Label Information Guided Graph Construction for Semi-Supervised Learning

Liansheng Zhuang, Zihan Zhou, Shenghua Gao, Jingwen Yin, Zhouchen Lin, Yi Ma, *IEEE Fellow*

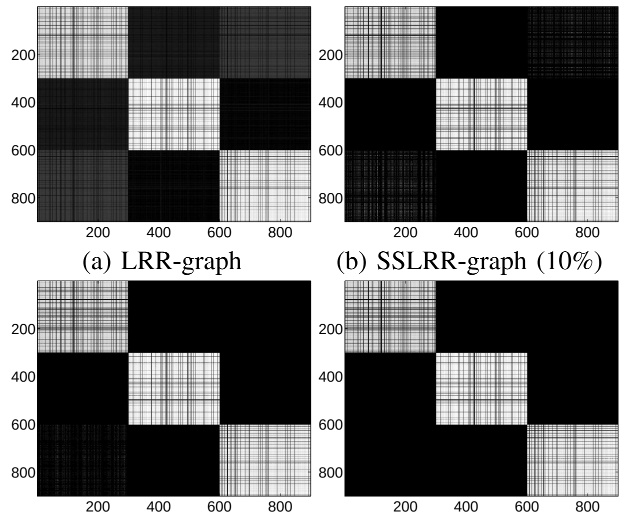
***Abstract*—In the literature, most existing graph-based semi-supervised learning (SSL) methods only use the label information of observed samples in the label propagation stage, while ignoring such valuable information when learning the graph. In this paper, we argue that it is beneficial to consider the label information in the graph learning stage. Specifically, by enforcing the weight of edges between labeled samples of different classes to be zero, we explicitly incorporate the label information into the state-of-the-art graph learning methods, such as the Low-Rank Representation (LRR), and propose a novel semi-supervised graph learning method called Semi-Supervised Low-Rank Representation (SSLRR). This results in a convex optimization problem with linear constraints, which can be solved by the linearized alternating direction method. Though we take LRR as an example, our proposed method is in fact very general and can be applied to any self-representation graph learning methods. Experiment results on both synthetic and real datasets demonstrate that the proposed graph learning method can better capture the global geometric structure of the data, and therefore is more effective for semi-supervised learning tasks.**

***Index Terms*—Semi-supervised Graph Learning, Low-Rank Representation, Label Information.**

I. INTRODUCTION

IN computer vision and machine learning research communities, semi-supervised learning (SSL) [1]–[3] has attracted numerous attention over the past decade because of its ability to make use of rich unlabeled data for training. It has been demonstrated that unlabeled data, when used in conjunction with a small set of labeled data, can often considerably improve the learning accuracy. Among the current methods, graph-based SSL is an appealing approach due to its low computation complexity and flexibility in practice.

In general, a graph-based SSL method consists of two key steps. First, a graph is built from all data samples (including both labeled and unlabeled samples) to model the relationships among the points. Then, label information of the labeled samples is propagated to the unlabeled samples over the graph. Though different graph-based SSL methods formulate the label propagation process via different objective functions, all of them share one common assumption (i.e., the *cluster assumption*), that is, points on the same structure (such as a cluster, a subspace, or a manifold) are likely to have the same label. Since one normally does not have an explicit model for the underlying structures, a graph constructed from the data samples often serves as an approximation to it. Therefore, constructing a good graph that best captures the essential data structure is critical for all graph-based SSL methods.



(a) LRR-graph (b) SSLRR-graph (10%)

(c) SSLRR-graph (30%) (d) SSLRR-graph (60%)

Fig. 1: Visualization of the unsupervised LRR-graph and semi-supervised SSLRR-graph with 10%, 30%, and 60% labeled data in a synthetic dataset. The "block-diagonal structure" of the graph is best preserved by the SSLRR-graph with 60% labeled data samples.

This paper proposes a novel framework to address the graph construction problem in SSL. Our key insight is that most graph-based SSL methods do not take advantage of label information when building a graph from the data samples, which limits their performance. Leveraging powerful tools from high-dimensional statistics and optimization, we successfully mitigate this issue by constructing the graph in a *semi-supervised* manner. Our framework is quite general and can be applied to many existing graph learning methods. As an example, Figure 1 compares the SSLRR-graphs constructed by our method to the original LRR-graph [4] on a synthetic dataset (see Section V-A about synthetic data generation).

This work is supported by the National Science Foundation of China (No.s 61472379 and 61371192). Zhouchen Lin is supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502) and National Natural Science Foundation (NSF) of China (grant nos. 61625301 and 61231002).

Liansheng Zhuang and Jingwen Yin are with University of Science and Technology of China, Hefei, China. Zihan Zhou is with the College of Information Sciences and Technology at the Pennsylvania State University, United States. Shenghua Gao and Yi Ma are with ShanghaiTech University, Shanghai, China. Zhouchen Lin is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, P. R. China. Also, Zhouchen Lin is with Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, P. R. China. Liansheng Zhuang is the corresponding author (lszhuang@ustc.edu.cn).

A. Background and Motivations

Intuitively, a good graph should reveal the true intrinsic complexity or dimensionality of the data points by capturing the global structures of the data (i.e., multiple clusters, subspaces, or manifolds). Traditional methods such as *k-nearest neighbors* (kNN) and *Locality Linear Embedding* (LLE) [3], [5], [6], however, mainly rely on pair-wise Euclidean distance to build such a graph, thus are unable to capture the global data structures. As a result, these methods tend to be sensitive to local data noise and errors. Moreover, traditional methods always use fixed global parameters to determine the graph structure and the edge weights, thus may fail to offer any datum-adaptive neighborhoods.

Recently, motivated by the advance in fast computational methods in signal processing (particularly, in the areas of compressive sensing, sparse representation, and low-rank matrix recovery [7]–[11]), several methods [4], [12]–[15] have been proposed to construct undirected graphs that exploit the global structures of the data. Different from traditional methods, these methods seek a representation of each datum as a linear combination of *all* the other data samples. By solving a high-dimensional convex optimization problem, these methods automatically select the most informative "neighbors" for each datum, and simultaneously obtain the graph adjacency structure and weights in a nearly parameter-free fashion. We call these methods *self-representation methods*.

While the self-representation methods are datum-adaptive and robust to local errors, they also suffer from some problems in practice. In particular, in the ideal case, the linear coefficients recovered via convex optimization should be "structure-sparse", that is, only those points belonging to the same structure as the target point should have nonzero coefficients. Unfortunately, this assumption only holds true if all the points lie in a union of independent or disjoint subspaces and are noiseless [16]. In other words, in the presence of dependent subspaces, nonlinear manifolds and/or data errors, these methods may select points from different structures (i.e., classes) to represent a data point, making the representation less informative.

To overcome this difficulty, our key insight is that all the aforementioned self-representation methods are *unsupervised*. That is, they do not utilize label information to learn the graph. Thus, when applied within the SSL framework, they could be further improved. To see why the label information provides valuable cues to the graph construction, consider the extreme case where the label information of all samples is available. In such case, one can directly enforce the coefficients belonging to different classes to be zero, so that the resulting representation is naturally structure-sparse. In fact, some researchers have explored the label information in graph construction [17]–[19]. However, these works are based on traditional graph construction methods, hence inevitably inherit the limits of traditional methods as mentioned above.

B. Our Contributions

Inspired by the above observations, we propose to explicitly incorporate label information into the self-representation methods. Specifically, we show that one can seamlessly integrate the label information of a subset of the samples into the state-of-the-art self-representation methods, such as the LRR-graph [4], by restricting the representation coefficients between labeled points from different classes to be zero. Intuitively, this information helps us prevent the structure-sparsity of the coefficients from being destroyed in challenging real world scenarios, i.e., small signal-to-noise ratio, dependent subspaces and/or nonlinear manifolds. Thus, by solving a convex optimization problem with linear constraints, we can obtain a new linear representation for each data point which respects the label information, and subsequently construct a graph that better captures the global data structures.

