A Framework of Joint Graph Embedding and Sparse Regression for Dimensionality Reduction

Xiaoshuang Shi, Zhenhua Guo, *Member, IEEE*, Zhihui Lai, Yujiu Yang, Zhifeng Bao, and David Zhang, *Fellow, IEEE*

***Abstract*— Over the past few decades, a large number of algorithms have been developed for dimensionality reduction. Despite the different motivations of these algorithms, they can be interpreted by a common framework known as graph embedding. In order to explore the significant features of data, some sparse regression algorithms have been proposed based on graph embedding. However, the problem is that these algorithms include two separate steps: 1) embedding learning and 2) sparse regression. Thus their performance is largely determined by the effectiveness of the constructed graph. In this paper, we present a framework by combining the objective functions of graph embedding and sparse regression so that embedding learning and sparse regression can be jointly implemented and optimized, instead of simply using the graph spectral for sparse regression. By the proposed framework, supervised, semisupervised, and unsupervised learning algorithms could be unified. Furthermore, we analyze two situations of the optimization problem for the proposed framework. By adopting an $L_{2,1}$-norm regularization for the proposed framework, it can perform feature selection and subspace learning simultaneously. Experiments on seven standard databases demonstrate that joint graph embedding and sparse regression method can significantly improve the recognition performance and consistently outperform the sparse regression method.**

***Index Terms*— Graph embedding, sparse regression, feature selection, subspace learning, $L_{2,1}$-norm.**

Manuscript received April 3, 2014; revised August 19, 2014 and November 28, 2014; accepted February 9, 2015. Date of publication February 19, 2015; date of current version March 3, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61101150, in part by the Shenzhen Special Fund for the Strategic Development of Emerging Industries under Grant JCYJ20120831165730901, and in part by the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. David Frakes.

X. Shi and Y. Yang are with the Shenzhen Key Laboratory of Broadband Network and Multimedia, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: sxs1234@163.com; yang.yujiu@sz.tsinghua.edu.cn).

Z. Guo is with the Shenzhen Key Laboratory of Broadband Network and Multimedia, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China, and also with the Key Laboratory of Measurement and Control of Complex System of Engineering, Ministry of Education, Southeast University, Nanjing 210018, China (e-mail: zhenhua.guo@sz.tsinghua.edu.cn).

Z. Lai is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518055, China (e-mail: lai_zhi_hui@163.com).

Z. Bao is with the School of Computer Science and Information Technology, RMIT University, Melbourne, VIC 3000, Australia (e-mail: zhifeng.bao@rmit.edu.au).

D. Zhang is with the Biometrics Research Centre, Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

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

Digital Object Identifier 10.1109/TIP.2015.2405474

I. Introduction

DIMENSIONALITY reduction is a key research topic in many fields of information processing, such as data mining, machine learning and pattern recognition. Many techniques have been proposed in the past years. Among these techniques, the linear algorithms including Principal Component Analysis (PCA) [1], Linear Discriminant Analysis (LDA) [2], [3], the nonlinear algorithms such as the kernel trick [4], ISOMAP [5], Locally Linear Embedding (LLE) [6], Laplacian Eigenmap (LE) [7], and the local geometric structure based algorithms like Locality Preserving Projection (LPP) [8] and Marginal Fisher Analysis (MFA) [9] have been very popular due to their relative simplicity and effectiveness. Besides, the tensorization algorithms [10], [11] have also been proposed to conduct dimensionality reduction on tensor data. Despite the different motivations of these algorithms, they can be unified into a graph embedding framework [9], [12].

One of the major disadvantages of all above algorithms is that the learned projective functions are linear combinations of all the original features, so it is hard to interpret the significance of the features [13]. To deal with this problem, many sparse subspace learning methods have been developed. For example, Sparse PCA (SPCA) [14] has been proposed based on elastic net [15] made up by an L_1-norm [16] and an L_2-norm regularization. A unified sparse subspace learning framework [13] has also been proposed based on elastic net. Furthermore, feature selection methods have been utilized to explore the significant features, typical algorithms include: PCA Score (PCAS) [17], Laplacian Score (LS) [18], Spectral Feature Selection (SFS) [19], Multi-Cluster Feature Selection (MCFS) [20] and Minimum Redundancy Spectral Feature Selection (MRSFS) [21]. Recently, joint feature selection and subspace learning algorithms [22], [23] have also been developed based on an $L_{2,1}$-norm regularization for performing feature selection and subspace learning simultaneously.

The algorithms mentioned above [17]–[23] can utilize both the manifold structure and learning mechanism, thus they are able to obtain better performance than traditional algorithms in many cases. But all of these algorithms first define the characterized manifold structure and then implement regression steps, as a result, the performance of regression steps is largely determined by the constructed graph. The reason is that once the graph spectral is obtained, it is fixed

in the following regression steps. If the graph spectral vectors and the optimal regression vectors can be jointly learned at the same time, the graph defined on the manifold data can be affected by the regression vectors, therefore, the performance of the algorithm might be improved [24], [25].

Based on the above motivations, in this paper, we propose a framework of joint graph embedding and sparse regression for dimensionality reduction, in which the graph spectral vectors can adaptively change with the optimal sparse regression vectors, and the framework can unify most of popular dimensionality reduction algorithms. To achieve this goal, we integrate the models of graph embedding and sparse regression simultaneously. For solving the new integrated model, two reformulation situations are adopted and analyzed. Since adaptive changed graph spectral approach has been applied to unsupervised learning [24], [25], in this paper, our experiments focus on supervised and semi-supervised learning.

Three main contributions of this paper are listed as follows:

1. We propose a unified framework of joint graph embedding and sparse regression, so most of popular dimensionality reduction algorithms including supervised, semi-supervised and unsupervised learning can be unified into this framework.
2. We analyze the optimization problem of the proposed framework and illustrate two transformation situations to solve this optimization problem.
3. Extensive experiments demonstrate that joint graph embedding and sparse regression method can significantly improve the recognition performance of the previous classical algorithms, and outperform the state of the art sparse regression method.

The rest of the paper is structured as follows. Section II introduces the notations and definitions used in this paper, and briefly reviews the graph embedding models and the sparse regression types. Section III presents the proposed framework of joint graph embedding and sparse regression, unifies the approaches in graph embedding [9] and the semi-supervised learning algorithms into the proposed framework, illustrates the relations between our study and other related work, and analyzes the computation complexity of the optimization problem in the proposed framework. Section IV reports and analyzes experimental results on benchmark databases. Finally, Section V gives the conclusion and discusses the future research work.

II. A Brief Review of Related Work

A. Notations and Definitions

For a matrix $M = (m_{ij}) \in R^{n\times d}$, its i-th row and j-th column are denoted by m^i and m_j, respectively. The L_p-norm of a vector $v \in R^n$ is defined as $\|v\|_p = (\sum_{i=1}^n |v_i|^p)^{\frac{1}{p}}$ $(p \in Z^+)$. The Frobenius norm of the matrix M is defined as

$$\|M\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} m_{ij}^2} = \sqrt{\sum_{i=1}^{n} \left\|m^i\right\|_2^2} \tag{1}$$

The L_1-norm of the matrix M is defined as

$$\|M\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{d} \left|m_{ij}\right| = \sum_{i=1}^{n} \left\|m^i\right\|_1 \tag{2}$$

The $L_{2,1}$-norm of a matrix was introduced in [26] and [27], it is defined as

$$\|M\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} m_{ij}^2} = \sum_{i=1}^{n} \left\|m^i\right\|_2 \tag{3}$$

An adaptive loss function for M is defined as

$$\|M\|_\sigma = \sum_{i=1}^{n} \frac{(1+\sigma)\left\|m^i\right\|_2^2}{\left\|m^i\right\|_2 + \sigma} \tag{4}$$

which is a nonnegative convex. When $\sigma \to 0$, $\|M\|_\sigma = \|M\|_{2,1}$; when $\sigma \to \infty$, $\|M\|_\sigma = \|M\|_F^2$ [28], [29].

B. Graph Embedding

Many approaches have been proposed for the task of dimensionality reduction. Although they have different motivations, they can be interpreted by a graph embedding framework [9], [12]. Given data $X = \{x_1, x_2, \cdots, x_n\} \in R^{d\times n}$, where n represents the number of observations and d is the number of predictors. In graph embedding [12], each vertex of a constructed graph $G = \{X, W\}$ corresponds to a data point $x_i \in X$. Let $W \in R^{n\times n}$ be a symmetric matrix with W_{ij} having the weight of the edge joining vertices i and j, and using W_{ij} to characterize the favorite similarity relationship between x_i and x_j. The purpose of graph embedding is to find an optimal low dimensional matrix $A \in R^{d\times c}$, which can best preserve the similarity relationship between the data points, the optimization problem for A is

$$\begin{aligned} &\arg\min_A \ Tr(A^T X L X^T A) \\ &s.t.\ A^T X D X^T A = I_c \end{aligned} \tag{5}$$

where $L = D - W$, D is a diagonal matrix and $D_{ii} = \sum_{j=1}^n W_{ij}$, $I_c \in R^{c\times c}$ is a unit matrix.

In graph embedding [9], there are two constructed graphs in each algorithm: intrinsic graph $G = \{X, W\}$ and penalty graph $G^p = \{X, W^p\}$. For each algorithm, the intrinsic graph in [9] is the same as the graph in [12]. The optimal A in graph embedding [9] is given by the following optimization problem:

$$\begin{aligned} &\arg\min_A \ Tr(A^T X L X^T A) \\ &s.t.\ A^T X B X^T A = I_c \end{aligned} \tag{6}$$

where $B = L^p = D^p - W^p$, D^p is also a diagonal matrix and $D_{ii}^p = \sum_{j=1}^n W_{ij}^p$.

It is worth noting that the different dimensionality reduction algorithms, including PCA, LDA, ISOMAP, LPP, LLE, have various intrinsic graphs $G = \{X, W\}$ and penalty graphs $G^p = \{X, W^p\}$, which lead to different L and B. The kernelization [4] and tensorization [10], [11] dimensionality reduction approaches can also be interpreted by Eq. (6).

Since the optimization model Eq. (6) is more flexible and more general than that of [9, eq. (5)], in this paper, we develop the proposed framework based on Eq. (6).

