Report No. UIUCDCS-R-2005-2636 **UILU-ENG-2005-1826**

Using Graph Model for Face Analysis

by

Deng Cai, Xiaofei He, and Jiawei Han

September 2005

Using Graph Model for Face Analysis*

Deng Cai† Xiaofei He‡ Jiawei Han†

† Department of Computer Science, University of Illinois at Urbana-Champaign

‡ Department of Computer Science, University of Chicago

Abstract

Previous work has demonstrated that the image variations of many objects (human faces in particular) under variable lighting can be effectively modelled by low dimensional linear spaces. The typical methods for learning a face subspace include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Locality Preserving Projection (LPP). Theoretical analysis shows that all these three methods can be obtained from different graph models which correspond to different geometrical structures. In this paper, we systematically analyze the relationship between these three subspace methods. We shows that LPP provides a more general framework for subspace learning and a natural solution to the small sample issue in LDA. Extensive experiments on face recognition and clustering are performed on Yale, ORL and PIE databases.

1 Introduction

There is currently a great deal of interest in appearance-based approaches to face recognition [24], [1], [7]. When using appearance-based approaches, we usually represent an image of size $m_1 \times m_2$ pixels by a vector in $\mathbb{R}^{m_1 \times m_2}$. Throughout this paper, we denote by *face space* the set of all the face images. The face space is generally a low dimensional manifold embedded in the ambient space [19], [22]. The typical linear algorithms for learning such a face manifold for recognition include Principal Component Analysis (PCA) [5], Linear Discriminant Analysis (LDA) [5], and Locality Preserving Projection (LPP) [9].

PCA projects the data along the direction of maximal variance. It is optimal in the sense of reconstruction error. The basis functions of PCA can be obtained by solving the eigenvectors of the covariance matrix. They are generally called Eigenfaces [24]. Unlike PCA which is unsupervised, LDA is supervised. PCA is optimal in representation, while LDA is optimal in discrimination.

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

LDA 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. The basis functions obtained by LDA are generally called Fisherfaces [1].

LDA (Fisherfaces approach) is successfully applied in many applications. However, the singularity of the within-class scatter matrix makes the computation of LDA complicated and less optimal, especially for high dimensional data. Moreover, the number of available projection directions of LDA is bounded by the class number. Many efforts have been devoted to improve the performance of Fisherfaces. Among them, Null subspace methods are the most popular one and it effectively applied the null subspace of the within-class scatter matrix [3], [27], [11]. However, the fundamental problem and limitations in LDA are still not solved in theory.

LPP is a recently proposed linear dimensionality reduction algorithm [9]. LPP builds a graph model which reflects the intrinsic geometrical structure of the data space, and finds a projection that respects this graph structure. LPPs are linear approximations to the Laplace-Beltrami operator on the face manifold. The basis functions obtained by LPP are generally called Laplacianfaces [7]. It has been shown that both PCA, LDA and LPP arise from the same principle applied to different choices of this graph structure [7].

It has been shown that LPP performs much better than PCA on face recognition and marginally better than LDA [7]. However, the reason that LPP outperforms LDA is still unclear. In this paper, we consider the following problems:

1. Under what situations LPP can outperform LDA?
2. What is the connection between LPP and null space LDA methods?
3. Finally, since LPP can be applied under either supervised, semi-supervised or unsupervised mode, the Laplacianfaces method can be naturally used for face clustering. Thus, how is Laplacianfaces compared with Eigenfaces? What is the relationship between clustering using LPP and canonical spectral clustering algorithms [20][16][2]?

In this paper, we provide a systematical analysis of the properties of LPP in unsupervised and supervised setting, the difference of which can be reflected on the constructions of the graphs. For supervised situation, we analyze the relationship between LPP and LDA. We shows that LPP provides a more general framework for subspace learning and a natural solution to the small sample size problem [3] in LDA. For unsupervised situation, we analyze the connection between LPP and spectral clustering. By constructing a nearest neighbor graph, LPP provides an unsupervised approximation to the supervised LDA, which intuitively explains why LPP can outperform PCA for clustering.

It is worthwhile to highlight several aspects of our analysis for LPP here:

1. LPP provides a general graph embedding approach for subspace learning, which can be performed in supervised, semi-supervised and unsupervised mode. The graph construction is the

key step of LPP.

2. LDA is a special case of LPP under supervised mode. The general graph embedding approach makes LPP more applicable and the small sample issue in LDA can be naturally solved in LPP approach.

