Semi-supervised Orthogonal Graph Embedding with Recursive Projections

Hanyang Liu[1], **Junwei Han**[1*], **Feiping Nie**[1,2]
[1] Northwestern Polytechnical University, Xi'an 710072, P. R. China
[2] University of Texas at Arlington, USA
ericstarkhan@gmail.com, jhan@nwpu.edu.cn, feipingnie@gmail.com

Abstract

Many graph based semi-supervised dimensionality reduction algorithms utilize the projection matrix to linearly map the data matrix from the original feature space to a lower dimensional representation. But the dimensionality after reduction is inevitably restricted to the number of classes, and the learned non-orthogonal projection matrix usually fails to preserve distances well and balance the weight on different projection direction. This paper proposes a novel dimensionality reduction method, called the semi-supervised orthogonal graph embedding with recursive projections (SOGE). We integrate the manifold smoothness and label fitness as well as the penalization of the linear mapping mismatch, and learn the orthogonal projection on the Stiefel manifold that empirically demonstrates better performance. Moreover, we recursively update the projection matrix in its orthocomplemented space to continuously learn more projection vectors, so as to better control the dimension of reduction. Comprehensive experiment on several benchmarks demonstrates the significant improvement over the existing methods.

1 Introduction

Dimensionality reduction has always been a major research topic in the domain of pattern recognition and machine learning. Over the past decades, many efficient and classical linear dimensionality reduction methods have been proposed, such as principle component analysis (PCA) [Turk and Pentland, 2011], linear discriminant analysis (LDA) [Belhumeur *et al.*, 1997] and maximum margin criterion [Li *et al.*, 2006]. Besides, some nonlinear methods such as locally linear embedding [Roweis and Saul, 2000] and Laplacian eigenmaps [Belkin and Niyogi, 2003] are also demonstrated to be suitable for practical applications, focusing on preserving the local or global geometric structure of data.

Although supervised algorithms generally outperform unsupervised algorithms, they require full labeled data, which usually costs huge human labor for labeling. In the last decade, semi-supervised learning algorithms have been proposed to utilize both unlabeled samples and limited number of labeled samples. Among those semi-supervised dimensionality reduction methods, the graph based algorithms, such as LLGC [Zhou *et al.*, 2004],GFHF [Zhu *et al.*, 2003], LLPG [Zhang *et al.*, 2014] exploit data-driven graphs to project data onto the embedded manifolds and thus have the advantage in preserving the geometry structure of data. For classification, a good subspace learned by graph based algorithms should be both smooth and discriminative. Graph based methods like LLGC and GFHF were proposed to simultaneously guarantee both the manifold smoothness and label fitness of data, yet fail to predict labels of the "out-of-sample" data that is not included in the training samples. In contrast, later works such as [Belkin *et al.*, 2006; Cai *et al.*, 2007; Liu *et al.*, 2008; Huang *et al.*, 2012; Yan *et al.*, 2016; Kim *et al.*, 2016] extended some classical models and are able to project the unseen data into lower dimensional subspace.

Despite the success of many graph based dimensionality reduction methods in tackling the partially labeled problem, they still have some limitations. Because it is difficult for nonlinear semi-supervised learning methods to develop an implicit function that can map the unseen data, most of the proposed semi-supervised methods adopt linear functions for data mapping, e.g. SDA [Cai *et al.*, 2007], TR-FSDA [Huang *et al.*, 2012], and linear LapRLS [Belkin *et al.*, 2006]. While the linear mapping function can simplify the learning process, it rigidly assumes that the lower dimension representation of the original data strictly lies on the its linearly spanned space. This formulation usually cannot well represent the real world data especially those embedded with nonlinear manifold. Another limitation that commonly exists in some linear semi-supervised approaches, such as linear LapRLS and FME [Nie *et al.*, 2010], is that the dimensionality after reduction is inevitably restricted to the number of classes, because the learned projection matrix has to satisfy the linear mapping function. Moreover, most of the semi-supervised algorithms use nonorthogonal projections because the orthogonal constraint increases the optimization difficulty, despite the fact that orthogonal projections tend to balance the weights on different projection directions and keep the Euclidean distance based similarity of original data [Cai *et al.*, 2006; Kokiopoulou and Saad, 2005].

This paper focuses on the three aforementioned challenges

*Corresponding authors.

we're mainly concerned with. In this paper, we propose the semi-supervised orthogonal graph embedding with recursive projections (SOGE), integrating the manifold smoothness and label fitness as well as a flexible regularization of mapping mismatch. The major contributions of our work are as follows,

- Inspired by [Abernethy *et al.*, 2008; Liu *et al.*, 2017], we employ the regression residue as a regularization, which helps to relax the strictly linear mapping in most linear semi-supervised methods, and flexibly adjust the mapping mismatch. This formulation can better fit the real world data embedded with nonlinear manifold.
- Due to the good property and empirically better performance of orthogonal projections [Cai *et al.*, 2006; Kokiopoulou and Saad, 2005], we obtain our projection matrix on the Stiefel manifold. We propose an optimization algorithm for our model, which can also be applied to some common optimization problems.
- We propose a recursive procedure to continuously update the projection matrix in its orthogonal complements and learn more projection vectors, so as to control the dimension of reduction freely and provide more choices of dimensionality after reduction for users.

2 Background

2.1 Notations

In this paper, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denotes the sample set in c classes. The corresponding label matrix is denoted by the binary matrix $\mathbf{Y} \in \mathbb{B}^{n\times c}$ with $Y_{ij} = 1$ if the data point $\mathbf{x}_i$ belongs to the j-th class, and $Y_{ij} = 0$ otherwise. In graph based semi-supervised learning, the Laplacian matrix $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix whose elements are the row (or column) sums of the symmetric similarity matrix $\mathbf{S}$. The normalized Laplacian matrix $\tilde{\mathbf{L}}$ is defined by $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ denotes the identity matrix.