To verify the effectiveness of our method, we conduct extensive experiments on both simulation datasets and public real datasets for two important tasks, namely nonlinear manifolds clustering and semi-supervised classification. The experimental results clearly demonstrate that, compared to graphs constructed via existing unsupervised self-representation methods, the graphs constructed by our method are more robust to data errors, and more informative and discriminative in practice, especially in the cases of complex data structures (e.g., dependent subspaces and nonlinear manifolds).

In summary, we make the following contributions in this paper:

1) We present a novel semi-supervised graph learning framework which seamlessly integrates the label information of data samples into the state-of-the-art graph learning methods. Compared with the graphs learned by existing methods, graphs learned by our method better capture the global structure of the data, especially when the data is subject to noises, the subspaces are not independent, or the data points lie in nonlinear manifolds.
2) We apply our method to both manifold clustering and semi-supervised learning tasks, and empirically demonstrate that the label information helps preserve the block-diagonal structure of the coefficient matrix, so that the learned graph is more informative and robust to data noises in practice.
3) While we use LRR as a representative example to illustrate our semi-supervised graph learning method, our method is in fact quite general and can be easily applied to other self-representation graph learning methods. To this end, we conduct experiments with three existing graph learning methods, namely LRR-graph, ℓ_1-graph [12] and non-negative low rank and sparse graph (NNLRS-graph) [14]. We demonstrate that one can significantly improve the performance of these methods by incorporating the label information.

The remainder of this paper is organized as follows. In Section III, we present our semi-supervised graph learning framework. We give details about how to construct the graph weight matrix in Section IV. Experiment results and analysis are presented in Section V. Finally, Section VI concludes our paper.

II. Related Work

Compared to label inference, graph construction has attracted much less attention in the machine learning community until recent years [5], [20]–[23]. Traditionally, ϵ-neighborhood and k-Nearest Neighbors (kNN) are commonly used in graph based SSL methods. An ϵ-neighborhood graph is built by connecting all the data points whose distance are smaller than the threshold ϵ. These graphs are often sensitive to the chosen parameter ϵ and produce undesirable degree distribution (e.g., disconnected components or almost-complete graph). On the other hand, a kNN graph links each node to its k nearest neighbors. Compared to ϵ-neighborhood graphs, kNN graphs enjoy some favorable properties in choosing the parameters, and tend to perform better than ϵ-neighborhood graphs in practice [5]. Various unsupervised methods have been proposed to improve the kNN graph construction [5], [6], [22]–[26]. For example, Wang *et al.* [24] estimated the graph edge weights in a manner of multi-wise edges instead of pairwise edges. Jebara *et al.* [5] proposed to use b-matching to produce a balanced or regular graph, which ensures that each node in the graph has the same number of edges. Other works focus on efficient algorithms to find the k nearest neighbors [27]–[31]. But a common limitation of all above methods is that the neighbors are selected "locally" – the decision is only based on the individual relation between the reconstructed data point and the other data points. Such neighbors can only capture the local data structure and thus greatly limit the performance of graph based SSL methods.

To remedy the above issue, *self-representation methods* [4], [12], [13], [32]–[36] have become increasingly popular in recent years. For example, ℓ_1-graph [12] proposes to encode each datum as a sparse representation of the other samples. It is shown to have several advantages in practice, including robustness to noise, sparsity for efficiency, and datum-adaptive neighborhood [12]. Since ℓ_1-graph is purely based on numerical solutions, Zhou *et al.* [35] propose to exploit the geometric data structure via a kNN fused lasso graph. Additionally, Fang *et al.* [32] impose the auto-grouped effect in the ℓ_1-graph by applying two sparse regularizations: Elastic net and Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR). Han *et al.* [34], [36] use a reduced size dictionary to preserve the locality and the geometry structure for data clustering applications. Yang *et al.* [33] use the Graph Laplacian regularization to improve the quality of ℓ_1-graph.

To further capture the global data structure, Liu *et al.* [4] propose the LRR-graph, which seeks a low-rank representation of the data. By jointly obtaining the representation of all the data under the low-rankness assumption, LRR-graph effectively impose global constraints on the data structure (e.g., multiple subspaces). Moreover, since each sample can be used to represent itself, there always exist a feasible solution for LRR-graph even if the data sampling is insufficient. These properties make LRR-graph a good candidate for various learning tasks including SSL. Extensions to the original LRR-graph include [14], [37], which further impose non-negative and sparse constraints on the low-rank representation, [38], which incorporates a block-diagonal prior, and [39], which combines the low-rank representation with the kernel trick. Other works introduce various regularized terms so as to explicitly consider the cases where the data lie on non-linear manifolds [40]–[43].

However, most self-representation methods ignore the label information of the data samples during graph construction. In this paper, we demonstrate how the label information can be integrated into the construction of LRR-graph, resulting in significant improvement on the performance of existing SSL methods. Further, our framework can be readily applied to other self-representation methods such as ℓ_1-graph, Least Squares Representation [44], Correlation Adaptive Subspace Segmentation [45], Correntropy Induced ℓ_2-graph [46], and the Smooth Representation [47].

III. Semi-Supervised Graph Learning

In this section, we use the LRR-graph [4] as a representative example to describe our semi-supervised graph learning framework. For completeness, we first give a brief overview of LRR.

A. Low-Rank Representation: An Overview

Low-Rank Representation (LRR) was originally proposed in [4] to segment data drawn from a union of multiple linear (or affine) subspaces. Specifically, given a set of sufficiently dense data vectors $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ (each column is a sample) drawn from a union of k subspaces, LRR seeks the lowest-rank representation among all the candidates that represent each data vector as the linear combination of the data themselves. It proposes to solve the following high-dimensional convex optimization problem:

$$\begin{aligned} \min_{Z,E} \quad & \|Z\|_* + \lambda\|E\|_{2,1} \\ s.t. \quad & X = XZ + E, \end{aligned} \tag{1}$$

where $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n]$ is the coefficient matrix with each $\boldsymbol{z}_i$ being the coefficients of $\boldsymbol{x}_i$. In addition, $\|\cdot\|_*$ is the nuclear norm, i.e., sum of singular values. The $\ell_{2,1}$-norm of E, $\|E\|_{2,1} = \sum_{j=1}^{n}\sqrt{\sum_{i=1}^{d}(e_{ij})^2}$ where e_{ij} is the (i,j)-th element of matrix E, is used to model the sample-specific corruptions and outliers. Finally, the parameter $\lambda > 0$ is used to balance the effects of the two terms. It can be chosen according to properties of the two norms, or tuned empirically.

As shown in [4], when data are clean and sampled from independent subspaces, the optimal solution Z^* of (1) is block-diagonal (ignoring the E term). That is, for each $\boldsymbol{x}_i$, only those entries of $\boldsymbol{z}_i$ which correspond to data points in the same subspace as $\boldsymbol{x}_i$ have nonzero values. In this way, LRR is able to capture the global structure (i.e., multiple subspaces) of the data. Further, by introducing the error term $\|E\|_{2,1}$, LRR achieves robust subspace segmentation results despite of corrupted data vectors or outliers. After solving the problem (1), we can define the affinity matrix W of an undirected graph as $W = (|Z^*| + |Z^{*T}|)/2$. Consequently, the undirected graph (called the LRR-graph) also captures the global structure of the data.

However, in real applications, the data is alway subject to noises, the subspaces could be dependent, and the data could even lie in nonlinear manifolds. In these cases, the block-diagonal structure of Z^* is often destroyed. To overcome this difficulty and improve the performance of the LRR-graph, we next propose a novel method to generalize the LRR model within the SSL framework by taking into account the label information of observed samples.