3. When performed in unsupervised mode, the p-nearest neighbor graph provides an approximate to the label information of the data points. Thus LPP can outperform PCA in unsupervised setting. Also, the connection between LPP and spectral clustering reveals that Laplacianfaces approach is also suitable for face clustering.

The rest of this paper is organized as follows: Section 2 provides a brief review of PCA, LDA and LPP. Section 3 gives the analysis of LPP in both supervised and unsupervised situation. Extensive experiments for face recognition and face clustering on Yale, ORL and PIE face databases are provided in Section 4. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2 A Brief review of PCA, LDA and LPP

PCA is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original m-dimensional data onto the $k(<< m)$-dimensional linear subspace spanned by the leading eigenvectors of the datas covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. Given a set of data points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, let $\mathbf{a}$ be the transformation vector and $y_i = \mathbf{a}^T\mathbf{x}_i$. The objective function of PCA is as follows:

$$\begin{aligned} \mathbf{a}_{opt} &= \arg\max_{\mathbf{a}} \sum_{i=1}^{n} (y_i - \overline{y})^2 \\ &= \arg\max_{\mathbf{a}} \mathbf{a}^T C \mathbf{a} \end{aligned}$$

where $\overline{y} = \frac{1}{n}\sum y_i$ and C is the data covariance matrix. The basis functions of PCA are the eigenvectors of the data covariance matrix associated with the largest eigenvalues. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Turk and Pentland [24] use Principal Component Analysis to describe face images in terms of a set of basis functions, or "eigenfaces."

While PCA seeks directions that are efficient for representation, Linear Discriminant Analysis (LDA) seeks directions that are efficient for discrimination. LDA is a supervised learning algorithm, 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. Suppose we have a set of n

samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, belonging to c classes. The objective function of LDA is as follows:

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_B \mathbf{a}}{\mathbf{a}^T S_W \mathbf{a}}$$

$$S_B = \sum_{i=1}^{c} n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T \tag{1}$$

$$S_W = \sum_{i=1}^{c} \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right) \tag{2}$$

where $\mathbf{m}$ is the total sample mean vector, n_i is the number of samples in the i-th class, $\mathbf{m}^{(i)}$ is the average vector of the i-th class, and $\mathbf{x}_j^{(i)}$ is the j-th sample in the i-th class. We call S_W the within-class scatter matrix and S_B the between-class scatter matrix. The basis functions of LDA are the eigenvectors of the following generalized eig-problem associated with the largest eigenvalues:

$$S_B \mathbf{a} = \lambda S_W \mathbf{a} \tag{3}$$

Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linearly separable space using bases that are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA. However, some recent work [14] shows that, when the training data set is small, PCA can outperform LDA, and also that PCA is less sensitive to different training data sets.

Similar to LDA, LPP also aims to extract the most discriminative features. Given a affinity matrix W, LPP can be obtained by solving the following minimization problem:

$$\begin{aligned} \mathbf{a}_{opt} &= \arg\min_{\mathbf{a}} \sum_{i=1}^{m} \left(\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j \right)^2 W_{ij} \\ &= \arg\min_{\mathbf{a}} \mathbf{a}^T X L X^T \mathbf{a} \end{aligned}$$

with the constraint

$$\mathbf{a}^T X D X^T \mathbf{a} = 1$$

where $L = D - W$ is the *graph Laplacian* [4] and $D_{ii} = \sum_j W_{ij}$. D_{ii} measures the local density around $\mathbf{x}_i$. LPP constructs the affinity matrix W as:

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

Thus, the objective function in LPP incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close" then y_i ($= \mathbf{a}^T \mathbf{x}_i$) and y_j ($= \mathbf{a}^T \mathbf{x}_j$) are close as well [8]. Finally, the basis functions of LPP are the eigenvectors associated with the smallest eigenvalues of the following generalized eig-problem:

$$X L X^T \mathbf{a} = \lambda X D X^T \mathbf{a} \tag{4}$$

3 Computational Analysis of LPP

The graph construction is the key of LPP. In this section, we systematically analysis the properties of LPP under different graph. Particularly, we will show that LDA is a special case of LPP under a specific supervised graph. LPP can naturally solve two intrinsic problem in tractional LDA, i.e. small sample size problem and only has $c-1$ available projection directions, where c is the number of categories. We will also show LPP in unsupervised mode has direct connection with spectral clustering. The graph model of LPP make it very easy to incorporate label information, thus, LPP provides a more general general graph embedding approach for subspace learning, for supervised, unsupervised and semi-supervised learning.