2.2 LLGC and GFHF

Learning with local and global consistency (LLGC) [Zhou *et al.*, 2004] addresses the problem of prior assumption of consistency [Zhou *et al.*, 2004] by combining the graph embedding term and the least square of difference between the soft prediction label matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$ and the label matrix $\mathbf{Y}$, with a trade-off parameter. The objective function is as follows:

$$\min_{\mathbf{F}} \sum_{i,j=1}^{n} S_{ij} \left\| \frac{\mathbf{f}_i}{\sqrt{D_{ii}}} - \frac{\mathbf{f}_j}{\sqrt{D_{jj}}} \right\|^2 + \mu \sum_{i=1}^{l} \|\mathbf{f}_i - \mathbf{y}_i\|^2 \quad (1)$$

Gaussian fields and harmonic functions (GFHF) [Zhu *et al.*, 2003] learns the predicted labels by minimizing the "harmonic" entropy function, with the constraint of label fitness:

$$\min_{\mathbf{F}} \sum_{i,j=1}^{n} \|\mathbf{f}_i - \mathbf{f}_j\|^2 S_{ij}, \;\; s.t. \sum_{i=1}^{l} \mathbf{f}_i = \mathbf{y}_i \quad (2)$$

Apparently the objective function of LLGC and GFHF share the same formulation

$$g(\mathbf{F}) = \mathrm{tr}\left(\mathbf{F}^T\mathbf{M}\mathbf{F}\right) + \mathrm{tr}\left(\mathbf{F}-\mathbf{Y}\right)^T\mathbf{U}\left(\mathbf{F}-\mathbf{Y}\right) \quad (3)$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ denotes the normalized Laplacian matrix $\tilde{\mathbf{L}}$ in LLGC, and the Laplacian matrix $\mathbf{L}$ in GFHF. Matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ is diagonal that has the fist l and the rest $n-l$ diagonal entries as μ and 0 respectively. For GFHF in Eq. (2), $\mu = \infty$.

2.3 Linear LapRLS

One of the algorithms proposed in [Belkin *et al.*, 2006], the Laplacian Regularized Least Squares (LapRLS) defines a linear regression function $f(\mathbf{x}_i) = \mathbf{W}^T\mathbf{x}_i + \mathbf{b}$, where $\mathbf{W} \in \mathbb{R}^{d\times c}$ is the projection matrix and $\mathbf{b} \in \mathbb{R}^{c\times 1}$ is the bias. The linear transform of data in the linear LapRLS is obtained by minimizing the following objective:

$$g(\mathbf{W},\mathbf{b}) = \lambda_A \mathrm{tr}\left(\mathbf{W}^T\mathbf{W}\right) + \lambda_I \mathrm{tr}\left(\mathbf{W}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W}\right) + \frac{1}{l}\sum_{i=1}^{l}\left\|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i^T\right\|^2 \quad (4)$$

where λ_A and λ_I balance the penalty of projection matrix $\mathbf{W}$, the manifold smoothness and the regression error. The closed-form solution of the above object minimization can be easily obtained.

Connections. LLGC and GFHF directly learn the predicted label matrix by combining the manifold smoothness and the label fitness. This semi-supervised formulation is efficient but can do little to the unseen testing data, because the predicted label matrix exclusively corresponds to the data involved in the training process, with or without labels. We note that the linear LapRLS in Eq.(4) is actually the "out-of-sample" extension of the formulation of LLGC and GFHF. It exploits the linear regression function for data mapping, and the learned projection matrix $\mathbf{W}$ can be further applied to project testing data.

3 Proposed Model

3.1 Regression Regularization

The predicted label matrix in many algorithms such as PCA and LPP is restricted to lie on the space spanned by the training samples $\mathbf{X}$, namely, $\mathbf{F} = \mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T$, where $\mathbf{1} \in \mathbb{R}^{m\times 1}$ denotes the vector with all elements as 1. Noted that this "hard" projection formulation may be too strict to fit the data samples mostly with nonlinear structure onto the linear manifold, we adopt the regression residue of the linear projection to relax the hard constraint. We assume the predicted labels lie on the space $\mathbf{F} = \mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T + \Delta\mathbf{F}$, where $\Delta\mathbf{F} \in \mathbb{R}^{n\times c}$ is the residue modeling the mismatch of the linear projection. In SOGE, we aim to find the optimal mapping with a proper projection residue $\Delta\mathbf{F}$, we add a regularization to the graph embedding semi-supervised model like Eq. (3):

$$R(\mathbf{W},\Delta\mathbf{F}) = \|\Delta\mathbf{F}\|_F^2 + \lambda \mathrm{tr}\left(\mathbf{W}^T\mathbf{W}\right) \quad (5)$$

where the second term is the penalty of the projection matrix similar to Eq. (4).

3.2 Objective with Orthogonal Constraint

Nonorthogonal projection matrix puts different weights on different projection directions, while orthogonal projections

help to preserve distances and the overall geometry of data [Kokiopoulou and Saad, 2005]. In our work, we obtain the projection matrix $\mathbf{W}$ in the Stiefel manifold $\mathcal{V} = \{\mathbf{W} \in \mathbb{R}^{n\times c} : \mathbf{W}^T\mathbf{W} = \mathbf{I}\}$. In the next subsection, we also show that the orthogonal constraint setting can also contribute to learning more projection vectors in our proposed recursive process.

Since $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, the penalty term in Eq.(5) becomes $\mathrm{tr}(\mathbf{W}^T\mathbf{W}) = d$. Adding the regularization in Eq.(5) with a parameter α to the graph embedding semi-supervised formulation, we have the final objective function with the orthogonal constraint for SOGE:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I},\mathbf{b},\mathbf{F}} \mathrm{tr}\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right) + \mathrm{tr}\left(\mathbf{F}-\mathbf{Y}\right)^T\mathbf{U}\left(\mathbf{F}-\mathbf{Y}\right) + \alpha\left\|\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{F}\right\|_F^2 \quad (6)$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the Laplacian matrix constructed from the data. We set $\mathbf{U} \in \mathbb{R}^{n\times n}$ as the diagonal matrix, in which the diagonal entries corresponding to labeled samples are set as μ, and those corresponding to unlabeled samples set as 0.

