

# Constructing a Non-Negative Low Rank and Sparse Graph with Data-Adaptive Features

Liansheng Zhuang, Shenghua Gao, Jinhui Tang, Jingjing Wang,  
Zhouchen Lin, *Senior Member*, and Yi Ma, *IEEE Fellow*,

**Abstract**—This paper aims at constructing a good graph for discovering intrinsic data structures in a semi-supervised learning setting. Firstly, we propose to build a non-negative low-rank and sparse (referred to as NNLS) graph for the given data representation. Specifically, the weights of edges in the graph are obtained by seeking a nonnegative low-rank and sparse matrix that represents each data sample as a linear combination of others. The so-obtained NNLS-graph can capture both the global mixture of subspaces structure (by the low rankness) and the locally linear structure (by the sparseness) of the data, hence is both generative and discriminative. Secondly, as good features are extremely important for constructing a good graph, we propose to learn the data embedding matrix and construct the graph jointly within one framework, which is termed as NNLS with embedded features (referred to as NNLS-EF). Extensive experiments on three publicly available datasets demonstrate that the proposed method outperforms the state-of-the-art graph construction method by a large margin for both semi-supervised classification and discriminative analysis, which verifies the effectiveness of our proposed method.

**Index Terms**—Graph Construction, Low Rank and Sparse Representation, Semi-Supervised Learning, Data Embedding.

## I. INTRODUCTION

IN many big data related applications, e.g., image based object recognition, one often lacks sufficient labeled training data which can be cost and time-prohibitive to obtain. On the other hand, a large number of unlabeled data are widely available, e.g., from the Internet. Semi-supervised learning (SSL) can utilize both labeled samples and richer yet unlabeled samples. Recently, it has received considerable attention in both computer vision and machine learning communities [2]. Among current SSL methods, graph based SSL is particularly appealing due to its success in practice and its computational efficiency.

A fundamental problem in graph based SSL is the construction of a graph from the observed data, which represents the underlying data structures. Graph based SSL methods treat both labeled and unlabeled samples from the dataset as nodes

in a graph, and then instantiate edges among these nodes which are weighted by the affinity between the corresponding pairs of samples. Label information of the labeled samples can then be efficiently and effectively propagated to the unlabeled data over the graph. Most learning methods formalize the propagation process through a regularized functional on the graph. Despite many forms used in the literature, the regularizers mainly try to accommodate the so-called *cluster assumption* [3], [4], i.e., points on the same low-dimensional smooth structure (such as a cluster, a subspace, or a manifold) are likely to share the same label. Since one normally does not have (or care about) an explicit parametric model for the underlying manifolds, many methods approximate them by constructing an undirected graph from the observed data points. Therefore, correctly constructing a graph that can best capture the essential data structures is critical for all graph based SSL methods [5]–[8].

Lots of efforts have been made to exploit ways of constructing a good graph for SSL [9]–[12]. According to Wright [9], an informative graph should have three characteristics: high discriminating power, low sparsity and adaptive neighborhood. Guided by these rules, lots of sparse representation (SR) based graph construction methods have been proposed [13]–[16]. However, these SR based methods usually do not characterize the global structure of data. To overcome this drawback, Liu *et al.* propose low-rank representation (LRR) to compute simultaneously the weights in an undirected graph (referred to as LRR-graph hereafter) that represent the affinities among all the data samples [17], [18]. But LRR usually results in a dense graph, and the negative values in LRR are not physical meaningful for constructing an affinity graph.

On the one hand, by understanding the characteristics of an informative graph and both the advantages and disadvantages of previous works, in this paper we propose to harness both sparsity and low rankness of high-dimensional data to construct an informative graph. In addition, we will explicitly enforce the representation to be non-negative so that coefficients of the representation can be directly converted to graph weights. Such a graph is called *nonnegative low-rank and sparse graph* (NNLS-graph). Specifically, given a set of data points we represent a data point as a linear combination of other points, where the coefficients should be both nonnegative and sparse. Nonnegativity ensures that every data point is in the convex hull of its neighbors, while sparsity ensures that the involved neighbors are as few as possible. Moreover, to make data vectors on the same subspace be clustered into the same cluster, we enforce the matrix constructed by the coefficient

A preliminary version of this paper was published in CVPR2012 [1]. This work is supported by the National Science Foundation of China (No.s 61371192 and 61103134), and the Science Foundation for Outstanding Young Talent of Anhui Province (BJ2101020001). Z. Lin is also supported by National Science Foundation of China (No.s 61272341, 61231002 and 61121002).

Liansheng Zhuang, and Jingjing Wang are with University of Science and Technology of China, Hefei 230027, China. Shenghua Gao and Yi Ma are with ShanghaiTech University, Shanghai, China. Zhouchen Lin is with Peking University, Beijing, China. Jinhui Tang is with Nanjing University of Science and Technology, Jiangsu, China. Liansheng Zhuang is the corresponding author (lszhuang@ustc.edu.cn)

vectors of all data points to be low-rank. On the other hand, previous works [19], [20] have shown that by projecting the data with PCA, the embedded data will greatly facilitate the subsequent sparse representation and improve the classification accuracy. Therefore, finding a good data embedding strategy is also very important for the sparse representation. However, these previous works do data embedding [19], [20] and the subsequent sparse representation separately. Therefore, the learnt features may not optimize the subsequent representation. Realizing that a good data representation is important for the good performance of graph construction and the possible improvement space [19], [20], we propose to simultaneously learn the data embedding matrix and graph, which further improves the performance of semi-supervised classification.

The contributions of this paper can be summarized as follows:

- We propose to learn an NNLSR-graph for SSL. The sparsity property ensures the NNLSR-graph to be sparse and capture the local low-dimensional linear structures of the data. The low-rank characteristic guarantees that the NNLSR-graph can better capture the global cluster or subspace structures of the data than SR based graphs [15], [16]. Thus the robustness of NNLSR-graph to noise and outliers can be enhanced.
- We propose to simultaneously learn the data embedding and graph. Such a strategy learns a better data representation, which is more suitable for building an NNLSR-graph, and consequently enhances the performance of semi-supervised classification.

Extensive experiments demonstrate that the NNLSR-graph significantly improves the performance of SSL – often reducing the error rates by multiple folds!

This article extends its preliminary version [1] in terms of both technique and performance evaluation. First, we extend our NNLSR framework by jointly learning the data embedding and NNLSR-graph, which enhances the robustness of NNLSR-graph for data analysis. Second, we conduct more experiments to evaluate the proposed algorithms. Third, more details about our methods are provided and the influence of different parameters is also discussed in the paper.

The rest of this paper is organized as follows. In Section II, we briefly review works related to graph construction in SSL. In Section III, we detail the construction of NNLSR-graph, and in Section IV, we extend NNLSR by simultaneously learning the data embedding and NNLSR-graph. We present experiments and analysis in Section V. Finally, we conclude our paper in Section VI.

