91.40 ±0.45 2*5	92.51 ±0.13 6*5	91.45 ±0.25 2*5	87.49 ±0.00 1*5	87.15 ±0.71 4*5	92.15 ±0.33 30*5	92.82 ±0.28 28*5	**93.08 ±0.31 25*5**	92.67 ±0.26 1*5	**96.09 ±0.21 20*5**
5	61.43 ±2.08 7*5	61.43 ±2.08 8*5	56.83 ±2.30 2*5	49.41 ±3.10 6*5	56.23 ±2.43 2*5	15.76 ±2.28 6*5	57.19 ±2.24 2*5	61.45 ±3.53 1*5	62.00 ±2.40 4*5	62.04 ±2.34 30*5	59.51 ±2.24 8*5	**67.83 ±2.39 10*5**	56.96 ±2.28 100*5	**66.96 ±1.72 10*5**
10	69.86 ±2.18 7*5	69.86 ±2.18 8*5	55.78 ±2.15 2*5	49.49 ±2.64 6*5	54.73 ±2.00 2*5	14.14 ±2.30 6*5	55.64 ±2.05 2*5	61.02 ±2.16 1*5	68.87 ±1.75 4*5	67.81 ±1.91 30*5	58.67 ±1.87 12*5	**76.22 ±2.64 20*5**	65.66 ±1.68 100*5	**78.15 ±1.64 20*5**

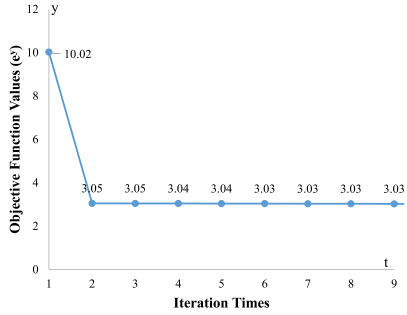
(a)



(b)



(c)



(d)

Fig. 3. (a) Sample images on the LFW database. The recognition rates (%) versus the dimensions of different methods on the (b) PIE67 $\times$ 170, (c) LFW databases, respectively. (d) An example of the convergence curve of JSLR on Yale database.

TABLE VII
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON PIE67 $\times$ 170 FACE DATABASE

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KS VD2	JSLR ($\alpha1=0$)	JSLR	Deep-NN	Deep-JSLR
10	45.75 ±3.71 40*5	45.75 ±3.72 40*5	78.63 ±3.55 13*5	59.72 ±3.33 40*5	76.14 ±4.03 13*5	69.71 ±3.69 21*5	58.85 ±3.01 13*5	72.74 ±2.70 13*5	82.72 ±2.86 40*5	78.78 ±2.11 40*5	81.47 ±2.93 40*5	**83.64 ±3.17 40*5**	90.90 ±1.21 40*5	**95.62 ±1.31 40*5**
20	62.89 ±3.05 40*5	62.89 ±3.06 40*5	88.41 ±2.09 13*5	73.32 ±3.12 40*5	87.84 ±2.11 13*5	83.38 ±3.09 30*5	72.75 ±2.66 13*5	84.29 ±2.11 13*5	86.84 ±1.42 40*5	86.19 ±1.23 40*5	90.84 ±1.42 40*5	**91.55 ±1.37 40*5**	93.98 ±1.07 40*5	**97.70 ±0.73 40*5**
30	69.85 ±3.57 40*5	69.85 ±3.58 40*5	91.46 ±2.22 13*5	77.99 ±2.76 40*5	91.05 ±2.30 13*5	87.42 ±2.72 36*5	83.74 ±3.13 13*5	87.98 ±2.78 13*5	87.81 ±1.61 40*5	87.96 ±1.27 39*5	93.38 ±1.82 40*5	**93.70 ±1.78 40*5**	95.36 ±1.00 40*5	**98.38 ±0.67 40*5**

There are total 11,390 images from 67 individuals and each individual has 170 images on this database. The experiment on this database is conducted to evaluate the performance of JSLR as well as the compared methods on the occasion when there are various facial expression, lighting condition and angle on the face images.

In this experiments, l ($l = 10, 20, 30$) images of each individual are randomly selected for training and the remaining are used for testing. The recognition rates of all methods are shown in Fig. 3(b) and Table VII. From Fig. 3(b), JSLR as well as JSLR($\alpha1 = 0$) obtain higher recognition rate than other methods, which indicates that the proposed method is superior to other methods even without the $L_{2,1}$-norm regularization term (this can also be verified by Fig. 2(c) and Table II).