B. Semi-Supervised Low-Rank Representation

In this subsection, we incorporate the label information of observed samples into the original LRR framework, and propose a new model called *Semi-Supervised Low-Rank Representation* (SSLRR). The key idea of SSLRR is to preserve the known global geometric structure of data when solving the LRR problem. Specially, since we still aim to group the samples into one cluster if and only if they lie on the same subspace, the collection of all coefficient vectors $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n]$ should remain low-rank and have a block-diagonal structure as in LRR. As we have some labeled samples, we can directly enforce the coefficients Z_{ij} between two labeled data points from different clusters to be zero. Therefore, the SSLRR model solves the following problem:

$$\begin{aligned}\min_{Z,E} \quad & \|Z\|_* + \lambda\|E\|_{2,1} \\ s.t. \quad & X = XZ + E, \\ & Z^T\mathbf{1} = \mathbf{1}, \\ & Z_{ij} = 0, \quad \forall (i,j) \in \Omega,\end{aligned} \tag{2}$$

where $\mathbf{1}$ is an all-one vector, Ω is the set of edges between two labeled samples from different classes, whereby $(i,j) \in \Omega$ indicates that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are not in the same class. As we can see from (2), similar to LRR, SSLRR also seeks the lowest-rank representation Z^* among all the data points. Meanwhile, by enforcing $Z_{ij} = 0, \forall (i,j) \in \Omega$, it makes use of the label information to help prevent the block-diagonal structure of Z^* from being destroyed in real world scenarios. By enforcing the sum-to-one constraint on the rows of the weight matrix, we hope to obtain the invariance to translations. The same trick is also used by existing methods, such as the popular Locally Linear Embedding (LLE) [3]. Since the SSLRR problem (2) is convex, it can be efficiently solved by fast first-order optimization methods, as we describe next.

C. Solving SSLRR via LADMAP

Recently, various methods have been proposed to solve the low-rank and sparse matrix recovery problem. In this paper, we adopt the Linearized Alternating Direction Method with Adaptive Penalty (LADMAP) [48] for its efficiency. LADMAP is a general method for solving convex programs with linear constraints. At each iteration, it first approximates the augmented Lagrangian function by linearizing the quadratic term and adding a proximal term. Then, it minimizes over the approximated function to update variables alternately. To apply LADMAP to our problem, we first define the linear mappings:

$$\mathcal{A}(Z) = \begin{pmatrix} \mathrm{vec}(XZ) \\ Z^T\mathbf{1} \\ \mathcal{P}_\Omega(Z) \end{pmatrix}, \quad \mathcal{B}(E) = \begin{pmatrix} \mathrm{vec}(E) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} \mathrm{vec}(X) \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix},$$

where $\mathrm{vec}(\cdot)$ is the vectorization operator that stacks columns of a matrix into a vector. $\mathcal{P}_\Omega(Z)$ is the projection operator that extracts the entries in Z whose indices are in Ω. Then, Eq. (2) can be rewritten as:

$$\begin{aligned}\min_{Z,E} \quad & \|Z\|_* + \lambda\|E\|_{2,1} \\ s.t. \quad & \mathcal{A}(Z) + \mathcal{B}(E) = \boldsymbol{c}.\end{aligned} \tag{3}$$

Then, applying LADMAP (Algorithm 1 in [48]) to the standard form (3) yields the following updating rules.

Updating Z. First, we update Z as:

$$Z_{k+1} = \mathrm{argmin}_Z \, \|Z\|_* + \frac{\beta_k \eta_\mathcal{A}}{2}\|Z - \tilde{Z}_k\|_F^2, \tag{4}$$

where k is the iteration number,

$$\tilde{Z}_k = Z_k - \mathcal{A}^*\left(\boldsymbol{y}_k + \beta_k[\mathcal{A}(Z_k) + \mathcal{B}(E_k) - \boldsymbol{c}]\right)/(\beta_k\eta_\mathcal{A}),$$

$\beta_k > 0$ is the penalty parameter, $\eta_\mathcal{A}$ is a relaxation parameter that satisfies $\eta_\mathcal{A} > \|\mathcal{A}\|_2$, in which $\|\mathcal{A}\|_2 = \max\limits_{Z \neq 0} \|\mathcal{A}(Z)\|_F / \|Z\|_F$ is the operator norm of $\mathcal{A}$, and $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$.

Here, we note that $\|\mathcal{A}\|_2 \leq \sqrt{\|X\|_2^2 + n + 1}$ and $\mathcal{A}^*(\boldsymbol{w}) = X^T\mathrm{mtx}(\boldsymbol{w}_1) + \mathbf{1}\boldsymbol{w}_2^T + \mathcal{P}_\Omega^*(\boldsymbol{w}_3)$, where

$$\boldsymbol{w} = \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w}_3 \end{pmatrix}$$

and the lengths of $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$ are dn, n, and $|\Omega|$, respectively. Additionally, $\mathrm{mtx}(\cdot)$ is the operator that reshapes an $dn \times 1$ vector into a $d \times n$ matrix, $\mathcal{P}_\Omega^*(\cdot)$ is the adjoint operator of $\mathcal{P}_\Omega(\cdot)$ which maps a $|\Omega| \times 1$ vector to an $n \times n$ matrix by inserting the entries of the vector at places of the matrix whose indices are in Ω. The rest of the entries of the matrix are all zeros. Roughly speaking, mtx and $\mathcal{P}_\Omega^*$ can be viewed as the inverse operations of vec and $\mathcal{P}_\Omega$, respectively.

Finally, the subproblem (4) has a closed form solution given by singular value thresholding (SVT) [49]:

$$Z_{k+1} = \tilde{U}_k \max\left(\tilde{\Sigma}_k - (\beta_k\eta_\mathcal{A})^{-1}I, 0\right)\tilde{V}_k^T, \tag{5}$$

where $\tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^T$ is the singular value decomposition (SVD) of $\tilde{Z}_k$.

Updating E. Next, we update E as:

$$E_{k+1} = \mathrm{argmin}_E \, \lambda\|E\|_{2,1} + \frac{\beta_k\eta_\mathcal{B}}{2}\|E - \tilde{E}_k\|_F^2, \tag{6}$$

where

$$\tilde{E}_k = E_k - \mathcal{B}^*(\boldsymbol{y}_k + \beta_k[\mathcal{A}(Z_{k+1}) + \mathcal{B}(E_k) - \boldsymbol{c}]/(\beta_k\eta_\mathcal{B})),$$

and $\eta_\mathcal{B} > 0$ is a relaxation parameter that satisfies $\eta_\mathcal{B} > \|\mathcal{B}\|_2$, in which $\|\mathcal{B}\|_2 = \max\limits_{E \neq 0} \|\mathcal{B}(E)\|_F / \|E\|_F$ is the operator norm

Algorithm 1 LADMAP for Solving the SSLRR Problem

Input: Data matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, balance parameter λ, and indices set Ω.

Steps:

1: Set parameters $0 < \varepsilon_1 \ll 1$, $0 < \varepsilon_2 \ll 1$, $\beta_{\max}$, $\rho_0 \in [1, 1.5]$, $\eta_{\mathcal{A}} = \|X\|_2^2 + n + 1$, $\eta_{\mathcal{B}} = 1$.
2: Initialize $Z_0 = 0$, $E_0 = 0$, $\boldsymbol{y}_0 = \boldsymbol{0}$, $\beta_0 \in (0, 1)$, $k \leftarrow 0$.
3: **Do**
4: Update Z by (5).
5: Update E by (7).
6: Update $\boldsymbol{y}$ by (8).
7: Update β by (9).
8: **While** $\|\mathcal{A}(Z_{k+1}) + \mathcal{B}(E_{k+1}) - \boldsymbol{c}\|_2 > \varepsilon_1$ or $\beta_k \max(\sqrt{\eta_{\mathcal{A}}}\|Z_{k+1} - Z_k\|_F, \sqrt{\eta_{\mathcal{B}}}\|E_{k+1} - E_k\|)/\|\boldsymbol{c}\|_2 > \varepsilon_2$

Output: The optimal solution (Z_{k+1}, E_{k+1}).

of $\mathcal{B}$, and $\mathcal{B}^*$ is the adjoint operator of $\mathcal{B}$. We note that $\|\mathcal{B}\|_2 \leq 1$ and $\mathcal{B}^*(\boldsymbol{w}) = \text{mtx}(\boldsymbol{w}_1)$, where $\boldsymbol{w}_1$ is the sub-vector of $\boldsymbol{w}$ consisting of the first dn entries of $\boldsymbol{w}$.