## II. RELATED WORK

**Euclidean distance based methods.** Conceptually, a good graph should reveal the intrinsic complexity or dimensionality of data (say through local linear relationship) and also capture certain global structures of data as a whole (i.e., multiple clusters, subspaces, or manifolds). Traditional methods (such as  $k$ -nearest neighbors and Locally Linear Reconstruction [11]) mainly rely on pair-wise Euclidean distances and construct a graph by a family of overlapped local patches. The so-obtained

graph only captures the local structures and cannot capture the global structures of the whole data (i.e. the clusters). Moreover, these methods cannot produce data-adaptive neighborhoods because of using fixed global parameters to determinate the graph structure and their weights. Finally, these methods are sensitive to local data noise and errors.

**Sparse representation based methods:** As pointed out in [9], sparsity is an important characteristics for an informative graph. Therefore, lots of researchers propose to improve the robustness of graph by enforcing sparsity. Specifically, Yan et al. [13], [14] proposed to construct an  $\ell_1$ -graph via sparse representation (SR) [19] by solving an  $\ell_1$  optimization problem. An  $\ell_1$ -graph over a data set is derived by encoding each sample as a sparse representation of the remaining samples, and automatically selecting the most informative neighbors for each sample. The neighborhood relationship and graph weights of an  $\ell_1$ -graph are simultaneously obtained during the  $\ell_1$  optimization in a parameter-free way. Different from traditional methods, an  $\ell_1$ -graph explores higher order relationships among more data points, hence is more powerful and discriminative. Benefitting from SR, the  $\ell_1$ -graph is sparse, data-adaptive and robust to data noise. Following  $\ell_1$ -graph, other graphs have also been proposed based on SR in recent years [15], [16]. However, all these SR based graphs find the sparsest representation of each sample *individually*, lacking global constraints on their solutions. So these methods may be ineffective in capturing the global structures of data. This drawback may reduce the performance when the data are grossly corrupted. When not enough “clean data” are available, SR based methods may not be robust to noise and outliers [17].

**Low-rank representation based methods.** To capture the global structure of data, Liu *et al.* propose low-rank representation (LRR) for data representation and use it to construct the affinities of an undirected graph (hereafter called LRR-graph) [17], [18]. An LRR-graph jointly obtains the representation of all data under a global low-rank constraint, thus is better at capturing the global data structures (such as multiple clusters and subspaces). It has been proven that, under mild conditions, LRR can correctly preserve the membership of samples that belong to the same subspace. However, compared to the  $\ell_1$ -graph, LRR often results in a dense graph (see Figure 2), which is undesirable for graph-based SSL [9]. Moreover, as the coefficients can be negative, LRR allows the data to “cancel out each other” by subtraction, which lacks physical interpretation for most visual data. In fact, non-negativity is more consistent with the biological modeling of visual data [21], [22], and often leads to better performance for data representation [22] and graph construction [15].

## III. NONNEGATIVE LOW-RANK AND SPARSE GRAPHS

### A. Nonnegative Low-Rank and Sparse Representation

Let  $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$  be a matrix whose columns are  $n$  data samples drawn from independent subspaces<sup>1</sup>. Then each column can be represented by a linear

<sup>1</sup>The subspaces  $S_1, \dots, S_k$  are independent if and only if  $\sum_{i=1}^k S_i = \bigoplus_{i=1}^k S_i$ , where  $\bigoplus$  is the direct sum.

combination of bases in dictionary  $A = [a_1, a_2, \dots, a_m]$ :

$$X = AZ, \quad (1)$$

where  $Z = [z_1, z_2, \dots, z_n]$  is the coefficient matrix with each  $z_i$  being the *representation* of  $x_i$ . The dictionary  $A$  is often overcomplete. Hence there can be infinitely many feasible solutions to problem (1). To address this issue, we impose the *most sparsity* and *lowest rank* criteria, as well as a nonnegative constraint. That is, we seek a representation  $Z$  by solving the following optimization problem

$$\min_Z \text{rank}(Z) + \beta \|Z\|_0, \quad \text{s.t. } X = AZ, Z \geq 0, \quad (2)$$

where  $\beta > 0$  is a parameter to trade off between low rankness and sparsity. As observed in [17], the *low rankness* criterion is better at capturing the global structure of data  $X$ , while the *sparsity* criterion can capture the local structure of each data vector. The optimal solution  $Z^*$  is called the nonnegative ‘‘lowest-rank and sparsest’’ representation (NNLRSR) of data  $X$  with respect to the dictionary  $A$ . Each column  $z_i^*$  in  $Z^*$  reveals the relationship between  $x_i$  and the bases in dictionary.

However, solving problem (2) is NP-hard. As a common practice (e.g., [23]) we may solve the following relaxed convex program instead

$$\min_Z \|Z\|_* + \beta \|Z\|_1, \quad \text{s.t. } X = AZ, Z \geq 0, \quad (3)$$

where  $\|\cdot\|_*$  is the nuclear norm of a matrix [24], i.e., the sum of the singular values of the matrix, and  $\|\cdot\|_1$  is the  $\ell_1$ -norm of a matrix, i.e., the sum of the absolute value of all entries in the matrix.

In real applications, the data are often noisy and even grossly corrupted. So we have to add a noise term  $E$  to (1). If a fraction of the data vectors are grossly corrupted, we may reformulate problem (3) as

$$\begin{aligned} \min_{Z,E} \|Z\|_* + \beta \|Z\|_1 + \lambda \|E\|_{2,1}, \\ \text{s.t. } X = AZ + E, Z \geq 0, \end{aligned} \quad (4)$$

where  $\|E\|_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^m ([E]_{ij})^2}$  is called the  $\ell_{2,1}$ -norm [25], and the parameter  $\lambda > 0$  is used to balance the effect of noise, which is set empirically. The  $\ell_{2,1}$ -norm encourages the columns of  $E$  to be zero, which assumes that the corruptions are ‘‘sample-specific’’, i.e., some data vectors are corrupted and the others are clean. For small Gaussian noise, we can relax the equality constraint in problem (2) as did in [26]. Namely, the Frobenious norm  $\|E\|_F$  is used instead. In this paper, we focus on the  $\ell_{2,1}$ -norm.

### B. LADMAP for Solving NNLRSR

The NNLRSR problem (4) could be solved by the popular alternating direction method (ADM) [17], [27]. However, ADM requires introducing two auxiliary variables when solving (4) and expensive matrix inversions are required in each iteration. So we adopt a recently developed method called the linearized alternating direction method with adaptive penalty (LADMAP) [28] to solve (4).