C. Experiments Based on Deep Learning

In this section, experiments on three database (AR, the standard subsets of the FERET and the LFW databases [74]) were

TABLE VIII
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON AR FACE DATABASE BASED ON DEEP LEARNING

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	Deep-NN	JSLR (α1=0)	JSLR
4	87.12 ±12.26 30*5	87.12 ±12.33 30*5	87.53 ±9.26 24*5	83.46 ±10.96 22*5	88.80 ±10.73 24*5	87.12 ±11.90 30*5	74.53 ±9.91 24*5	82.37 ±8.81 23*5	90.01 ±6.50 30*5	89.89 ±6.89 30*5	83.43 ±12.78 29*5	13.49 ±7.47 30*5	**90.23 ±9.77 28*5**
5	91.84 ±9.36 30*5	91.84 ±9.43 30*5	91.16 ±7.84 24*5	89.82 ±8.20 20*5	93.17 ±7.79 24*5	91.84 ±8.83 30*5	80.74 ±7.82 24*5	89.63 ±7.56 21*5	94.67 ±4.61 30*5	94.10 ±4.60 30*5	89.26 ±9.36 29*5	76.91 ±5.51 30*5	**94.70 ±7.02 29*5**
6	95.40 ±2.77 29*5	95.33 ±2.75 30*5	94.39 ±3.84 24*5	93.99 ±3.14 16*5	96.16 ±2.74 24*5	95.33 ±2.82 30*5	84.64 ±6.58 24*5	93.85 ±3.65 18*5	96.81 ±2.25 30*5	96.67 ±1.78 30*5	93.27 ±3.07 29*5	92.93 ±3.58 30*5	**97.52 ±1.84 30*5**

TABLE IX
THE MAXIMAL RECOGNITION RATE OF ALL METHODS ON THE Fb, Fc, Dup1, Dup2 FACE DATABASE BASED ON DEEP LEARNING

Algorithm	Fb (dim=216, 512)	Fc (dim=216, 512)	Dup1(dim=216, 512)	Dup2(dim=216, 512)
PCA	99.41, 99.41	99.48, 99.48	98.20, 98.20	98.29, 98.29
UDFS	80.75, 99.41	65.46, 99.48	43.07, 98.20	45.30, 98.29
SLDA	98.49, 98.91	**100.00**, **100.00**	84.63, 88.50	89.74, 92.31
LPP	98.58, 99.16	90.03, 92.66	86.75, 91.45	86.75, 91.45
RFS	99.58, 99.58	**100.00**, **100.00**	97.51, 98.48	98.72, 99.15
SpLPP	99.41, 99.50	98.97, 99.48	97.78, 98.06	98.72, 99.15
RR	98.74, 98.91	98.45, 98.45	94.46, 94.46	96.58, 96.58
LDA	99.41, 99.41	99.48, 99.48	98.20, 98.20	98.29, 98.29
LCLE-DL*	-	-	-	-
LC-KSVD2*	-	-	-	-
Deep-NN	99.41, 99.50	99.48, 99.48	98.48, 98.48	98.72, 98.72
JSLR (α1=0)	98.58, 99.41	99.48, **100.00**	86.01, 93.77	88.03, 94.44
JSLR	**99.67**, **99.67**	**100.00**, **100.00**	**98.75**, **98.89**	**99.57**, **99.57**

*Since there is only one sample in each class on the training set of Fa, the dictionary learning methods are not suitable to use in this case and the performance is too poor to be presented.

conducted based on deep learning. In the experiments, the Caffe deep learning framework [75] was used as the pre-processing to learn the deep features from the sample images. After the deep features were obtained, we further used the subspace learning methods (i.e. PCA, UDFS, SLDA, LPP, RFS, SpLpp, RR, LDA and the proposed JSLR) and dictionary learning methods (i.e. LCLE-DL and LC-KSVD2) to perform further feature extraction and then the nearest neighbor classifier was used for classification.

For AR and the standard subsets of the FERET databases, the dimension of extracted features based on the deep convolutional neural network (CNN) is 512 while that of the LFW database is 1024. For the standard FERET dataset, the Fa subset was used as the gallery set while the Fb, Fc, Dup1 and Dup2 were used as the probe sets. The LFW database contains images from 5,749 subjects in the uncontrolled environment, which makes it as a challenging recognition task. 158 subjects with total 4,324 images are selected from LFW-a subset and used in our experiment as the LFW-a subset is the aligned version of LFW database. The sample images on this database are shown in Fig. 3(a).