Finally, the subproblem (6) also has a closed form solution [4]. Let $\boldsymbol{e}_{k+1,i}$ and $\tilde{\boldsymbol{e}}_{k,i}$ be the i-th column of E_{k+1} and $\tilde{E}_k$, respectively, we have:

$$\boldsymbol{e}_{k+1,i} = \max\left(1 - \lambda/(\beta_k \eta_{\mathcal{B}} \|\tilde{\boldsymbol{e}}_{k,i}\|_2), 0\right) \tilde{\boldsymbol{e}}_{k,i}. \tag{7}$$

Updating $\boldsymbol{y}$. Third, the Lagrange multiplier $\boldsymbol{y}$ is updated as:

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \beta_k[\mathcal{A}(Z_{k+1}) + \mathcal{B}(E_{k+1}) - \boldsymbol{c}]. \tag{8}$$

Updating β. Fourth, the penalty β is updated adaptively as follows:

$$\beta_{k+1} = \min(\beta_{\max}, \rho\beta_k), \tag{9}$$

where

$$\rho = \begin{cases} \rho_0, & \beta_k \max(\sqrt{\eta_{\mathcal{A}}}\|Z_{k+1} - Z_k\|_F, \\ & \quad \sqrt{\eta_{\mathcal{B}}}\|E_{k+1} - E_k\|)/\|\boldsymbol{c}\|_2 \leq \varepsilon_2, \\ 1, & \text{otherwise}, \end{cases} \tag{10}$$

where $\rho_0 \geq 1$ is a constant and $0 < \varepsilon_2 \ll 1$ is a threshold.

The algorithm is summarized in Algorithm 1, in which the above iteration stops when the convergence criteria are met. More details about LADMAP can be referenced to [48].

Computational Complexity. Though label information is integrated, our SSLRR model has the same computational cost as the original LRR model. Specially, the computational cost of Algorithm 1 is mainly determined by updating the variables Z, E, and $\boldsymbol{y}$. For ease of analysis, let r_X be the lowest rank for X we can find with our algorithm, and k denote the number of iterations. Without loss of generality, we assume the sizes of X are $d \times n$ ($d < n$) in the following. In each iteration, SVT is used to update the low-rank matrix whose total complexity is $O(r_X n^2)$ when we use partial SVD. Then we compute XZ_{k+1} as $((X\hat{U}_k)\hat{\sum}_k)\hat{V}_k^T$ and employ soft thresholding to update the sparse error matrix E with the total complexity of $O(r_X n^2)$. The complexity of updating the Lagrange multiplier $\boldsymbol{y}$ is $O(dn)$. So, the total cost of Algorithm 1 is $O(2kr_X n^2 + dn) = O(kr_X n^2)$.

IV. Graph Construction via Semi-Supervised Low-Rank Representation

Given a data matrix X, let $G = (V, E)$ be a graph associated with a weight matrix $W = \{w_{ij}\}$, where $V = \{\boldsymbol{v}_i\}_{i=1}^n$ is the node set, $E = \{e_{ij}\}$ is the edge set, and w_{ij} is the weight of edge e_{ij} linking two nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The problem of graph construction is to determine the graph weight matrix W. In this paper, we are primarily concerned about the estimation of an undirected graph with nonnegative weights.

After solving problem (2), we may obtain the optimal coefficient matrix Z^*. Since each data point is represented by all the other samples, Z^* naturally characterizes the relationships among samples. Further, the low rank term of (2) encourages the coefficients of samples coming from the same affine subspace to be highly correlated and fall into the same cluster, so that Z^* captures the global structure (i.e. the subspaces) of the whole data. However, the immediate output of SSLRR is a directed graph (asymmetric similarity between nodes). In order to make use of existing graph-based semi-supervised classification algorithms, a simple symmetrization step is often employed to convert the directed graph to an undirected one:

$$W = (|Z^*| + |Z^*|^T)/2. \tag{11}$$

However, the above step discards useful information conveyed by the edge directions. To preserve the valuable structural information from directed pairwise relationship between vertices, we adopt the Co-linkage Similarity (CS) [50], which considers a second-order random walk on the directed graph. The key idea is that, if from node i, a random walker has a higher probability to reach node j, then there is a larger similarity between i and j. To this end, [50] defines four types of process, i.e, co-citation, co-reference, passage ($i \to j$) and passage ($j \to i$) on a directed graph, as shown in Figure 2. Each type defines a similarity between vertex pairs on a directed graph. Thus, the effective similarity between two

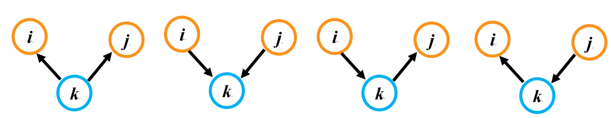


Fig. 2: Four fundamental second order processes on a directed graph [50]. From left to right: vertices i and j are co-cited by vertex k; vertices i and j co-reference vertex k; passage from vertices i to j; passage from vertices j to i.

nodes on the directed graph is given by

$$W = Z^{*T}Z^* + Z^*Z^{*T} + Z^*Z^* + Z^{*T}Z^{*T}, \tag{12}$$

where the four terms represent co-citation, co-reference, passage ($i \to j$), and passage ($j \to i$), respectively. Clearly, the obtained graph weight matrix is symmetrical. Further, in this way we enhance the pairwise relationships between the vertices by taking into account the mutual link reinforcement and make the topological structure more lucid. We refer readers to Section V-C for an empirical comparison of the

Algorithm 2 Graph Construction via SSLRR

Input: Data matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, balance parameter λ.

Steps:

1: Normalize all the samples $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i / \|\boldsymbol{x}_i\|_2$ to obtain $\hat{X} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_n\}$.

2: Assign the adjacency matrix Ω according to the label information of observed samples.

3: Solve the following problem by Algorithm 1:

$$
\begin{aligned}
(Z^*, E^*) = \arg\min_{Z,E} \; & \|Z\|_* + \lambda \|E\|_1, \\
\text{s.t. } & \hat{X} = \hat{X} Z + E, \mathbf{1}^T = \mathbf{1}^T Z, \\
& Z_{ij} = 0, (i,j) \in \Omega.
\end{aligned}
$$

4: Construct the graph weight matrix W by

$$W = Z^{*T} Z^* + Z^* Z^{*T} + Z^* Z^* + Z^{*T} Z^{*T}.$$

Output: The weight matrix W of SSLRR-graph.

graph weight matrix construction methods. More details about Co-linkage Similarity can be found in [50].

Finally, we summarize our method for constructing the SSLRR-graph in Algorithm 2.

V. Experiments

To demonstrate the effectiveness of the proposed method, we conduct benchmark experiments in two scenarios, namely, manifold clustering on synthetic data and semi-supervised classification on real image data. Besides LRR, to further show the good generalization ability of the proposed semi-supervised graph learning framework, we adapt our method into two other existing self-representation methods: ℓ_1-graph [12], and NNLRS-graph [14]. For these two methods, the label information is incorporated in the same way as in the LRR-graph.[1] We term the resultant methods as *Semi-supervised ℓ_1-graph* (denoted as **SSL1-graph**) and *Semi-supervised NNLRS-graph* (denoted as **SSNNLRS-graph**), respectively. As a result, we have six graph learning methods in our experiments (three unsupervised methods and three semi-supervised methods), as shown in Table I.

Note that each semi-supervised graph learning method shares the same parameter as its corresponding unsupervised version. Specifically, for ℓ_1-graph and SSL1-graph, as well as LRR-graph and SSLRR-graph, one needs to choose the weight λ for the error term. For NNLRS-graph and SSNNLRS-graph, the parameters for the two error terms, β and λ, need to be determined. Compared to the unsupervised method, the only difference in the semi-supervised version is the additional constraint on the label information. Therefore, to better study the benefits of considering the label information in the graph learning stage, we decide to use the same parameter values for both methods in each experiment, and tune the parameters according to each dataset. Further, for fair comparison, we first tune the parameters for each unsupervised learning method, and then apply the same parameters to its semi-supervised version. In this way, we make sure that the choices of parameters do not favor our proposed semi-supervised methods.