3.1 LPP in Supervised Mode

In section 2, we construct the affinity graph as: put edge between two nodes if they are neighbors and the weight of each edge is set to 1. This is a very simple graph construction method. For classification task, since the label information are available, we can utilize the label information to construct a better graph with respect to discriminant information. In the following, we will discuss two possible graph construction approaches which utilize the label information. One approach makes LPP identical to LDA while do not suffer the small sample size problem; the other approach makes LPP generate more than $c-1$ valuable projection directions which will be useful for face recognition.

3.1.1 LPP1

We construct the affinity matrix matrix as:

$$W_{ij} = \begin{cases} \frac{1}{n_l}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } l\text{-th class;} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where n_l is the number of samples in the l-th class. In such setting, we can easily check that $D = I$ and

$$L = D - W = I - W \tag{6}$$

where I is a $n \times n$ identity matrix. Thus, the eig-problem of LPP in Eq. 4 becomes:

$$XLX^T\mathbf{a} = \lambda XX^T\mathbf{a} \tag{7}$$

In LDA, the within-class scatter matrix S_B and the between-class scatter matrix S_W are defined as Eq. 1 and Eq 2. The data covariance matrix S_C can be defined as:

$$S_C = \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

and we have [23]

$$S_C = S_W + S_B$$

He et. al. [7] proved that with the affinity matrix as Eq. 5, $S_W = XLX^T$. If we remove the mean of the data at the preprocessing step, we have $S_C = XX^T$ and $S_B = XWX^T$. Thus the eig-problem of LDA in Eq. 3 becomes:

$$\begin{aligned} & S_B\mathbf{a} = \lambda S_W\mathbf{a} \\ \Rightarrow \quad & (S_C - S_W)\mathbf{a} = \lambda S_W\mathbf{a} \\ \Rightarrow \quad & S_W\mathbf{a} = \frac{1}{1+\lambda} S_C\mathbf{a} \\ \Rightarrow \quad & XLX^T\mathbf{a} = \frac{1}{1+\lambda} XX^T\mathbf{a} \\ \Rightarrow \quad & XLX^T\mathbf{a} = \lambda' XX^T\mathbf{a} \end{aligned} \tag{8}$$

where $\lambda' = 1/(1+\lambda)$. Eq. 8 provides a graph embedding point of view of canonical LDA. Thus, the eigenvector of the eig-problem (3) associated with largest eigenvalue λ (the solution of LDA) is equivalent to the eigenvector of the eig-problem (8) associated with smallest eigenvalue λ', the latter is exactly the the solution of LPP in Eq. 7.

The above analysis shows that with specific constructed graph, W in Eq. 5 particularly, the solution of LPP is identical to LDA. Thus, LDA is a special case of LPP.

It can also be proved that the rank of L in Eq. 6 is $n-c$ thus the rank of within-class scatter matrix $S_W = XLX^T$ is at most $n-c$ [7]. However, in many cases, in appearance-based face recognition, the number of pixels in an face image (or, the dimensionality of the face space) is larger than $n-c$, i.e., $m > n-c$. Thus, XLX^T is singular. In order to overcome the complication of a singular XLX^T , Belhumeur et al. [1] proposed the Fisherface approach that the face images are projected from the original image space to a subspace with dimensionality $n-c$ using PCA and then LDA is performed in this subspace. In small sample size situation, it is possible that $n-c \ll m$, the PCA projection step in Fisherface approach only keep the largest $n-c$ principle components and might lose a lot of information. In the LPP framework, the LDA can be written as a graph embedding approach in Eq. 8, we only need to ensure XX^T is full rank. Thus, we can keep all the non-zero eigenvalue in the PCA step, that is, there will be no information loss in the PCA step, even on small sample size situation. This is one of the main reasons why Laplacianface approach can outperform Fisherface approach, which will be shown in the experiment part. Moreover, the PCA step can be used to remove noise, as suggested in [7]. In the LPP framework, we can keep any percentage of eigenvalue in the PCA step which gives us maximal flexibility.

3.1.2 LPP2

The affinity matrix W as Eq. 5 makes LPP identical to LDA. The number of effective projection directions in such LPP are also $c-1$. This happens due to the rank of W is only c, since row i and

row j of W are exactly same if the data point i and j are belong to the same class.

To solve this problem, we can construct the W as

$$W_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Since it is hard to choose a suitable parameter t, we can also use the following W:

$$W_{ij} = \begin{cases} \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

In such setting, the D will be no longer identity matrix and the solution of the LPP will not longer be same with the solution of the LDA. Our experiments show that in such W setting, the recognition results of LPP using the first $c-1$ dimensions are nearly equal to the recognition results of LDA. Moreover, row i and row j of W will not likely to be same even if the data point i and j are belong to the same class. Thus, we can get more than $c-1$ eigenvectors for the solution of Eq. 4 which will lead better results in some cases.