The experimental results on AR databases is listed in Table VIII. For the standard subsets of the FERET database, the best recognition rates corresponding different methods are shown in Table IX, in which the accuracy on both the original dimensions (i.e. 512) and 216 dimensions (i.e. half of 512) are listed. The results in Table VIII and IX clearly show that the performance of the proposed JSLR is better than that of Deep-NN. This indicates that JSLR is able to extract discriminative information from deep features and further achieve higher recognition rate. The experimental results on LFW database are shown in Fig. 3(c) and Table X. In Fig. 3(c), the reason why no curves of LCLE-DL and LC-KSVD2 present is that no PCA is used as pre-processing to reduce the dimension of the input data to a specific value, the dictionary learning methods (i.e. LCLE-DL and LC-KSVD2) can only obtain recognition rate corresponding to the original dimension (i.e. 1024). Therefore, we cannot obtain the recognition rate curve versus the dimension variations for the two methods. From Fig. 3(c) and Table X, we can know that JSLR outperforms other compared methods again.

The convergence curves of the proposed JSLR on all databases are shown in Fig. 3(d) and Fig. 4. In these figures, the objective function value corresponding to each iteration is denoted as e^y where y is the values marked on the vertical coordinate. The convergence curves on all databases indicate that the proposed method can converge after several iterations.

TABLE X
THE PERFORMANCE (RECOGNITION RATE, STANDARD DEVIATION AND DIMENSION) OF ALL METHODS ON THE LFW FACE DATABASE BASED ON DEEP LEARNING

Training samples	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	Deep-NN	JSLR ($\alpha1=0$)	JSLR
3	96.46 ±0.18 15*5	96.50 ±0.18 16*5	97.59 ±0.23 31*5	91.13 ±0.67 19*5	96.96 ±0.19 31*5	89.29 ±1.02 9*5	95.06 ±0.43 31*5	55.60 ±3.36 29*5	95.39 ±0.57 40*5	93.94 ±0.96 40*5	96.45 ±0.16 197*5	95.51 ±0.39 40*5	**98.19 ±0.20 40*5**
5	96.79 ±0.23 14*5	96.80 ±0.22 13*5	98.58 ±0.22 31*5	97.82 ±0.20 21*5	98.39 ±0.26 31*5	96.67 ±0.24 9*5	96.48 ±0.32 31*5	69.64 ±2.22 25*5	97.51 ±0.32 40*5	95.48 ±0.30 40*5	96.78 ±0.20 198*5	98.39 ±0.14 40*5	**98.71 ±0.20 39*5**
7	97.06 ±0.11 20*5	97.12 ±0.15 13*5	**99.01 ±0.14 31*5**	98.85 ±0.17 25*5	98.96 ±0.10 31*5	98.10 ±0.20 9*5	97.07 ±0.32 25*5	70.47 ±2.71 26*5	97.95 ±0.35 40*5	97.16 ±0.38 40*5	97.08 ±0.12 197*5	98.91 ±0.14 39*5	98.92 ±0.13 39*5