We first introduce an auxiliary variable  $H$  in order to make the objective function separable:

$$\begin{aligned} \min_{Z,H,E} \|Z\|_* + \beta \|H\|_1 + \lambda \|E\|_{2,1}, \\ \text{s.t. } X = AZ + E, Z = H, H \geq 0. \end{aligned} \quad (5)$$

The augmented Lagrangian function of problem (5) is

$$\begin{aligned} L(Z, H, E, Y_1, Y_2, \mu) \\ = \|Z\|_* + \beta \|H\|_1 + \lambda \|E\|_{2,1} + \\ \langle Y_1, X - AZ - E \rangle + \langle Y_2, Z - H \rangle + \\ \frac{\mu}{2} (\|X - AZ - E\|_F^2 + \|Z - H\|_F^2) \\ = \|Z\|_* + \beta \|H\|_1 + \lambda \|E\|_{2,1} + \\ q(Z, H, E, Y_1, Y_2, \mu) - \frac{1}{2\mu} (\|Y_1\|_F^2 + \|Y_2\|_F^2), \end{aligned} \quad (6)$$

where

$$\begin{aligned} q(Z, H, E, Y_1, Y_2, \mu) \\ = \frac{\mu}{2} (\|X - AZ - E + Y_1/\mu\|_F^2 + \|Z - H + Y_2/\mu\|_F^2) \end{aligned} \quad (7)$$

LADMAP is to update the variables  $Z$ ,  $H$  and  $E$  alternately, by minimizing  $L$  with other variables fixed, where the quadratic term  $q$  is replaced by its first order approximation at the previous iterate and a proximal term is then added [28]. With some algebra, the updating schemes are as follows.

$$\begin{aligned} Z_{k+1} &= \underset{Z}{\text{argmin}} \|Z\|_* \\ &\quad + \langle \nabla_Z q(Z_k, H_k, E_k, Y_{1,k}, Y_{2,k}, \mu_k), Z - Z_k \rangle \\ &\quad + \frac{\eta_1 \mu_k}{2} \|Z - Z_k\|_F^2 \\ &= \underset{Z}{\text{argmin}} \|Z\|_* + \frac{\eta_1 \mu_k}{2} \|Z - Z_k\|_F^2 \\ &\quad + [-A^T(X - AZ_k - E_k + Y_{1,k}/\mu_k) \\ &\quad + (Z_k - H_k + Y_{2,k}/\mu_k)]/\eta_1 \|Z\|_F^2 \\ &= \Theta_{(\eta_1 \mu_k)^{-1}}(Z_k + [A^T(X - AZ_k - E_k + Y_{1,k}/\mu_k) \\ &\quad - (Z_k - H_k + Y_{2,k}/\mu_k)]/\eta_1), \\ H_{k+1} &= \underset{H \geq 0}{\text{argmin}} \beta \|H\|_1 + \frac{\mu_k}{2} \|Z_{k+1} - H + Y_{2,k}/\mu_k\|_F^2 \\ &= \max(S_{\beta \mu_k^{-1}}(Z_{k+1} + Y_{2,k}/\mu_k), 0), \\ E_{k+1} &= \underset{E}{\text{argmin}} \lambda \|E\|_{2,1} \\ &\quad + \frac{\mu_k}{2} \|X - AZ_{k+1} - E + Y_{1,k}/\mu_k\|_F^2 \\ &= \Omega_{\lambda \mu_k^{-1}}(X - AZ_{k+1} + Y_{1,k}/\mu_k), \end{aligned} \quad (8)$$

where  $\nabla_Z q$  is the partial differential of  $q$  with respect to  $Z$ ,  $\Theta$ ,  $S$  and  $\Omega$  are the singular value thresholding [24], shrinkage [27] and the  $\ell_{2,1}$  minimization operator [17], respectively, and  $\eta_1 = \|A\|_2^2$ . The complete algorithm is outlined in **Algorithm 1**.

### C. Nonnegative Low Rank and Sparse Graph Construction

Given a data matrix  $X$ , we may use the data themselves as the dictionary, i.e.,  $A$  in subsections III-A and III-B is simply chosen as  $X$  itself. With the optimal coefficient matrix  $Z^*$ , we may construct a weighted undirected graph  $G = (V, E)$  associated with a weight matrix  $W = \{w_{ij}\}$ , where  $V = \{v_i\}_{i=1}^n$  is the vertex set, each node  $v_i$  corresponding to a data point  $x_i$ , and  $E = \{e_{ij}\}$  is the edge set, each edge  $e_{ij}$  associating nodes  $v_i$  and  $v_j$  with a weight  $w_{ij}$ . As the vertex set  $V$  is given, the problem of graph construction is to determine the graph weight matrix  $W$ .

Since each data point is represented by other samples, a column  $z_i^*$  of  $Z^*$  naturally characterizes how other samples contribute to the reconstruction of  $x_i$ . Such information is useful for recovering the clustering relation among samples. The sparse constraint ensures that each sample is associated with only a few samples, so that the graph derived from  $Z^*$  is

**Algorithm 1** Efficient LADMAP Algorithm for NNLSR**Input:** data matrix  $X$ , parameters  $\beta > 0, \lambda > 0$ **Initialize:**  $Z_0 = H_0 = E_0 = Y_{1,0} = Y_{2,0} = 0, \mu_0 = 0.1, \mu_{\max} = 10^{10}, \rho_0 = 1.1, \varepsilon_1 = 10^{-6}, \varepsilon_2 = 10^{-2}, \eta_1 = \|A\|_2^2, k = 0.$ 

- 1: **while**  $\|X - AZ_k - E_k\|_F / \|X\|_F \geq \varepsilon_1$  or  $\mu_k \max(\sqrt{\eta_1} \|Z_k - Z_{k-1}\|_F, \|H_k - H_{k-1}\|_F, \|E_k - E_{k-1}\|_F) / \|X\|_F \geq \varepsilon_2$  **do**
- 2: Update the variables as (8).
- 3: Update Lagrange multipliers as follows:

$$Y_{1,k+1} = Y_{1,k} + \mu_k(X - AZ_{k+1} - E_{k+1}).$$

$$Y_{2,k+1} = Y_{2,k} + \mu_k(Z_{k+1} - H_{k+1}).$$

- 4: Update  $\mu$  as follows:

$$\rho = \begin{cases} \mu_{k+1} = \min(\mu_{\max}, \rho\mu_k), \text{ where} \\ \rho_0, & \text{if } \mu_k \max(\sqrt{\eta_1} \|Z_{k+1} - Z_k\|_F, \\ & \|H_{k+1} - H_k\|_F, \|E_{k+1} - E_k\|_F) / \|X\|_F \\ & < \varepsilon_2, \\ 1, & \text{otherwise.} \end{cases}$$

- 5: Update  $k: k \leftarrow k + 1.$

6: **end while****Output:** an optimal solution  $(Z^*, H^*, E^*).$ 

naturally sparse. The low rank constraint guarantees that the coefficients of samples coming from the same subspace are highly correlated and fall into the same cluster, so that  $Z^*$  can capture the global structure (i.e. clusters) of the whole data. Note here that, since each sample can be used to represent itself, there always exist feasible solutions even when the data sampling is insufficient, which is different from SR.

After obtaining  $Z^*$ , we can derive the graph adjacency structure and graph weight matrix from it. In practice, due to data noise the coefficient vector  $z_i^*$  of point  $x_i$  is often dense with small values. As we are only interested in the global structure of the data, we can normalize the reconstruction coefficients of each sample (i.e.  $z_i^* = z_i^* / \|z_i^*\|_2$ ) and make those coefficients under a given threshold zeros. After that, we can obtain a sparse  $\hat{Z}^*$  and define the graph weight matrix  $W$  as

$$W = (\hat{Z}^* + (\hat{Z}^*)^T) / 2. \quad (9)$$

The method for constructing an NNLSR-graph is summarized in **Algorithm 2**.

