Graph Embedding with Constraints

Xiaofei He
State Key Lab of CAD&CG
College of Computer Science
Zhejiang University, China
xiaofeihe@cad.zju.edu.cn

Ming Ji
State Key Lab of CAD&CG
College of Computer Science
Zhejiang University, China
jiming@cad.zju.edu.cn

Hujun Bao
State Key Lab of CAD&CG
College of Computer Science
Zhejiang University, China
bao@cad.zju.edu.cn

Abstract

Recently graph based dimensionality reduction has received a lot of interests in many fields of information processing. Central to it is a graph structure which models the geometrical and discriminant structure of the data manifold. When label information is available, it is usually incorporated into the graph structure by modifying the weights between data points. In this paper, we propose a novel dimensionality reduction algorithm, called *Constrained Graph Embedding*, which considers the label information as additional constraints. Specifically, we constrain the space of the solutions that we explore only to contain embedding results that are consistent with the labels. Experimental results on two real life data sets illustrate the effectiveness of our proposed method.

1 Introduction

In many real world applications like face recognition and text categorization, the data is often of very high dimensionality. Procedures that are analytically or computationally manageable in low-dimensional spaces can become completely impractical in a space of several hundreds or thousands dimensions [Duda *et al.*, 2000]. Thus, various techniques have been developed for reducing the dimensionality of the feature space, in the hope of obtaining a manageable problem. Two of the most popular techniques for this purpose are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [Duda *et al.*, 2000].

PCA is an unsupervised method. It aims to project the data along the direction of maximal variance. LDA is supervised. It searches for the project axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. Both of them are spectral methods, that is, methods based on eigenvalue decomposition of either the covariance matrix for PCA or the scatter matrices for LDA.

Recently, graph embedding has become a topic of significant interest for dimensionality reduction [Brand, 2003; Cai *et al.*, 2007; Li *et al.*, 2008; He *et al.*, 2005b; Sugiyama, 2007]. It usually constructs a graph to encode the geometrical information in the data. For some applications like Web search, the graph can be pre-defined by using hyperlinks. Using the notion of graph Laplacian [Chung, 1997], one can find a lower-dimensional representation which respects the graph structure. Many state-of-the-art dimensionality reduction algorithms such as Isomap [Tenenbaum *et al.*, 2000], Laplacian Eigenmap [Belkin and Niyogi, 2001], Locally Linear Embedding [Roweis and Saul, 2000], Neighborhood Preserving Embedding (NPE, [He *et al.*, 2005a]) and Locality Preserving Projections [He *et al.*, 2005b], as well as canonical algorithms like PCA and LDA, can be interpreted in a general graph embedding framework with different choices of the graph structure.

In some situations, there may be label information (or *within-class* and *between-class* advice) available. The most natural way to make use of such prior information is to incorporate it into the graph structure by modifying the weight matrix. Specifically, if two points share the same label or there is within-class advice for them, then the edge weight is increased. If two points have different labels or there is between-class advice for them, then the edge weight is decreased. The typical supervised graph embedding algorithms include Local Discriminant Embedding (LDE, [Chen *et al.*, 2005]). Note that, NPE and LPP can also be performed in supervised manner by incorporating the label information into the graph structure. The major disadvantage of these approaches is that there is no theoretical guarantee that data points from the same class are mapped into a lower dimensional space in which they are actually sufficiently close to each other.

In this paper, we propose a novel constrained dimensionality reduction algorithm, called *Constrained Graph Embedding* (CGE), that considers the label information or within-class/between-class advice as additional constraints. We constrain the space of the solutions that we explore only to contain embedding results that are consistent with the labels (advice). This way, the data points belonging to the same class are merged together in the embedding space in which better classification or clustering performance can be obtained. A key problem in graph embedding is the out-of-sample extension. We further propose that an explicit mapping function can be learned which is defined everywhere.

The paper is structured as follows: in Section 2, we provide a review of graph based dimensionality reduction. Our Constrained Graph Embedding (CGE) algorithm is introduced in

Section 3. In Section 4, we discuss how to perform out-of-sample extension. The experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2 A Review of Graph based Dimensionality Reduction

Suppose we have m data points $\{\mathbf{x}_i\}_{i=1}^m$. In the past decades, many dimensionality reduction algorithms have been proposed to find a lower-dimensional representation of $\mathbf{x}_i$. Despite the different motivations of these algorithms, they can be nicely interpreted in a general graph embedding framework [Brand, 2003; He *et al.*, 2005b].

