Constructing a Nonnegative Low-Rank and Sparse Graph With Data-Adaptive Features

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

***Abstract*— This paper aims at constructing a good graph to discover the intrinsic data structures under a semisupervised learning setting. First, we propose to build a nonnegative low-rank and sparse (referred to as NNLRS) graph for the given data representation. In particular, the weights of edges in the graph are obtained by seeking a nonnegative low-rank and sparse reconstruction coefficients matrix that represents each data sample as a linear combination of others. The so-obtained NNLRS-graph captures both the global mixture of subspaces structure (by the low-rankness) and the locally linear structure (by the sparseness) of the data, hence it is both generative and discriminative. Second, as good features are extremely important for constructing a good graph, we propose to learn the data embedding matrix and construct the graph simultaneously within one framework, which is termed as NNLRS with embedded features (referred to as NNLRS-EF). Extensive NNLRS experiments on three publicly available data sets demonstrate that the proposed method outperforms the state-of-the-art graph construction method by a large margin for both semisupervised 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 are costly and time-prohibitive to be obtained, while 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. Therefore it has received considerable attention in both computer vision and machine learning communities [2] recently. Among current SSL methods, graph based SSL is particularly appealing due to its successes in practice as well as its computational efficiency.

A fundamental problem in graph based SSL is to reconstruct a graph to characterize the underlying data structures among the observed data. In graph based SSL methods, labeled and unlabeled samples are the nodes in a graph, and the edges among these nodes are weighted by the affinity between the corresponding pairs of samples. Then the label information of the labeled samples can be efficiently and effectively propagated to the unlabeled data over the graph. Most learning methods formalize the propagation process through a regularized function on the graph. Despite many forms are used in the literature, the regularizers mainly try to accommodate the so-called *cluster assumption* [3], [4], i.e., points in 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. Consequently, properly constructing a graph that can well capture the essential data structures is critical for all graph based SSL methods [5]–[8].

Lots of efforts have been made to exploit the 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 a low-rank representation (LRR) to compute 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 physically meaningful for constructing an affinity graph.

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

Manuscript received September 3, 2014; revised January 1, 2015, March 9, 2015, and April 29, 2015; accepted May 11, 2015. Date of publication June 4, 2015; date of current version July 23, 2015. This work was supported by the National Science Foundation of China under Grant 61472379, Grant 61103134, and Grant 61371192. The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502, in part by the National Natural Science Foundation of China under Grant 61231002 and Grant 61272341, and in part by the Microsoft Research Asia Collaborative Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vladimir Stankovic. *(Corresponding author: Liansheng Zhuang.)*

L. Zhuang, J. Wang, and N. Yu are with the CAS Key Laboratory of Electromagnetic Space Information, School of Information Science and Technology, Unversity of Science and Technology of China, Hefei 230027, China (e-mail: lszhuang@ustc.edu.cn; kkwang@mail.ustc.edu.cn; ynh@ustc.edu.cn).

S. Gao and Y. Ma are with ShanghaiTech University, Shanghai 200031, China (e-mail: gaoshh@shanghaitech.edu.cn; mayi@shanghaitech.edu.cn).

J. Tang is with the Nanjing University of Science and Technology, Nanjing 210044, China (e-mail: jinhuitang@mail.njust.edu.cn).

Z. Lin is with the Key Laboratory of Machine Perception, School of Electronic Engineering and Computer Science, Ministry of Education, Peking University, Beijing 100871, China, and also with the Cooperative Medianet Innovation Center, Shanghai 201805, China (e-mail: zlin@pku.edu.cn).

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.2015.2441632

informative graph. In addition, we 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* (NNLRS-graph). Specifically, each data point is represented by a linear combination of other points, and the coefficients are nonnegative and sparse. Nonnegativity ensures that every data point is in the convex hull of its neighbors, and sparsity ensures that the involved neighbors are as few as possible. In addition to the nonnegative sparsity constraint, we enforce the matrix constructed by the coefficient vectors of all data points to be low-rank to make data vectors on the same subspace be clustered into the same cluster. Moreover, previous works [19]–[22] have shown that by projecting the data into the feature domains, the embedded data will greatly facilitate the subsequent sparse representation and low-rank representation, and improve the classification accuracy. Fox example, Yang *et al.* [21] proposed a new semi-supervised kernel low-rank presentation graph (SKLRG) by combining a low-rank representation with kernel trick. By exploring a low-rank structure in the projected domain, SKLRG can capture the global structure of complex data and achieve more robust subspace segmentation. In another work, Yang *et al.* [22] also sought a sparse presentation in the manifold domain, so that the local geometric information of the data can be better captured. Therefore, finding a good data embedding strategy is also very important for a sparse representation and a low-rank representation. However, these previous works do data embedding [19]–[22] and the subsequent sparse representation or the low-rank presentation 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 in [19]–[22], we propose to simultaneously learn the data embedding matrix and construct the 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 NNLRS-graph for SSL. The sparsity property captures the local low-dimensional linear structures of the data. The low-rank characteristic guarantees that the NNLRS-graph can better capture the global cluster or subspace structures of the data than SR based graphs [15], [16]. Thus the robustness of NNLRS-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 constructing an NNLRS-graph, and consequently enhances the performance of semi-supervised classification.