The first and the second term in Eq.(6) consider both the manifold smoothness and label fitness, similarly as in LGC, GFHF and LapRLS. The last term helps to learn projection matrix that can be used to map the out-of-sample data for dimensionality reduction, with the positive parameter α to adjust the mismatch of the projection. Therefore, the proposed model manages to integrate the manifold smoothness and label fitness, as well as the optimal projection with proper mismatch.

3.3 Recursive Projections

In many semi-supervised dimensionality reduction methods with linear projection method, such as LapRLS and FME [Nie *et al.*, 2010], the reduced dimensionality is restrained to be exactly the same as the number of classes, c, because the projection matrix $\mathbf{W}$ satisfies $\mathbf{F} = \mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T$. In many situations, it is not always necessary to reduce the dimensions of data to too low dimensions, especially for the high-dimensional data with small number of classes, because data in higher dimensions usually contains more information.

The recent work [Wang *et al.*, 2014] proposed that, if we use the same model to learn a new projection matrix $\tilde{\mathbf{W}}$ from the new sample set obtained as follows,

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X} \quad (7)$$

where $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, the new projection matrix $\tilde{\mathbf{W}}$ is orthogonal to $\mathbf{W}$. The proof can be found in Proposition 1 in [Wang *et al.*, 2014].

According to this proposition, we can recursively obtain new projection matrices $\tilde{\mathbf{W}}$ in the orthogonal complements of $\mathbf{W}$, by continuously solving Eq.(6) with new data matrix $\tilde{\mathbf{X}}$ updated by Eq.(7). Assume that we apply K times of the recursive process, finally we get the concatenation of all the diagonal projection matrices from each recursion, referred to as the recursive projection matrix

$$\mathbf{W}^* = [\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_k] \in \mathbb{R}^{d\times Kc} \quad (8)$$

We can exploit the recursive projection matrix $\mathbf{W}^*$ to project the original data to the subspace with dimensionality of Kc.

4 Optimization

4.1 Problem Analysis

It can be easily proved that the objective function in Eq.(6) is convex. To optimize the objective function with the orthogonal constraint on $\mathbf{W}$, our strategy is that we firstly express the optimal solutions of $\mathbf{F}$ and $\mathbf{b}$ with $\mathbf{W}$ and then find the optimal solution of $\mathbf{W}$ on the Stiefel manifold.

By setting the derivative of the objective with respect to $\mathbf{b}$ equal to zero, we get the solution $\mathbf{b} = 1/n\left(\mathbf{F}^T\mathbf{1} - \mathbf{W}^T\mathbf{X}\mathbf{1}\right)$. Then we have $\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{F} = \mathbf{H}_c\left(\mathbf{X}^T\mathbf{W} - \mathbf{F}\right)$, where $\mathbf{H}_c = \mathbf{I}_n - (1/n)\mathbf{1}\mathbf{1}^T$ is used for centering the data by subtracting the mean of the data. Apparently we also have $\mathbf{H}_c\mathbf{H}_c = \mathbf{H}_c = \mathbf{H}_c^T$. By setting the derivative of the function above respective to $\mathbf{F}$ equal to zero, we have the optimal solution for $\mathbf{F}$:

$$\begin{aligned}\mathbf{F} &= \left(\mathbf{L}+\mathbf{U}+\alpha\mathbf{H}_c\right)^{-1}\left(\mathbf{U}\mathbf{Y} + \alpha\mathbf{H}_c\mathbf{X}^T\mathbf{W}\right)\\ &= \alpha\mathbf{Q}\mathbf{X}_c^T\mathbf{W} + \mathbf{Q}\mathbf{U}\mathbf{Y}\end{aligned} \quad (9)$$

where $\mathbf{Q} = \left(\mathbf{L}+\mathbf{U}+\alpha\mathbf{H}_c\right)^{-1}$ and $\mathbf{X}_c = \mathbf{H}_c\mathbf{X}$. Apparently $\mathbf{Q}^T = \mathbf{Q}$. Replace $\mathbf{F}$ and $\mathbf{b}$ in Eq.(6) with their solutions above, and the objective function in Eq.(6) becomes

$$\begin{aligned}&\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \mathrm{tr}\left(\alpha\mathbf{Q}\mathbf{H}_c\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}\right)^T\mathbf{L}\left(\alpha\mathbf{Q}\mathbf{H}_c\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}\right)\\ &+ \mathrm{tr}\left(\alpha\mathbf{Q}\mathbf{H}_c\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}-\mathbf{Y}\right)^T\mathbf{U}\left(\alpha\mathbf{Q}\mathbf{H}_c\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}-\mathbf{Y}\right)\\ &+ \alpha\mathrm{tr}\left[\left(\alpha\mathbf{Q}\mathbf{H}_c-\mathbf{I}_n\right)\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}\right]^T\\ &\times \mathbf{H}_c\left[\left(\alpha\mathbf{Q}\mathbf{H}_c-\mathbf{I}_n\right)\mathbf{X}^T\mathbf{W}+\mathbf{Q}\mathbf{U}\mathbf{Y}\right]\end{aligned}$$