C. Sparse Regression

Given data $X = \{x_1, x_2, \cdots, x_n\} \in R^{d\times n}$, and $Y = \{y_1, y_2, \cdots, y_n\}^T \in R^{n\times c}$ is the corresponding low-dimensional matrix, the least square regression aims to find the projective matrix $A = (a_{ij}) \in R^{d\times c}$ by the following optimization model

$$\min_{A,b} \sum_{i=1}^{n} \left\| A^T x_i + b - y_i \right\|_2^2 \tag{7}$$

For simplicity, the bias b can be absorbed into A when the constant 1 is added as an additional dimension for each data point x_i. Thus Eq. (7) can be rewritten as

$$\min_{A} \sum_{i=1}^{n} \left\| A^T x_i - y_i \right\|_2^2 \tag{8}$$

On the basis of Eq. (8), other regression models have also been proposed, for example:

$$\min_{A} \sum_{i=1}^{n} \left\| A^T x_i - y_i \right\|_2 \tag{9}$$

which is named as robust loss function and is insensitive to data outliers [27].

And

$$\min_{A} \sum_{i=1}^{n} \left\| A^T x_i - y_i \right\|_\sigma \tag{10}$$

which can approximate Eq. (8) and Eq. (9) with the change of σ [28], [29].

We use the following regression model to represent these different regression models:

$$\min_{A} \sum_{i=1}^{n} \left\| A^T x_i - y_i \right\|_{\delta^*} \tag{11}$$

After adding a regularization term $R(A)$ with parameter γ on Eq. (11), it becomes

$$\min_{A} \sum_{i=1}^{n} \left\| A^T x_i - y_i \right\|_{\delta^*} + \gamma\, R(A) \tag{12}$$

There are several regularization terms stated as follows:

$$R_1(A) = \sum_{j=1}^{c} \left\| a_j \right\|_2^2, \quad R_2(A) = \sum_{j=1}^{c} \left\| a_j \right\|_1$$

$$R_3(A) = \sum_{i=1}^{d} \left\| a^i \right\|_2, \quad R_4(A) = \|A\|_\sigma = \sum_{i=1}^{d} \frac{(1+\sigma)\left\| a^i \right\|_2^2}{\left\| a^i \right\|_2 + \sigma}$$

$R_1(A)$ is the ridge regularization; $R_2(A)$ is the lasso regularization, combining $R_1(A)$ and $R_2(A)$ is the elastic net regularization. $R_3(A)$ penalizes all regression coefficients corresponding to a single feature as a whole. It can encourage row-sparsity and rank the significance of features [22], [24]; $R_4(A)$ smoothly interpolates between $L_{2,1}$-norm and L_2-norm [28]. All of these regularizations have been added on regression type and applied to graph embedding approaches [13], [22], [29]. For simplicity, we represent different regression models and varied regularizations by the following model:

$$\min_{A} \left\| X^T A - Y \right\|_{\delta^*} + \gamma\, \|A\|_{\rho^*} \tag{13}$$

Different δ^* and ρ^* can be selected according to different cases.

III. A Framework of Joint Graph Embedding and Sparse Regression

When graph embedding and sparse regression are jointly performed, the graph can be affected by the sparse regression vectors. In other words, the graph cannot only characterize the manifold structure, but also indicate the requirements of regression [24]. We formulate the optimization problem by combining the objective function Eq. (6) with Eq. (13) as

$$\begin{aligned} &arg\ \min_{Y,A}\ \alpha Tr(Y^T L Y) + \beta(\left\| X^T A - Y \right\|_{\delta^*} + \gamma\, \|A\|_{\rho^*}) \\ &s.t.\ Y^T B Y = I_c \end{aligned} \tag{14}$$

which is our proposed framework of joint graph embedding and sparse regression for dimensionality reduction. If Y is known, $\alpha = 0$, Eq. (14) will become the sparse regression model, which is easy to be solved. Otherwise, Eq. (14) is the joint graph embedding and sparse regression model, there are many choices for α, but they are equivalent to $\alpha = 1$. As $B = I_n$, where $I_n \in R^{n\times n}$ is a unit matrix, Eq. (14) can be directly solved, but as $B \neq I_n$, it is difficult to directly solve this optimization problem because the constraint is not convex [22]. Thus we will reformulate this problem to make it easy to be solved. The detailed transformation process will be shown in the following section.

Although there are many cases with different norms δ^* and ρ^*, in this paper, we select $\delta^* = \|\cdot\|_F^2$ and $\rho^* = \|\cdot\|_{2,1}$, respectively. The reason for this is that the L_2-norm regularization is common and simple, and the $L_{2,1}$-norm regularization can perform feature selection and subspace learning simultaneously, which usually has better performance than performing feature selection and subspace learning independently [22]. Also, $\|A\|_{2,1}$ is convex and could be easily optimized [27].

A. Transformation

In order to solve the optimization problem Eq. (14) with $B \neq I_n$, we need accomplish the following two tasks:

1. Transform Eq. (6) into the following optimization model

$$\begin{aligned} &arg\ \min_{\widetilde{Y}}\ Tr(\widetilde{Y}^T L \widetilde{Y}) \\ &s.t.\ \widetilde{Y}^T \widetilde{Y} = I_c \end{aligned} \tag{15}$$

   where $\widetilde{Y} \in R^{n\times c}$.
2. Find the relationship between $\widetilde{Y}$ and A.

In order to transform Eq. (6) into Eq. (15), we adopt Singular Value Decomposition (SVD) [30]. There are two matrices in Eq. (6) that can be decomposed by SVD,

1) the symmetric matrix B, which will place emphasis on the relationship between L and B, motivated by [13] and [22] that obtain the relationship between W and D through eigen-decomposition; 2) the symmetric matrix XBX^T, which studies the relationship between XLX^T and XBX^T, motivated by fisherface [23], [31]. According to the different matrices decomposed by SVD, we name the method that solves Eq. (14) via the SVD on B as Graph Joint Graph embedding and Sparse regression (GJGS), because B is the penalty graph in graph embedding [9]; the method solving Eq. (14) via the SVD of XBX^T is called Matrix Joint Graph embedding and Sparse regression (MJGS), because the XBX^T is similar to the between-class scatter matrix in fisherface [31].

1) SVD of Penalty Graph B: In Eq. (6), we do SVD of B, due to the fact that B is a symmetric matrix, thus

$$B = US_1U^T \tag{16}$$

Then

$$A^T XUS_1U^T X^T A = I_c \tag{17}$$

Let

$$Y_1 = S_1^{\frac{1}{2}} U^T X^T A \tag{18}$$

Thus $Y_1^T Y_1 = I_c$, and

$$X^T A = US_1^{-\frac{1}{2}} Y_1 \tag{19}$$

The transformation from Eq. (18) to Eq. (19) suggests that B must be a positive-definite matrix to make sure that the matrix $U \in R^{n\times n}$ is full rank. However, sometimes the matrix W^p is a singular matrix that leads to the singular matrix B. In that case, we set $B = B + \varepsilon I_n$, where $\varepsilon \to 0$.

Substituting Eq. (19) into Eq. (6), it can get

$$\begin{aligned} Y_1^* = arg\ \min_{Y_1}\ & Tr(Y_1^T S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} Y_1) \\ s.t.\ & Y_1^T Y_1 = I_c \end{aligned} \tag{20}$$

This is an eigen-analysis problem, whose solution is eigenvectors of $S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}}$.

According to $Y_1 = S_1^{\frac{1}{2}} U^T X^T A$, adding sparse regression into Eq. (20) and selecting $\delta^* = \|\cdot\|_F^2$ and $\rho^* = \|\cdot\|_{2,1}$, we can get

$$\begin{aligned} L(A, G, Y_1) = arg \min_{\substack{A,G,Y_1 \\ Y_1^T Y_1 = I_c}} & \alpha Tr(Y_1^T S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} Y_1) \\ & + \beta(\left\| S_1^{\frac{1}{2}} U^T X^T A - Y_1 \right\|_F^2 + \gamma\, Tr(A^T GA)) \end{aligned} \tag{21}$$

where the derivative of $Tr(A^T GA)$ with respect to A is equivalent to the derivative of $\|A\|_{2,1}$, $Tr(A^T GA) = \|A\|_{2,1}/2$ [24], G is a diagonal matrix with the i-th diagonal element equal to

$$g_{ii} = \frac{1}{2\left\| a^i \right\|_2} \tag{22}$$

where a^i is the i-th row vector of A, and when $a^i = 0$, we can define $g_{ii} = \frac{1}{2\sqrt{\|a^i\|_2^2 + \epsilon}}$ ($\epsilon \to 0$) as stated in [32].

If $\alpha = 0$, Eq. (21) is a special case of our proposed framework and becomes a sparse regression via $L_{2,1}$-norm regularization and its solution can be easily obtained. Here, we set $\alpha = 1$, it is joint graph embedding and sparse regression via $L_{2,1}$-norm regularization.

Taking the derivative of Eq. (21) with respect to A and setting the derivative to zero, we can get

$$\frac{\partial L(A, G, Y_1)}{\partial A} = XUS_1U^T X^T A - XUS_1^{\frac{1}{2}} Y_1 + \gamma GA = 0 \tag{23}$$

Due to $B = US_1U^T$, thus

$$A = (XBX^T + \gamma G)^{-1} XUS_1^{\frac{1}{2}} Y_1 \tag{24}$$

Substituting Eq. (24) into Eq. (21), we will get

$$\begin{aligned} & L(A, G, Y_1) \\ & = Tr(Y_1^T S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} Y_1) + \beta(Tr(A^T XUS_1U^T X^T A) \\ & \quad - 2Tr(A^T XUS_1^{\frac{1}{2}} Y_1) + Tr(Y_1^T Y_1) + \gamma\, Tr(A^T GA)) \\ & = Tr(Y_1^T S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} Y_1) \\ & \quad + \beta(-Tr(A^T (XBX^T + \gamma G)A) + Tr(Y_1^T Y_1)) \\ & = Tr(Y_1^T S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} Y_1) \\ & \quad + \beta(-Tr(Y_1^T S_1^{\frac{1}{2}} U^T X^T (XBX^T + \gamma G)^{-1} XUS_1^{\frac{1}{2}} Y_1) \\ & \quad + Tr(Y_1^T Y_1)) \\ & = Tr(Y_1^T (S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} - \beta S_1^{\frac{1}{2}} U^T X^T \\ & \qquad \times (XBX^T + \gamma G)^{-1} XUS_1^{\frac{1}{2}} + \beta I_n) Y_1) \end{aligned} \tag{25}$$

Considering the objective function in Eq. (25) and the constraint $Y_1^T Y_1 = I_c$, we can get the following optimization problem

$$\begin{aligned} Y_1^* = arg\ \min_{Y_1}\ & Tr(Y_1^T (S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} \\ & - \beta S_1^{\frac{1}{2}} U^T X^T (XBX^T + \gamma G)^{-1} XUS_1^{\frac{1}{2}} + \beta I_n) Y_1) \\ s.t.\ & Y_1^T Y_1 = I_c \end{aligned} \tag{26}$$

The solution of Eq. (26) is the eigenvectors of $S_1^{-\frac{1}{2}} U^T LUS_1^{-\frac{1}{2}} - \beta S_1^{\frac{1}{2}} U^T X^T (XBX^T + \gamma G)^{-1} XUS_1^{\frac{1}{2}} + \beta I_n$. Compared with Eq. (20), Eq. (26) shows the change of the graph after considering the sparse regression vectors, which leads to the different graph spectral vectors.