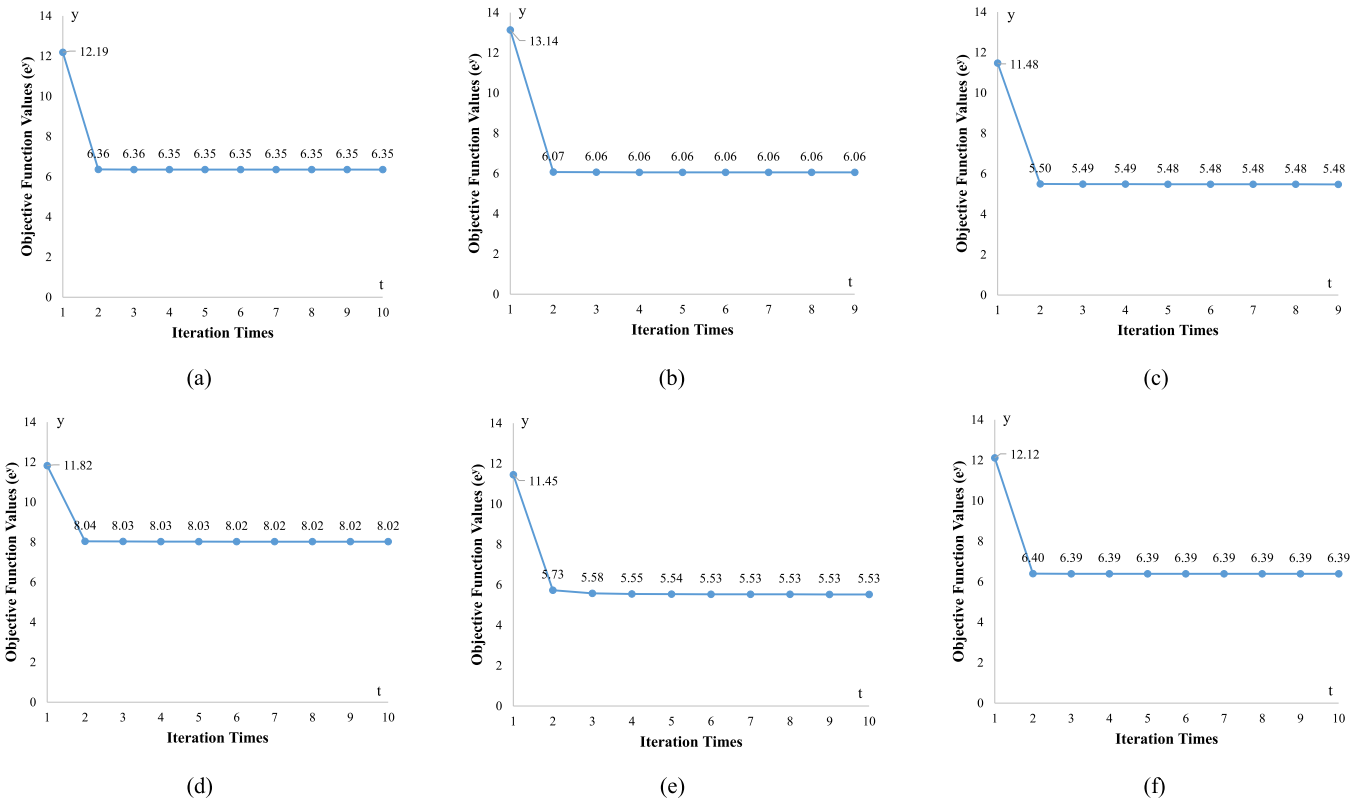


Fig. 4. Examples of the convergence curves of JSLR on (a) FERET, (b) AR, (c) CMU PIE, (d) USPS, (e) PIE67 × 170 and (f) LFW database, respectively.

D. Experimental Results and Discussions

The comparison among the proposed JSLR, classical PCA, RR, LDA, SLDA, LPP, SpLPP, $L_{2,1}$-norm based methods (UDFS, RFS) and dictionary learning methods (LCLE-DL, LC-KSVD2) has been presented using recognition rates on these databases: FERET, AR, CMU PIE, Yale and USPS. From the results, we reveal the following interesting points:

1) In all experiments, including the face databases (FERET, AR, CMU PIE, Yale) and non-facial database (USPS), JSLR consistently achieves higher recognition rates than other methods. These results are in line with the theoretical analysis of JSLR that it obtains discriminative information with joint sparsity and takes local geometric structure of dataset into consideration to perform feature selection and extraction.
2) JSLR is able to encode more discriminating information in the low-dimensional face subspace since the local geometric structure is considered to be more effective than the global structure for feature extraction and feature selection in some cases. The reason why JSLR outperforms the local structure learning method such as LPP and SpLpp is that JSLR utilizes -norm regularization for feature selection and feature extraction to obtain the discriminative information for face recognition.
3) As it can be seen from the Fig. 2(d), (e) and (f), the traditional regression methods and/or their extensions can obtain only c projections for feature extraction and classification, which is not enough to achieve high recognition rates. Note that the number of classes in CMU PIE and Yale is 68 and 15, respectively. Therefore the numbers of projections obtained by LDA are 67 and 14, and the numbers of projections obtained by RR are 68 and 15 on CMU PIE and Yale databases, respectively. Fig. 2(e) and (f) show that the recognition rates of RR and LDA achieve their top

TABLE XI
THE COMPUTATIONAL COST (UNIT: S) OF DIFFERENT METHODS

Data set (l)	PCA	UDFS	SLDA	LPP	RFS	SpLPP	RR	LDA	LCLE-DL	LC-KSVD2	JSLR (α1=0)	JSLR
FERET ($l=4$)	0.0720	0.0390	14.7585	**0.0201**	1.7227	23.8032	0.1456	0.0642	1.7157	21.1985	0.0334	0.0301
AR ($l=4$)	0.0305	0.0566	7.2524	**0.0097**	0.5550	39.4835	0.1484	0.0607	1.1984	11.4715	0.0242	0.0237
CMU PIE ($l=4$)	**0.0087**	0.0767	7.6509	0.0375	0.1741	457.7060	0.0383	0.0483	0.4201	6.1862	0.0341	0.0906
Yale ($l=4$)	0.0019	0.0053	0.1162	**0.0018**	0.0161	0.4617	0.1251	0.0039	0.0505	1.1682	0.0028	0.0035
USPS ($l=400$)	0.0105	0.0433	4.0517	0.2033	131.6732	240.4646	**0.0091**	0.0376	6.7102	347.7577	0.1423	0.1462
PIE67 × 170 ($l=80$)	0.1224	0.0738	168.2569	**0.0289**	285.8122	1.3378 e+03	0.1190	0.2580	7.7944	630.7555	0.2123	0.2864
LFW ($l=4$)	0.0229	0.0661	48.6211	**0.0157**	1.4418	255.8663	0.0319	0.0984	1.1866	18.8623	0.1565	0.2012
Fa	**0.0234**	0.4945	1.3758 e+04	0.0611	10.0917	3.8150 e+03	0.0236	0.0371	18.1813	1.5484 e+03	0.3679	0.2854