Extensive experiments on three popular datasets demonstrate that the NNLRS-graph can significantly improve the performance of SSL in most cases.

This article extends its preliminary work [1] in terms of both technique and performance evaluation. Firstly, we extend our NNLRS framework by simultaneously learning the data embedding and the NNLRS-graph, which enhances the robustness of NNLRS-graph for data analysis. Secondly, we conduct more experiments to evaluate the proposed algorithms. Thirdly, more details about our methods are provided and the effects of different parameters are also empirically evaluated in the paper.

The rest of this paper is organized as follows. In Section II, we briefly review the works related to graph construction in SSL. In Section III, we detail the construction of NNLRS-graph, and in Section IV, we extend NNLRS by simultaneously learning the data embedding and NNLRS-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 characteristic for an informative graph. Therefore, lots of researchers propose to improve the robustness of graph by enforcing sparsity. Specifically, Yan and Wang [13], Cheng *et al*. [14] proposed to construct an ℓ_1-graph via sparse representation (SR) [19] by solving an ℓ_1 optimization problem. An ℓ_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 ℓ_1-graph are simultaneously obtained during the ℓ_1 optimization in a parameter-free way. Different from traditional methods, an ℓ_1-graph explores higher order relationships among more data points, hence is more powerful and discriminative. Benefitting from SR, the ℓ_1-graph is sparse, data-adaptive and robust to data noise. Following ℓ_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.* proposed a low-rank representation (LRR) for data representation and use it to construct the affinities of an undirected graph (hereafter called LRR-graph) [17], [18], [21], [23]. 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 ℓ_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 [24], [25], and often leads to better performance for data representation [25], [26] and graph construction [15].

III. Nonnegative Low-Rank and Sparse Graphs

A. Nonnegative Low-Rank and Sparse Representation

Let $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$ be a matrix whose columns are n data samples drawn from independent subspaces.[1] Then each column can be represented by a linear combination of bases in dictionary $A = [a_1, a_2, \cdots, a_m]$:

$$X = AZ, \tag{1}$$

where $Z = [z_1, z_2, \cdots, z_n]$ 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

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

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 (see [27]) we may solve the following relaxed convex program instead

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

where $\|\cdot\|_*$ is the nuclear norm of a matrix [28], i.e., the sum of the singular values of the matrix, and $\|\cdot\|_1$ is the ℓ_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

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

where $\|E\|_{2,1} = \sum_{j=1}^{n}\sqrt{\sum_{i=1}^{m}([E]_{ij})^2}$ is called the $\ell_{2,1}$-norm [29], 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 [30]. 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]. 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) [31] to solve (4).

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

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

The augmented Lagrangian function of problem (5) is

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

where

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

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

[1] The subspaces $S_1, \cdots, S_k$ are independent if and only if $\sum_{i=1}^{k} S_i = \bigoplus_{i=1}^{k} S_i$, where $\bigoplus$ is the direct sum.

previous iterate and a proximal term is then added [31]. With some algebra, the updating schemes are as follows.

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

where $\nabla_Z q$ is the partial differential of q with respect to Z, Θ, S and Ω are the singular value thresholding [28], shrinkage and the $l_{2,1}$ 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 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.

Algorithm 1 Efficient LADMAP Algorithm for NNLRSR

```
Input: data matrix $X$, parameters $\beta > 0$, $\lambda > 0$
Initialize: $Z_0 = H_0 = E_0 = Y_{1,0} = Y_{2,0} = 0$, $\mu_0 = 0.1$, $\mu_{\max} = 10^{10}$, $\rho_0 = 1.1$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-2}$, $\eta_1 = \|A\|_2^2$, $k = 0$.
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
    Update the variables as (8).
    Update Lagrange multipliers as follows:
        $Y_{1,k+1} = Y_{1,k} + \mu_k(X - AZ_{k+1} - E_{k+1})$.
        $Y_{2,k+1} = Y_{2,k} + \mu_k(Z_{k+1} - H_{k+1})$.
    Update $\mu$ as follows:
        $\mu_{k+1} = \min(\mu_{\max}, \rho\mu_k)$, where
        $\rho = \begin{cases} \rho_0, & \text{if } \mu_k \max(\sqrt{\eta_1}\|Z_{k+1} - Z_k\|_F, \\ & \|H_{k+1} - H_k\|_F, \|E_{k+1} - E_k\|_F)/\|X\|_F \\ & < \varepsilon_2, \\ 1, & \text{otherwise.} \end{cases}$
    Update $k$: $k \leftarrow k + 1$.
end while
Output: an optimal solution $(Z^*, H^*, E^*)$.
```