3.2 LPP in Unsupervised Mode

In unsupervised learning, like clustering, the label information is not available. We can construct a p-nearest neighbor graph:

$$W_{ij} = \begin{cases} \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}, & \text{if } \mathbf{x}_i \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_i \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Since the neighboring data points probably belong to the same underlying class, the p-nearest neighbor graph gives us an approximation to the supervised graph of LDA in Eq. 5. Therefore, even though the label information is not available, we can still discover the discriminating structure to some extent by using LPP.

3.2.1 Connection to Spectral Clustering

The LPP in unsupervised mode can naturally be used as the first step of clustering: the high dimensional data can be projected to low dimension by LPP and then traditional clustering algorithm like K-means be applied. It is interesting to study what is the relation between K-means after LPP and spectral clustering.

It would be important to note that spectral clustering algorithms [20][17] can be simply thought of as a combination of Laplacian Eigenmaps [2] and K-means. Thus, the only difference between K-means after LPP and spectral clustering lies the difference between LPP and Laplacian Eigenmaps. The Laplacian Eigenmaps tries to solve the following minimum eigenvalue problem.

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{12}$$

where L and D can be defined same as in Eq. 4. After the spectral embedding step, the eigenvector $\mathbf{y}$ will be the low dimensional representation of original data points.

Recall the eigen-problem in LPP:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \tag{13}$$

The eigenvector $\mathbf{a}$ will be the mapping function and $X^T\mathbf{a}$ will be the low dimensional representation of original data points. Thus, the LPP impose a constraint on Eqn. (12) that $\mathbf{y}$ should be the linear combination of the column vectors of X^T.

The eigenvalue problem of LPP scales with the number of dimensions (m), while that of Laplacian Eigenmaps scales with the number of data points (n). The rank of X is no greater than $min(n, m)$. Thus, if $m > n$, we can reduce the data space into an n dimensional subspace without losing any information by using Singular Value Decomposition (SVD). Correspondingly, the data matrix X in such a subspace becomes a square matrix. We have the following proposition:

Proposition 1 *If X is a full rank square matrix, then LPP and Laplacian Eigenmap have the same result.*

Proof Recall that the eigenvalue problem of LPP is as follows:

$$XLX^T\mathbf{w} = \lambda XDX^T\mathbf{w} \tag{14}$$

Let $\mathbf{y} = X^T\mathbf{w}$. Equation (14) can be rewritten as follows:

$$XL\mathbf{y} = \lambda XD\mathbf{y} \tag{15}$$

Since X is a full rank square matrix, we get the following equation:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{16}$$

which is just the eigenvalue problem of Laplacian Eigenmaps. ∎

In many real world applications such as face clustering, the dimensionality of the face space is typically much larger than the number of faces. In such a case, LPP and Laplacian Eigenmaps will have the same embedding result if these face vectors are linearly independent. In such case, clustering using LPP will be identical to the spectral clustering.

4 Experimental Results

In this section, we investigate the performance of LPP for face recognition and face clustering. For face recognition, we focus on in which situation LPP in supervised mode (Laplacianface) can outperform LDA (Fisherface [1]). For face clustering, we are interested in how LPP in unsupervised mode can outperform PCA (Eigenface [24]).

4.1 Experiments on Face Recognition

We compared four methods in the face recognition experiment:

- Eigenface method (PCA) [25]
- Fisherface method (PCA+LDA) [1], Tthere are at most $c-1$ nonzero generalized eigenvalues and, so, an upper bound on the dimension of the reduced space is $c-1$.
- LPP1. Construct the graph as Eq. 5. LPP in such graph setting is identical to LDA. The upper bound on the dimension of the reduced space is still $c-1$. The difference between this LPP1 and Fisherface is that Fisherface keeps $n-c$ dimensions in the PCA step while LPP1 keeps all the non-zero eigenvalues in the PCA step.
- LPP2. Construct the graph as Eq. 10. We can keep more than $c-1$ dimensions since the graph matrix W is no longer rank c.

Three face databases were tested. The first one is the Yale database[1], the second is the ORL (Olivetti Research Laboratory) datebase[2], and the third is the PIE (pose, illumination, and expression) database from CMU [21]. In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Each image is represented by a $1,024$-dimensional vector in image space. Different pattern classifiers have been applied for face recognition, including nearest-neighbor [1], Bayesian [15], Support Vector Machine [18], etc. In this paper, we apply the nearest-neighbor classifier for its simplicity. The Euclidean metric is used as our distance measure.