recognition rates using all the projections (we copy the final recognition rate to full fill all the dimensions listed on the horizontal axis). Thus the recognition rates no more increase after the number of dimension reaches 67 and 14 for LDA and 68 and 15 for RR on CMU PIE and Yale face databases, respectively. These figures show that the lack of enough projection of LDA and RR limits their performances. However, JSLR can break through this limitation and obtain more projections. This is the potential reason for JSLR to achieve higher recognition rates. In addition, the experimental result on USPS database with more than 4000 samples (as shown in Table VI) indicates the robustness and effectiveness of JSLR in dealing with large-sample size problem.

4) The $L_{2,1}$-norm based methods such as JSLR, RFS and UDFS are robust to outliers in dataset and they guarantee the joint sparsity. However, JSLR obtains the best recognition rates when there are variations on lighting condition and face expressions. This indicates that JSLR is more robust than RFS and UDFS in feature extraction and selection when there exists variations on lighting condition and face expressions. In addition, the experimental results based on deep learning techniques presented in Table VIII and Table IX indicate the good performance of the proposed JSLR.
5) Experimental results indicate that the proposed JSLR performs better than the dictionary learning methods (LCLE-DL and LC-KSVD2). The reason is that JSLR guarantees the joint sparsity for discriminant feature selection or extraction in different cases. The Comparison between Deep-NN and Deep-JSLR shows that JSLR can further enhance the discriminative power of the deeply learned features based on CNNs in face recognition and character recognition tasks.
6) Table XI presents the computational time (unit: second) of each method on different data sets. From Table XI, we can know that the proposed JSLR based on $L_{2,1}$-norm minimization is fast convergent and the computational cost is much less than the L_1-norm based methods (i.e. SLDA, SpLPP). The essential reason is that both SLDA and SpLPP use the least-angle regression method to compute the sparse solution and the iteration times are more than the proposed method. Moreover, the projections of SLDA and SpLPP are computed one by one while JSLR can simultaneously compute a set of jointly sparse projections. Thus, the proposed JSLR is efficient and effective for computer vision and pattern recognition.
7) From Tables III and IV, we can see that when more training samples are used, the recognition rates of all methods are higher than that when less training samples are used. However, it does not mean that more traning samples can definitely help to obtain higher recognition rate. As shown in Table IV, when 20/24 training samples are used, the recognition rates of all methods become lower compared to the case when 19/24 training samples are used. The potential reason for this phenomenon is that too many training samples may lead to overfitting and thus all methods obtain poorer performance in the testing stage.

VII. CONCLUSION

Motivated by previous works that $L_{2,1}$-norm regularization is able to obtain joint sparsity, and the local geometric information can enhance feature selection capability, in this paper, we propose a novel method called JSLR for feature extraction and selection. With $L_{2,1}$-norm regularization and locality preserving property, JSLR can obtain any number of discriminative projections for feature selection, which addresses the drawback in LDA and ridge regression. Theoretical analyses show the close relationship of JSLR and ridge regression, which also guarantees the effectiveness of JSLR in feature extraction and selection. In order to obtain the optimal solution of JSLR, we propose an iterative algorithm which is proved to be convergent. In addition, the computational complexity of the algorithm is also presented. The performance of JSLR on several well-known face databases shows that it outperforms the classical principle component analysis methods, traditional sparse learning methods and recently proposed $L_{2,1}$-norm regularization methods.