In summary, since the optimization process of the case $\alpha = 0$ can be regarded as one particular case (Y_1 is known) and can be easily obtained, we present the algorithm for optimizing Eq. (21) as $\alpha = 1$ in Algorithm 1.

2) SVD of Matrix XBX^T: As we focus on doing SVD of XBX^T, the transformation is presented as follows.

Do SVD of XBX^T, thus

$$S_B = XBX^T = VS_2V^T \tag{27}$$

Algorithm 1 Graph Joint Graph Embedding and Sparse Regression (GJGS)

Input: $X \in R^{d\times n}$, $B \in R^{n\times n}$, $L \in R^{n\times n}$, β, γ.
Output: $A \in R^{d\times c}$.
Initialize: $G_{(0)} = I_d$, $t = 0$;
Do SVD of $B = US_1U^T$ (If B is a singular matrix, $B = B + \epsilon I_n$);
repeat
Calculate $Y_{1(t+1)}$ by the eigen-decomposition of
$S_1^{-\frac{1}{2}}U^TLUS_1^{-\frac{1}{2}} - \beta S_1^{\frac{1}{2}}U^TX^T(XBX^T + \gamma G_{(t)})^{-1}XUS_1^{\frac{1}{2}} + \beta I_n$;
Calculate $A_{(t+1)} = (XBX^T + \gamma G_{(t)})^{-1}XUS_1^{\frac{1}{2}}Y_{1(t+1)}$;
Calculate $G_{(t+1)}$ based on $A_{(t+1)}$,
where the i-th element is calculated by $G_{(t+1)}(i,i) = \frac{1}{2\left\|a^i_{(t+1)}\right\|_2}$;
$t = t + 1$;
until convergences

Algorithm 2 Matrix Joint Graph Embedding and Sparse Regression (MJGS)

Input: $X \in R^{d\times n}$, $B \in R^{n\times n}$, $L \in R^{n\times n}$, β, γ.
Output: $A \in R^{d\times c}$.
Initialize: $G_{(0)} = I_d$, $t = 0$;
Do SVD of $S_B = XBX^T = VS_2V^T$ (If XBX^T is a singular matrix, $S_B = XBX^T + \epsilon I_d$);
repeat
Calculate $Y_{2(t+1)}$ by the eigen-decomposition of
$S_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}} - \beta S_2^{\frac{1}{2}}V^T(S_B + \gamma G_{(t)})^{-1}VS_2^{\frac{1}{2}} + \beta I_d$;
Calculate $A_{(t+1)} = (S_B + \gamma G_{(t)})^{-1}VS_2^{\frac{1}{2}}Y_{2(t+1)}$;
Calculate $G_{(t+1)}$ based on $A_{(t+1)}$,
where the i-th element is calculated by $G_{(t+1)}(i,i) = \frac{1}{2\left\|a^i_{(t+1)}\right\|_2}$;
$t = t + 1$;
until convergences

when XBX^T is a singular value matrix, we set $S_B = XBX^T + \epsilon I_d$, where $I_d \in R^{d\times d}$ represents a unit matrix.

Then

$$A^TVS_2V^TA = I_c \tag{28}$$

Define

$$Y_2 = S_2^{\frac{1}{2}}V^TA \tag{29}$$

Thus $Y_2^TY_2 = I_c$, $Y_2 \in R^{d\times c}$, and

$$A = VS_2^{-\frac{1}{2}}Y_2 \tag{30}$$

Substituting Eq. (30) into Eq. (6), it can get

$$\begin{aligned} Y_2^* = arg\ \min_{Y_2}\ & Tr(Y_2^TS_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}}Y_2) \\ s.t.\ & Y_2^TY_2 = I_c \end{aligned} \tag{31}$$

Therefore, Y_2^* is the eigenvector of $S_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}}$.

Based on $Y_2 = S_2^{\frac{1}{2}}V^TA$, adding sparse regression into Eq. (31) and selecting $\delta^* = \|\cdot\|_F^2$ and $\rho^* = \|\cdot\|_{2,1}$, we will get

$$\begin{aligned} L(A,G,Y_2) = arg \min_{\substack{A,G,Y_2 \\ Y_2^TY_2=I_c}} & \alpha Tr(Y_2^TS_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}}Y_2) \\ & +\beta\left(\left\|S_2^{\frac{1}{2}}V^TA - Y_2\right\|_F^2 + \gamma\, Tr(A^TGA)\right) \end{aligned} \tag{32}$$

Similar to Algorithm 1, with fixed G and Y_2, it can get the solution of Eq. (32)

$$A = (S_B + \gamma\, G)^{-1}\, VS_2^{\frac{1}{2}}Y_2 \tag{33}$$

Substituting Eq. (33) into Eq. (32) and setting $\alpha = 1$, we can get

$$\begin{aligned} Y_2^* = arg\ \min_{Y_2}\ & Tr\Big(Y_2^T\Big(S_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}} \\ & -\beta S_2^{\frac{1}{2}}V^T(S_B+\gamma\, G)^{-1}VS_2^{\frac{1}{2}}+\beta I_d\Big)Y_2\Big) \\ s.t.\ & Y_2^TY_2 = I_c \end{aligned} \tag{34}$$

Thus, Y_2^* is the eigenvectors of $S_2^{-\frac{1}{2}}V^TXLX^TVS_2^{-\frac{1}{2}} - \beta S_2^{\frac{1}{2}}V^T(S_B + \gamma\, G)^{-1}VS_2^{\frac{1}{2}} + \beta I_d$.

In summary, we present the algorithm for optimizing Eq. (32) as $\alpha = 1$ in Algorithm 2.

Since Eq. (21) and Eq. (32) are the same regression type as the optimization problem in [24] and [25], whose convergence has been proved, therefore methods GJGS and MJGS are converged as well.

B. Semi-Supervised Learning for Dimensionality Reduction

Above section suggests that the proposed framework can cover the algorithms in graph embedding [9] including the supervised and unsupervised learning. In fact, the semi-supervised learning algorithms can also be unified into the proposed framework.

For supervised learning algorithms such as LDA, MFA and LPP, overfitting might happen when there are insufficient training samples [33]. A typical way to prevent overfitting is to impose a regularizer as the following optimization problem

$$arg\ \max_{A}\ Tr\left(\frac{A^TX_lB_lX_l^TA}{A^TX_lL_lX_l^TA + \mu J(A)}\right) \tag{35}$$

where $J(A)$ controls the learning complexity of the hypothesis family, and the coefficient μ controls the balance between the model complexity and empirical loss. $X_l \in R^{d\times m}$ represents the labeled data points, $L_l \in R^{m\times m}$ and $B_l \in R^{m\times m}$ constructed by the labeled data points corresponding to L and B in Eq. (6) respectively, where m is the number of labeled data points.

The regularizer $J(A)$ provides us with the flexibility for incorporating our prior knowledge to improve the performance according to the particular applications. Researchers can use the unlabeled data points to construct a $J(A)$ with incorporating manifold structure, such as Semi-supervised Discriminant Analysis (SDA) [33]. The direct graph of these manifold algorithms can be utilized to construct $J(A)$ as follows

$$J(A) = A^TXLX^TA \tag{36}$$

where $X \in R^{d\times n}$ contains labeled and unlabeled data points, $L = D - W$ is the direct graph constructed by the labeled and

Algorithm 3 Semi-Supervised Learning by Joint Graph Embedding and Sparse Regression (SJGS)

Input: $X \in R^{d\times n}$, $B_l \in R^{n\times n}$, $L_l \in R^{n\times n}$, $B \in R^{n\times n}$, $L \in R^{n\times n}$, μ, β, γ.
Output: $A \in R^{d\times c}$.
Initialize: $G_{(0)} = I_d$, $t = 0$;
Do SVD of $S_B = X_l B_l X_l^T = V S_2 V^T$;
repeat
Calculate $Y_{2(t+1)}$ by the eigen-decomposition of $S_2^{-\frac{1}{2}} V^T (X_l L_l X_l^T + \mu X L X^T) V S_2^{-\frac{1}{2}} - \beta S_2^{\frac{1}{2}} V^T (S_B + \gamma G_{(t)})^{-1} V S_2^{\frac{1}{2}} + \beta I_d$;
Calculate $A_{(t+1)} = (S_B + \gamma G_t)^{-1} V S_2^{\frac{1}{2}} Y_{2(t+1)}$;
Calculate $G_{(t+1)}$ based on $A_{(t+1)}$,
where the i-th element is calculated by $G_{(t+1)}(i,i) = \frac{1}{2\left\|a^i_{(t+1)}\right\|_2}$;
$t = t + 1$;
until convergences

