Non-Negative Low Rank and Sparse Graph for Semi-Supervised Learning

Liansheng Zhuang[1], Haoyuan Gao[1], Zhouchen Lin[2,3], Yi Ma[2], Xin Zhang[4], Nenghai Yu[1]
[1]MOE-Microsoft Key Lab., University of Science and Technology of China
[2]Microsoft Research Asia
[3]Key Lab. of Machine Perception (MOE), Peking University
[4]Dept. of Computer Sci. and Tech., Tsinghua University
Email:{lszhuang@ustc.edu.cn}

Abstract

Constructing a good graph to represent data structures is critical for many important machine learning tasks such as clustering and classification. This paper proposes a novel non-negative low-rank and sparse (NNLRS) graph for semi-supervised learning. The weights of edges in the graph are obtained by seeking a nonnegative low-rank and sparse matrix that represents each data sample as a linear combination of others. The so-obtained NNLRS-graph can capture both the global mixture of subspaces structure (by the low rankness) and the locally linear structure (by the sparseness) of the data, hence is both generative and discriminative. We demonstrate the effectiveness of NNLRS-graph in semi-supervised classification and discriminative analysis. Extensive experiments testify to the significant advantages of NNLRS-graph over graphs obtained through conventional means.

1. Introduction

For many applications of machine learning and computer vision, such as object recognition, one often lacks of sufficiently labeled training data, which are very costly and laborious to obtain. Nevertheless, today a large number of unlabeled data are widely available over the Internet. Semi-supervised learning (SSL) can utilize both limited labeled samples and rich yet unlabeled samples, and has recently received considerable attention in computer vision and machine learning communities [22]. Among current methods, graph based SSL is particularly appealing due to its empirical success in practice and its computational efficiency.

Graph based SSL relies on using a graph $G = (V, E)$ to represent data structures, where V is the set of vertices – each vertex corresponding to a data sample, and E is the set of edges associated with a weight matrix W. Label information of a subset of the samples can then be efficiently and effectively propagated to the remaining unlabeled data over the graph. Most learning methods formalize the propagation process through a regularized functional on the graph. Despite many forms used in the literature, the regularizers mainly try to accommodate the so-called *cluster assumption* [6, 17], which says that points on the same structure (such as a cluster, a subspace, or a manifold) are likely to share the same label. Since one normally does not have explicit model for the underlying manifolds, most methods approximate it by the construction of an undirected graph from observed data points. Therefore, correctly constructing a good graph that can best capture essential data structures is critical for all graph-based SSL methods [1, 23, 21]. In this paper, our main focus is on how to construct such a graph based on powerful new tools from high-dimensional statistics and optimization.

Motivations. Conceptually, a good graph should reveal the true intrinsic complexity or dimensionality of the data points (say through local linear relationships), and also capture certain global structures of the data as a whole (i.e. multiple clusters, subspaces, or manifolds). Traditional methods (such as k-nearest neighbors and Locally Linear Reconstruction [10]) mainly rely on pair-wise Euclidean distances and construct the graph by a family of overlapped local patches. The so-obtained graphs only capture local structures and cannot capture global structures of the whole data (i.e. the clusters). Moreover, these methods cannot produce datum-adaptive neighborhoods because of using fixed global parameters to determinate the graph structure and their weights. Finally, these methods are very sensitive to local data noise and errors.

According to Wright et al. [18], an informative graph should have three characteristics: high discriminating power, low sparsity, and adaptive neighborhood. Inspired by this insight, Yan et al. [20, 7] 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 datum as a sparse representation of

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

the remaining samples, and automatically selecting the most informative neighbors for each datum. 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, and hence is more powerful and discriminative. Benefitting from SR, the ℓ_1-graph is sparse, datum-adaptive and robust to data noise. Following ℓ_1-graph, other graphs were also proposed based on SR in recent years [8, 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 can greatly reduce the performance when the data is grossly corrupted. When no extra "clean data" are available, SR based methods may not be robust to noise and outliers [14].