APPENDIX
PROOF OF THEOREM 1

From Eq. (10), we have

$$\left\|\boldsymbol{Y}-\boldsymbol{X}^T\boldsymbol{B}\boldsymbol{A}^T\right\|_F^2 = Tr\left(\boldsymbol{Y}^T\boldsymbol{Y}-2\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}+\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{B}\right).$$

By setting the derivatives of the above problem with respect to $\boldsymbol{B}$ equaling to 0, we have

$$\boldsymbol{B}=(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}.$$

Let $\boldsymbol{B}^*$ represents the optimal solution of Eq. (10), then

$$\boldsymbol{B}^*=(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\mathbf{X}\mathbf{Y}\mathbf{A}.$$

Since $\boldsymbol{P}^0=(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y}$, we have $\boldsymbol{B}^*=\boldsymbol{P}^0\mathbf{A}$. As matrix $\boldsymbol{A}$ is a rotation matrix, then the subspace spanned by $\boldsymbol{B}^*$ in Eq. (10) is the same as that spanned by $\boldsymbol{P}^0$ in Eq. (1), namely, $\text{span}(\boldsymbol{B}^*)=\text{span}(\boldsymbol{P}^0)$.

Suppose $\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}$, $\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}$ and $\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}$, by the SVD of $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, we have $\boldsymbol{B}^*=\boldsymbol{U}\frac{1}{\boldsymbol{D}}\boldsymbol{V}^T\boldsymbol{Y}\boldsymbol{A}$. Since the optimal solution of Eq. (3) is $\boldsymbol{P}^*=\boldsymbol{U}\frac{\boldsymbol{D}}{\boldsymbol{D}^2+\alpha\boldsymbol{I}}\boldsymbol{V}^T\boldsymbol{Y}$, then we can find that the subspaces spanned by $\boldsymbol{B}^*$ and $\boldsymbol{P}^0$ have the same base matrix $\boldsymbol{U}$ and the only difference is that there is a weighted rotation matrix, which does not affect the spanned subspace. Thus, we can say $\text{span}(\boldsymbol{B}^*)=\text{span}(\boldsymbol{P}^*)$.

If $\alpha\to 0$, we have

$$\boldsymbol{P}^*\boldsymbol{A}=\boldsymbol{U}\frac{\boldsymbol{D}}{\boldsymbol{D}^2+\alpha\boldsymbol{I}}\boldsymbol{V}^T\boldsymbol{Y}\boldsymbol{A}\to\boldsymbol{U}\frac{1}{\boldsymbol{D}}\boldsymbol{V}^T\boldsymbol{Y}\boldsymbol{A}=\boldsymbol{B}^*.$$

Thus, for any two pattern vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, since $\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}$, the distance of the two points obtained by using the two subspaces (i.e. $\boldsymbol{B}^*$ and $\boldsymbol{P}^*$) for feature extraction is invariant. That is,

$$\left\|(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\boldsymbol{B}^*\right\|_2=\left\|(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\boldsymbol{P}^*\boldsymbol{A}\right\|_2=\left\|(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\boldsymbol{P}^*\right\|_2,$$

which indicates that the performance of using the two metric matrices derived by $\boldsymbol{B}^*$ and $\boldsymbol{P}^*$ for classification will be the same.

For Eq. (19), if $\alpha_1\to 0$ and $\alpha_2\to 0$, we have

$$\bar{\boldsymbol{B}}=\left(\boldsymbol{X}\boldsymbol{X}^T+\alpha_1\boldsymbol{D}_{\boldsymbol{B}}+\alpha_2\boldsymbol{X}\left(\overline{\boldsymbol{D}}-\overline{\boldsymbol{W}}\right)\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}\to(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{Y}\boldsymbol{A}=\boldsymbol{B}^*,$$

namely, $\text{span}(\boldsymbol{B}^*)\to\text{span}(\bar{\boldsymbol{B}})$.

Since $\text{span}(\boldsymbol{B}^*)=\text{span}(\boldsymbol{P}^0)$, $\text{span}(\boldsymbol{B}^*)=\text{span}(\boldsymbol{P}^*)$, then $\text{span}(\bar{\boldsymbol{B}})\to\text{span}(\boldsymbol{P}^0)$, $\text{span}(\bar{\boldsymbol{B}})\to\text{span}(\boldsymbol{P}^*)$.

REFERENCES

[1] G. V. Lashkia and L. Anthony, "Relevant, irredundant feature selection and noisy example elimination," *IEEE Trans. Syst., Man, Cybern., Part B Cybern.*, vol. 34, no. 2, pp. 888–897, Apr. 2004.

[2] T. W. S. Chow, P. Wang, and E. W. M. Ma, "A new feature selection scheme using a data distribution factor for unsupervised nominal data," *IEEE Trans. Syst., Man, Cybern., Part B Cybern.*, vol. 38, no. 2, pp. 499–509, Apr. 2008.

[3] J. Qian, J. Yang, and Y. Xu, "Local structure-based image decomposition for feature extraction with applications to face recognition," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3591–3603, Sep. 2013.