#### IV. JOINTLY LEARNING DATA REPRESENTATION AND NNLSR-GRAPH

The quality of data representation will greatly affect the quality of graph. The data representation which is robust to the data variance improves the robustness of the graph, and subsequently improves the performance of SSL. To improve the data representation, lots of endeavors have been made [29], [30]. For face data, the commonly used data representation are EigenFaces [31], LaplacianFaces [32], FisherFaces [33], and RandomFaces [19]. As shown in [19], these representation

**Algorithm 2** Nonnegative low rank and sparse graph construction**Input:** Data matrix  $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n},$ regularization parameters  $\beta > 0$  and  $\lambda > 0$ , threshold  $\theta > 0.$ **Steps:**

- 1: Normalize all the samples  $\hat{x}_i = x_i / \|x_i\|_2$  to obtain  $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}.$
- 2: Solve the following problem using Algorithm 1,

$$\begin{aligned} \min_{Z,E} \quad & \|Z\|_* + \beta \|Z\|_1 + \lambda \|E\|_{2,1} \\ \text{s.t.} \quad & \hat{X} = \hat{X}Z + E, Z \geq 0 \end{aligned}$$

and obtain an optimal solution  $(Z^*, E^*).$ 

- 3: Normalize all column vectors of  $Z^*$  by  $z_i^* = z_i^* / \|z_i^*\|_2,$  make small values under the given threshold  $\theta$  zeros by

$$\hat{z}_{ij}^* = \begin{cases} z_{ij}^*, & \text{if } z_{ij}^* \geq \theta, \\ 0, & \text{otherwise,} \end{cases}$$

and obtain a sparse  $\hat{Z}^*.$ 

- 4: Construct the graph weight matrix  $W$  by

$$W = (\hat{Z}^* + (\hat{Z}^*)^T) / 2.$$

**Output:** The weight matrix  $W$  of NNLSR-graph.

strategies greatly improve the data representation quality, and improve the classification accuracy. However, the data embedding and the subsequent sparse representation are conducted separately in [19], and a data embedding method in the previous step may not be the most suitable for the subsequent sparse representation.

Other than doing the data embedding and learning the NNLSR graph separately, we propose to learn the data representation and graph simultaneously to make the learnt data representation more suitable for the construction of NNLSR-graph. We first denote the data projection matrix as  $P.$  Similar to [34], we want the projected data to preserve the data information as much as possible. So we aim at minimizing  $\|X - P^T P X\|_F^2.$  By plugging the learning of  $P$  into the NNLSR-graph construction framework, we arrive at the following formulation:

$$\begin{aligned} \min_{Z,E,P} \quad & \|Z\|_* + \beta \|Z\|_1 + \lambda \|E\|_{2,1} + \gamma \|X - P^T P X\|_F^2, \\ \text{s.t.} \quad & P X = P X Z + E, Z \geq 0, \end{aligned} \quad (10)$$

where  $\gamma$  is a parameter to balance the reconstruction error, which is set empirically. For simplification, we term this formulation as NNLSR with embedded feature (referred to as NNLSR-EF). However, the objective function of NNLSR-EF is not convex. Therefore it is inappropriate to optimize all the variables in problem (10) simultaneously. Following the commonly used strategy in dictionary learning [35], [36], we alternatively update the unknown variables. Specifically, we first optimize the above objective w.r.t.  $Z$  and  $E$  by fixing  $P,$  then we update  $P$  and  $E$  while fixing  $Z.$

When  $P$  is fixed, (10) reduces to

$$\begin{aligned} \min_{Z,E} \quad & \|Z\|_* + \beta \|Z\|_1 + \lambda \|E\|_{2,1} \\ \text{s.t.} \quad & P X = P X Z + E, Z \geq 0, \end{aligned} \quad (11)$$

We use LADMAP to solve for  $Z$  and  $E$ . By introducing an auxiliary variable  $H$ , we obtain the following augmented Lagrangian function:

$$\begin{aligned} & \tilde{L}(Z, H, E, Y_1, Y_2, \mu) \\ = & \|Z\|_* + \beta \|H\|_1 + \lambda \|E\|_{2,1} + \\ & \langle Y_1, PX - PXZ - E \rangle + \langle Y_2, Z - H \rangle + \\ & \frac{\mu}{2} (\|PX - PXZ - E\|_F^2 + \|Z - H\|_F^2) \\ = & \|Z\|_* + \beta \|H\|_1 + \lambda \|E\|_{2,1} + \\ & \tilde{q}_1(Z, H, E, Y_1, Y_2, \mu) - \frac{1}{2\mu} (\|Y_1\|_F^2 + \|Y_2\|_F^2), \end{aligned} \quad (12)$$

where

$$\begin{aligned} & \tilde{q}_1(Z, H, E, Y_1, Y_2, \mu) \\ = & \frac{\mu}{2} (\|PX - PXZ - E + Y_1/\mu\|_F^2 + \|Z - H + Y_2/\mu\|_F^2) \end{aligned} \quad (13)$$

Then we can apply Algorithm 1 to (11) by simply replacing  $X$  and  $A$  in Algorithm 1 with  $PX$ .

After updating  $Z$  and  $E$ , we only fix  $Z$ . So (10) reduces to the following problem

$$\begin{aligned} & \min_{E, P} \lambda \|E\|_{2,1} + \gamma \|X - P^T P X\|_F^2, \\ & \text{s.t. } PX = PXZ + E, Z \geq 0. \end{aligned} \quad (14)$$

We can solve the above problem with inexact ALM [27]. The augmented Lagrange function is

$$\begin{aligned} & \tilde{L}(P, E, \mu) \\ = & \lambda \|E\|_{2,1} + \gamma \|X - P^T P X\|_F^2 \\ & + \frac{\mu}{2} \|PX - PXZ - E\|_F^2 + \langle Y_1, PX - PXZ - E \rangle. \end{aligned} \quad (15)$$

By minimizing  $\tilde{L}(P, E, \mu)$  with other variables fixed, we can update the variables  $E$  and  $P$  alternately as follows<sup>2</sup>.

$$\begin{aligned} E_{k+1} &= \underset{E}{\operatorname{argmin}} \lambda \|E\|_{2,1} \\ & \quad + \frac{\mu_k}{2} \|P_k X - P_k X Z - E + Y_{1,k}/\mu_k\|_F^2 \\ &= \Omega_{\lambda/\mu_k^{-1}}(P_k X - P_k X Z + Y_{1,k}/\mu_k), \\ P_{k+1} &= \underset{P}{\operatorname{argmin}} \gamma \|X - P^T P X\|_F^2 \\ & \quad + \frac{\mu_k}{2} \|PX - PXZ - E_{k+1} + Y_{1,k}/\mu_k\|_F^2. \end{aligned} \quad (16)$$

We alternatively solve problem (11) and problem (14) until convergence. The whole process of the optimization of NNLS-EF is summarized in Algorithm 4. After getting the optimal solution  $Z^*$ , we use the same strategy as that of NNLS-graph to construct a graph.