To capture the global structure of the whole data, Liu *et al.* has proposed the low-rank representation (LRR) for the data and use it to construct the affinities of an undirected graph (here called LRR-graph) [14]. LRR-graph jointly obtains the representation of all the data under a global low-rank constraint, and 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 the 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 [18]. Moreover, as the coefficients can be negative, LRR allows the data to "cancel each other out" by substraction, which lacks physical interpretation for many visual data. In fact, non-negativity is more consistent with the biological modeling of visual data [9, 11], and often lead to better performance for data representation [11] and graph construction [8].

Contributions. Inspired by above insights, we propose to harness both sparsity and low rankness of high-dimensional data to construct an informative graph. In addition, we will explicitly enforce the representation to be non-negative so that coefficients of the representation can be directly converted to graph weights. Such a graph is called *nonnegative low-rank and sparse graph* (NNLRS-graph). Specifically, given a set of data points, we represent a data point as a linear combination of the other points, where the coefficients should be nonnegative and sparse. Nonnegativity ensures that every data point is in the convex hull of its neighbors, while sparsity ensures that the involved neighbors are fewest possible. Moreover, since we require that data vectors on the same subspace can be clustered in the same cluster, we require that the coefficient vectors of all data points collectively form a low-rank matrix. The NNLRS has many conceptual advantages: The sparsity property ensures NNLRS-graph to be sparse and capture the local low-dim linear relationships of the data. By imposing the low-rankness, the NNLRS-graph can better capture the global cluster or subspace structures of the data than SR based graphs [8, 16], and is more robust to noise and outliers.

Computing the NNLRS representation for a large set of data points is in general an NP-hard problem. Nevertheless, recent breakthroughs in high-dimensional optimization suggest that such a sparse and low-rank representation can be very effectively obtained through convex relaxation. As we will show in this paper, similar to the LRR-graph and the ℓ_1-graph, the convex program associated with the NNLRS model can be solved very efficiently by the alternating direction method (ADM) [12]. To improve its speed and scalability, we further adopt a novel method called the linearized ADM with adaptive penalty (LADMAP) [13]. LADMAP uses less auxiliary variables and no matrix inversions, hence converges faster than usual ADM and results in less computation load.

We have conducted extensive experiments on public databases for various SSL tasks. In many of these experiments, we see that the NNLRS-graph can significantly improve the performance of semi-supervised learning – often reducing the error rates by multiple folds! These results clearly demonstrate that NNLRS-graph is more informative and discriminative than graphs constructed by conventional methods.

The remainder of this paper is organized as follows. In Section 2, we give the details of how to construct a nonnegative low-rank and sparse graph. Our experiments and analysis are presented in Section 3. Finally, Section 4 concludes our paper.

2. Nonnegative Low-Rank and Sparse Graphs

2.1. 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 a basis $A = [a_1, a_2, \cdots, a_m]$:

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

where $Z = [z_1, z_2, \cdots, z_n]$ is the coefficient matrix with each z_i being the *representation* of x_i. The basis (also called dictionary) 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

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

[1]The subspaces are independent iff $\sum_{i=1}^k S_i = \bigoplus_{i=1}^k S_i$, where $\bigoplus$ is the direct sum.

where $\beta > 0$ is a parameter to trade off between low rankness and sparsity. As observed in [14], 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 atoms in the basis.

However, solving problem (2) is NP-hard. Fortunately, as suggested by matrix completion methods [4], we can solve the following relaxed convex program instead

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

where $\|\cdot\|_*$ is the nuclear norm of a matrix [3], 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

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

where $\|E\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{m}([E]_{ij})^2}$ is called the $\ell_{2,1}$-norm [15], 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) similar to [5]. Namely, the Frobenious norm $\|E\|_F$ is used instead. In this paper, we focus on the $\ell_{2,1}$-norm.

2.2. LADMAP for Solving NNLRSR

The NNLRSR problem (4) could be solved by the popular ADM method [14, 12]. 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) [13] to solve (4).