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

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

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

D. Complexity Analysis

The computational cost of constructing NNLRS-graph in Algorithm 2 is mainly determined by Step 2, while the major computation of Algorithm 1 is to update the variables as (8). For ease of analysis, let r_A be the lowest rank for A we can find with our algorithm, and k denote the number of iterations. Without loss of generality, we assume the sizes of both A and X are $d \times n$ $(d < n)$ in the following. In each iteration, SVT is applied to update the low-rank matrix whose total complexity is $O(r_A n^2)$ when we use partial SVD. Then, we employ soft thresholding to update the sparse error matrix whose complexity is $O(dn)$. And the complexity of $\ell_{2,1}$ minimization operator is $O(d^2 n)$. So, the total cost of Algorithm 1 is $O(kr_A n^2 + kd^2 n)$. Since $r_A \leq min(d, n)$, the complexity of constructing NNLRS-graph is at most $O(kdn^2)$.

Algorithm 2 Nonnegative Low-Rank and Sparse Graph Construction

```
Input: Data matrix $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$,
    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 $\hat{X} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n\}$.
2: Solve the following problem using Algorithm 1,
       $\min_{Z,E} \ \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_{2,1}$
       s.t. $\hat{X} = \hat{X}Z + E, Z \geq 0$
   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 NNLRS-graph.
```

When the number of data samples is large, it will be time consuming.

IV. Jointly Learning Data Representation and NNLRS-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 [32], [33]. For face data, the commonly used data representation are EigenFaces [34], LaplacianFaces [35], FisherFaces [36], and RandomFaces [19]. As shown in [19], these representation 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 NNLRS graph separately, we propose to learn the data representation and graph simultaneously to make the learnt data representation more suitable for the construction of NNLRS-graph. We first denote the data projection matrix as P. Similar to [37], we want the projected data to preserve the data information as much as possible. So we aim at minimizing $\|X - P^T PX\|_F^2$. By plugging the learning of P into the NNLRS-graph construction framework, we arrive at the following formulation:

$$\min_{Z,E,P} \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_{2,1} + \gamma\,\|X - P^T PX\|_F^2, \quad \text{s.t. } PX = PXZ + E, \quad Z \geq 0, \tag{10}$$

where γ is a parameter to balance the reconstruction error, which is set empirically (γ is fixed to 0.5 in our experiments). For simplification, we term this formulation as NNLRS with embedded feature (referred to as NNLRS-EF). However, the objective function of NNLRS-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 [38], [39], 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

$$\min_{Z,E} \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_{2,1} \quad \text{s.t. } PX = PXZ + E, \quad Z \geq 0, \tag{11}$$

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

$$\begin{aligned}&\tilde{L}(Z, H, E, Y_1, Y_2, \mu)\\ &\quad= \|Z\|_* + \beta\|H\|_1 + \lambda\|E\|_{2,1}\\ &\qquad+ \langle Y_1, PX - PXZ - E\rangle + \langle Y_2, Z - H\rangle\\ &\qquad+ \frac{\mu}{2}\Big(\|PX - PXZ - E\|_F^2 + \|Z - H\|_F^2\Big)\\ &\quad= \|Z\|_* + \beta\|H\|_1 + \lambda\|E\|_{2,1}\\ &\qquad+ \tilde{q_1}(Z, H, E, Y_1, Y_2, \mu) - \frac{1}{2\mu}\Big(\|Y_1\|_F^2 + \|Y_2\|_F^2\Big),\end{aligned} \tag{12}$$

where

$$\begin{aligned}&\tilde{q_1}(Z, H, E, Y_1, Y_2, \mu)\\ &= \frac{\mu}{2}\Big(\|PX - PXZ - E + Y_1/\mu\|_F^2 + \|Z - H + Y_2/\mu\|_F^2\Big)\end{aligned} \tag{13}$$

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

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

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

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

Algorithm 3 Efficient Inexact ALM Algorithm for Problem (14)

```
Input: data matrix $X$, parameters $\lambda > 0$, $\gamma > 0$, $Z$
Initialize: $E_0 = Y_{1,0} = 0$, $P_0 = I$, $\mu_0 = 0.1$, $\mu_{\max} = 10^{10}$, $\rho = 1.1$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-3}$, $k = 0$.
while $\|P_kX - P_kXZ - E_k\|_F/\|P_kX\|_F \geq \varepsilon_1$ or $\|E_k - E_{k-1}\|_F/\|P_kX\|_F \geq \varepsilon_2$ or $\|P_k - P_{k-1}\|_F/\|P_kX\|_F \geq \varepsilon_2$ do
    Update the variables as (16).
    Update the Lagrange multiplier as follows:
        $Y_{1,k+1} = Y_{1,k} + \mu_k(P_{k+1}X - P_{k+1}XZ - E_{k+1})$.
    Update $\mu$ as follows:
        $\mu_{k+1} = \min(\mu_{\max}, \rho\mu_k)$.
    Update $k$: $k \leftarrow k + 1$.
end while
Output: an optimal solution $(E^*, P^*)$.
```