## V. EXPERIMENTS

In this section, we evaluate the performance of our proposed methods on publicly available databases, and compare them with currently popular graphs under the same SSL setting. Two typical SSL tasks are considered, semi-supervised classification and semi-supervised dimensionality reduction. All algorithms are implemented with Matlab 2010. All experiments are run 50 times (unless otherwise stated) on a server with an Intel Xeon5680 8-Core 3.50GHz processor and 16GB memory.

<sup>2</sup>we solve the subproblem for  $P$  with the L-BFGS [37] algorithm. The codes can be found at <http://www.di.ens.fr/~mschmidt/Software/minFunc.html>

---

**Algorithm 3** Efficient Inexact ALM Algorithm for problem (14)

---

**Input:** data matrix  $X$ , parameters  $\lambda > 0, \gamma > 0, Z$

**Initialize:**  $E_0 = Y_{1,0} = 0, P_0 = I, \mu_0 = 0.1, \mu_{\max} = 10^{10}, \rho = 1.1, \varepsilon_1 = 10^{-6}, \varepsilon_2 = 10^{-3}, k = 0.$

- 1: **while**  $\|P_k X - P_k X Z - E_k\|_F / \|P_k X\|_F \geq \varepsilon_1$  **or**  $\|E_k - E_{k-1}\|_F / \|P_k X\|_F \geq \varepsilon_2$  **or**  $\|P_k - P_{k-1}\|_F / \|P_k X\|_F \geq \varepsilon_2$  **do**
- 2: Update the variables as (16).
- 3: Update the Lagrange multiplier as follows:

$$Y_{1,k+1} = Y_{1,k} + \mu_k (P_{k+1} X - P_{k+1} X Z - E_{k+1}).$$

- 4: Update  $\mu$  as follows:

$$\mu_{k+1} = \min(\mu_{\max}, \rho \mu_k)$$

- 5: Update  $k: k \leftarrow k + 1.$

6: **end while**

**Output:** an optimal solution  $(E^*, P^*)$ .

---



---

**Algorithm 4** Optimization Algorithm for NNLS-EF (10)

---

**Input:** data matrix  $X$ , parameters  $\beta > 0, \lambda > 0, \gamma > 0, \varepsilon_3 > 0$

- 1: **while** difference between successive  $Z, P$  or  $E$  is greater than  $\varepsilon_3$  **do**
- 2: Update the variables  $E$ , and  $Z$  by solving problem (11) with **Algorithm 1**.
- 3: Update  $E$  and  $P$  by solving problem (14) with **Algorithm 3**
- 4: **end while**

**Output:** an optimal solution  $(Z^*, P^*, E^*)$ .

---

### A. Experiment Setup

**Databases:** We test our proposed methods on three public databases<sup>3</sup> for evaluation: YaleB, PIE, and USPS. YaleB and PIE are face databases and USPS is a hand-written digit database. We choose them because NNLS-graph aims at extracting a linear subspace structure of data. So we have to select databases that roughly have linear subspace structures. It is worth pointing out that these datasets are commonly used in the SSL literature. Existing methods have achieved rather decent results on these data sets. So surpassing them on these three data sets is very challenging and convincing enough to justify the advantages of our method.

- **The YaleB Database:** This face database has 38 individuals, each subject having about 64 near frontal images under different illuminations. We simply use the cropped images of first 15 individuals, and resize them to  $32 \times 32$  pixels.
- **The PIE Database:** This face database contains 41368 images of 68 subjects with different poses, illumination and expressions. We select the first 15 subjects and

<sup>3</sup>Available at <http://www.zjucadcg.cn/dengcai/Data/>

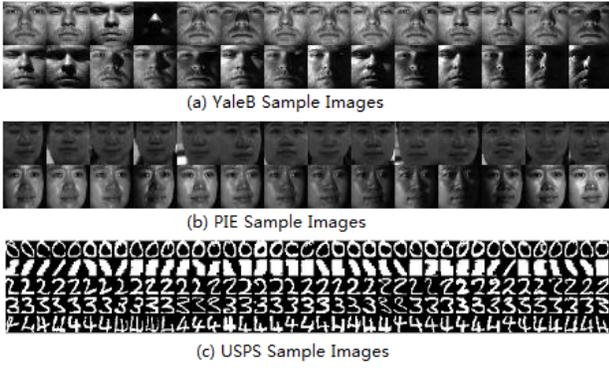


Fig. 1: Sample images used in our experiments.

only use their images in five near frontal poses (C05, C07, C09, C27, C29) and under different illuminations and expressions. Each image is manually cropped and normalized to a size of  $32 \times 32$  pixels.

- **The USPS Database:** This handwritten digit database contains 9298 handwritten digit images in total, each having  $16 \times 16$  pixels. We only use the images of digits 1, 2, 3 and 4 as four classes, each having 1269, 926, 824 and 852 samples, respectively. So there are 3874 images in total.

Fig. 1 shows the sample images of the three databases. As suggested by [19], we normalize the samples so that they have a unit  $\ell_2$  norm.

**Comparison Methods:** We compare our proposed graph construction methods with the following baseline methods:

- **$k$ NN-graph:** We adopt Euclidean distance as the similarity measure, and use a Gaussian kernel to re-weight the edges. The Gaussian kernel parameter  $\sigma$  is set to 1. There are two configurations for constructing graphs, denoted as  $k$ NN0 and  $k$ NN1, where the numbers of nearest neighbors are set to 5 and 8, respectively.
- **LLE-graph [4]:** Following the lines of [4], we construct two LLE-graphs, denoted as **LLE0** and **LLE1**, where the numbers of nearest neighbors are 8 and 10, respectively. Since the weights  $W$  of LLE-graph may be negative and asymmetric, similar to [14] we symmetrize them by  $W = (|W| + |W^T|)/2$ .
- **$\ell_1$ -graph [14]:** Following the lines of [14], we construct the  $\ell_1$ -graph. Since the graph weights  $W$  of  $\ell_1$ -graph is asymmetric, we also symmetrize it as suggested in [14].
- **SPG [15]:** In essence, the SPG problem is a lasso problem with the nonnegativity constraint, without considering corruption errors. Here we use an existing toolbox<sup>4</sup> to solve the lasso problem, and construct the SPG graph following the lines of [15].
- **LRR-graph:** Following [17], we construct the LLR-graph, and symmetrize it as we do for  $\ell_1$ -graph. The parameters of LRR are the same as those in [17].
- **NNLRS-graph:** Similar to our graph construction methods, for our NNLRS-graph and its extension, we empirically tune the regularization parameters according to

different data sets, so as to achieve the best performance. Without loss of generality, we fix the reduced dimensionality to 100 in all our experiments.