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

$$\begin{array}{l} \min_{Z,E} \|Z\|_* + \beta\|W\|_1 + \lambda\|E\|_{2,1}, \\ \text{s.t. } X = AZ + E, Z = W, W \geq 0. \end{array} \tag{5}$$

The augmented Lagrangian function of problem (5) is

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

where

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

The LADMAP is to update the variables Z, W, and E alternately, by minimizing L with other variables fixed, where the quadratic term q is replaced by its first order approximation at the previous iterate and then adding a proximal term [13]. With some algebra, the updating schemes are as follows.

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

where $\nabla_Z q$ is the partial differential of q with respect to Z, Θ, S and Ω are the singular value thresholding [3], shrinkage [12] and the $l_{2,1}$ minimization operator [14], respectively, and $\eta_1 = \|A\|_2^2$. The complete algorithm is outlined in **Algorithm** 1.

2.3. 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 2.1 and 2.2 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 the 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

Algorithm 1 Efficient LADMAP Algorithm for NNLRSR

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

Initialize: $Z_0 = W_0 = E_0 = Y_{1,0} = Y_{2,0} = 0$, $\mu_0 = 0.1$, $\mu_{\max} = 10^{10}$, $\rho_0 = 1.1$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-2}$, $\eta_1 = \|A\|_2^2$, $k = 0$.

1: **while** $\|X - AZ_k - E_k\|_F/\|X\|_F \geq \varepsilon_1$ or $\mu_k \max(\sqrt{\eta_1}\|Z_k - Z_{k-1}\|_F, \|W_k - W_{k-1}\|_F, \|E_k - E_{k-1}\|_F)/\|X\|_F \geq \varepsilon_2$ **do**

2: Update the variables as (8).

3: Update Lagrange multipliers as follows:

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

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

4: Update μ as follows:

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

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

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

6: **end while**

Output: an optimal solution (Z_k, W_k, E_k).

Algorithm 2 Nonnegative low rank and sparse graph construction

Input: Data matrix $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$, regularized parameters β and λ, threshold θ

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,

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

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

3: Normalize all column vectors of Z^* by $z_i^* = z_i^*/\|z_i^*\|_2$, and make small values under given threshold θ 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.

cluster, so that Z^* can capture the global structure (i.e. the clusters) of the whole data. Note here that, since each sample can be used to represent itself, there always exist feasible solutions even when the data sampling is insufficient, which is different from SR.

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

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

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

3. Experiments and Analysis

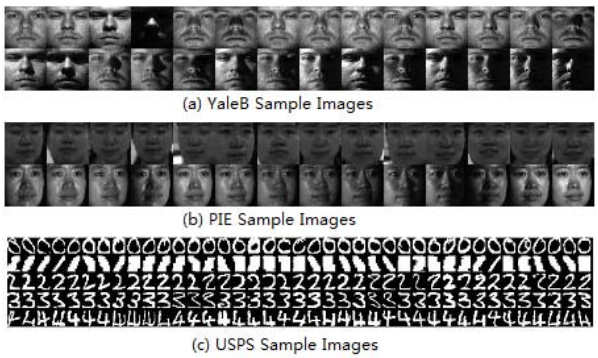
In this section, we evaluate the performance of NNLRS-graph on public databases, and compare it with currently popular graphs in the same SSL setting. Two typical semi-supervised learning tasks are considered, semi-supervised classification and semi-supervised dimensionality reduction. All algorithms are implemented in Matlab 2010. All experiments are run 50 times (unless otherwise stated) on a Windows 2008 Server, with an Intel Xeon5680 8-Core 3.50GHz processor and 16GB memory.

3.1. Experiment Settings

Databases: We select three public databases[2] for our experiments: YaleB, PIE, and USPS. YaleB and PIE are face images and USPS is digit images. We choose them because NNLRS-graph aims at extracting a linear subspace structure of data. So we have to select databases that roughly have linear subspace structures.