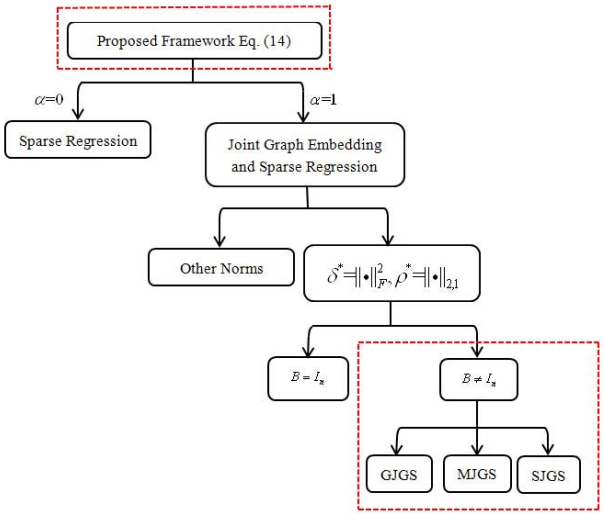


Fig. 1. The derivative cases from the proposed framework Eq. (14) (The main contributions are circled by red line)†.

unlabeled data points together, W is a weight matrix, D is a diagonal matrix and $D_{ii} = \sum_{j=1}^{n} W_{ij}$.
Thus Eq. (35) can be rewritten as

$$\underset{A}{arg\ max}\ Tr\left(\frac{A^T X_l B_l X_l^T A}{A^T X_l L_l X_l^T A + \mu A^T X L X^T A}\right) \quad (37)$$

If we let $A^T X_l B_l X_l^T A = I_c$, then Eq. (37) becomes

$$\underset{A}{arg\ min}\ Tr\left(A^T (X_l L_l X_l^T + \mu X L X^T) A\right)$$
$$s.t.\ A^T X_l B_l X_l^T A = I_c \quad (38)$$

The matrix $X_l L_l X_l^T + \mu X L X^T$ is a symmetric matrix, which can be written as the type of XLX^T in Eq. (6), and $X_l B_l X_l^T$ can be written as the type of XBX^T. Therefore, we can unify Eq. (38) into the proposed framework Eq. (14). Then we use the method similar to Algorithm 2 to solve this optimization problem.

In summary, we present the case $\alpha = 1$ of the whole optimization process of semi-supervised learning with $\delta^* = \|\cdot\|_F^2$ and $\rho^* = \|\cdot\|_{2,1}$ in Algorithm 3.

C. Relations to Other Approaches

In recent sparse regression research, one algorithm with adaptively changed graph spectral has been proposed in [24]. It uses LLE to select features for unsupervised learning. But LLE is simply one particular case of graph embedding, in which $B = I_n$. Although the extended version of unsupervised joint embedding learning and sparse regression framework [25] also mentioned the sparse regression case as $B \neq I_n$, it just considered the relationship between B and L, and did not consider the relationship between XBX^T and XLX^T. In addition, compared with the optimization model in [25], the optimization model of Eq. (14) is a totally different optimization problem, because the constraint in Eq. (14) is not convex [22]. And Eq. (14) can also be transformed into the same regression type as the optimization model in [25] when $B \neq I_n$, which suggests that the optimization model Eq. (14) is more general than the optimization model in [25]. Moreover, [24] and [25] focus on the unsupervised learning only, but we aim to unify all the algorithms including supervised and unsupervised learning in graph embedding [9] into our proposed framework, and we also want to unify the semi-supervised learning algorithms into our framework. Thus this paper has a completely different goal from [25].

Although [13] and [22] also unified the algorithms in graph embedding into one framework, they unified the algorithms into sparse regression framework based on the optimization model Eq. (5). Besides, [23] that transformed LDA into sparse regression type via $L_{2,1}$-norm minimization, and [34] that was developed based on LPP, are sparse regression methods as well. In this paper, we mainly unify the algorithms into joint graph embedding and sparse regression framework based on Eq. (6) that is more flexible and more general than Eq. (5).

In order to better illustrate the relationship to other approaches and our main contributions, we present the derivative cases of the proposed framework in Fig. 1. From it, we can see that our main contributions (circled by red line) include the proposed framework Eq. (14), and three proposed algorithms: GJGS, MJGS and SJGS. GJGS and MJGS are joint graph embedding and sparse regression methods via an L_2-norm loss function and an $L_{2,1}$-norm regularization as $B \neq I_n$, while SJGS unifies the semi-supervised learning algorithm into the proposed framework.

Recently, Sparse Representation based Classifier (SRC) [35], [36], has been proposed. It assumes that each test data point could be linear combined by the training set, and it explores classification information through these combined coefficients. Many related techniques including [37]–[39] have been developed. By contrast, the algorithms covered in the proposed framework do not explore the information of test data points, so they require much less test time than SRC and its variants. Only with much higher dimensions, could SRC and its variants obtain better

†The source codes of the proposed algorithm can be downloaded from https://drive.google.com/file/d/0BwRQCMfgq102VGRXVWZuNmFxbTQ/view?usp=sharing

Fig. 2. Ten image samples in each image database. (a) COIL20, (b) UMIST, (c) ORL, (d) USPS, (e) AR, (f) Extended Yale-B, (g) PIE.

classification than most of algorithms covered in the proposed framework.

D. Computation Complexity

In this subsection, we mainly analyze the time complexity of our proposed methods, and make a comparison with the sparse regression methods like MCFS, MRSFS and [23].

In the proposed method GJGS, the SVD of the penalty graph B is of order $O(n^3)$. Computing the matrix XBX^T requires $max\left\{O(nd^2), O(n^2d)\right\}$ operations and the inversion of $XBX^T+\gamma G$ costs at most $O(d^3)$ operations. So, computation of the matrix $S_1^{-\frac{1}{2}}U^TLUS_1^{-\frac{1}{2}}-\beta S_1^{\frac{1}{2}}U^TX^T(XBX^T+\gamma G)^{-1}XUS_1^{\frac{1}{2}}+\beta I_n$ requires $max\left\{O(n^3), O(nd^2)\right\}$ operations, and its eigen-decomposition needs $O(n^3)$ operations. Next, computing the matrix $(XBX^T+\gamma G)^{-1}XUS_1^{\frac{1}{2}}Y_1$ costs $max\left\{O(n^2d), O(nd^2)\right\}$. Lastly, the operations for computing G are $O(d^2)$. Therefore, as $d >> n$, the time complexity of the method GJGS is $O(kd^3)$, where k is the number of iterations. Similar to GJGS, as $d >> n$, the time complexity of both MJGS and SJGS is $O(kd^3)$. However, as $d >> n$, the dimension d is usually reduced to $n-c$ before performing our methods, where c is the number of classes, as a result, the time complexity of our methods becomes $O(kn^3)$, which suggests that the proposed methods are suitable for high-dimensional data.

In contrast with the sparse regression methods focusing the relationship between B and L, like MCFS and MRSFS, the method GJGS mainly contains two more steps: 1) computing SVD of the penalty graph B, which requires $O(n^3)$ operations; 2) computing the matrix $S_1^{-\frac{1}{2}}U^TLUS_1^{-\frac{1}{2}}-\beta S_1^{\frac{1}{2}}U^TX^T(XBX^T+\gamma G)^{-1}XUS_1^{\frac{1}{2}}+\beta I_n$. But those sparse methods usually need to obtain the inversion of B, whose cost is $O(n^3)$, thus the additional computation cost for GJGS is the step 2), which costs $max\left\{O(kn^3), O(knd^2)\right\}$. Similarly, in comparison with the method [23] containing the SVD of XBX^T, the additional cost for MJGS is $max\left\{O(kd^3), O(kdn^2)\right\}$.

IV. Experiments

In [24] and [25], joint graph embedding and sparse regression method has been applied to unsupervised learning algorithms. Therefore, we focus on supervised and semi-supervised learning algorithms, and apply the joint graph embedding and sparse regression method to two representative linear discriminant analysis algorithms LDA [31] and MFA [9] in our experiments.

In the following parts, we use seven standard databases for experiments:

COIL20 object database [40], [41]: this database contains 1440 images of 20 objects, and each object is captured from varying angles with a 5-degree interval. Here, we resize each image to 32×32 pixels. Ten images of one object are shown in Fig. 2a.

UMIST face database [42]: this database contains 575 multi-view images of 20 human subjects under a wide range of poses from profile to frontal views. Here, all images are selected and each face image is resized to 28×23 pixels. Ten images of one human are presented in Fig. 2b.

ORL face database [43]: this database contains 400 face images of 40 human subjects under a dark homogenous ground with the subjects in an upright, frontal position. Here, all images are chosen and each face image is resized to 32×32 pixels. Ten images of one subject are displayed in Fig. 2c.

USPS handwritten digits database [44]: contains 8-bit gray-scale images of "0" through "9", the size of each image is 16×16 pixels. Thus each digit image is represented as a 256D vector. Here, 9298 handwritten digit images are selected, which were used in [45]. Ten sample images from USPS database are shown in Fig. 2d.

AR face database [46]: this database contains over 4000 color images of 126 human subjects (70 men and 56 women), the images contain frontal view faces with different facial expressions, illumination conditions and occlusions. Here, we choose 2600 face images of 100 individuals (50 men and 50 women) under different facial expressions, illumination conditions and occlusions. All the face images are manually cropped and resized to 33×30 pixels. Ten face images of one human subject are shown in Fig. 2e.

Extended Yale-B face database [47]: this database contains 16128 face images of 38 human subjects under 9 poses and 64 illumination conditions. We choose the frontal pose with different illuminations. There are 2414 face images in total. All the face images are manually aligned and cropped, and they are also resized to 32×32 pixels. Ten face images of one human subject are presented in Fig. 2f.

CMU PIE face database [48]: this database contains 41368 face images of 68 human subjects under 13 different poses and 43 illumination conditions, and with 4 different expressions. For the current study, we select the images from the frontal pose (C27) and each subject has around 49 images from varying illuminations and facial expressions, and they are manually cropped and resized to 32×32 pixels. Ten images of one human subject are displayed in Fig. 2g.

A. Supervised Learning Experiments

In supervised learning experiments, as $\alpha = 0$, Eq. (14) becomes sparse regression, we name the method that solves Eq. (14) via the SVD of B as Graph Sparse Regression (GSR); the method solving Eq. (14) via the SVD of XBX^T as Matrix Sparse Regression (MSR). Then we apply GSR, GJGS, MSR and MJGS to LDA and MFA, the derivative methods are divided into two groups based on LDA and MFA. As a side note, the GSR (LDA) and MSR (LDA) are FSSL (LDA) and L21FLDA, which are presented in [22] and [23], respectively. In addition, we present a sparse regression method Unified Sparse Subspace Learning (USSL) [13] with LDA, which adopts an L_1-norm regularization. Moreover, in order to better show the performance of our proposed methods, we also present the classical methods: PCA and LPP, and the methods Collaborative Representation based Classification (CRC) [37] and Two-phase Sparse Representation (TSR) [38], which are derivative from the popular method SRC with less test time.