Unsupervised	ℓ_1-graph	LRR-graph	NNLRS-graph
Semi-Supervised	SSL1-graph	SSLRR-graph	SSNNLRS-graph

TABLE I: The six graphs used in our experiments.

A. Manifold Clustering on Synthetic Data

We consider the manifold clustering application by applying standard spectral clustering algorithm to the learned weight matrix W. The experiment is conducted on a series of synthetic data sets which contain Gaussian noises and data corruption. First, we evenly sample 900 noise-free points from three sinusoid manifolds. Then, the sample points are embedded into a 100-dimensional space and occupy the first two dimensions. We further add Gaussian noise with zero mean and variance 0.01 in all the 100 dimensions. Finally, we randomly select 10% of the samples and corrupt each sample with a much higher Gaussian noise with zero mean and variance $0.3\|\boldsymbol{x}\|_2$, where $\|\boldsymbol{x}\|_2$ is the ℓ_2-norm of the sample. One example of the noisy data is shown in Figure 3.

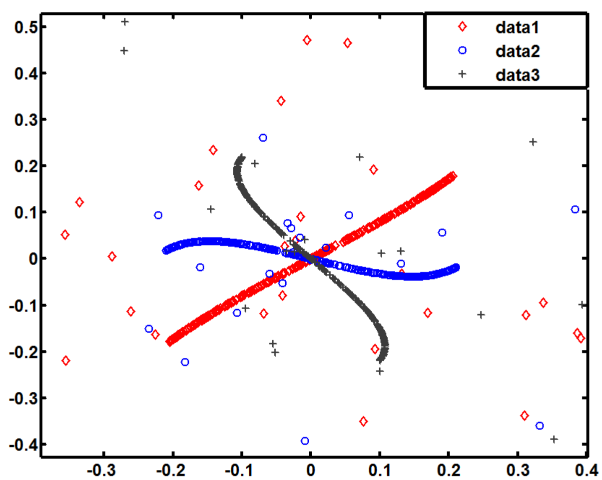


Fig. 3: An example of 900 points sampled in $\mathbb{R}^2$ and embedded in 100-D space with added noise and corruption.

After building six graphs with different methods, Normalized Cuts [51] is used for data segmentation. To apply the semi-supervised graph learning methods, we randomly select and label 30% of the observed samples from each cluster. As clustering methods do not provide the class label of each cluster, we use a post-processing step to assign each cluster a label: Given the ground truth classification results, the label of a cluster is the index of the ground truth class that contributes the maximum number of samples to the cluster. In this way, we can obtain the segmentation accuracy by computing the percentage of correctly classified samples, which is shown in Figure 4. It shows that the semi-supervised learning methods significantly outperform their unsupervised counterparts,

[1] Though there are other self-presentation methods (such as [13], [15]), most of them are derived from ℓ_1-graph and LRR-graph. Therefore, we choose ℓ_1-graph, LRR-graph and NNLRS-graph as our reference methods.

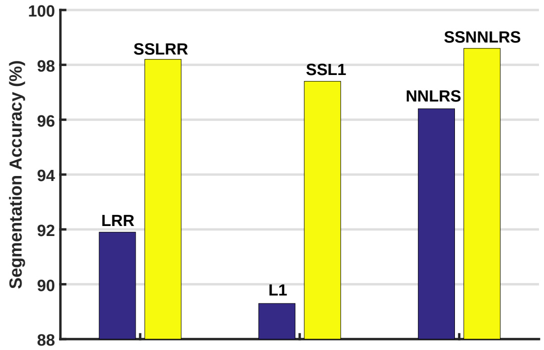


Fig. 4: Clustering results on the synthetic data.

suggesting that the label information can indeed improve the performance of self-representation graph learning methods.

Sensitivity to the percentage of labeled samples. To further evaluate the influence of label information, we vary the percentage of labeled samples from 10% to 60% for both LRR-graph and SSLRR-graph. As shown in Figure 5, SSLRR-graph consistently outperforms LRR-graph. In addition, the segmentation accuracy of SSLRR-graph increases steadily as the percentage of labeled samples increases, resulting a larger performance gap between these two methods. This suggests that adding more label information can help better preserve the underlying global data structures while we construct the graph. In Figure 1, we further visualize the graph weight matrix obtained by the LRR-graph and the SSLRR-graph. As one can see, with the increase of the percentage of labeled samples, block-diagonal structure of the constructed graph becomes more evident.

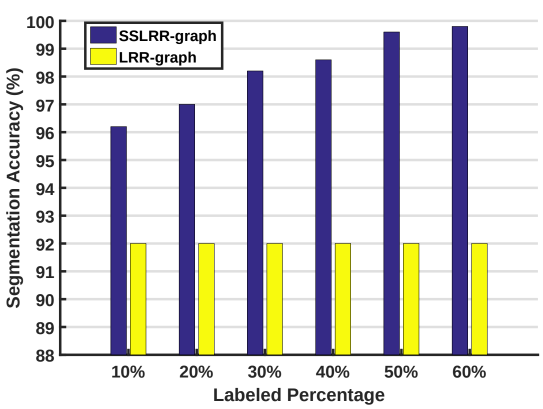


Fig. 5: Clustering results on the synthetic data with the variation of labeled percentage.

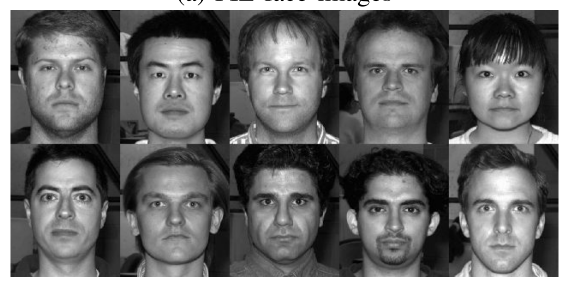
B. Semi-supervised Classification on Real Image Datasets

Besides synthetic data, we also evaluate our proposed semi-supervised graph-learning methods in applications including face recognition, handwritten digit recognition, and object recognition under the transductive learning setting. We adopt the popular *Local and Global Consistency* (LGC) [2] as the classification framework. Specifically, LGC builds upon an undirected graph, and utilizes the graph and known labels to recovery a continuous classification function $F \in \mathbb{R}^{|V|\times c}$ by optimizing the following energy function:

$$\min_{F\in\mathbb{R}^{|V|\times c}} tr\{F^T L_W F + \mu(F-Y)^T(F-Y)\}, \qquad (13)$$

where c is the number of classes, $Y \in \mathbb{R}^{|V|\times c}$ is the label matrix, in which $Y_{ij} = 1$ if sample $\boldsymbol{x}_i$ is associated with label j for $j \in \{1, 2, \cdots, c\}$, and $Y_{ij} = 0$ otherwise. L_W is the normalized graph Laplacian $L_W = D^{-1/2}(D-W)D^{-1/2}$, in which D is a diagonal matrix with $D_{ij} = \sum_j W_{ij}$. The weight $\mu \in [0, \infty)$ balances the local fitting and the global smoothness of the function F. As suggested in [2], we fix $\mu = 0.01$ in all the experiments.

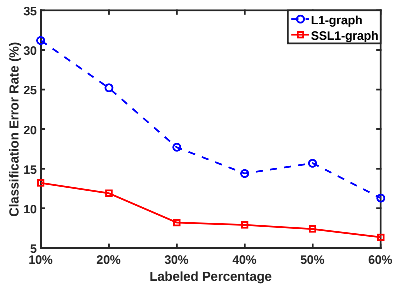
(a) PIE face images



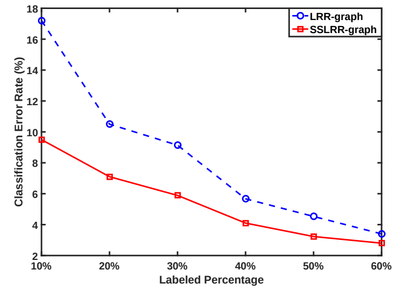
(b) YaleB face images

Fig. 6: Sample images in the PIE and YaleB face datasets.