- **YaleB Database**: This face database has 38 individuals, each subject having around 64 near frontal images under different illuminations. We simply use the cropped images of first 15 individuals, and resize them to 32×32 pixels.
- **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.
- **USPS Database**: This handwritten digit database contains 9298 16×16 handwritten digit images in total. We only use the images of digits 1, 2, 3 and 4 as four

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



(a) YaleB Sample Images

(b) PIE Sample Images

(c) USPS Sample Images

Figure 1: Sample images used in our experiments.

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 norm.

Compared graphs: The graphs used in our experiments for comparison include:

- k**NN-graph:** We adopt Euclidean distance as our 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 [17]:** Following the lines of [17], 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 [7] we symmetrize them by $W = (|W| + |W^T|)/2$.
- ℓ_1**-graph [7]:** Following the lines of [7], we construct the ℓ_1-graph. Since the graph weights W of ℓ_1-graph is asymmetric, we also symmetrize it as suggested in [7].
- **SPG [8]:** In essence, the SPG problem is a lasso problem with the nonnegativity constraint, without considering corruption errors. Here, we use an existing toolbox[3] to solve the lasso problem, and construct the SPG graph following the lines of [8].
- **LRR-graph:** Following [14], we construct the LLR-graph, and symmetrize it as ℓ_1-graph. The parameters of LRR are the same as those in [14].
- **NNLRS-graph:** For our NNLRS-graph, the two regularization parameters are empirically set to $\beta = 0.2$ and $\lambda = 10$.

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

3.2. 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) [23] and *Local and Global Consistency* (LGC) [21] 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:

$$\min_{F\in\mathbb{R}^{|V|\times c}} tr(F^T L_W F),\ \text{s.t. } L_W F_u = 0, F_l = Y_l, \quad (10)$$

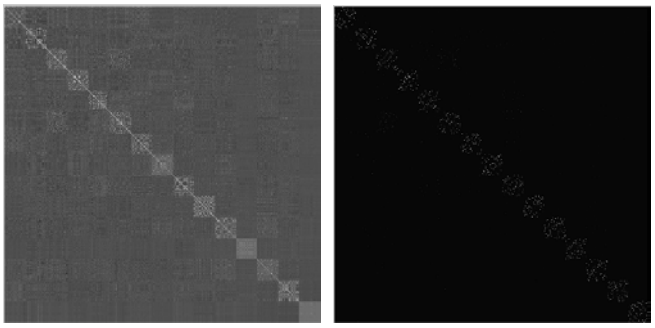
where $L_W = D - W$ is the graph Laplacian, in which D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. Instead of clamping the classification function on labeled nodes by setting hard constraints $F_l = Y_l$, LGC introduces an elastic fitness term as follows:

$$\min_{F\in\mathbb{R}^{|V|\times c}} tr\{F^T \tilde{L}_W F + \mu(F - Y)^T(F - Y)\}, \quad (11)$$

where $\mu \in [0\ \infty)$ balances the tradeoff 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 [20, 7, 18, 8]. For 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 USPS database, we randomly select 200 images for each category, and randomly label them. Different from [20, 8], the percentage of labeled samples ranges from 10% to 60%, instead of ranging from 50% to 80%. This is because the goal of SSL is to reduce the number of labeled images. So we are more interested in the performance of SSL methods with low labeling percentages. The final results are reported in Table 1 and Table 2, respectively. From these results, we can observe that:

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



(a) LRR-graph Weights (b) ℓ_1-graph Weights

Figure 2: Visualization of different graph weights W on YaleB face database.

2) 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 methos. 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 YaleB data set 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.

3.3. 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) [2] 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 geometrical 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 3 tabulates the recognition error rates for different graphs under different labeling percentages. We can see that NNLRS-graph almost consistently outperforms other graphs.