Then we remove the constant term and rewrite the above problem in the form as follows:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \mathrm{tr}\left(\mathbf{W}^T\mathbf{A}\mathbf{W}\right) - 2\mathrm{tr}\left(\mathbf{W}^T\mathbf{B}\right) \quad (10)$$

where

$$\mathbf{A} = \alpha^2\mathbf{X}_c\left(\frac{1}{\alpha}\mathbf{I}_n - \mathbf{Q}\right)\mathbf{X}_c^T \quad (11)$$

$$\mathbf{B} = \alpha\mathbf{X}_c\mathbf{Q}\mathbf{U}\mathbf{Y} \quad (12)$$

We note that $\mathrm{tr}\left(\mathbf{W}^T\lambda_{max}\mathbf{I}_d\mathbf{W}\right) = \lambda_{max}c$, where λ_{max} is the greatest eigenvalue of matrix $\mathbf{A}$ and $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ is an identity matrix. We can convert the problem in Eq.(10) into a convex optimization problem on the Stiefel manifold, by rewriting Eq.(10) to the following

$$\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \mathrm{tr}\left[\mathbf{W}^T\left(\lambda_{max}\mathbf{I}_d - \mathbf{A}\right)\mathbf{W}\right] + 2\mathrm{tr}\left(\mathbf{W}^T\mathbf{B}\right) \quad (13)$$

Obviously the matrix $(\lambda_{max}\mathbf{I}_d - \mathbf{A})$ is positive semi-definite. It can be readily proved that the objective function in Eq.(13) is also convex.

4.2 Optimization with Orthogonal Constraint

To solve the optimization problem of SOGE in Eq.(6), we need to obtain the optimal solution of $\mathbf{W}$ by solving Eq.(13) first. Similarly as in [Nie *et al.*, 2014; Enhbat, 1996; Nie *et al.*, 2017], we consider a general version of the problem in Eq.(13) as follows

$$\max_{\mathbf{X}\in\mathcal{V}} \mathcal{F}(\mathbf{X}) \quad (16)$$

Algorithm 1 Algorithm to solve problem in Eq.(16)

Initialize $\mathbf{X}$: $\mathbf{X} \in \{\mathbf{X} \in \mathbb{R}^{m\times n} : \mathbf{X}^T\mathbf{X} = \mathbf{I}\}$.
Repeat

1. Obtain the derivative of $\mathcal{F}$ at point $\mathbf{X}^{(t)}$:
$$\mathbf{D}^{(t)} = \mathcal{F}'(\mathbf{X}^{(t)}) \quad (14)$$
2. Update $\mathbf{X}$ by solving the following problem
$$\max_{\mathbf{X}^T\mathbf{X}=\mathbf{I}} \langle \mathbf{D}^{(t)}, \mathbf{X} - \mathbf{X}^{(t)}\rangle \Leftrightarrow \max_{\mathbf{X}^T\mathbf{X}=\mathbf{I}} \mathrm{tr}(\mathbf{D}^{(t)T}X) \quad (15)$$
Apply SVD decomposition: $\mathbf{D}^{(t)} = \mathbf{\Lambda\Sigma V}^T$, and the optimal solution is $\mathbf{X}^{(t+1)} = \mathbf{\Lambda V}^T$.

Until $\mathrm{tr}(\mathbf{D}^{(t)T}\mathbf{X}^{(t+1)}) \leq \mathrm{tr}(\mathbf{D}^{(t)T}\mathbf{X}^{(t)})$
Return $\mathbf{X}$.

where $\mathcal{V} = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \mathbf{X}^T\mathbf{X} = \mathbf{I}\}$ is referred to as the set of orthogonal k-frames, also known as the Stiefel manifold, and $\mathcal{F} : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a convex function. Since function $\mathcal{F}$ is convex, obviously its derivative (gradient) at a certain point $\mathbf{X}^*$, referred to as $\mathbf{D}^* = \mathcal{F}'(\mathbf{X}^*)$, is a subgradient of $\mathcal{F}$. According to the optimality characterization of subgradient [Rockafellar, 1970], the point $\mathbf{X}^*$ is an optimal solution of problem in Eq.(16) if and only if the condition as follows holds:

$$\langle \mathcal{F}'(\mathbf{X}^*), \mathbf{X} - \mathbf{X}^*\rangle \leq 0, \ \ \forall \mathbf{X} \in \mathcal{V} \quad (17)$$

where $\langle\cdot\rangle$ denotes the Euclidean inner product of matrices. The condition above is equivalent to

$$\max_{\mathbf{X}\in\mathcal{V}} \langle \mathcal{F}'(\mathbf{X}^*), \mathbf{X} - \mathbf{X}^*\rangle = \max_{\mathbf{X}\in\mathcal{V}} \mathrm{tr}\left[\mathbf{D}^{*T}(\mathbf{X} - \mathbf{X}^*)\right] \leq 0 \quad (18)$$

Hence $\mathbf{X}^*$ is an optimal solution to Eq.(16) if and only if $\mathrm{tr}\left(\mathbf{D}^{*T}\mathbf{X}\right) \leq \mathrm{tr}\left(\mathbf{D}^{*T}\mathbf{X}^*\right), \forall \mathbf{X} \in \mathcal{V}$.

We combine the subgradient ascent together with simultaneously enforcing the orthogonal constraint on $\mathbf{X}$, and develop an iterative optimization algorithm for the problem in Eq(16), as shown in Algorithm 1. The empirical analysis in the next section shows that Algorithm 1 can converge very fast.

Theorem 1 *Algorithm 1 will monotonically increase the objective of the problem in Eq.(16) in each iteration until the algorithm converges.*

Proof. Because function $\mathcal{F}$ is convex, the derivative $\mathbf{D}^{(t)}$ is a subgradient of $\mathcal{F}$. According to the definition of subgradient [Rockafellar, 1970], we have

$$\mathcal{F}(\mathbf{X}^{(t+1)}) \geq \mathcal{F}(\mathbf{X}^{(t)}) + \langle \mathbf{D}^{(t)}, \mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}\rangle \quad (19)$$

When $\mathbf{X}^{(t)}$ is not the optimal solution to the problem in Eq.(16), according to the condition of optimal solution in Eq.(17), we have $\langle \mathbf{D}^{(t)}, \mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}\rangle > 0$, hence $\mathcal{F}(\mathbf{X}^{(t+1)}) > \mathcal{F}(\mathbf{X}^{(t)})$. So the objective will increase in each iteration before it converges. □

According to Theorem 1 we can conclude that Algorithm 1 monotonically increase the objective of the problem in Eq.(16) in each iteration until the objective function converges. The convergence of Algorithm 1 is also shown by Figure 1 in the experiment.