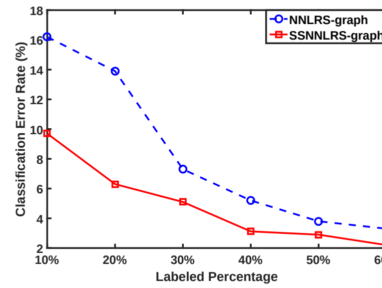
1) Evaluation on Face Recognition: We investigate the performance of our semi-supervised graph learning for face recognition on two well-known public face datasets: YaleB and CMU PIE. In the literature, both datasets have been frequently used to evaluate the performance of semi-supervised learning methods. The YaleB face database consists of 38 individuals, and each subject has around 64 near frontal images under different illuminations. We use the cropped images of first 15 individuals for our experiment, and resize them to 32×32 pixels. The PIE face database contains 41368 images of 68 people, each person under 13 different poses, 43 different illumination conditions and with 4 different expressions. In our experiments, we only use the frontal images with neural expression of the first 20 individual. The images are cropped using affine transformations based on locations of the eyes and nose, and resize them to 100×100 pixels. Sample images from these two face datasets are shown in Figure 6. Empirically, we have found that images of each subject in YaleB and PIE has a roughly linear subspace structure.



(a) L1-graph vs SSL1-graph

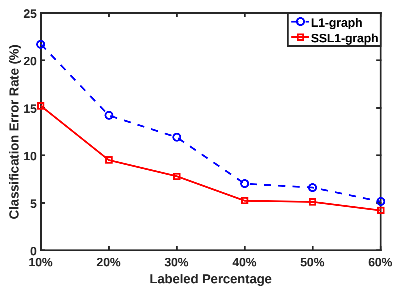


(b) LRR-graph vs SSLRR-graph

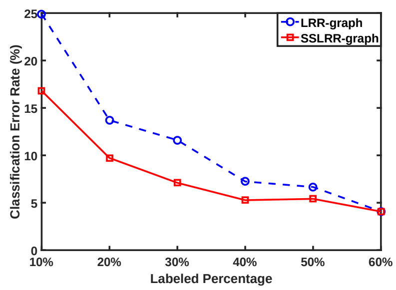


(c)NNLRS-graph vs SSNNLRS-graph

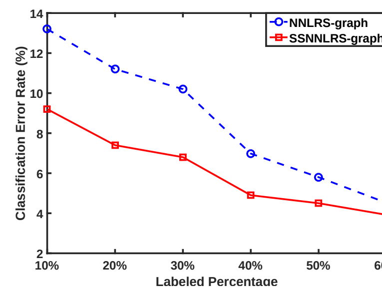
Fig. 7: Face recognition error rates on the YaleB dataset.



(a) L1-graph vs SSL1-graph



(b) LRR-graph vs SSLRR-graph



(c)NNLRS-graph vs SSNNLRS-graph

Fig. 8: Face recognition error rates on the PIE dataset.

In Figure 7 and Figure 8, we report the classification accuracy based on six different graph learning methods on YaleB and PIE datasets. The percentage of the labeled samples varies from 10% to 60%. As expected, in most cases, the classification error rates for all methods decrease with the increase of the percentage of labeled samples. More importantly, in *all* cases, the semi-supervised graph learning methods outperform their unsupervised counterparts. For example, by incorporating the labels of 10% data samples in YaleB into the ℓ_1-graph, the classification error rate is reduced from 31.2% to 13.2%. This demonstrates the importance of incorporating label information in graph construction.

2) Evaluation on Handwritten Digit Recognition: In this experiment, we address the handwritten digit recognition problem on the USPS dataset. The dataset contains normalized grey scale images of size 16×16, divided into a training set of 7291 images and a test set of 2007 images. Following the setting of previous works, we only use the images of digits 1, 2, 3, and 4 as the four classes, which have 1296, 926, 824 and 852 samples, respectively. Some sample images from USPS digit dataset are shown in Figure 9.

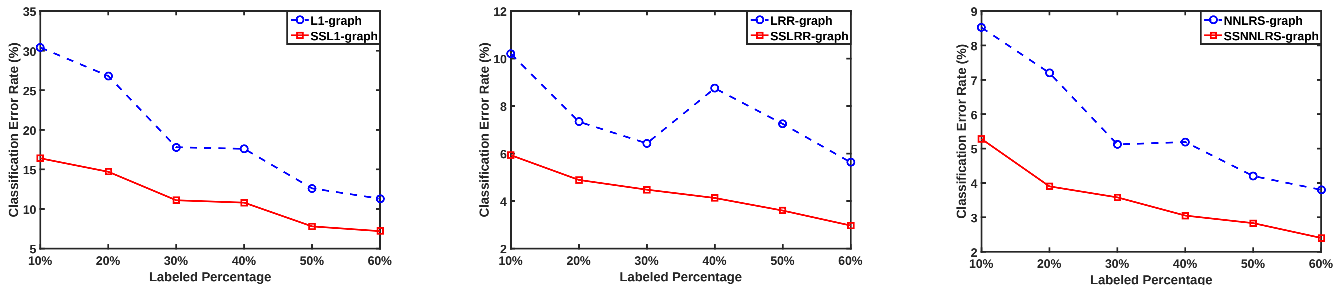
Figure 10 shows the recognition error rates of all the six methods on the USPS dataset. As we can see, the proposed semi-supervised graph learning methods consistently outperform the unsupervised versions. Moreover, as the percentage of labeled samples increases, the classification error rates for SSL1-graph, SSLRR-graph, and SSNNLRR-graph decreases monotonically. This suggests that labeled samples help recover the underlying data structure of the learned graphs. In Figure 11, we further compare the weight matrices obtained by

Fig. 9: Sample images in the USPS dataset.

the LRR-graph and SSLRR-graph on USPS dataset. We can see that, with the increase of the number of labeled samples, the block-diagonal structure of the graphs is indeed better preserved.

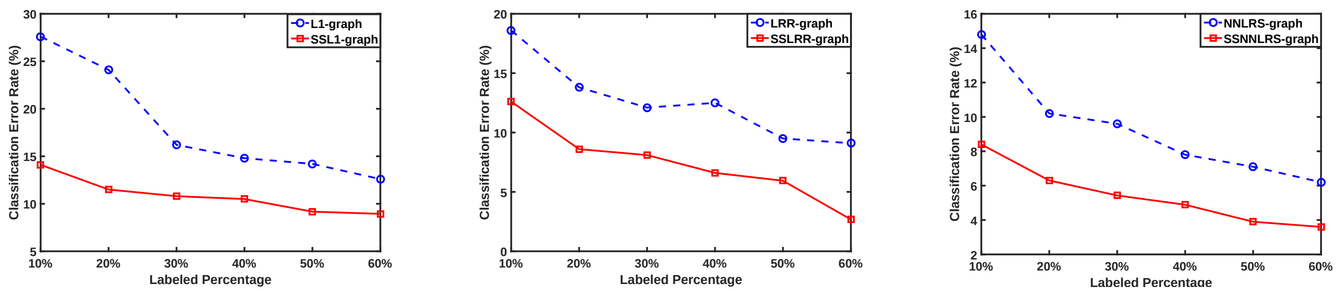
3) Evaluation on Visual Object Recognition: We verify the importance of label information for graph learning for nonlinear manifolds by conducting visual object recognition experiments on the COIL20 dataset. The dataset contains 20 objects. The images of each objects were taken 5 degrees apart as the object is rotated on a turntable, resulting in 72 images for each object. The size of the grayscale image is 32×32 pixels. Figure 12 shows some sample images in the COIL20 dataset. We have empirically found that data samples in the COIL20 dataset lie on non-linear manifolds. This makes the graph learning and classification tasks more challenging, because most self-representation methods (such as ℓ_1-graph, LRR-graph, NNLRS-graph) are based on linear models.