3.4. Parameters Sensitivity of NNLRS-graph

Finally, 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 PIE face database. Since the percentage of gross corruption errors in data should be fixed, we set $\lambda = 10$ empirically[4] and only vary β. Because here we test many parametric settings, like above experiments, here we only average the rates over 5 random trials. The results are shown in Table 4. From this table, we can see that, the performance of NNLRS-graph based SDA obviously decreases when $\beta > 1$. If we ignore the sparsity properties ($\beta = 0$), 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 $\beta = 0.2$.

4. Conclusion

This paper proposes a novel informative graph, called the nonnegative low rank and sparse graph (NNLRS-graph), for graph-based semi-supervised learning. NNLRS-graph mainly uses two important properties of high-dimensional data, sparsity and low rankness, both of which capture the structure of the whole data. It simultaneously derives the graph structure and the graph weights, by solving a problem of nonnegative low rank and sparse representation of the whole data. Extensive experiments on both classification and dimensionality reduction show that, NNLRS-graph is better at capturing the globally linear structure of data, and thus is more informative and more suitable than other graphs for graph-based semi-supervised learning.

Acknowledgement

This work is supported by the National Science Foundation of China (No.60933013, No.61103134), the National Science and Technology Major Project (No.2010ZX03004-003), the Fundamental Research Funds for the Central Universities (WK210023002, WK2101020003), and the Science Foundation for Outstanding Young Talent of Anhui Province (BJ2101020001).

References

[1] M. Belkin, P. Nigogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. *J. Machine Learning Research*, 7:2399–2434, 2006.

[2] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *ICCV*, 2007.

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

Table 1: Classification error rates (%) of various graphs combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis after the dataset names). The bold numbers are the lowest error rates under different sampling percentages.

Dataset	kNN0	kNN1	LLE0	LLE1	ℓ_1-graph	SPG	LRR-graph	NNLRS-graph
YaleB (10%)	33.51	38.27	29.21	29.94	46.13	15.57	28.22	**3.75**
YaleB (20%)	34.66	38.97	30.63	30.63	45.54	17.56	24.46	**9.84**
YaleB (30%)	33.71	37.87	28.17	28.17	46.14	16.54	22.33	**10.54**
YaleB (40%)	33.00	37.34	28.36	28.36	43.39	17.16	19.42	**9.38**
YaleB (50%)	33.10	37.38	28.38	28.38	42.25	18.99	18.04	**9.64**
YaleB (60%)	32.48	37.78	28.53	28.53	41.52	20.50	16.09	**8.13**
PIE (10%)	34.84	37.54	33.06	33.44	22.88	20.50	33.98	**11.11**
PIE (20%)	37.46	40.31	35.05	35.81	22.94	**20.30**	34.35	22.81
PIE (30%)	35.30	37.80	32.52	32.88	22.33	20.60	31.81	**17.86**
PIE (40%)	35.81	38.22	32.51	32.99	23.14	20.81	32.39	**16.25**
PIE (50%)	34.39	37.38	31.41	31.64	23.01	21.43	31.33	**19.25**
PIE (60%)	35.63	38.00	32.76	32.85	25.76	23.82	32.50	**21.56**
USPS (10%)	1.87	2.20	17.10	27.31	43.27	3.95	2.25	**1.57**
USPS (20%)	2.51	2.67	22.92	30.83	41.27	5.28	3.10	**1.93**
USPS (30%)	5.88	6.10	21.26	27.54	38.31	10.48	8.91	**4.95**
USPS (40%)	7.87	8.44	19.21	22.78	34.86	14.22	13.44	**7.44**
USPS (50%)	17.19	18.44	18.41	19.48	29.42	20.38	21.88	**11.27**
USPS (60%)	11.04	15.20	14.80	14.94	23.36	15.89	17.75	**6.09**

Table 2: Classification error rates (%) of various graphs combined with the LGC label propagation method under different percentages of labeled samples (shown in the parenthesis after the dataset names). The bold numbers are the lowest error rates under different sampling percentages.