## B. Semi-supervised Classification

In this subsection, we carry out the classification experiments on the above databases using the existing graph based SSL frameworks. We select two popular methods, *Gaussian Harmonic Function* (GHF) [6] and *Local and Global Consistency* (LGC) [7] to compare the effectiveness of different graphs. Let  $Y = [Y_l \ Y_u]^T \in \mathbb{R}^{|V| \times c}$  be a label matrix, where  $Y_{ij} = 1$  if sample  $x_i$  is associated with label  $j$  for  $j \in \{1, 2, \dots, c\}$  and  $Y_{ij} = 0$  otherwise. Both GHF and LGC realize the label propagation by learning a classification function  $F = [F_l \ F_u]^T \in \mathbb{R}^{|V| \times c}$ . They utilize the graph and the known labels to recover the continuous classification function by optimizing different predefined energy functions. GHF combines Gaussian random fields and harmonic function for optimizing the following cost on a weighted graph to recover the classification function  $F$ :

$$\min_{F \in \mathbb{R}^{|V| \times c}} \text{tr}(F^T L_W F), \text{ s.t. } L_W F_u = 0, F_l = Y_l, \quad (17)$$

where  $L_W = D - W$  is the graph Laplacian, in which  $D$  is a diagonal matrix with  $D_{ii} = \sum_j W_{ij}$ . Instead of clamping the classification function on labeled nodes by setting hard constraints  $F_l = Y_l$ , LGC introduces an elastic fitness term as follows:

$$\min_{F \in \mathbb{R}^{|V| \times c}} \text{tr}\{F^T \tilde{L}_W F + \mu(F - Y)^T (F - Y)\}, \quad (18)$$

where  $\mu \in [0, +\infty)$  trades off between the local fitting and the global smoothness of the function  $F$ , and  $\tilde{L}_W$  is the normalized graph Laplacian  $\tilde{L}_W = D^{-1/2} L_W D^{-1/2}$ . In our experiments, we simply fix  $\mu = 0.99$ .

We combine different graphs with these two SSL frameworks, and quantitatively evaluate their performance by following the approaches in [9], [13]–[15]. For the YaleB and PIE databases, we randomly select 50 images from each subject as our data sets in each run. Among these 50 images, images are randomly labeled. For the USPS database, we randomly select 200 images for each category, and randomly label them. Different from [13], [15], the percentage of labeled samples ranges from 10% to 60%, instead of ranging from 50% to 80%. This is because the goal of SSL is to reduce the number of labeled images. So we are more interested in the performance of SSL methods with low labeling percentages. The final results are reported in Tables I and II, respectively. From these results, we can observe that:

- 1) In most cases, NNLRS-graph and its extension (i.e. NNLRS-EF) consistently achieve the lowest classification error rates compared to the other graphs, even at low labeling percentages. In many cases, the improvements are rather significant – cutting the error rates by multiple folds! This suggests that NNLRS-graph and its extension are more informative and thus more suitable for semi-supervised classification.

<sup>4</sup><http://sparselab.stanford.edu/>

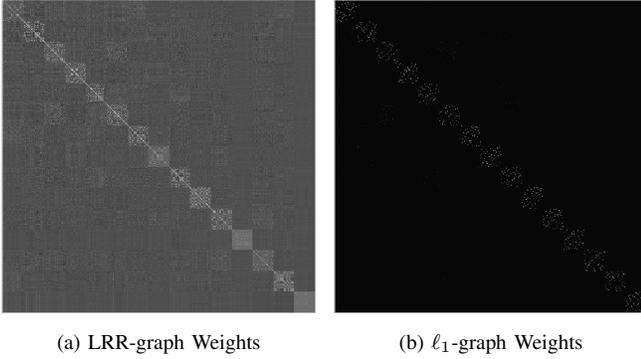
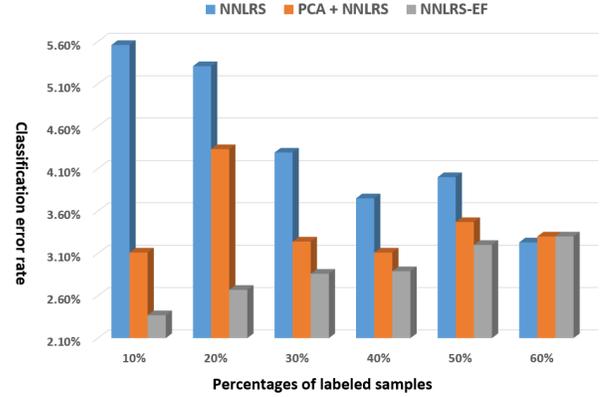


Fig. 2: Visualization of different graph weights  $W$  on the YaleB face database.

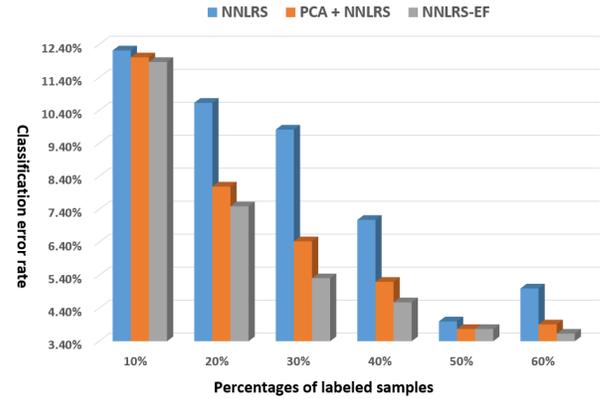
- 2) Compared with NNLRS-graph, NNLRS-EF also has a significant improvements in most cases. This demonstrates that a good data representation can markedly improve the performance of graph construction methods. This is because good representation is robust to data noise and helps to reveal the relationship among data points.
- 3) Though LRR always results in dense graphs, the performance of LRR-graph based SSL methods is not always inferior to that of  $\ell_1$ -graph based SSL methods. On the contrary, LRR-graph performs as well as  $\ell_1$ -graph in many cases. As illustrated in Fig. 2, the weights  $W$  of LRR-graph on the YaleB database is denser than that of  $\ell_1$ -graph. However, LRR-graph outperforms  $\ell_1$ -graph in all cases. This proves that the low rankness property of high-dimensional data is as important as the sparsity property for graph construction.

### C. Semi-supervised Discriminant Analysis

To further examine the effectiveness of NNLRS-graph, we use NNLRS-graph for semi-supervised dimensionality reduction (SSDR), and take semi-supervised discriminant analysis (SDA) [38] for instance. We use SDA to do face recognition on the face databases of YaleB and PIE. SDA aims to find a projection which respects the *discriminant* structure inferred from the labeled data points, as well as the intrinsic *geometric* structure inferred from both labeled and unlabeled data points. We combine SDA with different graphs to learn the subspace, and employ the nearest neighbor classifier. We run the algorithms multiple times with randomly selected data sets. In each run, 30 images from each subject are randomly selected as training images, while the rest images as test images. Among these 30 training images, some images are randomly labeled. Note here that different from the above transductive classification, the test set is not available in the subspace learning stage. Table III tabulates the recognition error rates for different graphs under different labeling percentages. We can see that NNLRS-graph almost consistently outperforms other graphs.



(a) Classification error rate on the YaleB face database



(b) Classification error rate on the PIE face database

Fig. 3: Classification error rate on the YaleB face database and the PIE face database, using the LGC label propagation method under different percentages of labeled samples.