Figure 13 reports the classification results of the six graph learning methods. As we can see, as the percentage of labeled samples increases, the classification error rates for our semi-



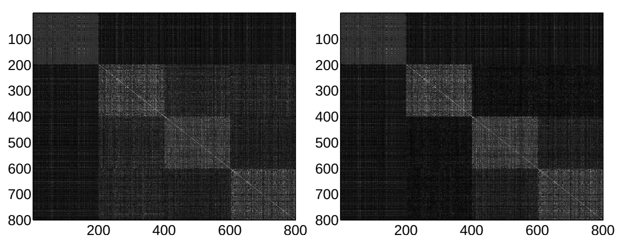
(a) ℓ_1-graph vs SSL1-graph (b) LRR-graph vs SSLRR-graph (c)NNLRS-graph vs SSNNLRS-graph

Fig. 10: Handwritten digit recognition error rates on the USPS dataset.

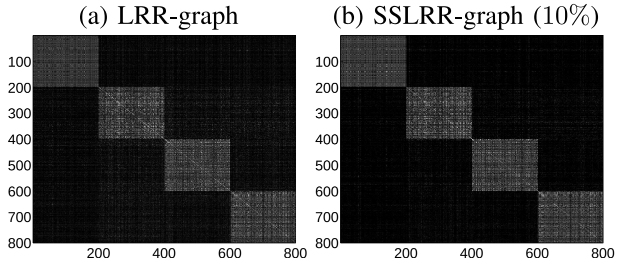


(a) ℓ_1-graph vs SSL1-graph (b) LRR-graph vs SSLRR-graph (c)NNLRS-graph vs SSNNLRS-graph

Fig. 13: Visual object recognition error rates on the COIL20 dataset.



(a) LRR-graph (b) SSLRR-graph (10%)



(c) SSLRR-graph (30%) (d) SSLRR-graph (60%)

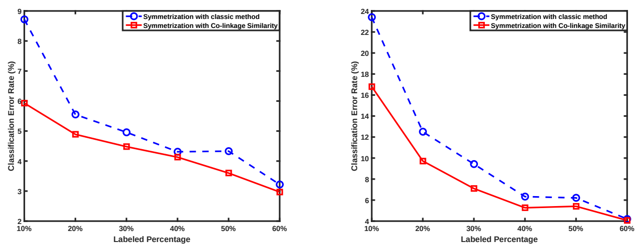
Fig. 11: Visualization of unsupervised LRR-graph and semi-supervised SSLRR-graph with 10%, 30%, and 60% of labeled data from USPS database.

supervised learning methods decrease. Further, they significantly outperform their corresponding unsupervised versions. For example, the error rate for LRR-graph on the COIL20 dataset with 60% labeled samples is 9.10%, whereas the error rate for SSLRR-graph is only 2.68%. This again demonstrates the importance of the incorporating label information in graph learning. By enforcing the weights between points from different categories to be zero, we can preserve part of the global data structure.

Fig. 12: Sample images in the COIL dataset.

C. Influence of the Graph Construction Methods

In this subsection, we evaluate the influence of Co-linkage Similarity (CS) [50] on the graph construction. Taking SSLRR-graph as example, we use the classic method (i.e., Eq. (11)) and the Co-linkage Similarity (i.e., Eq. (12)) to symmetrize the directed graph, and obtain SSLRR-graph$_{classic}$ and SSLRR-graph$_{CS}$, respectively. We evaluate SSLRR-graph$_{classic}$ and SSLRR-graph$_{CS}$ under the same settings on both the USPS dataset and the PIE dataset. The classification error rate on these two datasets are shown in Figure 14. As we can see that, the Co-linkage Similarity does improve the performance of SSLRR-graph, especially when the percentage of labeled samples is small. Generally, as the



(a) USPS (b) PIE

Fig. 14: Classification error rates using different graph weight matrix construction methods.

labeled percentage increases, the influence becomes weaker.

VI. Conclusion

In this paper, we propose a new graph learning framework, which is able to obtain highly informative graphs for graph-based SSL methods. Different from existing graph learning methods, our method explicitly takes advantage of label information in both the graph learning and label propagation stages. In particular, by restricting the coefficient between any two labeled samples from different classes to be zero, our method seamlessly incorporates the label information of the data samples into any self-representation methods (e.g., ℓ_1-graph, LRR-graph, and NNLRR-graph), and keep the same computational cost. As a result, the new graphs can better capture the global geometric structure of the data, therefore is more informative and discriminative, especially when the data is subject to noise, the subspaces are not independent, or the data points lie in nonlinear manifolds. Experiment results on both synthetic and real datasets demonstrate that the label information indeed helps preserve the block-diagonal structure of the coefficient matrices, and significantly improves the performance of existing graph learning methods.

As for future work, we plan to further investigate efficient algorithms for constructing large-scale SSLRR-graphs. Also, current methods conduct label propagation for classification after graph construction. It is interesting to develop principled method to solve the graph construction and label propagation problems at the same time.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 888–905, 2004. 1

[2] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2004, pp. 595–602. 1, 7

[3] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locality linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. 1, 2, 4

[4] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE TPAMI*, vol. 35, no. 1, pp. 171–184, January 2013. 1, 2, 3, 5

[5] T. Jebara, J. Wang, and S. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML*, 2009, pp. 441–448. 2, 3

[6] D. Kong, C. H. Ding, H. Huang, and F. Nie, "An iterative locally linear embedding algorithm," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, New York, NY, USA, 2012, pp. 1647–1654. 2, 3

[7] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005. 2

[8] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006. 2

[9] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baranuik, "Single-pixel imaging via compressed sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, 2008. 2

[10] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE TPAMI*, vol. 31, no. 2, pp. 210–227, 2008. 2

[11] E. Candés, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM*, vol. 58, no. 3, 2011. 2

[12] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with ℓ_1-graph for image analysis," *IEEE TIP*, vol. 19, no. 4, 2010. 2, 3, 6

[13] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *CVPR*, 2011, pp. 792–801. 2, 3, 6

[14] L. Zhuang, H. Gao, Z. Liu, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *CVPR*, 2012. 2, 3, 6

[15] Y. Zheng, X. Zhang, S. Yang, and L. Jiao, "Low-rank representation with local constraint for graph construction," *Neurocomputing*, vol. 122, no. 24, pp. 398–405, December 2013. 2, 6

[16] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, 2011. 2

[17] M. Rohban and H. R. Rabiee, "Supervised neighborhood graph construction for semi-supervised classification," *Pattern Recognition*, vol. 45, pp. 1362–1372, April 2012. 2

[18] L. Berton and A. de Andrade Lopes, "Graph construction based on labeled instances for semi-supervised learning," in *22nd International Conference on Pattern Recognition (ICPR)*, 2014. 2

[19] P. S. Dhillon, P. P. Talukdar, and K. Crammer, "Inference driven metric learning (idml) for graph construction," 2010, university of Pennsylvania Department of Computer and Information Science Technical Report No.MS-CIS-10-18. 2

[20] M. Maier, U. von Luxburg, and M. Hein, "Influence of graph construction on graph-based clustering measures," in *NIPS*, 2008. 3

[21] P. P. Tlukdar, "Topics in graph construction for semi-supervised learning," Department of Computer & Information Science, University of Pennsylvania, Tech. Rep. MS-CIS-09-13, August 2009. [Online]. Available: http://repository.upenn.edu/cis_reports/936 3

[22] D. A. Vega-Oliveros, L. Berton, A. Mantini Eberle, A. de Andrade Lopes, and L. Zhao, "Regular graph construction for semi-supervised learning," *2nd International Conference on Mathematical Modeling in Physical Sciences 2013*, vol. 490, no. 1, 2014. 3

[23] L. Berton and A. de Andrade Lopes, "Graph construction for semi-superivsed learning," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015, pp. 4343–4344. 3

[24] J. Wang, F. Wang, C. Zhang, H. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE TPAMI*, vol. 31, no. 9, pp. 1600–1615, 2009. 3