1) Parameter Settings: The sets of training images are randomly selected from each database, and the remained images are used for testing. On COIL20 and ORL database, $p = [4, 5, 6]$; $p = [6, 8, 10]$ in UMSIT database; $p = [10, 20, 30]$ in USPS database; $p = [4, 6, 8]$ in AR database; p=[5, 10, 20] in Extended Yable-B database and PIE database, where p is the number of training images of each class. We repeat this process 20 times and calculate the mean recognition accuracy.

Different preprocessing methods can result in different recognition accuracy for each dataset, there are two common preprocessing ways: 1) scaling features to [0, 1]; 2) normalizing each face image vector to be a unit vector. In order to get better recognition accuracy, empirically, we preprocess the data on COIL20, UMIST and ORL image databases using the former way, and the data sets in USPS, AR, Extended Yale-B and PIE are preprocessed by the latter way to explore the algorithms' performance. Then we use PCA to reduce the dimensionality of the data to $n - c$ before performing our methods to reduce the dimensionality to $c-1$, where n is the number of observations and c is the number of classes. It is a popular scheme for fisherface [31] and MFA [9]. While PCA, LPP and CRC usually obtain the best results on higher dimension than $c-1$, we respectively set their dimension to $min(n-1, 200)$, $min(n-1, 200)$ and $n/2$ in our experiments, respectively. TSR selects some representative data points for obtaining better results, thus the selected number of representative data points is $min(n, 300)$ in experiments. Here, the method PCA adopts the nearest neighbor classifier [49], CRC and TSR utilize the nearest subspace classifier [50], and other methods use the nearest centroid classifier [51] for classification.

Because the parameters in sparse regression and joint graph embedding have different ranges, in order to get their best recognition accuracy, we tune the parameter γ in sparse regression by searching the grid [0.01, 0.02, 0.05, 0.1, 0.2, 1, 2, 10] for the algorithms GSR and MSR, and searching the grid [10, 20, $\cdots$, 100] for USSL according to [13]; for the algorithms GJGS and MJGS, the parameters β and γ are searched on the grid [10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 10^2]. Then we report the top-2 recognition accuracy from the best parameter configuration.

2) Results and Discussions: Table 1 shows the recognition accuracy of LDA, MFA and their variants on COIL20, UMIST, ORL, AR, Extended Yale-B and PIE databases. The algorithms are divided into two groups according to LDA and MFA, the results shown in boldface are the best accuracy in each group. The dimension of LDA, MFA and their variants is $c - 1$ in Table 1. In order to better present the algorithms' recognition performance, we also present Fig. 3 to show recognition accuracy with different dimensions for LDA and its variants, and Fig. 4 to illustrate relationship between accuracy and

TABLE I

RECOGNITION ACCURACY (MEAN ACCURACY±STANDARD DERIVATION) OF LDA, MFA AND THEIR VARIANTS FOR SUPERVISED LEARNING ON SEVEN STANDARD DATABASES. (a) COIL20, (b) UMIST, (c) ORL, (d) USPS

(a)

Group	Method	$p=4$		$p=5$		$p=6$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	81.07±1.84	-	83.62±1.25	-	85.91±1.20	-
	LPP[8]	76.35±2.81	-	77.63±1.95	-	79.85±2.36	-
	CRC[37]	77.37±1.84	-	80.46±1.82	-	83.33±1.29	-
	TSR[38]	76.30±1.76	-	80.02±2.08	-	82.51±1.76	-
LDA	fisherface[31]	76.87±2.59	-	78.45±1.90	-	81.58±1.47	-
	USSL[13]	78.03±2.96	30	80.90±2.37	30	83.10±1.88	40
	GSR[22]	79.13±3.10	1	81.39±2.19	1	83.26±1.94	1
	GJGS	**83.49±2.82**	1, 10^{-1}	**85.56±1.92**	1, 10^{-1}	**87.95±1.60**	1, 10^{-1}
	MSR[23]	77.59±2.04	1	81.67±2.40	1	84.58±2.02	1
	MJGS	78.59±1.95	1, 1	82.34±2.73	1, 1	85.42±1.82	5, 10^{-1}
	MFA[9]	83.33±2.96	-	84.87±1.94	-	86.96±1.81	-
	GSR	81.09±2.79	10	83.43±2.19	10	85.48±1.91	10
MFA	GJGS	**83.71±2.66**	10^{-1}, 1	**85.68±2.02**	10^{-1}, 1	**88.11±1.57**	10^{-1}, 1
	MSR	77.98±1.99	10	82.15±2.29	10	84.65±1.71	10
	MJGS	78.10±1.88	$10^{-2}, 10$	82.19±2.40	$10^{-2}, 10$	85.06±1.70	$10^{-2}, 10$

(b)

Group	Method	$p=6$		$p=8$		$p=10$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	88.70±2.21	-	92.48±1.61	-	95.77±1.75	-
	LPP[8]	89.42±2.21	-	92.53±2.15	-	94.77±1.92	-
	CRC[37]	88.64±2.52	-	92.78±1.75	-	95.52±1.84	-
	TSR[38]	89.26±2.25	-	92.37±1.96	-	95.24±2.18	-
	fisherface[31]	88.02±3.25	-	92.59±2.06	-	92.21±1.19	-
	USSL[13]	88.63±3.45	60	93.80±2.09	70	95.23±1.72	70
LDA	GSR[22]	89.82±3.05	0.2	94.22±1.79	0.2	95.85±1.18	0.1
	GJGS	**95.14±2.31**	5, 10^{-2}	**97.87±0.86**	5, 10^{-2}	**98.31±1.02**	5, 10^{-2}
	MSR[23]	91.10±2.75	0.2	95.64±1.70	0.2	97.55±1.06	0.2
	MJGS	91.60±2.36	1, 10^{-1}	96.02±1.47	1, 10^{-1}	97.76±1.20	1, 10^{-1}
	MFA[9]	95.14±2.24	-	97.83±0.98	-	98.15±1.08	-
	GSR	91.73±3.04	2	95.49±1.74	2	96.77±1.23	2
MFA	GJGS	**95.15±2.21**	5, 10^{-2}	**97.89±1.00**	5, 10^{-2}	**98.40±0.89**	5, 10^{-2}
	MSR	90.88±2.73	2	95.64±1.81	2	97.51±1.14	2
	MJGS	91.38±2.48	$10^{2}, 10^{-3}$	95.75±1.78	$10^{2}, 10^{-3}$	97.71±1.02	$10^{2}, 10^{-3}$

(c)

Group	Method	$p=4$		$p=5$		$p=6$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	84.67±2.15	-	87.78±2.02	-	89.50±3.05	-
	LPP[8]	89.23±2.55	-	92.93±1.29	-	94.87±1.28	-
	CRC[37]	91.48±1.28	-	93.08±1.93	-	93.38±1.65	-
	TSR[38]	92.85±1.66	-	94.30±1.87	-	95.16±1.85	-
	fisherface[31]	90.65±1.87	-	93.05±1.83	-	94.59±1.81	-
	USSL[13]	89.46±2.75	70	94.48±1.80	80	94.72±1.60	90
LDA	GSR[22]	89.77±3.01	0.2	92.55±1.66	0.2	95.19±2.07	0.2
	GJGS	**93.90±2.20**	10^{2}, 10^{-3}	**96.07±1.05**	10^{2}, 10^{-3}	**97.31±1.73**	10^{2}, 10^{-3}
	MSR[23]	90.58±2.17	0.2	93.04±1.85	0.2	95.94±1.46	0.2
	MJGS	90.69±2.31	5, 10^{-1}	93.55±1.90	5, 10^{-1}	96.22±1.75	5, 10^{-1}
	MFA[9]	**93.98±2.24**	-	96.33±1.14	-	97.22±1.50	-
	GSR	91.12±2.79	10	94.40±1.68	10	96.34±1.29	10
MFA	GJGS	**93.98±2.56**	10^{2}, 10^{-3}	**96.60±1.00**	10^{2}, 10^{-3}	**97.78±1.29**	10^{2}, 10^{-3}
	MSR	90.90±2.25	10	93.73±1.96	10	96.19±1.67	10
	MJGS	90.94±1.36	$10^{-2}, 5$	93.80±2.00	$10^{-2}, 5$	96.31±1.36	$10^{-2}, 5$

(d)

Group	Method	$p=10$		$p=20$		$p=30$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	82.18±1.33	-	87.06±0.80	-	89.24±0.62	-
	LPP[8]	64.15±2.97	-	49.04±3.33	-	47.75±3.64	-
	CRC[37]	83.44±1.22	-	85.55±0.97	-	87.02±0.52	-
	TSR[38]	81.19±1.20	-	85.66±0.74	-	87.84±0.62	-
	fisherface[31]	65.53±2.38	-	51.29±2.77	-	49.19±1.81	-
	USSL[13]	82.68±1.88	20	85.90±1.62	30	86.91±1.58	40
LDA	GSR[22]	82.89±1.61	0.1	85.99±1.36	0.05	87.62±0.83	0.1
	GJGS	**84.00±1.65**	$5, 10^{2}$, 10^{-3}	**87.37±0.84**	$5, 10^{2}$, 10^{-3}	**88.48±0.72**	10^{-1}, 10^{-1}
	MSR[23]	82.83±1.79	0.2	86.15±1.33	0.1	87.86±0.70	0.05
	MJGS	83.97±1.67	1, 10^{-1}	87.35±0.85	10, 10^{-2}	88.45±1.42	5, 10^{-2}
	MFA[9]	79.95±1.55	-	83.28±0.93	-	85.55±0.67	-
	GSR	83.79±1.48	1	87.24±0.94	1	88.36±0.76	1
MFA	GJGS	**84.44±1.60**	1, 10^{-1}	**87.81±0.88**	10^{-2}, 1	**88.86±0.73**	10^{-1}, 1
	MSR	83.71±1.65	2	87.15±0.97	2	88.46±0.69	1
	MJGS	84.28±1.64	$10^{-1}, 1$	87.62±0.88	$10^{-1}, 1$	88.85±0.73	$10^{-1}, 1$