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

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[6] W. Zheng and X. Tang, "A robust algorithm for generalized orthonormal discriminant vectors," in *Proc. IEEE 18th Int. Conf. Pattern Recognit.*, 2006, vol. 2, pp. 784–787.

[7] G. Dai and Y. Qian, "A gabor direct fractional-step LDA algorithm for face recognition," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2004, vol. 1, pp. 61–64.

[8] I. Dagher, "Incremental PCA-LDA algorithm," in *Proc. IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl.*, 2010, pp. 97–101.

[9] R. Raghavendra, B. Dorizzi, A. Rao, and G. H. Kumar, "Designing efficient fusion schemes for multimodal biometric systems using face and palmprint," *Pattern Recognit.*, vol. 44, no. 5, pp. 1076–1088, 2011.

[10] M. Eskandari and Ö. Toygar, "Selection of optimized features and weights on face-Iris fusion using distance images," *Comput. Vision Image Understanding*, vol. 137, pp. 63–75, 2015.

[11] F. Zhang, J. Yang, J. Qian, and Y. Xu, "Nuclear norm-based 2-DPCA for extracting features from images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2247–2260, Oct. 2015.

[12] W. Wang, Y. Yan, F. Nie, S. Yan, and N. Sebe, "Flexible manifold learning with optimal graph for image and video representation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2664–2675, Jun. 2018.

[13] F. Nie, S. Yang, R. Zhang, and X. Li, "A general framework for auto-weighted feature selection via global redundancy minimization," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2428–2438, Dec. 2018.

[14] H. Zhang, J. Yang, J. Xie, J. Qian, and B. Zhang, "Weighted sparse coding regularized nonconvex matrix regression for robust face recognition," *Inf. Sci.*, vol. 394, pp. 1–17, 2017.

[15] X. Ning, W. Li, B. Tang, and H. He, "BULDP: Biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2575–2586, Feb. 2018.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. Roy. Stat. Soc.: Ser. B (Stat. Method.)*, vol. 73, no. 3, pp. 273–282, 2011.

[17] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc.: Ser. B (Stat. Method.)*, vol. 67, no. 2, pp. 301–320, 2005.

[18] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the lasso," *J. Comput. Graph. Stat.*, vol. 12, no. 3, pp. 531–547, 2003.

[19] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 41–48.

[20] J. Feng, L. Jiao, F. Liu, T. Sun, and X. Zhang, "Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images," *Pattern Recognit.*, vol. 51, pp. 295–309, 2016.

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

[22] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[23] Z. Qiao, L. Zhou, and J. Z. Huang, "Sparse linear discriminant analysis with applications to high dimensional low sample size data," *Int. J. Appl. Math.*, vol. 39, no. 1, 2009.

[24] Z. Zeng, X. Wang, J. Zhang, and Q. Wu, "Semi-supervised feature selection based on local discriminative information," *Neurocomputing*, vol. 173, pp. 102–109, 2016.

[25] C. Shi, Q. Ruan, G. An, and R. Zhao, "Hessian semi-supervised sparse feature selection based on $l_{2,1}/_2$-matrix norm," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 16–28, Jan. 2014.

[26] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2049–2058, Nov. 2015.

[27] A. Majumdar and R. K. Ward, "Classification via group sparsity promoting regularization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2009, pp. 861–864.

[28] A. Majumdar and R. K. Ward, "Robust classifiers for data reduced via random projections," *IEEE Trans. Syst., Man, Cybern., Part B Cybern.*, vol. 40, no. 5, pp. 1359–1371, Oct. 2010.

[29] A. Majumdar and R. K. Ward, "Fast group sparse classification," *Can. J. Elect. Comput. Eng.*, vol. 34, no. 4, pp. 136–144, 2009.

[30] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int. Conf. Comput. Vision Volume 1*, 2005, vol. 2, pp. 1208–1213.

[31] D. Cai *et al.*, "Isometric projection," in *Proc. Assoc. Advancement Artif. Intell.*, 2007, pp. 528–533.

[32] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Advances Neural Inf. Process. Syst.*, 2004, pp. 153–160.

[33] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[34] D. Cai, X. He, and J. Han, "Spectral regression: A unified approach for sparse subspace learning," in *Proc. IEEE Int. Conf. Data Mining*, 2007, pp. 73–82.

[35] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.

[36] S. Liao *et al.*, "Discriminant analysis via joint euler transform and $\ell_{2,1}$ -norm," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5668–5682, Nov. 2018.

[37] J. Yang *et al.*, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 156–171, Jan. 2017.

