Statistical and Computational Analysis of Locality Preserving Projection

Xiaofei He XIAOFEI@CS.UCHICAGO.EDU
Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637

Deng Cai DENGCAI2@CS.UIUC.EDU
Department of Computer Science, UIUC, 201 N. Goodwin Avenue Urbana, IL 61801

Wanli Min WANLIMIN@US.IBM.COM
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

Abstract

Recently, several manifold learning algorithms have been proposed, such as ISOMAP (Tenenbaum et al., 2000), Locally Linear Embedding (Roweis & Saul, 2000), Laplacian Eigenmap (Belkin & Niyogi, 2001), Locality Preserving Projection (LPP) (He & Niyogi, 2003), etc. All of them aim at discovering the meaningful low dimensional structure of the data space. In this paper, we present a statistical analysis of the LPP algorithm. Different from Principal Component Analysis (PCA) which obtains a subspace spanned by the *largest* eigenvectors of the *global* covariance matrix, we show that LPP obtains a subspace spanned by the *smallest* eigenvectors of the *local* covariance matrix. We applied PCA and LPP to real world document clustering task. Experimental results show that the performance can be significantly improved in the subspace, and especially LPP works much better than PCA.

1. Introduction

Manifold learning algorithms based on geometrical analysis have received a lot of attention in recent years. The typical algorithms include ISOMAP (Tenenbaum et al., 2000), Locally Linear Embedding (Roweis & Saul, 2000), Laplacian Eigenmap (Belkin & Niyogi, 2001), Locality Preserving Projection (LPP) (He & Niyogi, 2003), etc. The former three are non-linear algorithms, while LPP is linear. The basic assumption of these algorithms is that, in many cases of interest, the observed data are sampled from a underlying sub-manifold which is embedded in high dimensional space. The task of manifold learning algorithms is to discover the meaningful low dimensional structure.

LPP is originally derived by linearly approximating the eigenfunctions of the Laplace Beltrami operator on the manifold. Its applications on face recognition (He et al., 2005) and document representation (He et al., 2004) have shown its effectiveness in discovering the local geometrical structure of the data space. PCA is a classical technique for linear dimensionality reduction. Different from LPP which aims at preserving the local structure, PCA aims at preserving the global structure. It projects the data along the directions of maximal variances. The basis functions obtained by PCA are the eigenvectors of the data covariance matrix. Bregler and Omohundro have proposed a *local* PCA approach to discover the local geometrical structures of the data space (Bregler & Omohundro, 1995). The main difference between local PCA and LPP is that local PCA is globally non-linear while LPP is globally linear.

In this paper, we show that the basis functions obtained by LPP are the eigenvectors of the *local* covariance matrix. We first define a ϵ *covariance matrix*. Thus, the standard covariance matrix used in PCA and the local covariance matrix used in LPP can be characterized by different choices of ϵ. Specifically, in PCA, ϵ is chosen to be infinity, while in LPP, ϵ is chosen to be sufficiently small. We also show that, under certain conditions, LPP and Laplacian Eigenmap can give the same result.

The rest of this paper is organized as follows: Section 2

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

provides a brief review of PCA and LPP. In Section 3, we provide a statistical analysis of the LPP algorithm. In Section 4, we show that LPP and Laplaican Eigenmap can give the same result under certain conditions. Some experimental evaluations on document clustering are provided in Section 5. Finally, we conclude our paper in Section 6.

2. A Brief Review of PCA and LPP

PCA is a canonical linear dimensionality reduction algorithm. The basic idea of PCA is to project the data along the directions of maximal variances so that the reconstruction error can be minimized. Given a set of data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, let $\mathbf{w}$ be the transformation vector and $y_i = \mathbf{w}^T\mathbf{x}_i$. Let $\overline{\mathbf{x}} = \frac{1}{n}\sum \mathbf{x}_i$ and $\overline{y} = \frac{1}{n}\sum y_i$. The objective function of PCA is as follows:

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

where $C = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x} - \overline{\mathbf{x}}) (\mathbf{x} - \overline{\mathbf{x}})^T$ is the data covariance matrix. The basis functions of PCA are the eigenvectors of the data covariance matrix associated with the largest eigenvalues.

Different from PCA which aims to preserve the global structure, LPP aims to preserve the local structure. Given a local similarity matrix S, the optimal projections can be obtained by solving the following minimization problem (He & Niyogi, 2003):

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

where $L = D - S$ is the *graph Laplacian* (Chung, 1997) and $D_{ii} = \sum_j S_{ij}$. For the detailed derivation of LPP, please see (He & Niyogi, 2003). The basis functions of LPP are the eigenvectors of the matrix XLX^T associated with the smallest eigenvalues. Let $\mathbf{e} = (1, \cdots, 1)^T$. It is interesting to note that, when the similarity matrix $S = \frac{1}{n}\mathbf{e}\mathbf{e}^T$ whose entries are all $\frac{1}{n}$, the corresponding matrix XLX^T is just the data covariance matrix $C = X(I - \frac{1}{n}\mathbf{e}\mathbf{e}^T)X^T$. In such a case, the constructed graph is a complete graph whose weights on all the edges are equal to $\frac{1}{n}$. This observation motivates us to consider the connection between PCA and LPP, and especially the connection between the data covariance matrix and the matrix XLX^T.