Algorithm 4 Optimization Algorithm for NNLRS-EF (10)

```
Input: data matrix $X$, parameters $\beta > 0$, $\lambda > 0$, $\gamma > 0$, $\varepsilon_3 > 0$
while difference between successive $Z$, $P$ or $E$ is greater than $\varepsilon_3$ do
    Update the variables $E$, and $Z$ by solving problem (11) with Algorithm 1.
    Update $E$ and $P$ by solving problem (14) with Algorithm 3.
end while
Output: an optimal solution $(Z^*, P^*, E^*)$.
```

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

$$\begin{aligned} E_{k+1} &= \arg\min_{E} \lambda\|E\|_{2,1} \\ &\quad + \frac{\mu_k}{2}\|P_kX - P_kXZ - E + Y_{1,k}/\mu_k\|_F^2 \\ &= \Omega_{\lambda\mu_k^{-1}}(P_kX - P_kXZ + Y_{1,k}/\mu_k), \\ P_{k+1} &= \arg\min_{P} \gamma\,\|X - P^TPX\|_F^2 \\ &\quad + \frac{\mu_k}{2}\|PX - PXZ - E_{k+1} + Y_{1,k}/\mu_k\|_F^2. \quad (16) \end{aligned}$$

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

A. Experiment Setup

Databases: We test our proposed methods on three public databases[3] for evaluation: YaleB, PIE, and USPS. YaleB and PIE are face databases and USPS is a hand-written digit database. We choose them because NNLRS-graph and its extension aim 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 × 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 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 × 32 pixels.
- *The USPS Database:* This handwritten digit database contains 9298 handwritten digit images in total, each having 16 × 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.

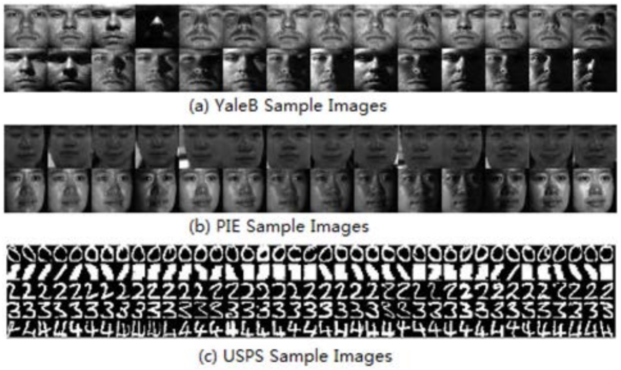
Fig. 1 shows the sample images of the three databases. As suggested by [19], we normalize the samples so that they have a unit ℓ_2 norm.

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

- *kNN-Graph:* We adopt Euclidean distance as the similarity measure, and use a Gaussian kernel to re-weight the edges. The Gaussian kernel parameter σ 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 $W = (|W| + |W^T|)/2$.

[2] 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

[3] Available at http://www.zjucadcg.cn/dengcai/Data/



(a) YaleB Sample Images

(b) PIE Sample Images

(c) USPS Sample Images

Fig. 1. Sample images used in our experiments.

- *ℓ_1-Graph [14]:* Following the lines of [14], we construct the ℓ_1-graph. Since the graph weights W of ℓ_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[4] 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 ℓ_1-graph. The parameters of LRR are the same as those in [17].
- *NNLRS-Graph:* Similar to other 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 $j \in \{1, 2, \cdots, c\}$ 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:

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

where $L_W = D - W$ is the graph Laplacian, in which D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. Δ_{uu} is the lower right $|V_u| \times |V_u|$ submatrix of L_W. 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:

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

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] and [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] and [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. Moreover, in order to understand the robustness of our proposed methods, we also show the statistical variation of the classification error under different percentage of labeled samples. 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.
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 ℓ_1-graph based SSL methods. On the contrary, LRR-graph performs as well as ℓ_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 ℓ_1-graph. However, LRR-graph outperforms ℓ_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.
4) Finally, according to the statistical variation of the classification error, we can see that both NNLRS-graph and NNLRS-EF are robust under different labeling percentages. In most cases, with the growth of labeled samples, the statistical variation will decrease.