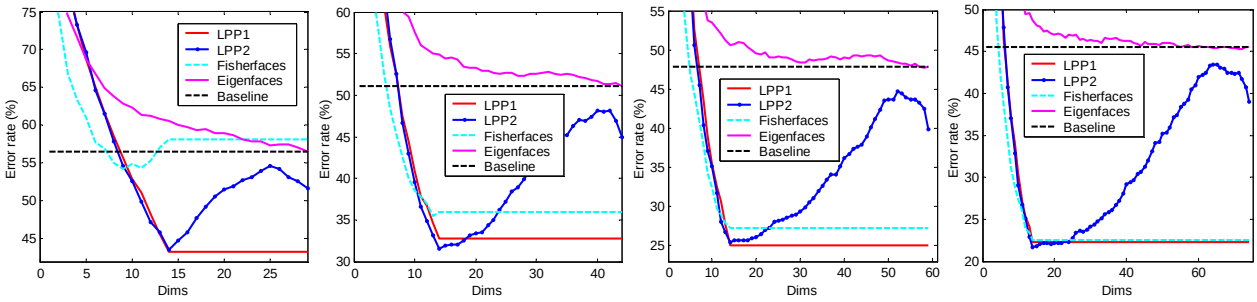
In short, the recognition process has three steps. First, we calculate the face subspace from the training samples; then the new face image to be identified is projected into d-dimensional subspace by using PCA, LDA and LPP; finally, the new face image is identified by a nearest neighbor classifier.

4.1.1 Yale Database

The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 gray scale images of 15 individuals. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). Figure 1 shows the 11 images of one individual in Yale data base. A random subset with $l(= 2, 3, 4, 5)$ images per individual was taken with labels to form the training set, and the rest of the database was

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[2] http://www.uk.research.att.com/facedatabase.html

Figure 1: Sample face images from the Yale database. For each subject, there are 11 face images under different lighting conditions with facial expression.



(a) 2 Train (b) 3 Train (c) 4 Train (d) 5 Train

Figure 2: Error rate vs. dimensionality reduction on Yale database

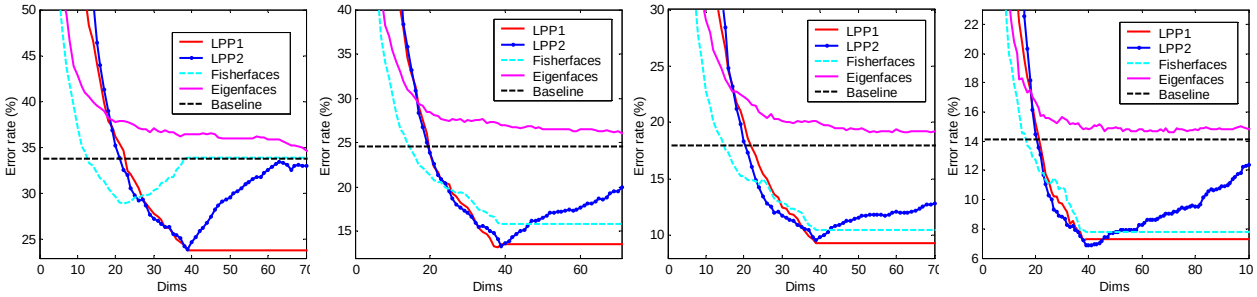
Table 1: Performance comparisons on the Yale database

Method	2 Train	3 Train	4 Train	5 Train
Baseline	56.5%	51.1%	47.8%	45.6%
Eigenfaces	56.5% (29)	51.1% (44)	47.8% (58)	45.2% (71)
Fisherfaces	54.3% (9)	35.5% (13)	27.3%(14)	22.5% (14)
LPP1	**43.3% (14)**	32.7% (14)	**25.1% (14)**	22.3% (14)
LPP2	43.5% (14)	**31.5% (14)**	25.4% (14)	**21.7% (14)**

considered to be the testing set. For each given l, we average the results over 20 random splits. In general, the performance of all these methods varies with the number of dimensions. We show the best results and the optimal dimensionality obtained by Eigenface, Fisherface, LPP1, LPP2, and baseline methods in Table 1. For the baseline method, the recognition is simply performed in the original 1024-dimensional image space without any dimensionality reduction.