Given a graph G with m vertices, each vertex represents a data point. Let W be a a symmetric $m \times m$ matrix with W_{ij} having the weight of the edge joining vertices i and j. The G and W can be defined to characterize certain statistical or geometrical properties of the data set. The purpose of graph embedding is to represent each vertex of the graph as a low dimensional vector that preserves similarities between the vertex pairs, where similarity is measured by the edge weight.

Let $\mathbf{y} = (y_1, \cdots, y_m)^T$ be the map from the graph to the real line. The optimal $\mathbf{y}$ is given by minimizing [Belkin and Niyogi, 2001]:

$$\begin{aligned} &\sum_{i,j=1}^{m} (y_i - y_j)^2 W_{ij} \\ s.t. \quad &\mathbf{y}^T D \mathbf{y} = 1, \end{aligned} \tag{1}$$

where D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ij}$. The constraint $\mathbf{y}^T D \mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. This objective function incurs a heavy penalty if neighboring vertices i and j are mapped far apart. Therefore, minimizing it is an attempt to ensure that if vertex i and j are connected with high weight then y_i and y_j are close in the embedding space [Belkin and Niyogi, 2001; Guattery and Miller, 2000]. With some simple algebraic formulations, we have

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} = 2\mathbf{y}^T L \mathbf{y}$$

where $L = D - W$ is the *graph Laplacian* [Chung, 1997]. Finally, the minimization problem reduces to find

$$\mathbf{y}^* = \arg\min_{\mathbf{y}^T D \mathbf{y} = 1} \mathbf{y}^T L \mathbf{y} = \arg\min \frac{\mathbf{y}^T L \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \tag{2}$$

The optimal $\mathbf{y}$'s can be obtained by solving the minimum eigenvalue problem:

$$L\mathbf{y} = \lambda D \mathbf{y} \tag{3}$$

It would be interesting to note that graph embedding has a close connection to differential geometry. Suppose the data points are sampled from an underlying submanifold $\mathcal{M}$ which is embedded in the ambient space. Let $f : \mathcal{M} \to \mathbb{R}$ be a smooth one-dimensional map. Belkin and Niyogi showed that the optiaml map preserving locality can be obtained by solving the following optimization problem on the manifold:

$$\min_{\|f\|_{\mathcal{L}^2(\mathcal{M})}=1} \int_{\mathcal{M}} \|\nabla f\|^2 \tag{4}$$

By Stokes' theorem, we have:

$$\int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} < \mathcal{L}f, f > \tag{5}$$

where $\mathcal{L}$ is the Laplace-Beltrami operator, i.e. $\mathcal{L}f = -\text{div}\nabla(f)$. Therefore, the optimal f to the objective function (4) has to be an eigenfunction of $\mathcal{L}$. Belkin and Niyogi have shown that the optimal solution $\mathbf{y}^*$ gives a discrete approximation to the eigenfunction of $\mathcal{L}$ [Belkin and Niyogi, 2001].

3 Constrained Graph Embedding

There is no theoretical guarantee for previous graph embedding algorithms that two data points sharing the same label are mapped into a low dimensional space in which they are actually sufficiently close to each other. In this section, we introduce our *Constrained Graph Embedding* algorithm which makes use of the label information as additional constraints. We begin with a formal definition of the problem of constrained graph embedding.

3.1 The Problem

The generic graph embedding problem is the following. Given a graph $G(V, W)$ where $V = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ is the set of nodes and W is the weight matrix, find a Euclidean embedding of the m nodes $\mathbf{y}_1, \cdots, \mathbf{y}_m$ in $\mathbb{R}^d$, such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$. Our method considers the particular situation that there is label information, or within-class/between-class advice, available. Thus, it is necessary to constrain the space of the solutions that we explore only to contain embedding results that are consistent with this prior knowledge.

3.2 The Algorithm

Given m data points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\} \subset \mathbb{R}^n$ sampled from the underlying submanifold $\mathcal{M}$, one can build a nearest neighbor graph G to model the local geometrical structure of $\mathcal{M}$. For each data point $\mathbf{x}_i$, we find its k nearest neighbors and put an edge between $\mathbf{x}_i$ and its neighbors. There are many choices of the weight matrix W. A simple definition is as follows:

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_j, \\ & \text{or } \mathbf{x}_j \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Let $\mathbf{y} = (y_1, \cdots, y_m)$ be a one-dimensional map of $\mathbf{x}_i$'s. Without loss of generality, suppose there is label information available for the first p data points $\mathbf{x}_1, \cdots, \mathbf{x}_p$. The rest $m-p$ data points $\mathbf{x}_{p+1}, \cdots, \mathbf{x}_m$ are unlabeled. Suppose the labeled data points are from c classes.