[4] http://sparselab.stanford.edu/

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. THE NUMBERS WITH ± ARE THE VARIATIONS OF THE CLASSIFICATION ERROR IN OUR EXPERIMENTS

Dataset	kNN0	kNN1	LLE0	LLE1	ℓ_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(±1.28)	**2.37**(±0.82)
YaleB (20%)	34.66	38.97	30.63	30.63	45.54	17.56	24.46	9.84(±0.78)	**3.33**(±0.67)
YaleB (30%)	33.71	37.87	28.17	28.17	46.14	16.54	22.33	10.54(±0.72)	**2.67**(±0.52)
YaleB (40%)	33.00	37.34	28.36	28.36	43.39	17.16	19.42	9.38(±0.83)	**2.44**(±0.55)
YaleB (50%)	33.10	37.38	28.38	28.38	42.25	18.99	18.04	9.64(±0.93)	**2.67**(±0.73)
YaleB (60%)	32.48	37.78	28.53	28.53	41.52	20.50	16.09	8.13(±1.27)	**2.67**(±0.67)
PIE (10%)	34.84	37.54	33.06	33.44	22.88	20.50	33.98	**11.11**(±3.92)	11.83(±2.36)
PIE (20%)	37.46	40.31	35.05	35.81	22.94	20.30	34.35	22.81(±1.86)	**7.12**(±1.54)
PIE (30%)	35.30	37.80	32.52	32.88	22.33	20.60	31.81	17.86(±1.44)	**5.6**(±1.23)
PIE (40%)	35.81	38.22	32.51	32.99	23.14	20.81	32.39	16.25(±1.00)	**4.5**(±1.06)
PIE (50%)	34.39	37.38	31.41	31.64	23.01	21.43	31.33	19.25(±1.00)	**3.8**(±0.98)
PIE (60%)	35.63	38.00	32.76	32.85	25.76	23.82	32.50	21.56(±1.22)	**3.79**(±1.17)
USPS (10%)	1.87	2.20	17.10	27.31	43.27	3.95	2.25	**1.57**(±0.49)	2.36(±0.73)
USPS (20%)	2.51	2.67	22.92	30.83	41.27	5.28	3.10	1.93(±0.85)	**1.90**(±0.52)
USPS (30%)	5.88	6.10	21.26	27.54	38.31	10.48	8.91	4.95(±0.76)	**1.79**(±0.42)
USPS (40%)	7.87	8.44	19.21	22.78	34.86	14.22	13.44	7.44(±0.73)	**1.53**(±0.50)
USPS (50%)	17.19	18.44	18.41	19.48	29.42	20.38	21.88	11.27(±0.87)	**1.58**(±0.63)
USPS (60%)	11.04	15.20	14.80	14.94	23.36	15.89	17.75	6.09(±1.15)	**1.40**(±0.54)

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. THE NUMBERS WITH ± ARE THE VARIATIONS OF THE CLASSIFICATION ERROR IN OUR EXPERIMENTS

Dataset	kNN0	kNN1	LLE0	LLE1	ℓ_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(±0.59)	**3.30**(±1.28)
YaleB (20%)	31.09	35.59	25.84	26.65	50.53	12.39	24.46	5.31(±0.87)	**3.20**(±0.90)
YaleB (30%)	28.56	33.54	22.24	22.83	52.33	9.57	22.33	4.29(±1.17)	**2.89**(±0.69)
YaleB (40%)	26.35	30.97	19.82	19.90	57.16	7.07	19.42	3.75(±0.90)	**2.86**(±0.55)
YaleB (50%)	24.78	29.73	17.61	17.65	65.79	5.63	18.04	4.00(±1.30)	**2.67**(±0.83)
YaleB (60%)	22.98	28.58	15.75	15.94	77.56	4.42	16.09	3.23(±0.95)	**2.37**(±0.75)
PIE (10%)	34.28	36.42	32.25	32.53	21.71	19.75	31.26	12.22(±2.91)	**11.87**(±2.31)
PIE (20%)	33.06	36.11	30.42	30.83	17.18	15.45	29.82	10.63(±2.42)	**7.49**(±1.52)
PIE (30%)	30.11	33.51	26.52	27.01	12.06	10.71	25.61	9.82(±1.69)	**5.31**(±1.06)
PIE (40%)	28.46	32.15	23.62	24.01	9.01	8.25	23.86	7.08(±1.21)	**4.58**(±1.06)
PIE (50%)	26.96	30.45	21.65	22.22	6.61	6.29	21.24	4.00(±1.18)	**3.77**(±1.00)
PIE (60%)	25.09	29.09	19.56	20.02	5.13	4.95	20.05	5.00(±1.25)	**3.63**(±0.86)
USPS (10%)	3.13	3.21	27.69	35.06	33.52	6.92	3.49	2.80(±1.03)	**2.62**(±0.94)
USPS (20%)	2.22	2.10	22.43	28.96	26.42	4.04	1.83	1.62(±0.72)	**1.58**(±0.61)
USPS (30%)	1.55	1.53	19.18	25.30	18.92	2.69	1.22	1.13(±0.51)	**1.05**(±0.47)
USPS (40%)	1.20	1.18	16.62	22.53	16.64	1.88	0.92	0.88(±0.44)	**0.87**(±0.41)
USPS (50%)	0.82	0.86	14.28	20.01	11.67	1.14	0.61	0.59(±0.44)	**0.53**(±0.42)
USPS (60%)	0.65	0.72	12.61	17.69	8.89	0.83	0.49	**0.48**(±0.3)	**0.48**(±0.3)