Algorithm 2 Algorithm of SOGE

Input: Dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1} \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$, diagonal matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$, label matrix $\mathbf{Y} \in \mathbb{B}^{n\times c}$, the regularization parameter α and the number of recursion times K.
Output: Recursive projection matrix $\mathbf{W}^* \in \mathbb{R}^{d\times Kc}$.

1: $j = 1$
2: **for** $j < K$ **do**
3: Initialize $\mathbf{W}_j$ randomly to satisfy $\mathbf{W}^T\mathbf{W} = \mathbf{I}$.
4: **while** *not converge* **do**
5: 1) Obtain the greatest eigenvalue λ_{max} of $\mathbf{A}$;
6: 2) Obtain gradient: $\mathbf{D} = (\lambda_{max}\mathbf{I} - \mathbf{A})\mathbf{W}_i + \mathbf{B}$;
7: 3) Update $\mathbf{W}_j$: $\mathbf{D} = \mathbf{\Lambda\Sigma V}^T$, $\mathbf{W}_j = \mathbf{\Lambda V}^T$.
8: **end while**
9: Update data matrix $\mathbf{X} \leftarrow \mathbf{X} - \mathbf{W}_j^T\mathbf{W}_j\mathbf{X}$.
10: **return** $\mathbf{W}_j$, $\mathbf{X}$
11: $j \leftarrow j + 1$
12: **end for**
13: Recursive projections $\mathbf{W}^* = [\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_K]$.
Return $\mathbf{W}^*$.

4.3 Algorithm of SOGE

To optimize the objective of our SOGE method, we apply Algorithm 1 to solve the maximization problem in Eq.(13). Thus we can optimize the objective function of SOGE in Eq.(6). Then we update the data matrix following Eq.(7), and continue obtaining new $\mathbf{W}$ in its orthocomplemented space. After repeating the recursive process for K times, finally we get the recursive projection matrix $\mathbf{W}^*$ by Eq.(8). The number of recursion times K can be set by users as they need, according to different types of data. The algorithm of SOGE is shown in Algorithm 2.

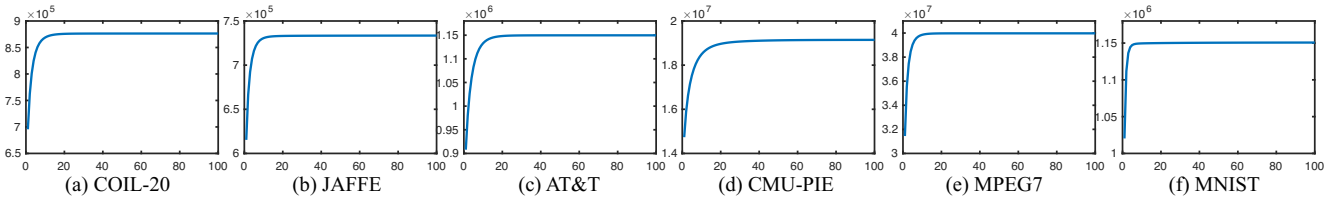
Time Complexity. In each iteration in Algorithm 2, the major computational burden lies on computing the SVD decomposition to matrix $\mathbf{D} \in \mathbb{R}^{d\times c}$ (step 6) with time complexity $O(d^2c + dc^2 + c^3)$, and updating data matrix by Eq.(7) (step 7) with time complexity $O(d^2c + d^2n)$. For most of the data, usually $d < n$ and $c << n$, so the SOGE method is very efficient for dimensionality reduction.

5 Experiment

5.1 Datasets

In our experiments, we use six real world benchmarks including three face benchmarks (JAFFE[1], AT&T[2], and CMU-PIE), a handwritten digits dataset MNIST, and two object benchmarks (COIL-20 and MPEG7[3]). For CMU-PIE database that contains more than 40,000 faces, we choose the frontal pose group (C27) from varying illuminations and facial expressions. For the dataset MNIST that contains more than 70,000 facial images, we randomly select 15,000 of them from all 10 classes. For other datasets, all samples are used in the experiment. The detailed information of the benchmark datasets used in the experiment are listed in Table 2.

[1] http://www.kasrl.org/jaffe.html
[2] http://www.cl.cam.ac.uk/research/dtg/attarchive.html
[3] http://www.dabi.temple.edu/ shape/MPEG7/dataset.html



(a) COIL-20 (b) JAFFE (c) AT&T (d) CMU-PIE (e) MPEG7 (f) MNIST

Figure 1: Convergence curves of the objective in Eq.(13) on each dataset.

Table 1: Recognition performance (Mean Accuracy ± Std %) of each algorithm over 20 random splits on six benchmark datasets. The optimal parameters for each algorithm under different experimental settings are also shown in the table.

Dataset	Method	1 Labeled Sample			2 Labeled Samples			3 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
COIL-20	LapRLS	**74.37±3.7**	**74.59±3.2**	$(10^{-3}, 10^{0})$	82.39±1.6	82.42±1.4	$(10^{-3}, 10^{0})$	83.80±2.3	83.92±2.4	$(10^{-3}, 10^{0})$
	SDA	67.34±2.6	66.86±3.1	$(10^{-3}, 10^{0})$	76.40±2.6	76.27±2.4	$(10^{-3}, 10^{0})$	80.31±2.8	80.08±2.9	$(10^{-3}, 10^{0})$
	TCA	65.27±3.2	65.17±3.7	$(10^{-3}, 10^{-6})$	70.77±4.2	70.60±4.1	$(10^{-3}, 10^{-6})$	71.91±2.9	67.99±2.5	$(10^{-3}, 10^{-6})$
	FME	61.44±3.9	61.20±3.4	$(10^{6}, 10^{9})$	71.53±2.6	70.83±2.7	$(10^{3}, 10^{6})$	76.25±2.0	75.24±1.7	$(10^{3}, 10^{6})$
	TR-FSDA	52.23±4.1	54.00±3.5	$(10^{-3}, 10^{0})$	75.77±5.1	75.57±5.5	$(10^{-3}, 10^{0})$	83.73±4.3	81.91±4.4	$(10^{-3}.10^{0})$
	SOGE	72.01±2.3	71.43±2.39	(10^{-6})	**82.68±2.7**	**82.86±2.1**	(10^{-9})	**84.09±2.4**	**84.31±2.2**	(10^{-9})