As we have described earlier, the label information can be introduced into graph embedding as additional constraints.

Let $\mathbf{u}_i$ be a m-dimensional indicator vector for the i-th class. That is, $\mathbf{u}_{i,j} = 1$ if and only if $\mathbf{x}_j$ is labeled with the i-th class; $\mathbf{u}_{i,j} = 0$ otherwise. Let $\mathbf{e}_i$ be a m-dimensional vector whose i-th element is 1 and all other elements are 0. We introduce the label constraint matrix U as follows:

$$U = \left(\mathbf{u}_1, \cdots, \mathbf{u}_c, \mathbf{e}_{p+1}, \cdots, \mathbf{e}_m\right) \tag{7}$$

As an example, consider that $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are labeled with the first class, $\mathbf{x}_4, \mathbf{x}_5$ are labeled with the second class, and the rest $m-5$ points are unlabeled. Thus, the constraint matrix U can be represented as follows:

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{m-5} \end{pmatrix}$$

where I_{m-5} is a $(m-5)\times(m-5)$ identity matrix. Using the label constraint matrix U, we can impose the label constraints explicitly by introducing an auxiliary vector $\mathbf{z}$:

$$\mathbf{y} = U\mathbf{z} \tag{8}$$

With the above constraint, it is clearly to see that if $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same label, then $y_i = y_j$. Thus, we have:

$$\sum_{i,j=1}^{m} (y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y} = \mathbf{z}^T U^T L U \mathbf{z}$$

and

$$\mathbf{y}^T D \mathbf{y} = \mathbf{z}^T U^T D U \mathbf{z}$$

Finally, the minimization problem reduces to finding:

$$\begin{aligned} \max \quad & \mathbf{z}^T U^T L U \mathbf{z} \\ s.t. \quad & \mathbf{z}^T U^T D U \mathbf{z} = 1, \end{aligned} \tag{9}$$

The optimal vector $\mathbf{z}$ that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$U^T L U \mathbf{z} = \lambda U^T D U \mathbf{z} \tag{10}$$

It is easy to check that L is positive semi-definite and D is positive definite. Since the column vectors of U are linearly independent, for any non-zero $\mathbf{z}$, $U\mathbf{z}$ is not a zero vector. Therefore, $U^T L U$ is still positive semi-definite and $U^T D U$ is positive definite. This implies that $\lambda \geq 0$. Let $\mathbf{1}_d$ be d-dimensional vector of all ones. Note that U is a $m \times (c + m - p)$ matrix and $U\mathbf{1}_{c+m-p} = \mathbf{1}_m$. It is easy to check that $L\mathbf{1}_m = 0$, thus $U^T L U \mathbf{1}_{c+m-p} = 0$. This implies that $\mathbf{1}_{c+m-p}$ is an eigenvector associated with the zero eigenvalue. This eigenvector should be removed since it leads to a constant embedding and all the maps collapse to a single point. Once $\mathbf{z}$ is solved, the embedding result $\mathbf{y}$ can be obtained by Eq. (8). When there is no label information available, $U = I_m$. In this case, our algorithm reduces to the ordinary Laplacian Eigenmap.

4 Out-of-Sample Extension

The approach described so far yields a map which is defined only on the training points. For real applications such as face recognition, we need to have an explicit mapping function which is defined everywhere. In this section, we discuss how to perform out-of-sample extension.

Consider a linear map $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. Let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$ be a $n \times m$ data matrix. Thus, the optimal $\mathbf{a}$ should satisfy:

$$X^T\mathbf{a} = \mathbf{y}$$

However, in reality, such $\mathbf{a}$ may not exist. A possible way is to find $\mathbf{a}$ which can best fit the equation in the least squares sense:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \sum_{i=1}^{m} \left(\mathbf{a}^T\mathbf{x}_i - y_i\right)^2 \tag{11}$$

The technique for solving the least square problem is already matured [Golub and Loan, 1996] and there exist many efficient iterative algorithms (e.g., LSQR [Paige and Saunders, 1982]) that can handle very large scale least square problems.