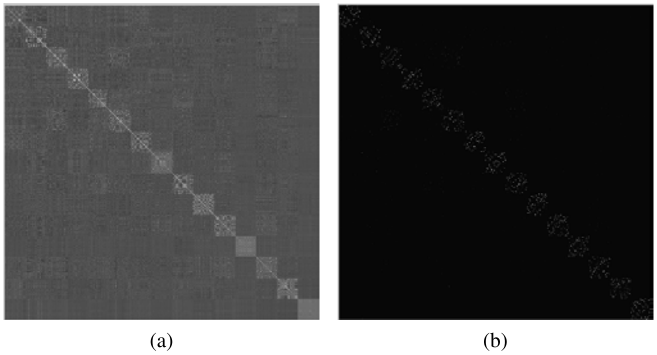


Fig. 2. Visualization of different graph weights W on the YaleB face database. (a) LRR-graph Weights. (b) ℓ_1-graph Weights.

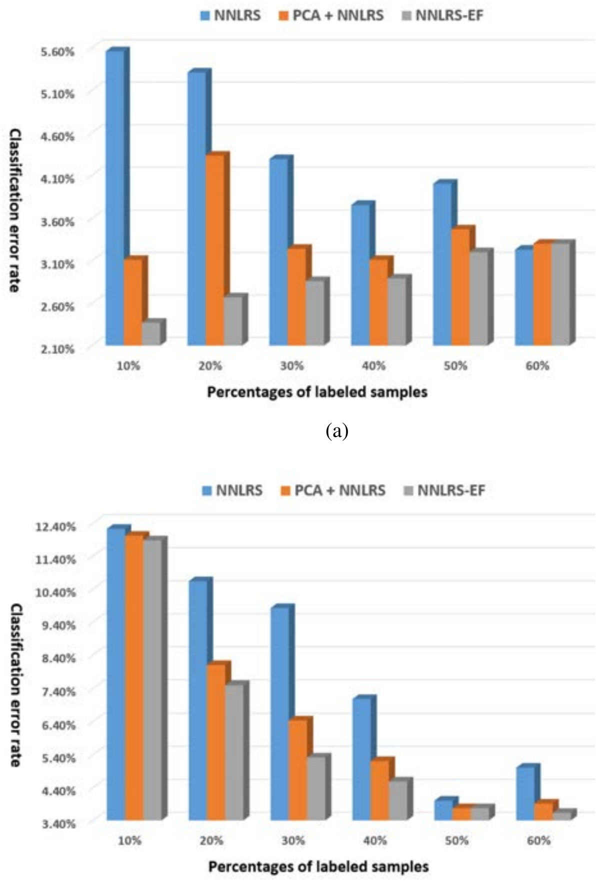
C. Semi-Supervised Discriminant Analysis

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) [40] 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

TABLE III

RECOGNITION ERROR RATES (%) OF VARIOUS GRAPHS FOR SEMI-SUPERVISED DISCRIMINATIVE ANALYSIS UNDER DIFFERENT PERCENTAGES OF LABELED SAMPLES

Dataset	kNN0	kNN1	LLE0	LLE1	ℓ_1-graph	SPG	LRR	NNLRS
YaleB (10%)	43.79	48.55	39.06	39.43	37.29	36.88	40.18	**34.46**
YaleB (20%)	30.31	34.37	25.30	25.59	23.87	23.56	27.96	**22.43**
YaleB (30%)	20.14	23.16	16.04	16.23	14.69	14.58	18.38	**14.09**
YaleB (40%)	13.95	16.01	10.57	10.84	9.87	9.68	12.60	**9.40**
YaleB (50%)	9.89	11.69	7.34	7.42	6.78	6.78	9.03	**6.49**
YaleB (60%)	7.56	9.78	5.71	5.79	5.32	5.30	7.09	**5.16**
PIE (10%)	44.53	48.80	38.79	39.30	35.82	35.02	42.20	**34.40**
PIE (20%)	29.16	33.60	23.57	24.02	21.33	20.84	27.35	**20.74**
PIE (30%)	16.26	19.26	12.58	12.76	11.37	11.13	15.30	**11.11**
PIE (40%)	10.74	13.05	8.26	8.44	7.55	**7.42**	10.28	7.47
PIE (50%)	7.26	8.55	5.70	5.77	5.30	5.23	6.93	**5.17**
PIE (60%)	5.36	6.23	4.38	4.42	4.11	**4.08**	5.22	**4.08**



(a)

(b)

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. (a) Classification error rate on the YaleB face database. (b) Classification error rate on the PIE face database.

see that NNLRS-graph almost consistently outperforms other graphs.

D. Parameters Sensitivity of NNLRS-Graph

In this subsection, we examine the parameter sensitivity of NNLRS-graph, which includes two main parameters, β and λ. β is to balance the sparsity and the low-rankness, while λ is to deal with the gross corruption errors in data. Large β 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[5] and only vary β. 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 $\beta = 0.2$.