TABLE I

(Continued.) RECOGNITION ACCURACY (MEAN ACCURACY±STANDARD DERIVATION) OF LDA, MFA AND THEIR VARIANTS FOR SUPERVISED LEARNING ON SEVEN STANDARD DATABASES. (e) AR, (f) EXTENDED YALE-B, (g) PIE

(e)

Group	Method	$p=4$		$p=6$		$p=8$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	25.83±1.05	-	32.30±0.87	-	38.05±1.06	-
	LPP[8]	82.53±0.92	-	87.74±0.86	-	88.17±0.63	-
	CRC[37]	82.23±0.81	-	90.19±0.96	-	93.85±0.69	-
	TSR[38]	86.00±0.83	-	92.41±0.74	-	95.32±0.75	-
LDA	fisherface[31]	80.17±1.00	-	82.50±0.98	-	76.36±1.10	-
	USSL[13]	78.30±1.56	10^2	87.95±1.03	10^2	91.52±1.01	10^2
	GSR[22]	83.90±1.41	0.02	91.62±0.92	0.02	94.67±0.71	0.02
	GJGS	**87.15±1.25**	$10,10^{-3}$	**93.43±1.18**	$10,10^{-3}$	**95.98±1.52**	$1,10^{-2}$
	MSR[23]	85.03±1.44	0.02	92.50±0.79	0.02	95.48±0.60	0.02
	MJGS	85.81±1.29	$1,10^{-2}$	93.09±1.84	$1,10^{-2}$	95.89±2.01	$1,\ 10^{-2}$
	MFA[9]	82.90±1.40	-	91.37±0.97	-	95.03±0.73	-
	GSR	84.07±1.39	1	91.62±0.70	1	94.57±0.62	1
MFA	GJGS	**87.07±1.24**	$5,10^{-3}$	**93.45±1.22**	$10^{-1}, 10^{-2}$	**96.51±1.07**	$5,10^{-3}$
	MSR	84.58±1.31	1	91.51±0.94	1	94.78±0.72	10
	MJGS	85.71±2.06	$5, 10^{-3}$	93.09±1.23	$10^{-1}, 10^{-1}$	95.81±0.92	$5, 10^{-3}$

(f)

Group	Method	$p=5$		$p=10$		$p=20$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	37.46±1.22	-	53.08±1.17	-	69.62±1.13	-
	LPP[8]	75.13±1.61	-	86.68±1.24	-	90.94±0.67	-
	CRC[37]	78.08±1.72	-	90.08±0.82	-	96.41±0.47	-
	TSR[38]	76.50±2.06	-	89.20±0.89	-	95.69±0.51	-
LDA	fisherface[31]	76.70±1.16	-	86.72±1.03	-	86.69±0.88	-
	USSL[13]	71.74±2.35	10^2	84.66±1.47	10^2	92.58±0.86	10^2
	GSR[22]	75.34±1.37	0.01	88.02±1.17	0.02	95.63±0.69	0.02
	GJGS	**77.65±0.51**	$5,10^{-3}$	**89.70±0.74**	$10,10^{-3}$	**96.28±5.04**	$1,10^{-2}$
	MSR[23]	74.63±2.23	0.02	82.98±0.99	0.02	94.37±0.60	0.02
	MJGS	76.17±0.88	$1,10^{-2}$	84.45±1.72	$1,10^{-2}$	95.39±3.57	$1,\ 10^{-2}$
	MFA[9]	74.98±1.88	-	89.10±1.20	-	96.12±0.59	-
	GSR	75.96±1.38	0.1	88.60±1.19	0.2	95.90±0.66	0.2
MFA	GJGS	**77.81±1.55**	$10^{-1}, 10^{-2}$	**89.89±0.98**	$5, 10^{-3}$	**96.51±0.70**	$1,10^{-2}$
	MSR	72.15±1.96	0.2	82.51±1.50	1	94.19±0.76	1
	MJGS	76.72±2.38	$10, 10^{-3}$	84.19±1.20	$10^{-1}, 10^{-1}$	95.44±0.55	$10^{-2}, 1$

(g)

Group	Method	$p=5$		$p=10$		$p=20$	
		Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
	PCA[1]	59.11±1.21	-	77.04±1.21	-	90.97±0.91	-
	LPP[8]	92.98±0.81	-	95.05±0.43	-	95.95±0.46	-
	CRC[37]	94.36±0.80	-	96.38±0.32	-	97.33±0.37	-
	TSR[38]	94.42±0.73	-	97.05±0.29	-	98.24±0.31	-
LDA	fisherface[31]	92.84±0.86	-	94.55±0.38	-	95.86±0.47	-
	USSL[13]	91.43±1.56	10^2	95.34±1.03	10^2	96.95±0.36	10^2
	GSR[22]	93.09±0.70	0.01	96.08±0.44	0.02	97.49±0.47	0.02
	GJGS	**94.59±1.43**	$10,10^{-3}$	**96.80±2.37**	$1,10^{-2}$	**97.98±1.84**	$1,10^{-2}$
	MSR[23]	92.77±0.82	0.02	96.42±0.34	0.02	97.78±0.29	0.02
	MJGS	93.45±1.29	$1,10^{-2}$	96.72±1.82	$1,10^{-2}$	97.91±2.46	$5,\ 10^{-2}$
	MFA[9]	94.07±0.80	-	96.81±0.32	-	97.83±0.28	-
	GSR	93.28±0.76	0.2	96.12±0.39	1	97.62±0.32	1
MFA	GJGS	**94.64±1.70**	$5,\ 10^{-3}$	**96.89±0.34**	$10^{-1}, 10^{-1}$	**98.08±0.27**	$10^{-2}, 1$
	MSR	92.35±0.84	0.2	96.32±0.37	1	97.71±0.30	1
	MJGS	93.31±0.86	$1, 10^{-2}$	96.79±0.26	$10^{-1}, 10^{-1}$	98.04±0.27	$10^{-2}, 1$

feature dimensions of MFA and its variants. Moreover, the comparison of test time among GJGS, PCA, CRC and TSR is shown in Table 2, the time is obtained by Matlab running on a 2.8 GHz Intel Core. Since all the methods in LDA and MFA groups adopt the nearest centroid classifier, and LPP also uses this classifier, they have the same test time when they have the same reduced dimension. For brief, we just present the test time of GJGS. The performance of all algorithms shown in Fig. 3, Fig. 4 and Table 2 is on COIL20, UMIST, ORL, USPS, AR, Extended Yale-B and PIE databases with $p = 6$, $p = 8$, $p = 5$, $p = 10$, $p = 4$, $p = 5$ and $p = 5$ respectively. From Tables 1-2, Fig. 3 and Fig. 4, we could have the following observations:

1. In most cases, sparse regression method can significantly improve the classification performance of LDA, especially for the large number of training data, while it cannot improve the classification accuracy of MFA. The reason may be that sparse regression method has upper limit of classification accuracy with a certain number of training data, which can be inferred from the

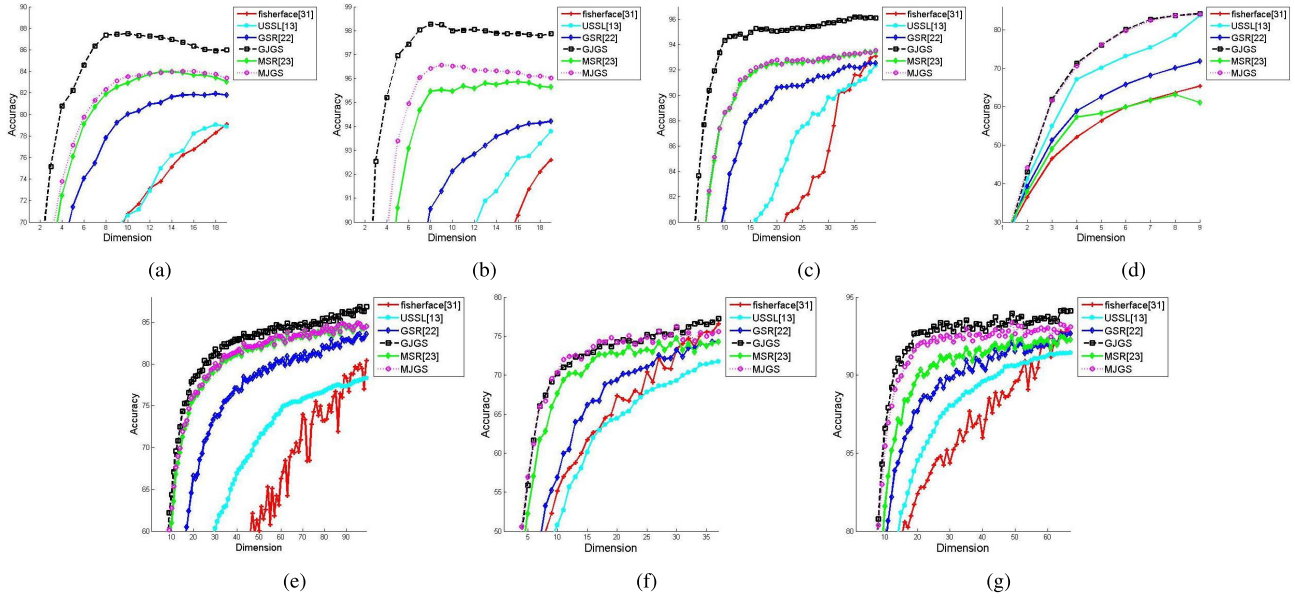


Fig. 3. Recognition accuracy vs. dimension for LDA and its variants on six standard databases: (a) COIL20 (p=6), (b) UMIST (p=8), (c) ORL (p=5), (d) USPS (p=10), (e) AR (p=4), (f) Extended Yale-B (p=5), (g) PIE (p=5).