### D. Parameters Sensitivity of NNLRS-graph

In this subsection, we examine the parameter sensitivity of NNLRS-graph, which includes two main parameters,  $\beta$  and  $\lambda$ .  $\beta$  is to balance the sparsity and the low-rankness, while  $\lambda$  is to deal with the gross corruption errors in data. Large  $\beta$  means that we emphasize the sparsity property more than the low-rankness property. We vary the parameters and evaluate the classification performance of NNLRS-graph based SDA on the PIE face database. Since the percentage of gross corruption errors in data should be fixed, we set  $\lambda = 10$  empirically<sup>5</sup> and only vary  $\beta$ . Because here we test many parametric settings, like the above experiments here we only average the rates over 5 random trials. The results are shown in Table IV. From this table, we can see that the performance of NNLRS-graph based SDA decreases when  $\beta > 1$ . If we ignore the sparsity property (i.e.,  $\beta = 0$ ), the performance also decreases. This means that both sparsity property and low-rankness property are important for graph construction. An informative graph should reveal the global structure of the whole data and be as sparse as possible. In all of our experiments above, we always set  $\beta = 0.2$ .

<sup>5</sup>In the above experiments, we did not tune  $\lambda$  either.

TABLE I: Classification error rates (%) of various graphs combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis after the dataset names). The bold numbers are the lowest error rates under different sampling percentages.

Dataset	$k$ NN0	$k$ NN1	LLE0	LLE1	$\ell_1$ -graph	SPG	LRR	NNLRS	NNLRS-EF
YaleB (10%)	33.51	38.27	29.21	29.94	46.13	15.57	28.22	3.75	<b>2.37</b>
YaleB (20%)	34.66	38.97	30.63	30.63	45.54	17.56	24.46	9.84	<b>3.33</b>
YaleB (30%)	33.71	37.87	28.17	28.17	46.14	16.54	22.33	10.54	<b>2.67</b>
YaleB (40%)	33.00	37.34	28.36	28.36	43.39	17.16	19.42	9.38	<b>2.44</b>
YaleB (50%)	33.10	37.38	28.38	28.38	42.25	18.99	18.04	9.64	<b>2.67</b>
YaleB (60%)	32.48	37.78	28.53	28.53	41.52	20.50	16.09	8.13	<b>2.67</b>
PIE (10%)	34.84	37.54	33.06	33.44	22.88	20.50	33.98	<b>11.11</b>	11.83
PIE (20%)	37.46	40.31	35.05	35.81	22.94	20.30	34.35	22.81	<b>7.12</b>
PIE (30%)	35.30	37.80	32.52	32.88	22.33	20.60	31.81	17.86	<b>5.6</b>
PIE (40%)	35.81	38.22	32.51	32.99	23.14	20.81	32.39	16.25	<b>4.5</b>
PIE (50%)	34.39	37.38	31.41	31.64	23.01	21.43	31.33	19.25	<b>3.8</b>
PIE (60%)	35.63	38.00	32.76	32.85	25.76	23.82	32.50	21.56	<b>3.79</b>
USPS (10%)	1.87	2.20	17.10	27.31	43.27	3.95	2.25	<b>1.57</b>	2.36
USPS (20%)	2.51	2.67	22.92	30.83	41.27	5.28	3.10	1.93	<b>1.90</b>
USPS (30%)	5.88	6.10	21.26	27.54	38.31	10.48	8.91	4.95	<b>1.79</b>
USPS (40%)	7.87	8.44	19.21	22.78	34.86	14.22	13.44	7.44	<b>1.53</b>
USPS (50%)	17.19	18.44	18.41	19.48	29.42	20.38	21.88	11.27	<b>1.58</b>
USPS (60%)	11.04	15.20	14.80	14.94	23.36	15.89	17.75	6.09	<b>1.40</b>

TABLE II: Classification error rates (%) of various graphs combined with the LGC label propagation method under different percentages of labeled samples (shown in the parenthesis after the dataset names). The bold numbers are the lowest error rates under different sampling percentages.

Dataset	$k$ NN0	$k$ NN1	LLE0	LLE1	$\ell_1$ -graph	SPG	LRR	NNLRS	NNLRS-EF
YaleB (10%)	32.89	36.84	29.00	29.76	46.82	16.37	28.22	5.56	<b>2.37</b>
YaleB (20%)	31.09	35.59	25.84	26.65	50.53	12.39	24.46	5.31	<b>2.67</b>
YaleB (30%)	28.56	33.54	22.24	22.83	52.33	9.57	22.33	4.29	<b>2.86</b>
YaleB (40%)	26.35	30.97	19.82	19.90	57.16	7.07	19.42	3.75	<b>2.89</b>
YaleB (50%)	24.78	29.73	17.61	17.65	65.79	5.63	18.04	4.00	<b>3.2</b>
YaleB (60%)	22.98	28.58	15.75	15.94	77.56	4.42	16.09	<b>3.23</b>	3.3
PIE (10%)	34.28	36.42	32.25	32.53	21.71	19.75	31.26	12.22	<b>11.87</b>
PIE (20%)	33.06	36.11	30.42	30.83	17.18	15.45	29.82	10.63	<b>7.49</b>
PIE (30%)	30.11	33.51	26.52	27.01	12.06	10.71	25.61	9.82	<b>5.31</b>
PIE (40%)	28.46	32.15	23.62	24.01	9.01	8.25	23.86	7.08	<b>4.58</b>
PIE (50%)	26.96	30.45	21.65	22.22	6.61	6.29	21.24	4.00	<b>3.77</b>
PIE (60%)	25.09	29.09	19.56	20.02	5.13	4.95	20.05	5.00	<b>3.63</b>
USPS (10%)	3.13	3.21	27.69	35.06	33.52	6.92	3.49	2.80	<b>2.62</b>
USPS (20%)	2.22	2.10	22.43	28.96	26.42	4.04	1.83	1.62	<b>1.58</b>
USPS (30%)	1.55	1.53	19.18	25.30	18.92	2.69	1.22	1.13	<b>1.05</b>
USPS (40%)	1.20	1.18	16.62	22.53	16.64	1.88	0.92	0.88	<b>0.87</b>
USPS (50%)	0.82	0.86	14.28	20.01	11.67	1.14	0.61	0.59	<b>0.53</b>
USPS (60%)	0.65	0.72	12.61	17.69	8.89	0.83	0.49	<b>0.48</b>	<b>0.48</b>

### E. Joint Learning vs. Independent Learning

In this subsection, we further examine the effectiveness of joint feature learning. As our feature learning method is mostly related to PCA, we propose to compare our method with the following baseline. We first reduce the dimensionality of the data with PCA. Then we use the embedded data by applying PCA. For fair comparison, we also keep the dimensionality of the data to be 100, which is exactly the same as that of our>NNLRS-EF. For simplicity, we denote such baseline method as PCA+>NNLRS. We show the performance of>NNLRS, PCA+>NNLRS, and>NNLRS-EF in semi-supervised learning in Fig. 3. In all the experiments, we keep the same setting. From this figure, we can have the following observations:

- The performance of>NNLRS with embedded data as features (PCA+>NNLRS and>NNLRS-EF) is better than that of>NNLRS with raw pixels. This observation demonstrates the necessity of data embedding for data structure

discovery.