Dataset	Method	1 Labeled Sample			2 Labeled Samples			3 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
JAFFE	LapRLS	92.88±5.8	83.15±5.4	$(10^{-3}, 10^{0})$	96.25±3.6	86.45±4.3	$(10^{-3}, 10^{0})$	97.64±1.8	89.65±2.4	$(10^{-3}, 10^{0})$
	SDA	93.61±3.4	84.7±4.7	$(10^{-3}, 10^{0})$	96.06±2.1	87.50±4.5	$(10^{-3}, 10^{0})$	96.28±2.0	90.55±4.1	$(10^{-3}, 10^{0})$
	TCA	90.77±4.4	90.75±4.8	$(10^{-3}, 10^{-6})$	92.94±4.7	92.95±4.1	$(10^{6}, 10^{6})$	92.43±4.3	94.20±3.3	$(10^{6}, 10^{6})$
	FME	94.22±3.7	93.75±3.3	$(10^{3}, 10^{6})$	97.00±2.1	96.05±2.2	$(10^{3}, 10^{6})$	98.20±0.9	98.15±0.8	$(10^{3}, 10^{6})$
	TR-FSDA	85.55±4.6	86.05±4.7	$(10^{-3}, 10^{3})$	95.06±2.9	95.25±2.4	$(10^{-3}, 10^{3})$	97.43±2.4	97.60±2.3	$(10^{-3}, 10^{3})$
	SOGE	**95.16±2.8**	**94.60±3.2**	(10^{-3})	**97.12±3.6**	**96.65±3.3**	(10^{-3})	**98.36±2.7**	**98.70±2.3**	(10^{-3})

Dataset	Method	1 Labeled Sample			2 Labeled Samples			3 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
AT&T	LapRLS	67.34±3.1	67.32±2.8	$(10^{3}, 10^{0})$	81.41±3.0	80.10±3.4	$(10^{3}, 10^{0})$	85.81±3.5	87.03±3.1	$(10^{3}, 10^{0})$
	SDA	65.56±3.6	67.83±2.9	$(10^{6}, 10^{0})$	80.54±3.7	78.65±3.3	$(10^{6}, 10^{0})$	86.68±3.5	87.30±2.6	$(10^{6}, 10^{0})$
	TCA	63.22±3.4	62.20±4.3	$(10^{6}, 10^{6})$	68.58±4.9	71.83±3.4	$(10^{6}, 10^{6})$	70.00±4.2	73.45±3.21	$(10^{6}, 10^{9})$
	FME	68.34±3.4	66.92±3.2	$(10^{3}, 10^{0})$	79.08±3.2	77.38±3.9	$(10^{3}, 10^{0})$	84.00±3.1	81.87±3.0	$(10^{3}, 10^{0})$
	TR-FSDA	65.93±4.4	65.97±3.0	$(10^{-3}, 10^{3})$	79.83±4.0	80.15±3.0	$(10^{-3}, 10^{3})$	87.37±4.3	87.95±2.1	$(10^{-3}, 10^{3})$
	SOGE	**69.72±3.7**	**70.25±2.9**	(10^{0})	**83.88±3.5**	**82.33±3.1**	(10^{0})	**88.81±3.5**	**88.60±2.6**	(10^{0})

Dataset	Method	1 Labeled Sample			2 Labeled Samples			3 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
CMU-PIE	LapRLS	19.30±1.2	18.41±1.1	$(10^{-3}, 10^{-3})$	30.88±1.5	30.44±1.6	$(10^{-3}, 10^{-3})$	37.39±1.3	38.94±1.7	$(10^{-3}, 10^{-3})$
	SDA	28.41±2.6	26.67±3.2	$(10^{6}, 10^{9})$	46.66±2.6	45.11±3.0	$(10^{6}, 10^{9})$	56.39±2.8	58.10±3.1	$(10^{6}, 10^{9})$
	TCA	52.45±1.7	51.29±2.1	$(10^{-3}, 10^{0})$	67.85±1.7	66.66±1.4	$(10^{-3}, 10^{0})$	74.82±1.6	76.99±1.2	$(10^{-3}, 10^{0})$
	FME	54.20±1.0	52.26±1.0	$(10^{3}, 10^{0})$	**69.26±2.0**	**68.35±2.3**	$(10^{3}, 10^{0})$	**75.74±1.7**	**78.57±1.4**	$(10^{3}, 10^{0})$
	TR-FSDA	45.72±6.2	43.66±6.6	$(10^{3}, 10^{-3})$	56.80±6.2	57.49±5.7	$(10^{3}, 10^{-3})$	64.95±5.2	65.69±5.9	$(10^{3}, 10^{-3})$
	SOGE	**54.62±2.5**	**55.44±2.7**	(10^{0})	66.77±1.4	66.562.0	(10^{0})	74.05±1.7	76.03±1.2	(10^{0})