In the situation that the number of data points is smaller than the number of features, the minimization problem (11) is *ill posed*. There can be infinitely many solutions to the linear equations system $X^T\mathbf{a} = \mathbf{y}$ (the system is under-determined). The most popular way to solve this problem is to impose a Tikhonov regularizer:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left(\sum_{i=1}^{m} \left(\mathbf{a}^T\mathbf{x}_i - y_i\right)^2 + \gamma\|\mathbf{a}\|^2 \right) \tag{12}$$

The regularized least square is also called ridge regression [Hastie *et al.*, 2001]. The $\gamma \geq 0$ is a parameter to control the amounts of shrinkage [Hastie *et al.*, 2001]. The regularized least squares in (12) can be rewritten in the matrix form as follows:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left(\left(X^T\mathbf{a} - \mathbf{y}\right)^T \left(X^T\mathbf{a} - \mathbf{y}\right) + \gamma\mathbf{a}^T\mathbf{a} \right) \tag{13}$$

Requiring the derivative of the right hand side with respect to $\mathbf{a}$ vanish, we get:

$$\mathbf{a} = \left(XX^T + \gamma I\right)^{-1} X\mathbf{y} \tag{14}$$

5 Experimental Results

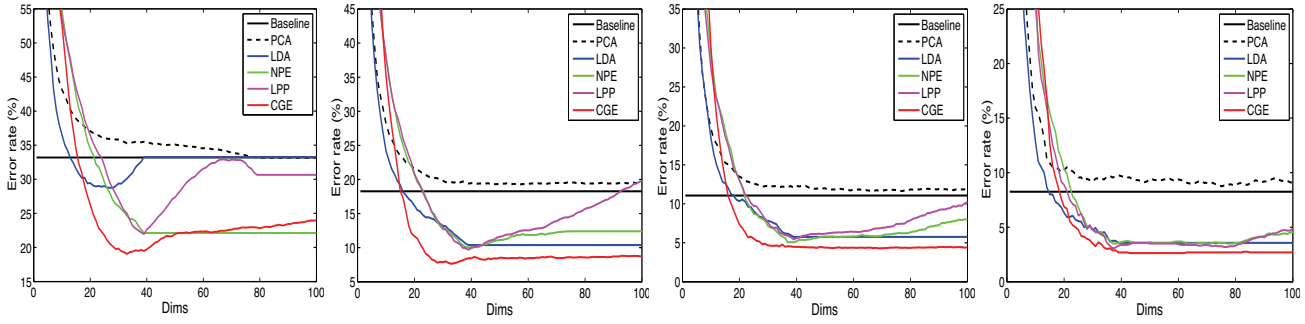
In this section, we investigate the use of CGE on face recognition. We compare our proposed algorithm with Eigenface (PCA, [Turk and Pentland, 1991]), Fisherface (LDA, [Belhumeur *et al.*, 1997]), Laplacianface (LPP, [He *et al.*, 2005b]), and Neighborhood Preserving Embedding (NPE, [He *et al.*, 2005a]). We begin with a brief discussion about data preparation.

5.1 Data Preparation

Two face database were tested. The first one is the AT&T database [1], and the second one is the CMU PIE database [Sim *et al.*, 2003]. In all the experiments, preprocessing to locate the faces were applied. Original images were normalized (in

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Figure 1: The sample cropped face images from AT&T database. The original face images are taken under varying expressions and facial details.



(a) 2 labels (b) 4 labels (c) 6 labels (d) 8 labels

Figure 2: Error rate vs. dimensionality reduction on AT&T database.

Table 1: Recognition error rate of different algorithms on the AT&T database (mean±std-dev%)

Method	2 labels	4 labels	6 labels	8 labels
Baseline	33.2±3.3 (1024)	18.3±2.2 (1024)	11.1±2.3 (1024)	8.25±2.0 (1024)
PCA	33.2±3.3 (79)	18.3±2.2 (159)	11.1±2.3 (238)	8.25±2.0 (316)
LDA	28.7±3.4 (28)	10.4±2.0 (39)	5.75±2.4 (39)	3.56±1.6 (39)
NPE	22.1±3.4 (39)	10.0±1.6 (39)	5.03±2.1 (37)	3.50±2.0 (37)
LPP	22.0±3.0 (39)	9.69±1.6 (39)	5.44±2.0 (39)	3.06±1.8 (37)
CGE	**19.1±2.9** (33)	**7.63±2.2** (33)	**4.28±1.8** (69)	**2.62±1.6** (43)

scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final image for matching. The size of each cropped image in all the experiments is 32×32 pixels, with 256 gray levels per pixel. Thus, each image can be represented by a 1024-dimensional vector in image space. No further preprocessing is done. The nearest neighbor classifier is applied for its simplicity. In our algorithm, the parameter k (number of nearest neighbors) is empirically set to 3, and γ is set to 0.1.