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 NNLSR 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

[5] In the above experiments, we did not tune λ either.

TABLE IV

RECOGNITION ERROR RATES (%) OF NNLRS-GRAPH FOR SEMI-SUPERVISED DISCRIMINATIVE ANALYSIS ON THE PIE FACE DATABASE UNDER DIFFERENT PERCENTAGES OF LABELED SAMPLES. λ IS FIXED AT 10

β	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

subsequent data structure discovery, which demonstrates the effectiveness of our NNLRS-EF framework.

VI. CONCLUSION

This paper proposes a novel informative graph, i.e., nonnegative low-rank and sparse graph (NNLRS-graph), for graph-based semi-supervised learning. NNLRS-graph mainly leverages two important properties of high-dimensional data, sparsity and low-rankness, both of which capture the structure of the whole data. Furthermore, as good features are robust to data noise and thus help to reveal the relationship among data points, we propose to simultaneously learn the data embedding and construct the graph within one framework, which is termed as NNLRS-EF. Extensive experiments on both classification and dimensionality reduction show that, NNLRS-graph and NNLRS-EF are better at capturing the globally linear structure of data, and thus are more informative and more suitable than other graphs for graph-based semi-supervised learning. Also, experiments show that joint feature learning does significantly improve the performance of NNLRS-graph.

However, though our proposed methods have obtained impressive performance, they have high computational complexity, which mainly comes from the optimization procedure. Therefore, our methods currently are only suitable to applications with small scale data. To apply current models to large scale data, one can develop new paralleled version of our methods and(or) use GPU to accelerate the methods. Moreover, current methods are mainly based on linear model, while data in real application always lies in nonlinear manifolds. So, one can improve the proposed methods by incorporating the geometric structure of nonlinear manifolds. Finally, we currently figure out qualitatively the settings which our methods can cope with. Rigorous quantitative analysis will give more insight and help to improve our model. We will follow some existing work, such as [41], to develop theoretical analysis of our methods in our future work.

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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2328–2335.

[2] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.

[3] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 129–143, Jan. 2011.

[4] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.

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

[6] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, vol. 20. Aug. 2003, pp. 912–919.

[7] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. Cambridge, MA, USA: MIT Press, Dec. 2003, pp. 595–602.

[8] A. Azran, "The rendezvous algorithm: Multiclass semi-supervised learning with Markov random walks," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, Jun. 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," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[10] S. I. Daitch, J. A. Kelner, and D. A. Spielman, "Fitting a graph to vector data," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, Jun. 2009, pp. 201–208.

[11] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, Jun. 2009, pp. 441–448.

[12] P. P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Sep. 2009, pp. 442–457.

[13] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, Jun. 2009, pp. 792–801.

[14] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with ℓ^1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.

[15] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2849–2856.

[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 Trans. Intell. Syst. Technol.*, vol. 2, no. 2, Feb. 2011, Art. ID 14.

[17] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 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 Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[19] 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.

[20] S. Gao, I. W. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, Feb. 2013.

[21] S. Yang, Z. Feng, Y. Ren, H. Liu, and L. Jiao, "Semi-supervised classification via kernel low-rank representation graph," *Knowl.-Based Syst.*, vol. 69, pp. 150–158, Oct. 2014.

[22] S. Yang, P. Jin, B. Li, L. Yang, W. Xu, and L. Jiao, "Semisupervised dual-geometric subspace projection for dimensionality reduction of hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3587–3593, Jun. 2014.

[23] L. Zhuang, H. Gao, J. Luo, and Z. Lin, “Regularized semi-supervised latent Dirichlet allocation for visual concept learning,” *Neurocomputing*, vol. 119, pp. 26–32, Nov. 2013.

[24] P. O. Hoyer, “Modeling receptive fields with non-negative sparse coding,” *Neurocomputing*, vols. 52–54, pp. 547–552, Jun. 2003.

[25] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[26] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*. Cambridge, MA, USA: MIT Press, 2001, pp. 556–562.

[27] E. J. Candés, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *J. ACM*, vol. 58, no. 3, May 2011, Art. ID 11.

[28] J.-F. Cai, E. J. Candés, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[29] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient $l_{2,1}$-norm minimization,” in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 339–348.

[30] E. J. Candès and Y. Plan, “Matrix completion with noise,” *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[31] 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*. Red Hook, NY, USA: Curran Associates, Dec. 2011, pp. 612–620.

[32] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Wiley, 2005.

[33] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in Neural Information Processing Systems 16*, vol. 16. Cambridge, MA, USA: MIT Press, Dec. 2003, pp. 234–241.

[34] M. Turk and A. Pentland, “Eigenfaces for recognition,” *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

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

[36] P. N. Belhumeur, J. P. Hespanha, and D. 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.

[37] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, “ICA with reconstruction cost for efficient overcomplete feature learning,” in *Proc. NIPS*, 2011, pp. 1017–1025.