[38] J. Qian, L. Lei, Y. Jian, F. Zhang, and Z. Lin, "Robust nuclear norm regularized regression for face recognition with occlusion," *Pattern Recognit.*, vol. 48, no. 10, pp. 3145–3159, 2015.

[39] L. Luo, J. Yang, J. Qian, Y. Tai, and G. F. Lu, "Robust image regression based on the extended matrix variate power exponential distribution of dependent noise," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 2168–2182, Sep. 2017.

[40] L. Luo, J. Yang, J. Qian, and Y. Tai, "Nuclear-l1 norm joint regression for face reconstruction and recognition with mixed noise," *Pattern Recognit.*, vol. 48, no. 12, pp. 3811–3824, 2015.

[41] J. Chen, J. Yang, L. Luo, J. Qian, and W. Xu, "Matrix variate distribution-induced sparse representation for robust image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2291–2300, Oct. 2015.

[42] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Regularized robust coding for face recognition," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1753–1766, May 2013.

[43] Y. Li, J. Si, G. Zhou, S. Huang, and S. Chen, "FREL: A stable feature selection algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1388–1402, Jul. 2015.

[44] W. Yang, C. Sun, and W. Zheng, "A regularized least square based discriminative projections for feature extraction," *Neurocomputing*, vol. 175, pp. 198–205, 2016.

[45] H. Pu and G. Gao, "Parameterless reconstructive discriminant analysis for feature extraction," *Neurocomputing*, vol. 190, pp. 50–59, 2016.

[46] Y. Yang, Z.-J. Zha, Y. Gao, X. Zhu, and T.-S. Chua, "Exploiting web images for semantic video indexing via robust sample-specific loss," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1677–1689, Oct. 2014.

[47] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Nov. 2012.

[48] J. Han, Z. Sun, and H. Hao, "l_0-norm based structural sparse least square regression for feature selection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3927–3940, 2015.

[49] J. Yang and C. Ong, "An effective feature selection method via mutual information estimation," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 42, no. 6, pp. 1550–1559, Dec. 2012.

[50] Z. Zeng, H. Zhang, R. Zhang, and C. Yin, "A novel feature selection method considering feature interaction," *Pattern Recognit.*, vol. 48, no. 8, pp. 2656–2666, 2015.

[51] Y. Hui and Y. Jian, "Sparse discriminative feature selection," *Pattern Recognit.*, vol. 48, no. 5, pp. 1827–1835, 2015.

[52] Y. Liu, F. Tang, and Z. Zeng, "Feature selection based on dependency margin," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1209–1221, Jun. 2015.

[53] 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.

[54] F. Nie, H. Huang, C. Xiao, and C. H. Q. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[55] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Berlin/Heidelberg: Springer, Jul. 2003.

[56] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "L2,1-norm regularized discriminative feature selection for unsupervised learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[57] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1083–1094, May 2015.

[58] E. V. Den Berg and M. P. Friedlander, "Theoretical and empirical results for recovery from multiple measurements," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2516–2527, May 2010.

[59] A. Majumdar and R. K. Ward, "Synthesis and analysis prior algorithms for joint-sparse recovery," in *Proc. IEEE Int. Conf. Acoust.*, 2012, pp. 3421–3424.

[60] J. He, L. Ding, L. Jiang, and L. Ma, "Kernel ridge regression classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 2263–2267.

[61] Z. Zheng *et al.*, "Regression analysis of locality preserving projections via sparse penalty," *Inf. Sci.*, vol. 303, pp. 1–14, 2015.

[62] G. Shikkenawis and S. K. Mitra, "On some variants of locality preserving projection," *Neurocomputing*, vol. 173, pp. 196–211, 2016.

[63] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[64] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Multilinear principal component analysis of tensor objects for recognition," in *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. 776–779.

[65] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 818–833.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv preprint arXiv:1409.1556*.

[67] Q. Wang, Z. Qin, F. Nie, and Y. Yuan, "Convolutional 2D LDA for nonlinear dimensionality reduction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2929–2935.

[68] Z. Jiang, L. Zhe, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 1697–1704.

[69] Z. Li, Z. Lai, Y. Xu, J. Yang, and D. Zhang, "A locality-constrained and label embedding dictionary learning algorithm for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 278–293, Feb. 2017.

[70] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 499–515.

[71] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[72] A. Martinez and R. Benavente, "The AR face database," CVC, West Lafayette, IN, USA, Tech. Rep. 24, 1998.

[73] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination and expression database of human faces," Carnegie Mellon Univ.: Pittsburgh, Pennsylvania, USA, Tech. Rep. CMU-RI-TR-OI-02, 2001.

[74] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," Tech. Rep. 07-49, UMass, 2007.

[75] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.