Dataset	kNN0	kNN1	LLE0	LLE1	ℓ_1-graph	SPG	LRR-graph	NNLRS-graph
YaleB (10%)	32.89	36.84	29.00	29.76	46.82	16.37	28.22	**5.56**
YaleB (20%)	31.09	35.59	25.84	26.65	50.53	12.39	24.46	**5.31**
YaleB (30%)	28.56	33.54	22.24	22.83	52.33	9.57	22.33	**4.29**
YaleB (40%)	26.35	30.97	19.82	19.90	57.16	7.07	19.42	**3.75**
YaleB (50%)	24.78	29.73	17.61	17.65	65.79	5.63	18.04	**4.00**
YaleB (60%)	22.98	28.58	15.75	15.94	77.56	4.42	16.09	**3.23**
PIE (10%)	34.28	36.42	32.25	32.53	21.71	19.75	31.26	**12.22**
PIE (20%)	33.06	36.11	30.42	30.83	17.18	15.45	29.82	**10.63**
PIE (30%)	30.11	33.51	26.52	27.01	12.06	10.71	25.61	**9.82**
PIE (40%)	28.46	32.15	23.62	24.01	9.01	8.25	23.86	**7.08**
PIE (50%)	26.96	30.45	21.65	22.22	6.61	6.29	21.24	**4.00**
PIE (60%)	25.09	29.09	19.56	20.02	5.13	**4.95**	20.05	5.00
USPS (10%)	3.13	3.21	27.69	35.06	33.52	6.92	3.49	**2.80**
USPS (20%)	2.22	2.10	22.43	28.96	26.42	4.04	1.83	**1.62**
USPS (30%)	1.55	1.53	19.18	25.30	18.92	2.69	1.22	**1.13**
USPS (40%)	1.20	1.18	16.62	22.53	16.64	1.88	0.92	**0.88**
USPS (50%)	0.82	0.86	14.28	20.01	11.67	1.14	0.61	**0.59**
USPS (60%)	0.65	0.72	12.61	17.69	8.89	0.83	0.49	**0.48**

[3] J.-F. Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, (4):1956–1982, 2010.

[4] E. Candés, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, (3), 2011.

[5] E. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[6] K. Chen and S. Wang. Frobnication. *IEEE Trans. on PAMI,*,

Table 3: 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-graph	NNLRS-graph
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**

Table 4: Recognition error rates (%) of NNLRS-graph for semi-supervised discriminative analysis on 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

33(1):129–143, January 2011.

[7] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with ℓ_1-graph for image analysis. *IEEE Trans. on Image Processing*, 19(4), 2010.

[8] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong. Nonnegative sparse coding for discriminative semi-supervised learning. In *CVPR*, pages 792–801, 2011.

[9] P. O. Hoyer. Modeling receptive fields with non-negative sparse coding. *Computational Neuroscience: Trends in Research 2003, Neurocomputing*.

[10] T. Jebara, J. Wang, and S. Chang. Graph construction and b-matching for semi-supervised learning. In *ICML*, pages 441–448. ACM, 2009.

[11] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative marix factorization. *Nature*, (6755):788–791, 1999.

[12] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, 2009. UIUC Technical Report UILU-ENG-09-2215, arxiv:1009.5055.

[13] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low rank representation. In *NIPS*, 2011.

[14] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*. Citeseer, 2010.

[15] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence*. ACM, 2009.

[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. on Intelligent Systems and Technology*, (2), 2011.

[17] J. Wang, F. Wang, C. Zhang, H. Shen, and L. Quan. Linear neighborhood propagation and its applications. *IEEE Trans. on PAMI*, 31(9):1600–1615, 2009.

[18] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of IEEE*, 98(6):1031–1044, 2010.

[19] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on PAMI*, 31(2):210–227, 2008.

[20] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *SIAM Int'l Conf. on Data Mining, SDM*, pages 792–801, 2009.

[21] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 595–602, 2004.

[22] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006.

[23] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, volume 20, pages 912–919, 2003.