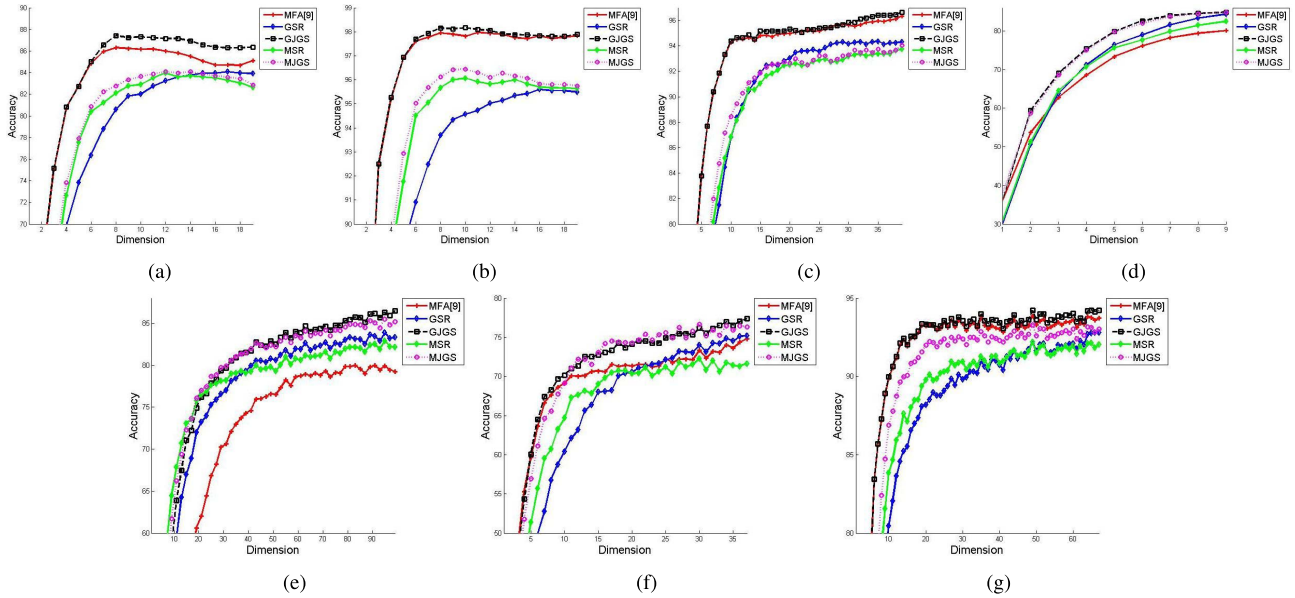


Fig. 4. Recognition accuracy vs. dimension for MFA and its variants on six standard databases: (a) COIL20 (p=6), (b) UMIST (p=8), (c) ORL (p=5), (d) USPS (p=10), (e) AR (p=4), (f) Extended Yale-B (p=5), (g) PIE (p=5).

TABLE II

TEST TIME (ms) FOR A SINGLE IMAGE ON SEVEN TYPICAL DATABASES

Method	COIL20	UMIST	ORL	USPS	AR	Yale-B	PIE
PCA[1]	0.24	0.28	0.51	0.14	2.97	0.40	1.82
CRC [37]	0.27	0.27	0.67	0.14	1.98	0.63	1.05
TSR [38]	9.42	10.47	20.08	7.18	47.33	20.98	33.96
GJGS	**0.037**	**0.038**	**0.079**	**0.020**	**0.20**	**0.075**	**0.14**

comparison of the classification accuracy of the sparse method on LDA and MFA.

2. For LDA, sparse regression via $L_{2,1}$-norm regularization can get better recognition performance than sparse regression via L_1-norm regularization in most cases, the main reason is that $L_{2,1}$-norm regularization can perform feature selection and subspace learning simultaneously, which encourages row-sparsity, but the selected features by L_1-norm regularization are independent and generally different for each dimension of the subspace.

TABLE III

RECOGNITION RESULTS (MEAN ACCURACY±STANDARD DEVIATION %) OF LDA, MFA AND THEIR VARIANTS FOR SEMI-SUPERVISE LEARNING ON THREE STANDARD DATABASES. (a) COIL20, (b) UMIST, (c) ORL

(a)

Method	$p=1$		$p=2$		$p=3$	
	Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
fisherface[31]	60.77±2.53	-	68.95±3.24	-	72.97±2.16	-
MFA[9]	-	-	73.17±2.92	-	78.45±2.72	-
SDA[33]	77.37±2.24	$10^{-6}, 10^{3}$	80.18±2.68	$10^{-6}, 10^{3}$	82.35±2.14	$10^{-6}, 1$
SJGS(LDA)	**77.70±2.39**	$10^{3},\ 10^{3},\ 10^{-3}$	80.70±2.39	$10,\ 10^{3},\ 10^{-6}$	**83.22±1.63**	$1,\ 10^{2},\ 10^{-6}$
SJGS(MFA)	77.19±2.28	$10,\ 10^{6},\ 10^{-9}$	**81.47±2.23**	$10^{3},\ 10^{6},\ 10^{-3}$	82.42±2.17	$10^{3},\ 10^{6},\ 10^{-3}$

(b)

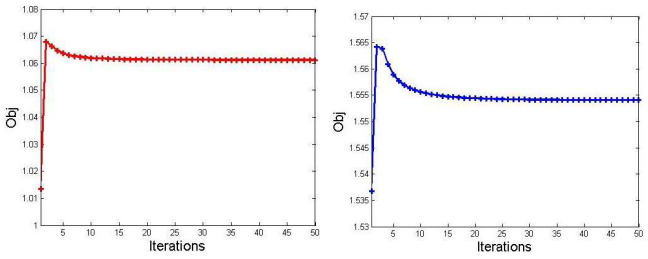
Method	$p=1$		$p=2$		$p=3$	
	Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy (%)	Parameters
fisherface[31]	44.86±5.34	-	63.72±3.40	-	75.39±3.82	-
MFA[9]	-	-	74.39±4.22	-	85.05±3.53	-
SDA[33]	58.87±3.64	$10^{-3}, 10^{-2}$	74.42±4.21	$10^{-3}, 10^{-9}$	84.42±3.42	$10^{-3}, 10^{-9}$
SJGS(LDA)	**61.14±4.84**	$10^{2},\ 10^{2},\ 10^{2}$	76.05±5.80	$1,\ 10^{3},\ 10^{-3}$	87.35±2.49	$10^{-2},\ 10,\ 10^{-6}$
SJGS(MFA)	60.14±5.84	$10^{6},\ 10^{9},\ 10$	**78.00±3.29**	$10^{-3}, 10^{-2}, 10^{-6}$	**87.61±3.10**	$10^{-3}, 10^{-2}, 10^{-6}$

(c)

Method	$p=1$		$p=2$		$p=3$	
	Accuracy (%)	Parameters	Accuracy (%)	Parameters	Accuracy(%)	Parameters
fisherface[31]	56.20±4.50	-	79.75±3.64	-	86.83±2.57	-
MFA[9]	-	-	82.10±3.25	-	90.69±2.93	-
SDA[33]	60.33±4.82	$10^{-3}, 10^{-2}$	82.22±3.68	$10^{-3}, 10^{-3}$	90.85±2.74	$10^{-3}, 10^{-9}$
SJGS(LDA)	**64.32±4.12**	$1, 10^{-2}, 1$	**82.98±3.08**	$10^{-3}, 10^{-2}, 10^{-6}$	**91.25±2.09**	$10^{-1}, 10^{-1}, 10^{-2}$
SJGS(MFA)	61.45±2.69	$10^{-3}, 10^{-9}, 10^{-2}$	82.25±4.11	$10^{-1}, 10^{-3}, 1$	90.84±2.46	$10^{-1}, 10^{-3}, 1$

3. For both LDA and MFA, joint graph embedding and sparse regression method outperforms the sparse regression method, which demonstrates that the adaptively changed graph spectral method, which learns graph spectral vectors and the optimal sparse regression vectors simultaneously, outperforms fixed graph spectral method. The reason is that the sparse regression vectors can affect the graph defined on the manifold data in joint graph embedding and sparse regression method [24].
4. For the sparse regression method GSR and MSR, there is no consistent winner on all seven databases. However, for the joint graph embedding and sparse regression method on both LDA and MFA, GJGS has better classification performance than MJGS, which suggests that GJGS is the superior choice for joint graph embedding and sparse regression method. Probably because MJGS is slightly overtraining as the training data are included for obtaining the graph spectral.
5. Our proposed methods GJGS (LDA), MJGS (LDA), GJGS (MFA) and MJGS (MFA) always have better classification accuracy than LPP on seven databases. Moreover, the proposed methods could obtain better classification results than PCA, CRC and TSR in most cases, especially for low-dimension. Furthermore, Table 2 illustrates that GJGS requires less test time than PCA, CRC and TSR, which further demonstrates the superior performance of the proposed methods.

3) Convergence Results: When we run the experiments, we record the objective value for each iteration. Since the convergence process on different databases is similar, for brief, we just show the convergence process on Extended Yale-B database in Fig 5, which shows the convergence results of GJGS and MJGS on LDA as p=5. Fig. 5 shows that GJGS and MJGS can converge fast, usually within 10 iterations, which is mainly because the algorithm has the closed form solution in each iteration. For MFA, similar iteration process could be found. Generally, we set the number of iterations to 20 for obtaining better and stable classification results.



(a) (b)

Fig. 5. The convergence results of two methods on Extended Yale-B database (p=5). (a) GJGS, (b) MJGS.

B. Semi-Supervised Learning Experiments

As shown in Table 1, the joint graph embedding and sparse method is effective for both preprocessing methods, and consistently outperforms the sparse regression method. To demonstrate the effectiveness of joint graph embedding and sparse regression method for semi-supervised learning

algorithms, we just present the recognition performance of applying SJGS to LDA and MFA on COIL20, UMIST and ORL databases. Here, we scale features to [0, 1]. In addition, we compare SJGS(LDA) and SJGS(MFA) with LDA, MFA and SDA [33] to illustrate recognition performance better. More precisely, the weight matrix W in the regularizer $J(A)$ adopted for SJGS(LDA) and SJGS(MFA) is defined as

$$W = \begin{cases} e^{\frac{\|x_i - x_j\|_2^2}{2}} & if\ x_j \in N_k(x_i)\ or\ x_i \in N_k(x_j) \\ 0 & otherwise \end{cases} \quad (39)$$

where $N_k(x_i)$ denotes the set of k nearest neighbors of x_i.