As can be seen, both two versions' LPP outperform Fisherface, especially in small training sample case. Figure 2 shows the plots of error rate versus dimensionality reduction. It is worthwhile to note that when kept $c-1$ dimensions in 2 training case, Fisherfaces is even worse than baseline. This result is consistent with the observation in [14] that Eigenface method can outperform Fisherface method when the training set is small. Our analysis in section 3.1 indicate the only difference between LPP1 and Fisherface is that Fisherface keeps $n-c$ dimensions in the PCA step while LPP1 keeps all the non-zero eigenvalues. In small training sample case, some essential information might be lost when only keeping $n-c$ dimensions. In LPP framework, we can keep all the information in the PCA step, thus, can achieve better performance. Note that, although LPP2 can use more than $c-1$ projection directions, it seems that the extra dimensions are useless in this database.

Figure 3: Sample face images from the ORL database. For each subject, there are 10 face images with different facial expression and details.



(a) 2 Train (b) 3 Train (c) 4 Train (d) 5 Train

Figure 4: Error rate vs. dimensionality reduction on ORL database

Table 2: Performance comparisons on the ORL database

Method	2 Train	3 Train	4 Train	5 Train
Baseline	33.8%	24.6%	18.0%	14.1%
Eigenfaces	33.7% (78)	24.6% (119)	18.0% (159)	14.1% (199)
Fisherfaces	28.9% (22)	15.8% (39)	10.5% (39)	7.75% (39)
LPP1	**23.9% (39)**	**13.3% (38)**	**9.33% (39)**	7.30% (39)
LPP2	**23.9% (39)**	13.4% (39)	9.58% (39)	**6.85% (40)**

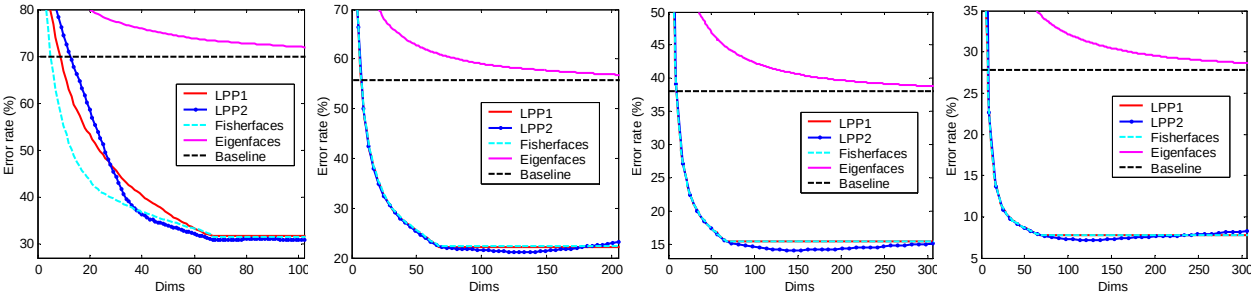
4.1.2 ORL Database

The ORL (Olivetti Research Laboratory) face database is used for this test. It contains 400 images of 40 individuals. Some images were captured at different times and have different variations including expression (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. 10 sample images of one individual in the ORL database are displayed in Figure 3. A random subset with $l(= 2, 3, 4, 5)$ images per individual was taken with labels to form the training set. The rest of the database was considered to be the testing set. For each given l, we average the results over 20 random splits. The experimental protocol is the same as before. The recognition results are shown in Table 2 and Figure 4. We can get a similar conclusion as before.

4.1.3 PIE Database

The CMU PIE face database contains 68 individuals with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination, and expression. We choose the five near frontal poses (C05, C07, C09, C27, C29) and use all

Figure 5: Sample face images from the CMU PIE database. For each subject, there are 170 near frontal face images under varying pose, illumination, and expression.



(a) 5 Train (b) 10 Train (c) 20 Train (d) 30 Train

Figure 6: Error rate vs. dimensionality reduction on PIE database

Table 3: Performance comparisons on the PIE database

Method	5 Train	10 Train	20 Train	30 Train
Baseline	69.9%	55.7%	38.2%	27.9%
Eigenfaces	69.9% (338)	55.7% (654)	38.1% (889)	27.9% (990)
Fisherfaces	31.5% (67)	22.4% (67)	15.4% (67)	7.77% (67)
LPP1	31.6% (67)	22.2% (67)	15.4% (67)	7.77% (67)
LPP2	**30.8% (67)**	**21.1% (134)**	**14.1% (146)**	**7.13% (131)**

the images under different illuminations, lighting and expressions which leaves us 170 near frontal face images for each individual. Figure 5 shows several sample images of one individual with different poses, expressions and illuminations. We randomly chose $l(= 5, 10, 20, 30)$ images for each individual as training data. We repeated this process 20 times and computed the average performance. Table 3 shows the recognition results.