[25] K. Ozaki, M. Shimbo, M. Komachi, and Y. Matsumoto, "Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 154–162. 3

[26] M. Zhang, C. Ding, and D. Kong, "Collective kernel construction in noisy environment," in *SIAM International Conference on Data Mining 2013 (SMD 2013)*, Austin, USA, 2013, pp. 64–72. 3

[27] Y.-M. Zhang, K. Huang, G. Geng, and C.-L. Liu, *Fast kNN Graph Construction with Locality Sensitive Hashing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 660–674. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40991-2_42 3

[28] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 577–586. 3

[29] D. C. Anastasiu and G. Karypis, "L2knng: Fast exact k-nearest neighbor graph construction with l2-norm pruning," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management(CIKM)*, 2015. 3

[30] Y. Park, S. Park, S.-g. Lee, and W. Jung, "Scalable k-nearest neighbor graph construction based on greedy filtering," in *Proceedings of the 22th international conference on World wide web*, 2013, pp. 227–228. 3

[31] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li, "Scalable k-nn graph construction for visual descriptors," in *CVPR*, 2012. 3

[32] Y. Fang, R. Wang, B. Dai, and X. Wu, "Graph-based learning via auto-grouped sparse regularization and kernelized extension," *TKDE*, vol. 27, no. 1, pp. 142–154, 2015. 3

[33] Y. Yang, Z. Wang, J. Yang, J. Wang, S. Chang, and T. S. Huang, "Data clustering by laplacian regularized l1-graph," in *AAAI*, 2014, pp. 3148–3149. 3

[34] S. Han, H. Huang, H. Qin, and D. Yu, "Locality-preserving l1-graph and its application in clustering," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 813–818. 3

[35] G. Zhou, Z. Lu, and Y. Peng, "L1-graph construction using structured sparsity," *Neurocomputing*, vol. 120, pp. 441–452, 2013. 3

[36] S. Han and H. Qin, "Structure aware l1-ggraph for data clustering," in *AAAI*, 2016, pp. 4214–4215. 3

[37] L. Zhuang, S. Gao, J. Tang, J. Wang, Z. Liu, Y. Ma, and N. Yu, "Constructing a nonnegative low-rank and sparse graph with data-adaptive features," *IEEE TIP*, vol. 24, no. 11, pp. 3717–3728, 2015. 3

[38] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *CVPR*, 2014, pp. 3818–3825. 3

[39] S. Yang, Z. Feng, H. Liu, and L. Jiao, "Semi-supervised classification via kernel low-rank representation graph," *Knowledge-Based Systems*, vol. 69, pp. 150–158, 2014. 3

[40] Y. Zheng, X. Zhang, S. Yang, and L. Jiao, "Low-rank representation with local constraint for graph construction," *Neurocomputing*, vol. 122, pp. 398–405, 2013. 3

[41] X. Lu, Y. Wang, and Y. Yuan, "Graph-regularized low-rank representation for destriping of hyperspectral images," *Neurocomputing*, vol. 51, no. 7, pp. 4009–4018, 2013. 3

[42] Y. Peng, B.-L. Lu, and S. Wang, "Enhanced low-rank representation via sparse manifold adaption for semi-supervised learning," *Neural Networks*, vol. 65, pp. 1–17, 2015. 3

[43] L. Zhuang, J. Wang, Z. Liu, A. Yang, Y. Ma, and N. Yu, "Locality-preserving low-rank representation for graph construction from nonlinear manifolds," *Neurocomputing*, vol. 175, pp. 715–722, 2016. 3

[44] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *ECCV*, 2012, pp. 347–360. 3

[45] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace lasso," in *ICCV*, 2013, pp. 1345–1352. 3

[46] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin, "Correntropy induced L2 graph for robust subspace clustering," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 1801–1808. 3

[47] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *CVPR*, 2014, pp. 3834–3841. 3

[48] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *NIPS*, 2011. 4, 5

[49] J.-F. Cai, E. J. Candés, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. 4

[50] H. Wang, H. Huang, and C. Ding, "Image categorization using directed graphs," in *ECCV*, 2010, pp. 762–775. 5, 6, 9

[51] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, August 2000. 6

Liansheng ZHUANG is an associate professor in the School of Information Science and Technology, USTC. He received his Ph.D. degree and bachelor's degree respectively in 2006 and 2001, from University of Science and Technology of China (USTC), China. During 2012-2013, he was a visiting research scientist in Dept. of EECS at University of California, Berkeley. His main research interesting is in computer vision, and machine learning. He is a member of ACM, IEEE, and CCF.

Zihan ZHOU is currently a faculty member in the College of Information Sciences and Technology at the Pennsylvania State University. His research interest lies in computer vision, image processing and machine learning. He received the bachelor's degree in Automation from Tsinghua University in 2007, and Ph.D degree in Electrical and Computer Engineering from University of Illinois at Urbana-Champaign in 2013. He is a member of IEEE.

Shenghua GAO is an assistant professor in ShanghaiTech University, China. He received the B.E. degree from the University of Science and Technology of China in 2008 (outstanding graduates), and received the Ph.D. degree from the Nanyang Technological University in 2012. From Jun 2012 to Aug 2014, he worked as a research scientist in Advanced Digital Sciences Center, Singapore. From Jan 2015 to June 2015, he visited UC Berkeley as a visiting scholar. His research interests include computer vision and machine learning. He has published more than 30 papers on object and face recognition related topics in many international conferences and journals, including IEEE T-PAMI,IJCV, IEEE TIP, IEEE TNNLS, IEEE TMM, IEEE TCSVT, CVPR, ECCV, etc. He has organized tutorials in VCIP2015 and ACCV2014. He was awarded the Microsoft Research Fellowship in 2010, and ACM Shanghai Young Research Scientist in 2015, and he is a recipient of National 1000 Young Talents Program in 2016.

Jingwen Yin is currently a undergraduate students in the School of the Gifted Young at University of Science and Technology of China. Her research interest lies in computer vision and data mining.

Zhouchen LIN (M'00-SM'08) received the Ph.D. degree in applied mathematics from Peking University in 2000. Currently, he is a professor at the Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University. He was a chair professor at Northeast Normal University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence and International Journal of Computer Vision. He is an IAPR Fellow.

Yi Ma (F13) is a Professor and Executive Dean of the School of Information and Science and Technology, ShanghaiTech University, China. From 2009 to early 2014, he was a Principal Researcher and the Research Manager of the Visual Computing group at Microsoft Research in Beijing. From 2000 to 2011, he was an Associate Professor at the Electrical & Computer Engineering Department of the University of Illinois at Urbana-Champaign. His main research interest is in computer vision, high-dimensional data analysis, and systems theory. He has written two textbooks "An Invitation to 3-D Vision" published by Springer in 2004, and "Generalized Principal Component Analysis" published by Springer in 2016. Yi Ma received his Bachelors degree in Automation and Applied Mathematics from Tsinghua University (Beijing, China) in 1995, a Master of Science degree in EECS in 1997, a Master of Arts degree in Mathematics in 2000, and a PhD degree in EECS in 2000, all from the University of California at Berkeley. Yi Ma received the David Marr Best Paper Prize at the International Conference on Computer Vision 1999, the Longuet-Higgins Best Paper Prize (honorable mention) at the European Conference on Computer Vision 2004, and the Sang Uk Lee Best Student Paper Award with his students at the Asian Conference on Computer Vision in 2009. He also received the CAREER Award from the National Science Foundation in 2004 and the Young Investigator Award from the Office of Naval Research in 2005. He was an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), the International Journal of Computer Vision (IJCV), and IEEE transactions on Information Theory. He is currently an associate editor of the IMA journal on Information and Inference, SIAM journal on Imaging Sciences, IEEE Signal Processing Magazine. He served as a Program Chair for ICCV 2013 and is a General Chair for ICCV 2015. He is a Fellow of IEEE. He is ranked the World's Highly Cited Researchers of 2016 by Clarivate Analytics of Thomson Reuters.