5.2 Face Recognition on AT&T Database

The AT&T face database consists of a total of 400 face images, of a total of 40 subjects (10 samples per subject). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. Fig. 1 shows some of the faces with varying expressions and facial details in the AT&T database.

For each subject, $l(= 2, 4, 6, 8)$ images are randomly selected as labeled set and the rest are considered as unlabeled set for testing. By applying CGE, LPP, PCA, LDA, and NPE, we can learn a face subspace in which the recognition is then performed. For each given l, we average the results over 20 random splits.

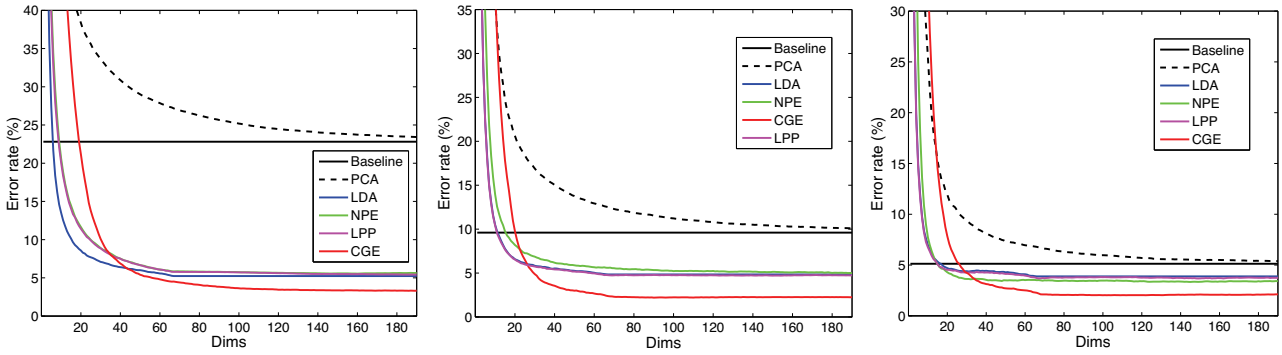
Fig. 2 shows the plots of error rate versus dimensionality reduction for different algorithms. For the baseline method, the recognition is simply performed in the original 1024-dimensional image space without any dimensionality reduction. Note that, the upper bound of the dimensionality of Fisherface is $c - 1$ where c is the number of subjects [Duda *et al.*, 2000]. As can be seen, the performance of these algorithms varies with the number of dimensions. We show the best results obtained by them in Table 1.

Our algorithm outperforms all other five methods. PCA performs the worst in all cases. It does not obtain any improvement over the baseline method. LPP and NPE significantly outperform LDA when there are only 2 labeled samples per subject. As the number of labeled samples increases, the performance difference between LDA, NPE, and LPP gets smaller.

5.3 Face Recognition on CMU PIE Database

The CMU PIE face database [Sim *et al.*, 2003] contains 68 subjects with 41,368 face images as a whole. The face

Figure 3: The sample cropped face images from CMU PIE database, with frontal pose and varying lighting and illumination.



(a) 10 labels (b) 20 labels (c) 30 labels

Figure 4: Error rate vs. dimensionality reduction on CMU PIE database.

Table 2: Recognition error rate of different algorithms on the CMU PIE database (mean±std-dev%)

Method	10 labels	20 labels	30 labels
Baseline	22.8±1.2 (1024)	9.60±0.76 (1024)	5.12±0.62 (1024)
PCA	22.8±1.2 (647)	9.60±0.76 (572)	5.12±0.62 (642)
LDA	5.23±0.41 (67)	4.83±0.45 (67)	3.88±0.40 (67)
NPE	5.55±0.43 (160)	4.99±0.53 (189)	3.33±0.40 (145)
LPP	5.43±0.41 (189)	4.70±0.47 (154)	3.67±0.33 (149)
CGE	**3.31±0.34** (189)	**2.20±0.28** (90)	**2.02±0.33** (102)

images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. In this experiment, we choose the frontal pose (C27) with varying lighting and illumination which leaves us 49 images per subject. Fig. 3 shows the sample cropped face images from the chosen database. For each subject, $l = (10, 20, 30)$ images are randomly selected as labeled samples and the rest are considered as unlabeled samples for testing.