As can be seen, in 20 and 30 training sample cases, the LPP1 and Fisherfaces are identical. This is because with so many training sample, $n - c$ is larger than the feature number, which is 1024. LPP2 approach, which can keep more than $c - 1$ projection directions, outperforms LPP1 in this database. Figure 6 shows a plot of error rate versus dimensionality reduction.

Figure 7: Sample face images from the CMU PIE database. For each subject, there are 22 face images under different lighting conditions with fixed pose (C27) and expression.

4.2 Experiments on Face Clustering

In this subsection, we evaluate the use of our algorithm on face clustering [6] [10][12]. The CMU PIE face database [21] is used in this experiment. In this experiment, we fixed the pose and expression. Thus, for each subject, we got 22 images under different lighting conditions. Figure 7 shows some sample images for a certain subject.

For each given number k, k classes were randomly selected from the face database. This process was repeated 20 times (except for $k = 68$) and the average performance was computed. For each single test (given k classes of faces), two algorithms, i.e. PCA and LPP are used to learn the face subspace. The K-means was then performed in the subspace as well as the original face space. And the K-means was repeated 10 times with different initializations and the best result in terms of the objective function of K-means was recorded.

4.2.1 Evaluation Metrics

The clustering result is evaluated by comparing the obtained label of each data point with that provided by the data corpus. Two metrics, the accuracy (AC) and the normalized mutual information metric ($\overline{MI}$) are used to measure the clustering performance [26]. Given a data point $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the data corpus, respectively. The AC is defined as follows:

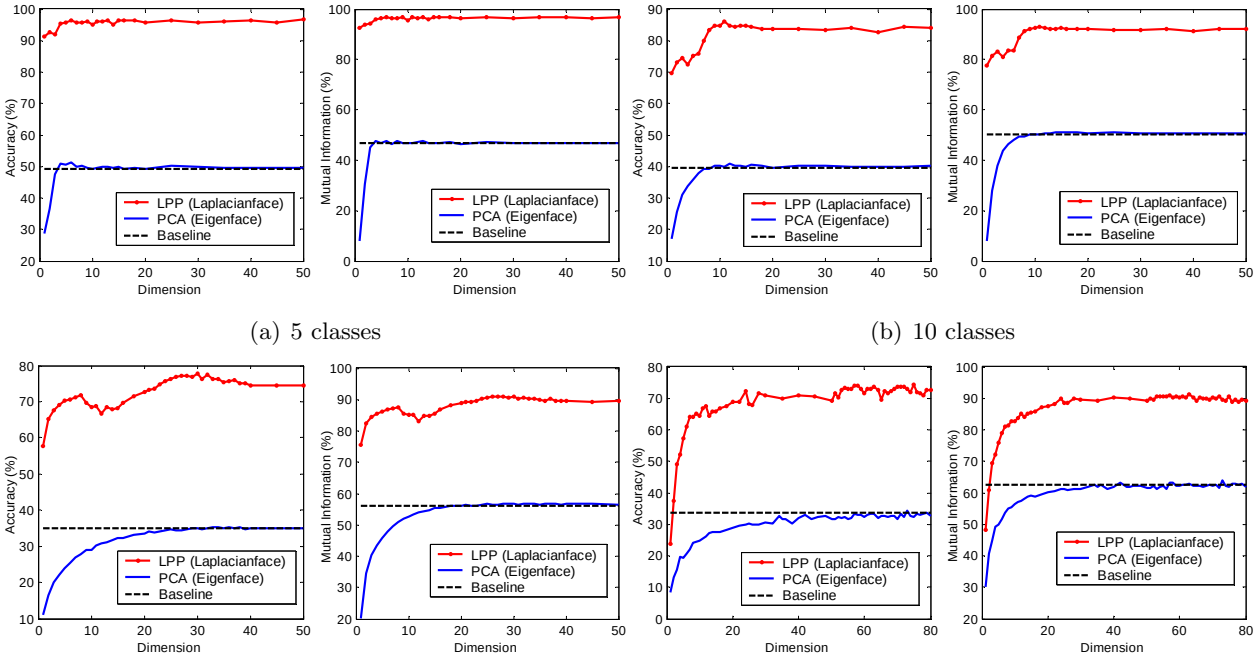
$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{17}$$

where n is the total number of data points and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [13].