Dataset	Method	1 Labeled Sample			2 Labeled Samples			3 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
MPEG7	LapRLS	54.50±1.8	49.93±1.4	$(10^{-}6, 10^{-3})$	62.89±2.1	58.24±1.7	$(10^{-6}, 10^{-3})$	67.47±1.5	63.17±1.1	$(10^{-6}, 10^{-3})$
	SDA	51.57±1.8	51.02±1.8	$(10^{6}, 10^{-3})$	62.35±2.0	63.41±1.5	$(10^{6}, 10^{-3})$	68.46±2.2	67.20±1.5	$(10^{6}, 10^{-3})$
	TCA	47.67±2.7	46.80±2.7	$(10^{6}, 10^{6})$	49.24±1.9	51.07±1.9	$(10^{6}, 10^{6})$	50.00±1.6	52.24±1.5	$(10^{6}, 10^{6})$
	FME	50.53±2.0	48.54±2.4	$(10^{3}, 10^{0})$	57.33±1.8	54.03±1.2	$(10^{3}, 10^{0})$	59.75±1.5	56.86±1.3	$(10^{3}, 10^{0})$
	TR-FSDA	51.56±1.4	51.67±2.3	$(10^{-6}, 10^{0})$	57.92±1.7	58.11±1.8	$(10^{-6}, 10^{0})$	59.65±2.2	60.59±1.5	$(10^{-6}, 10^{0})$
	SOGE	**53.92±1.4**	**52.96±1.7**	(10^{0})	**63.05±1.4**	**63.95±1.1**	(10^{0})	**69.36±1.3**	**68.90±1.3**	(10^{0})

Dataset	Method	10 Labeled Samples			20 Labeled Samples			30 Labeled Samples		
		Unlabel	Test	Param	Unlabel	Test	Param	Unlabel	Test	Param
MNIST	LapRLS	73.29±2.1	72.75±2.2	$(10^{-3}, 10^{-3})$	77.87±0.9	76.97±1.1	$(10^{-3}, 10^{0})$	82.94±0.7	83.68±0.8	$(10^{-3}, 10^{0})$
	SDA	71.27±1.4	71.24±1.4	$(10^{6}, 10^{0})$	75.74±1.1	75.63±1.4	$(10^{6}, 10^{0})$	79.46±0.9	79.37±1.2	$(10^{6}, 10^{0})$
	TCA	63.03±2.5	63.11±2.1	$(10^{-3}, 10^{0})$	63.21±1.3	62.94±1.4	$(10^{-3}, 10^{0})$	64.41±1.3	61.45±1.5	$(10^{-6}, 10^{-3})$
	FME	69.96±2.7	69.22±2.8	$(10^{-3}, 10^{0})$	74.95±1.4	74.01±1.4	$(10^{-6}, 10^{0})$	79.29±0.8	77.65±1.0	$(10^{-6}, 10^{0})$
	TR-FSDA	–	–		–	–		–	–	
	SOGE	**74.04±1.2**	**73.71±1.2**	(10^{-6})	**79.46±1.2**	**78.64±1.3**	(10^{-3})	**85.31±1.2**	**84.92±1.2**	(10^{0})

Table 2: Description of Benchmark Datasets

Dataset	Type	#Sample	#Dim	#Class
COIL-20	Object	1,440	1,024	20
JAFFE	Face	200	4,096	10
AT&T	Face	400	644	40
CMU-PIE	Face	3,332	1,024	68
MPEG7	Object	1,400	4,096	70
MNIST	Digits	15,000	784	10

5.2 Experiment Setup

We compare SOGE with several graph based semi-supervised methods: LapRLS [Belkin *et al.*, 2006], SDA [Cai *et al.*, 2007], TCA [Liu *et al.*, 2008], FME [Nie *et al.*, 2010], and TR-FSDA [Huang *et al.*, 2012]. For all the algorithms except TR-FSDA, we follow the constrained Laplacian rank (CLR) in [Nie *et al.*, 2016] to construct the Laplacian matrix $\mathbf{L}$ that exactly has rank $(n - c)$ and perform well empirically. For TR-FSDA, we construct the Laplacian matrices $\tilde{\mathbf{L}}_a$ and $\tilde{\mathbf{L}}_b$ directly following the paper [Huang *et al.*, 2012]. In SOGE, we set the weight μ in the diagonal matrix $\mathbf{U}$ as 100 for all datasets. In order to fairly compare SOGE with other algorithms, we tuned all the regularization parameters involved in each algorithms with grid search within $\{10^{-9}, 10^{-6}, 10^{-3}, 10^{0}, 10^{3}, 10^{6}, 10^{9}\}$.

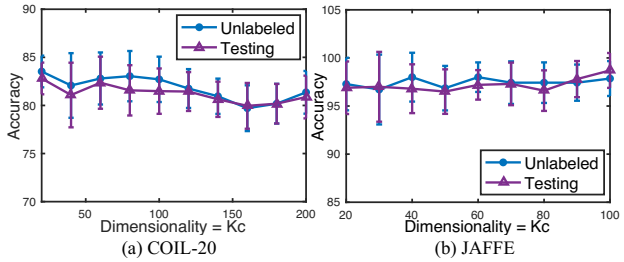
For all the algorithms, we employ the k-nearest neighbor (kNN) classifier to evaluate the performance of dimensionality reduction, and set $k = 1$ in kNN for all the algorithms. For all the datasets, we use PCA as a preprocessing procedure to denoise all the data with 95% of the information preserved, similarly as in [Yan *et al.*, 2007]. We follow the experiment setting as in [Cai *et al.*, 2007], to split the data for comparing different dimensionality reduction algorithms. Firstly, we randomly select 50% of the data as the training set and use the remained 50% as the testing set for the semi-supervised methods. Next, among the training set, we randomly label p samples in each class and treat the rest without labels as the unlabeled data. Thus, the original dataset is split into three part: labeled, unlabeled, and testing samples.

In the experiment, we set p as 10, 20 and 30 for MNIST and 1, 2 and 3 for the others, considering MNIST has much more samples per class. The selected p labeled samples per class are used to train the kNN classifier. Algorithms like LapRLS and FME can only reduce the dimensionality of data to c dimensions. So the number of final dimensions after reduction in all the algorithms is fixed as c for fair comparison.