[38] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in *Advances in Neural Information Processing Systems 20*, vol. 19. Cambridge, MA, USA: MIT Press, Dec. 2007, p. 801.

[39] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[40] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis,” in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–7.

[41] Y.-X. Wang, H. Xu, and C. Leng, “Provable subspace clustering: When LRR meets SSC,” in *Advances in Neural Information Processing Systems 26*. Red Hook, NY, USA: Curran Associates, 2013, pp. 64–72.

Liansheng Zhuang (M’11) received the bachelor’s and Ph.D. degrees from the University of Science and Technology of China (USTC), China, in 2001 and 2006, respectively. Since 2006, he has served as a Lecturer with the School of Information Science and Technology, USTC. From 2012 to 2013, he was a Visiting Research Scientist with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. His main research interesting is in computer vision and machine learning. He is a member of Association for Computing Machinery and China Computer Federation.

Shenghua Gao received the B.E. degree from the University of Science and Technology of China in 2008 (outstanding graduates), and the Ph.D. degree from Nanyang Technological University, in 2012. He is currently an Assistant Professor with ShanghaiTech University, China. From 2012 to 2014, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. His research interests include computer vision and machine learning, and he is focusing on face and object recognition. He has authored over 20 papers on object and face recognition related topics in many international conferences and journals, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, *International Journal of Computer Vision*, the IEEE Transactions on Image Processing, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Multimedia, the IEEE Transactions on Circuits and Systems for Video Technology, Computer Vision and Pattern Recognition, and the European Conference on Computer Vision. He was a recipient of the Microsoft Research Fellowship in 2010.

Jinhui Tang (M’08–SM’14) received the B.E. and Ph.D. degrees from the University of Science and Technology of China (USTC), in 2003 and 2008, respectively. From 2008 to 2010, he was a Research Fellow with the School of Computing, National University of Singapore. During that period, he visited the School of Information and Computer Science, University of California, Irvine, in 2010, as a Visiting Research Scientist. From 2011 to 2012, he visited Microsoft Research Asia, as a Visiting Researcher. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He has authored over 80 journal and conference papers in these areas. His current research interests include large-scale multimedia search, social media mining, and computer vision. He is a member of Association for Computing Machinery and China Computer Federation. He serves as an Editorial Board Member of Pattern Analysis and Applications, Multimedia Tools and Applications, Information Sciences, and Neurocomputing, a Technical Committee Member of about 30 international conferences, and a Reviewer for about 30 prestigious international journals. He was a co-recipient of the best paper award from ACM Multimedia in 2007, PCM 2011, and ICIMCS 2011.

Jingjing Wang received the B.S. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, in 2010, where he is currently pursuing the Ph.D. degree in electronic engineering and information science. His main research interests are machine learning and computer vision.

Zhouchen Lin (M’00–SM’08) received the Ph.D. degree in applied mathematics from Peking University, in 2000. He was a Guest Professor with Shanghai Jiao Tong University, Beijing Jiaotong University, and Southeast University. He was also a Guest Researcher with the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is also a Chair Professor with Northeast Normal University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is also an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and the *International Journal of Computer Vision*.

Yi Ma (F'13) received the bachelor's degree in automation and applied mathematics from Tsinghua University, Beijing, China, in 1995, and the master's degree in electrical engineering and computer sciences in 1997, the second master's degree in mathematics in 2000, and the Ph.D. degree in electrical engineering and computer sciences in 2000, from the University of California at Berkeley. From 2000 to 2011, he served as a tenured Associated Professor of the Department of Electrical and Computer Engineering with the University of Illinois at Urbana–Champaign, where he holds an Adjunct Professorship. He also serves as a Research Associate Professor with the Decision and Control Group of the Coordinated Science Laboratory and the Image Formation and Processing Group of the Beckman Institute. He was a Visiting Senior Researcher with Microsoft Research Asia, Beijing, China, in 2006, and a Visiting Professor with the Department of Electrical Engineering and Computer Sciences, UC Berkeley, in Spring 2007. From 2009 to 2014, he has served as a Research Manager with the Visual Computing Group, Microsoft Research Asia, Beijing, China. He is currently the Executive Dean of the School of Information Science and Technology with ShanghaiTech University, Shanghai, China. He has served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*. He has also served as the Chief Guest Editor for special issues for the IEEE PROCEEDINGS and the *IEEE Signal Processing Magazine* in 2010 and 2011. He currently serves as the Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY, the *IMA Journal on Information and Inference*, and the *Society of Industrial and Applied Mathematics (SIAM) Journal on Imaging Sciences*. He has served as the Area Chair for NIPS 2011, ICCV 2011, and CVPR 2013, and the Program Chair for ICCV 2013, and the General Chair for ICCV 2015. He is a member of Association for Computing Machinery and SIAM.

Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications, in 1987, and the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China, in 2004, where he is currently a Professor. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.