1) Parameter Setting: We randomly select 50% data as training data and the remained 50% data are used for testing. Among the training data, we randomly label p samples per class and the other samples are treated as unlabeled samples. We run the algorithms 20 times and calculate the mean accuracy.

In all the algorithms, three data sets are preprocessed by scaling features to [0, 1], and perform PCA to reduce the dimensionality of the data to $n - c$ before performing classification. For MFA, SDA, SJGS(LDA) and SJGS(MFA), we fix k=5. For fair comparison, the parameters μ and β of SDA and the parameters μ, β and γ in SJGS(LDA) and SJGS(MFA) are tuned by searching the grid $[10^{-9}, 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^6, 10^9]$, and then we report the best recognition accuracy and their corresponding parameters. Besides, all methods adopt the nearest centroid classifier for classification.

2) Results and Observations: In Table 3, the results shown in boldface are better than the others. Our observations are listed as follows:

1. From Table 3, we can see that MFA cannot be applied to the case in which p=1 due to L=0.
2. SJGS(LDA) and SDA, and SJGS(MFA) outperforms LDA and MFA in all the cases in term of mean recognition accuracy, respectively, because the unlabeled data can be used to improve the recognition performance.
3. In most cases, SJGS(LDA) has better recognition accuracy than SDA and LDA, and SJGS(MFA) has superior recognition performance than MFA, which demonstrates the effectiveness of joint graph embedding and sparse regression method for semi-supervised learning.

V. Conclusion

In this paper, we propose a framework of joint graph embedding and sparse regression for dimensionality reduction, which can unify most of popular dimensionality reduction algorithms including supervised, semi-supervised and unsupervised learning. Experiments on seven standard databases demonstrate the effectiveness of joint graph embedding and sparse regression method. The supervised learning experiments illustrate that joint graph embedding and sparse regression method with varied graph spectral outperforms sparse regression method with fixed graph spectral, and it can significantly improve the recognition performance; the experiments on semi-supervised learning also indicate that joint graph embedding and sparse regression method can obtain superior performance in recognition tasks. Due to the space limitation, in this paper, we just apply joint graph embedding and sparse regression method with the L_2-norm and $L_{2,1}$-norm regularization to two classic linear algorithms LDA and MFA. Many other algorithms such as PCA, ISOMAP, LPP or kernel methods, and the regularization with different norms are also worth studying based on the proposed framework. Therefore, in the future work, we will study the framework with different norms and apply this framework to other dimensionality reduction algorithms.

References

[1] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[2] A. M. Martínez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[3] J. Ye, R. Janardan, C. H. Park, and H. Park, "An optimization criterion for generalized discriminant analysis on undersampled problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 982–994, Aug. 2004.

[4] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[6] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[7] M. Belkin, and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2001, pp. 585–591.

[8] 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.

[9] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[10] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum, "Concurrent subspaces analysis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 203–208.

[11] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 526–532.

[12] M. Brand, "Continuous nonlinear dimensionality reduction by kernel eigenmaps," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 547–554.

[13] D. Cai, X. He, and J. Han, "Spectral regression: A unified approach for sparse subspace learning," in *Proc. 7th IEEE Int. Conf. Data Mining*, Oct. 2007, pp. 73–82.

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

[15] H. Zou and T. Hastie, "Regression shrinkage and selection via the elastic net, with applications to microarrays," *J. Roy. Statist. Soc., Ser. B (Statistical Methodology)*, vol. 67, pp. 301–320, Dec. 2003.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., Ser. B (Statistical Methodology)*, vol. 58, no. 1, pp. 267–288, 1996.

[17] W. J. Krzanowski, "Selection of variables to preserve multivariate data structure, using principal component analysis," *Appl. Statist.*, vol. 36, no. 1, pp. 22–33, 1987.

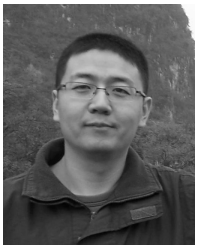
[18] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2005, pp. 507–514.

[19] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.

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

[21] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 673–678.
[22] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1294–1299.
[23] X. Shi, Y. Yang, Z. Guo, and Z. Lai, "Face recognition by sparse discriminant analysis via joint $L_{2,1}$-norm minimization," *Pattern Recognit.*, vol. 47, no. 7, pp. 2447–2453, 2014.
[24] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1324–1329.
[25] 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.
[26] C. Ding, D. Zhou, X. He, and H. Zha, "R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 281–288.
[27] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran & Associates Inc., 2010, pp. 1813–1821.
[28] C. Ding, "A new robust function that smoothly interpolates between L_1 and L_2 error functions," Dept. Comput. Sci. Eng., Texas Univ., Austin, TX, USA, Tech. Rep., 2013.
[29] F. Nie, H. Wang, H. Huang, and C. Ding, "Adaptive loss minimization for semi-supervised elastic embedding," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1565–1571.
[30] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proc. Nat. Acad. Sci. USA*, vol. 97, no. 18, pp. 10101–10106, 2000.
[31] 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.
[32] F. Nie, H. Wang, H. Huang, and C. Ding, "Early active learning via robust representation and structured sparsity," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1572–1578.
[33] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–7.
[34] Z. Zheng, "Sparse locality preserving embedding," in *Proc. 2nd Int. Congr. Image Signal Process.*, 2009, pp. 1–5.
[35] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[36] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
[37] D. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.
[38] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.
[39] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 553–560.
[40] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (Coil-20)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep., Feb. 1996.
[41] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
[42] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in *Face Recognition: From Theory to Applications* (NATO ASI Series F). New York, NY, USA: Springer-Verlag, 1998, pp. 446–456.
[43] F. S. Samaria, and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 1994, pp. 138–142. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed Feb. 21, 2015.
[44] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, 1994. [Online]. Available: http://www.cs.toronto.edu/~roweis/data.html, accessed Feb. 21, 2015.
[45] D. Cai, X. He, and J. Han, "Speed up kernel discriminant analysis," *VLDB J.*, vol. 20, no. 1, pp. 21–33, 2011.
[46] A. Martinez and R. Benavente. (2003). *The AR Face Database*. [Online]. Available: http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html, accessed Feb. 21, 2015.
[47] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[48] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2013.
[49] L. Devroye, L. Györfi, and G. Lugosi, "A probabilistic theory of pattern recognition," *Stochastic Modelling and Applied Probability*, vol. 31. New York, NY, USA: Springer-Verlag, 1996.
[50] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.
[51] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 10, pp. 6567–6572, 2002.

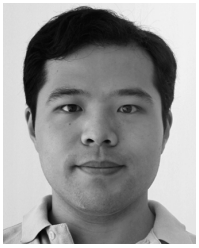
Xiaoshuang Shi received the B.S. degree in automation from Northwestern Polytechnical University, China, in 2009, and the M.S. degree in automation form Tsinghua University, China, in 2013. He is currently a Research Assistant with the Shenzhen Key Laboratory of Broadband Network and Multimedia, Graduate School at Shenzhen, Tsinghua University, China. His current research interests include pattern recognition and machine learning.



Zhenhua Guo received the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology and The Hong Kong Polytechnic University, in 2004 and 2010, respectively. Since 2010, he has been with the Graduate School at Shenzhen, Tsinghua University. His research interests include pattern recognition, texture classification, biometrics, and video surveillance.

Zhihui Lai received the B.S. degree in mathematics from South China Normal University, in 2002, the M.S. degree from Jinan University, in 2007, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NUST), China, in 2011. He has been a Research Associate with The Hong Kong Polytechnic University since 2010. He is currently an Associate Professor with Shenzhen University, China. He has authored over 30 scientific papers in pattern recognition and computer vision. His research interests include face recognition, image processing, content-based image retrieval, pattern recognition, compressive sense, human vision modernization, and applications in the fields of intelligent robot research.

Yujiu Yang received the Ph.D. degree in pattern recognition from the Institute of Automation, Chinese Academy of Sciences, in 2008. He is currently a Lecturer with the Graduate School at Shenzhen, Tsinghua University. His current research interests include statistical learning theory, web data mining, big data analytics, and machine learning. He has served as a member of the IEEE Computer Society and the Association for Computing Machinery from 2008.



Zhifeng Bao received the Ph.D. degree in computer science from the National University of Singapore, in 2011. He is currently an Assistant Professor with the School of Engineering and ICT, University of Tasmania, Australia. He is also a Faculty Member with the Human Interaction Technology Laboratory Australia. He was a recipient of the Best Ph.D. Thesis Award from the School of Computing, and was the winner of the Singapore Infocomm Development Authority Gold Medal. He has committed to the task of how to make data usable, and enhance data and knowledge sharing over the social network. His data usability works span across heterogeneous data, including structured data (e.g., relational data), semistructured data, unstructured text data, spatial data, multimedia data, and graph data (e.g., social network). He focused on building general yet efficient frameworks to support these usability modules, without breaking the traditional storage and indexing scheme for the underlying data.

David Zhang (F'08) received the degree in computer science from Peking University, the M.Sc. degree in computer science and the Ph.D. degree from the Harbin Institute of Technology (HIT), in 1982 and 1985, respectively, and the second Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 1994. From 1986 to 1988, he was a Post-Doctoral Fellow with Tsinghua University and an Associate Professor with Academia Sinica, Beijing. He is currently a Chair Professor with The Hong Kong Polytechnic University, where he is the Founding Director of the Biometrics Technology Centre supported by the Hong Kong Government in 1998. He also serves as a Visiting Chair Professor with Tsinghua University, and an Adjunct Professor with Shanghai Jiao Tong University, Peking University, HIT, and the University of Waterloo. He has authored over 10 books and 300 journal papers. He is a Croucher Senior Research Fellow, Distinguished Speaker of the IEEE Computer Society, and fellow of the International Association for Pattern Recognition. He is also the Founder and Editor-in-Chief of the *International Journal of Image and Graphics*, a Book Editor of *International Series on Biometrics* (Springer), an Organizer of the first International Conference on Biometrics Authentication, an Associate Editor of over 10 international journals, including the IEEE TRANSACTIONS and *Pattern Recognition*, and the Technical Committee Chair of the IEEE Computational Intelligence Society.