- The performance of>NNLRS-EF is better than PCA+>NNLRS which does PCA and learns the>NNLRS separately. Such an observation proves that joint learning can learn more proper data representation for the subsequent data structure discovery, which demonstrates the effectiveness of our>NNLRS-EF framework.

## VI. CONCLUSION

This paper proposes a novel informative graph, called the nonnegative low rank and sparse graph (>NNLRS-graph), for graph-based semi-supervised learning.>NNLRS-graph mainly uses two important properties of high-dimensional data, sparsity and low-rankness, both of which capture the structure of the whole data. It simultaneously derives the graph structure and the graph weights, by solving a problem of nonnegative

TABLE III: Recognition error rates (%) of various graphs for semi-supervised discriminative analysis under different percentages of labeled samples.

Dataset	$k$ NN0	$k$ NN1	LLE0	LLE1	$\ell_1$ -graph	SPG	LRR	NNLRS
YaleB (10%)	43.79	48.55	39.06	39.43	37.29	36.88	40.18	<b>34.46</b>
YaleB (20%)	30.31	34.37	25.30	25.59	23.87	23.56	27.96	<b>22.43</b>
YaleB (30%)	20.14	23.16	16.04	16.23	14.69	14.58	18.38	<b>14.09</b>
YaleB (40%)	13.95	16.01	10.57	10.84	9.87	9.68	12.60	<b>9.40</b>
YaleB (50%)	9.89	11.69	7.34	7.42	6.78	6.78	9.03	<b>6.49</b>
YaleB (60%)	7.56	9.78	5.71	5.79	5.32	5.30	7.09	<b>5.16</b>
PIE (10%)	44.53	48.80	38.79	39.30	35.82	35.02	42.20	<b>34.40</b>
PIE (20%)	29.16	33.60	23.57	24.02	21.33	20.84	27.35	<b>20.74</b>
PIE (30%)	16.26	19.26	12.58	12.76	11.37	11.13	15.30	<b>11.11</b>
PIE (40%)	10.74	13.05	8.26	8.44	7.55	<b>7.42</b>	10.28	7.47
PIE (50%)	7.26	8.55	5.70	5.77	5.30	5.23	6.93	<b>5.17</b>
PIE (60%)	5.36	6.23	4.38	4.42	4.11	<b>4.08</b>	5.22	<b>4.08</b>

TABLE IV: Recognition error rates (%) of>NNLRS-graph for semi-supervised discriminative analysis on the PIE face database under different percentages of labeled samples.  $\lambda$  is fixed at 10.

$\beta$	0	0.001	0.01	0.2	0.8	1	5	10	100
10%	38.93	26.48	26.48	26.43	26.62	27.05	39.10	39.52	40.24
20%	24.03	20.10	20.10	20.05	20.05	20.24	28.90	30.76	31.81
30%	13.56	12.10	12.10	12.19	12.19	12.19	16.57	16.95	18.48
40%	9.48	8.14	8.14	8.14	8.10	8.14	13.05	13.38	13.33
50%	6.57	5.48	5.52	5.52	5.57	5.57	7.19	6.71	6.52
60%	6.10	5.86	5.86	5.86	5.86	5.86	6.76	7.10	7.95

low rank and sparse representation of the *whole* data. Extensive experiments on both classification and dimensionality reduction show that,>NNLRS-graph is better at capturing the globally linear structure of data, and thus is more informative and more suitable than other graphs for graph-based semi-supervised learning. Moreover, as good features are robust to data noise and thus help to reveal the relationship among data points, we propose to include the data embedding and construct the graph within one framework. Experiments show that joint feature learning does significantly improve the performance of>NNLRS-graph.

## REFERENCES

- [1] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2328–2335.
- [2] X. Zhu, "Semi-supervised learning literature survey," 2005, Technical Report 1530, Department of Computer Sciences, University of Wisconsin-Madison.
- [3] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 1, pp. 129–143, January 2011.
- [4] J. Wang, F. Wang, C. Zhang, H. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 9, pp. 1600–1615, September 2009.
- [5] M. Belkin, P. Nigogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, November 2006.
- [6] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, vol. 20, no. 2, August 2003, pp. 912–919.
- [7] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16 (NIPS)*, December 2003, pp. 595–602.
- [8] A. Azran, "The rendezvous algorithm: multiclass semi-supervised learning with Markov random walks," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, June 2007, pp. 49–56.
- [9] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, June 2010.
- [10] S. I. Daitch, J. A. Kelner, and D. A. Spielman, "Fitting a graph to vector data," in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, June 2009, pp. 201–208.
- [11] T. Jebara, J. Wang, and S. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, June 2009, pp. 441–448.
- [12] P. P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, September 2009, pp. 442–457.
- [13] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *SIAM International Conference on Data Mining (SDM)*, June 2009, pp. 792–801.
- [14] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with  $\ell_1$ -graph for image analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 858–866, April 2010.
- [15] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 792–801.
- [16] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, "Image annotation by knn-sparse graph-based label propagation over noisily-tagged web images," *ACM Transactions on Intelligent Systems and Technology*, no. 2, February 2011.
- [17] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, June 2010, pp. 663–670.
- [18] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, no. 1, pp. 171–184, January 2013.
- [19] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 2, pp. 210–227, February 2009.
- [20] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with

- kernels," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 423–434, February 2013.
- [21] P. O. Hoyer, "Modeling receptive fields with non-negative sparse coding," *Computational Neuroscience: Trends in Research 2003, Neurocomputing*, pp. 547–552, June 2003.
- [22] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, no. 6755, pp. 788–791, October 1999.
- [23] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM (JACM)*, no. 3, May 2011.
- [24] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, no. 4, pp. 1956–1982, March 2010.
- [25] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient  $l_{2,1}$ -norm minimization," in *Proceedings of the 25<sup>th</sup> Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 339–348.
- [26] E. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, June 2010.
- [27] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2009, uUC Technical Report UILU-ENG-09-2215, arxiv:1009.5055.
- [28] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, December 2011, pp. 612–620.
- [29] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.
- [30] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, vol. 16, December 2003, pp. 234–241.
- [31] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [32] H. Zhang, P. Niyogi, S. Yan, X. He, and Y. Hu, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no. 3, pp. 328–340, March 2005.
- [33] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 19, no. 7, pp. 711–720, 1997.
- [34] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *NIPS*, 2011, pp. 1017–1025.
- [35] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Advances in Neural Information Processing Systems 20 (NIPS2007)*, vol. 19, p. 801, December 2007.
- [36] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [37] Q. Le, A. Karpenko, J. Ngiam, and A. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, December 2011, pp. 1017–1025.
- [38] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proceedings of IEEE 11<sup>th</sup> International Conference on Computer Vision (ICCV)*, October 2007, pp. 1–7.