The experimental design is the same as that in the last subsection. For each given l, we average the results over 20 random splits. Fig. 4 shows the plots of error rate versus dimensionality reduction for the PCA, LDA, NPE, LPP, CGE, and baseline methods. The best results obtained in the optimal subspace for each method are shown in Table 2.

As can be seen, our CGE algorithm consistently outperforms the other five algorithms in all the cases. The error rates obtained by LDA, NPE, and LPP are very close to each other.

5.4 Discussion

Several experiments on two standard face database have been systematically performed. We would like to highlight several points:

1. It is clear that the use of dimensionality reduction is beneficial in face recognition. There is a significant increase in performance from using LDA, NPE, LPP, and CGE. However, PCA fails to gain improvement over the baseline. This is because that PCA does not encode the discriminative information.

2. Our CGE algorithm significantly outperforms the canonical subspace learning algorithms (e.g. PCA and LDA) and the state-of-the-art subspace learning algorithms (e.g. NPE and LPP). The reason lies in the fact that CGE constrains the space of the solutions that we explore only to contain embedding results that are consistent with the labels. This way, discriminative information can be encoded in the learned subspace more accurately.

3. The NPE and LPP algorithms are only slightly better than LDA. All of these three algorithms aim to discover the intrinsic manifold structure. They encode the label information in the graph model by assigning higher weights between data points sharing the same label. However, it remains unclear how to select the optimal

weight in a principled manner.

6 Conclusions

This paper introduces a novel dimensionality reduction algorithm called Constrained Graph Embedding to enable more effective discriminant analysis. CGE shares some similar properties to LPP, such as a locality preserving character. However, unlike previous manifold learning algorithms which simply incorporate prior knowledge into the graph structure, our proposed algorithm makes full use of the prior knowledge to constrain the solution space. Thus, the obtained embedding results are consistent with the prior knowledge such that data points sharing the same label are merged together and simultaneously respect the geometrical structure. The experimental results on two standard databases have shown that our algorithm can significantly improve the face recognition accuracy.

The presented algorithm is linear. However, it can be easily extended to nonlinear embedding by using kernel tricks [Schölkopf and Smola, 2002]. Moreover, in this work we use a nearest neighbor graph. It would be interesting to explore other ways to construct the graph for better describing the geometrical and discriminating structure in the data.

Acknowledgements

This work is supported by Program for Changjiang Scholars and Innovative Research Team in University (IRT0652,PCSIRT), National Science Foundation of China under Grant 60875044, and National Key Basic Research Foundation of China under Grant 2009CB320801.

References

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, J. P. Hepanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[Belkin and Niyogi, 2001] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001.

[Brand, 2003] Matthew Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression for efficient regularized subspace learning. In *Proc. IEEE International Conference on Computer Vision*, Rio de Janeriro, Brazil, 2007.

[Chen *et al.*, 2005] Hwann-Tzong Chen, Huang-Wei Chang, and Tyng-Luh Liu. Local discriminant embedding and its variants. In *Proc. 2005 Internal Conference on Computer Vision and Pattern Recognition*, 2005.

[Chung, 1997] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[Duda *et al.*, 2000] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[Golub and Loan, 1996] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[Guattery and Miller, 2000] Stephen Guattery and Gary L. Miller. Graph embeddings and laplacian eigenvalues. *SIAM Journal on Matrix Analysis and Applications*, 21(3):703–723, 2000.

[Hastie *et al.*, 2001] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[He *et al.*, 2005a] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *Proc. International Conference on Computer Vision (ICCV'05)*, Beijing, China, 2005.

[He *et al.*, 2005b] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[Li *et al.*, 2008] Xuelong Li, Stephen Lin, Shuicheng Yan, and Dong Xu. Discriminant locally linear embedding with high-order tensor data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(2):342–352, 2008.

[Paige and Saunders, 1982] Christopher C. Paige and Michael A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[Roweis and Saul, 2000] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Schölkopf and Smola, 2002] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2002.

[Sim *et al.*, 2003] T. Sim, S. Baker, and M. Bsat. The CMU pose, illuminlation, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[Sugiyama, 2007] Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

[Tenenbaum *et al.*, 2000] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.