5.3 Experiment Results

We split the data randomly and apply kNN classifier to both the unlabeled samples and testing samples respectively, and repeat the random splitting and recognition for 20 times. The mean recognition accuracy and the standard deviation are reported in Table 1 (the recognition accuracy of TR-FSDA on MINIST is skipped because it takes TR-FSDA too long to process data in such a large size). Figure 1 shows the convergence of the objective in Eq.(13) on each dataset.

Moreover, we take the datasets COIL-20 and JAFFE with small number of classes as examples to demonstrate the performance of the recursive projections in SOGE. We set the recursion times K from 1 through 10 and plot the recognition accuracy and the standard deviation on both the unlabeled and testing sets, as shown in Figure 2. We have the following observations from the experiment result:



(a) COIL-20 (b) JAFFE

Figure 2: Recognition performance (Mean Accuracy ± Std %) of SOGE over 20 random splits on COIL-20 and JAFFE with different feature dimensions. The recursion times K is set from 1 through 10, and $p = 3$.

- Algorithms like SOGE, LapRLS and FME that use linear projection functions generally outperform the LDA-like algorithms like SDA and TR-FSDA on most datasets. It shows the advantage of linear projection functions to better preserve the information and structure of data in c feature dimensions.
- Our method SOGE generally outperforms other graph based semi-supervised methods involved in the experiment, which implies the significant advantage of the orthogonal projections employed in SOGE. Besides, Figure 1 demonstrates that the optimization algorithm of SOGE converges fast and effectively.
- Figure 2 shows that the recognition accuracy of SOGE changes little with different value of K. In SOGE, the data information and structure is well projected into every c feature dimensions, so its good performance is stable for different dimensions. The recursive projections makes SOGE more flexible in controlling the dimension of reduction without reducing the performance.

6 Conclusion

In this paper, we propose a novel orthogonal graph embedding semi-supervised method for dimensionality reduction, which integrates manifold smoothness and label fitness as well as proper projections. We employ the regression residue as the regularization to relax the strictly linear mapping and flexibly adjust the mapping mismatch. Moreover, we obtain the projection matrix for SOGE on the Stiefel manifold, which empirically demonstrates better performance. To optimize our model, we introduce an optimization algorithm that can also solve general convex maximization problems with orthogonal constraint. The adoption of recursive projections helps SOGE to flexibly learn more feature dimensions. The experiment on six benchmarks demonstrate the significant improvement of the proposed method over other semi-supervised dimensionality reduction methods.

Acknowlegments

This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

References

[Abernethy *et al.*, 2008] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. Web spam identification through content and hyperlinks. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 41–44. ACM, 2008.

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, Joro P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *TPAMI*, 19(7):711–720, 1997.

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7(1):2399–2434, 2006.

[Cai *et al.*, 2006] Deng Cai, Xiaofei He, Jiawei Han, and Hong Jiang Zhang. Orthogonal laplacianfaces for face recognition. *TIP*, 15(11):3608–14, 2006.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *ICCV*, pages 1–7, 2007.

[Enhbat, 1996] R. Enhbat. An algorithm for maximizing a convex function over a simple set. *Journal of Global Optimization*, 8(4):379–391, 1996.

[Huang *et al.*, 2012] Yi Huang, Dong Xu, and Feiping Nie. Semi-supervised dimension reduction using trace ratio criterion. *IEEE Transactions on neural networks and learning systems*, 23(3):519–526, 2012.

[Kim *et al.*, 2016] Kwang In Kim, James Tompkin, Hanspeter Pfister, and Christian Theobalt. Semi-supervised learning with explicit relationship regularization. In *CVPR*, pages 2188–2196, 2016.

[Kokiopoulou and Saad, 2005] Effrosini Kokiopoulou and Yousef Saad. Orthogonal neighborhood preserving projections. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[Li *et al.*, 2006] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *TNN*, 17(1):157–65, 2006.

[Liu *et al.*, 2008] Wei Liu, Dacheng Tao, and Jianzhuang Liu. Transductive component analysis. In *ICML*, pages 433–442, 2008.

[Liu *et al.*, 2017] Hanyang Liu, Junwei Han, Feiping Nie, and Xuelong Li. Balanced clustering with least square regression. In *AAAI*, 2017.

[Nie *et al.*, 2010] Feiping Nie, Dong Xu, Wai Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *TIP*, 19(7):1921–1932, 2010.

[Nie *et al.*, 2014] Feiping. Nie, Jiajun Yuan, and Heng Huang. Optimal mean robust principal component analysis. In *ICML*, pages 1062–1070, 2014.

[Nie *et al.*, 2016] Feiping Nie, Xiaoqian Wang, Michael I Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 2016.

[Nie *et al.*, 2017] Feiping Nie, Rui Zhang, and Xuelong Li. A generalized power iteration method for solving quadratic problem on the stiefel manifold. *Science China Information Science (SCIS)*, 60, 2017.

[Rockafellar, 1970] R. Tyrrell Rockafellar. Convex analysis. *Princeton Mathematical Series*, 29(17):5C101, 1970.

[Roweis and Saul, 2000] Sam. T. Roweis and Lawrence. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, 2000.

[Turk and Pentland, 2011] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. *Proc. CVPR*, volume 84(9):586–591, 2011.

[Wang *et al.*, 2014] Hua Wang, Feiping Nie, and Heng Huang. Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *ICML*, pages 1836–1844, 2014.

[Yan *et al.*, 2007] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *TPAMI*, 29(1), 2007.

[Yan *et al.*, 2016] Yan Yan, Zhongwen Xu, Ivor W Tsang, Guodong Long, and Yi Yang. Robust semi-supervised learning through label aggregation. In *AAAI*, 2016.

[Zhang *et al.*, 2014] Yan-Ming Zhang, Kaizhu Huang, Xinwen Hou, and Cheng-Lin Liu. Learning locality preserving graph from data. *IEEE transactions on cybernetics*, 44(11):2088–2098, 2014.

[Zhou *et al.*, 2004] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Sch Olkopf. Learning with local and global consistency. *Advances in NIPS*, 16(4):321–328, 2004.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.