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a data point arbitrarily selected from the corpus



(a) 5 classes

(b) 10 classes

(c) 30 classes

(d) 68 classes

Figure 8: Clustering performance vs. dimensionality reduction on CMU PIE face database

Table 4: Clustering performance comparisons on CMU PIE face database

k	Accuracy (%)			Mutual Information (%)		
	Kmeans	PCA+Kmeans	LPP+Kmeans	Kmeans	PCA+Kmeans	LPP+Kmeans
5	49.3	51.3	**96.6**	46.7	47.8	**97.0**
10	39.7	40.8	**86.1**	50.1	51.1	**92.9**
30	34.9	35.4	**77.8**	56.1	56.9	**90.9**
68	33.6	34.4	**74.5**	62.6	63.9	**91.3**

belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \tag{18}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}(C, C')$ ranges from 0 to 1. $\overline{MI} = 1$ if the two sets of clusters are identical, and $\overline{MI} = 0$ if the two sets are independent.

4.2.2 Results

We compared Laplacianface (LPP) with Eigenface (PCA) for clustering. Several tests were performed with different numbers of clusters (k=5, 10, 30, 68). In all the tests, the graph is constructed as Eq. 11 and the number of nearest neighbors in LPP is taken to be 5. The experimental results are shown in Fig. 8 and Table 4. As can be seen, in all these cases, LPP+K-means performs much better than K-means and PCA+K-means. Little improvement can be gained by PCA based clustering. This is because that, PCA can extract the most representative features, but it fails to extract the most discriminative features.

5 Conclusions

In this paper, we systematically analysis the properties of LPP both in unsupervised and supervised setting, the difference of which can be reflected on the constructions of the graphs. For supervised situation, we analyze the relationship between LPP and LDA. We shows that LPP provides a more general framework for subspace learning and a natural solution to the small sample size problem [3] in LDA. For unsupervised situation, we analyze the connection between LPP and spectral clustering. By constructing a nearest neighbor graph, LPP provides an unsupervised approximation to the supervised LDA, which intuitively explains why LPP can outperform PCA for clustering.

Several questions remain to be investigated in our future work:

1. As we show in section 3.2, LPP in unsupervised mode provides an optimal approximation to LDA which is supervised. Specifically, we approximate the graph defined in Eq. by a p-nearest neighbor graph. The more accurate the approximation is, the more discriminating power LPP has. However, it is unclear if there exists better approximation and how to obtain it.

2. Face recognition is an supervised learning process and clustering is inherently an unsupervised learning process. Sometimes, despite the large amount of unlabeled data, a small set of labeled data points might be available. In such a case, the unsupervised learning becomes semi-unsupervised (or, semi-supervised) learning. Thus, the optimal projection can be obtained by preserving locality as well as separating the data points with different labels. It is unclear how to obtain such constraint LPP, though it seems to be promising.

References

[1] P.N. Belhumeur, J.P. Hepanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*,

19(7):711–720, 1997.

[2] 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.

[3] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

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

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

[6] A. Fitzgibbon and A. Zisserman. Joint manifold distance: a new approach to appearance based clustering. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[7] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3), 2005.

[8] Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality preserving indexing for document representation. In *Proc. 2004 Int. Conf. on Research and Development in Information Retrieval (SIGIR'04)*, pages 96–103, Sheffield, UK, July 2004.

[9] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2003.

[10] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[11] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of lda. In *16th International Conference on Pattern Recognition (ICPR'02)*, 2002.

[12] J. Lim, J. Ho, M-H. Yang, K-C. Lee, and D. Kriegman. Image clustering with metric, local linear structure and affinity symmetry. In *Proceedings of the European Conference on Computer Vision*, 2004.

[13] L. Lovasz and M. Plummer. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest, 1986.

[14] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Trans. on PAMI*, 23(2):228–233, 2001.

[15] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on PAMI*, 19(7):696–710, 1997.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[17] Andrew Y. Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, Cambridge, MA, 2001.

[18] P. J. Phillips. Support vector machines applied to face recognition. *Advances in Neural Information Processing Systems*, 11:803–809, 1998.

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

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.

[21] T. Sim, S. Baker, and M. Bsat. The CMU pose, illuminlation, and expression database. *IEEE Trans. on PAMI*, 25(12):1615–1618, 2003.

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

[23] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

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

[25] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.

[26] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proc. 2003 Int. Conf. on Research and Development in Information Retrieval (SIGIR'03)*, pages 267–273, Toronto, Canada, Aug. 2003.

[27] H. Yu and J. Yang. A direct LDA algorithm for high dimensional data-with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.