3. Statistical Analysis of LPP

In this section, we provide a statistical analysis of LPP. We begin with a definition of ϵ *Covariance Matrix*.

3.1. ϵ Covariance Matrix

Let $\mathbf{x}, \mathbf{y}$ be two independent random variables in the data space. We define

$$C_\epsilon = \frac{1}{2} E\left[(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T | \|\mathbf{x} - \mathbf{y}\| < \epsilon\right] \quad (1)$$

where ϵ is a real number. Throughout this paper, we call C_ϵ ϵ *covariance matrix*. Let C denote the covariance matrix and $\mathbf{m}$ denote the mean vector. Thus,

$$C = E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T] \quad (2)$$

We have the following theorem:

Theorem 3.1 $\lim_{\epsilon \to \infty} C_\epsilon = C$.

Proof Since $\mathbf{x}$ and $\mathbf{y}$ are independent, we have

$$\begin{aligned}&\lim_{\epsilon \to \infty} C_\epsilon \\ =\ & \frac{1}{2} E\left[(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T\right] \\ =\ & \frac{1}{2} E\left[E\left[(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T | \mathbf{x}\right]\right] \\ =\ & \frac{1}{2} E\left[E\left[\mathbf{x}\mathbf{x}^T + \mathbf{y}\mathbf{y}^T - \mathbf{x}\mathbf{y}^T - \mathbf{y}\mathbf{x}^T | \mathbf{x}\right]\right] \\ =\ & \frac{1}{2} E\left[\mathbf{x}\mathbf{x}^T + E\left[\mathbf{y}\mathbf{y}^T\right] - \mathbf{x}\mathbf{m}^T - \mathbf{m}\mathbf{x}^T\right] \\ =\ & E\left[\mathbf{x}\mathbf{x}^T\right] - \mathbf{m}\mathbf{m}^T \\ =\ & E\left[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\right]\end{aligned}$$

This completes the proof. ∎

3.2. Convergence of XLX^T

Given a set of data points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ in $\mathbf{R}^d$, we define a $n \times n$ similarity matrix as follows:

$$S_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Let $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$. Let D be a $n \times n$ diagonal matrix such that $D_{ii} = \sum_j S_{ij}$. Thus, the Laplacian matrix is $L = D - S$. We have the following theorem:

Theorem 3.2 $\lim_{n \to \infty} \frac{1}{n^{2\beta}} XLX^T = C_\epsilon$, *where* $\beta = Prob(\|\boldsymbol{x} - \boldsymbol{y}\| < \epsilon)$.

Proof

$$XSX^T = \sum_{ij} S_{ij}\mathbf{x}_i\mathbf{x}_j^T = \sum_{i=1}^{n} \mathbf{x}_i (\sum_{j=1}^{n} S_{ij}\mathbf{x}_j^T)$$

Note that, $S_{ij} = 1_{\|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon}$. Let $p(\mathbf{x})$ be the density function. By the Law of Large Number, we have

$$\begin{aligned} & \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j^T S_{ij} = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j^T \mathbf{1}_{\|\mathbf{x}_j - \mathbf{x}_i\| < \epsilon} \\ = & E\left[\mathbf{y}^T \cdot \mathbf{1}_{\|\mathbf{y}-\mathbf{x}_i\|<\epsilon)}\right] = \int_{\|\mathbf{y}-\mathbf{x}_i\|<\epsilon} \mathbf{y} p(\mathbf{y}) d\mathbf{y} \\ \doteq & g_\epsilon(\mathbf{x}_i) \end{aligned}$$

g_ϵ is a function from $\mathbf{R}^d$ to $\mathbf{R}^d$. Again, by the Law of Large Number, we have

$$\begin{aligned} & \lim_{n\to\infty} \frac{1}{n^2} XSX^T \\ = & \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i g_\epsilon(\mathbf{x}_i) \\ = & \int \mathbf{x} g_\epsilon(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\ = & \int \mathbf{x} p(\mathbf{x}) \left(\int_{\|\mathbf{y}-\mathbf{x}\|<\epsilon} \mathbf{y}^T p(\mathbf{y}) d\mathbf{y} \right) d\mathbf{x} \\ = & \int_{\|\mathbf{y}-\mathbf{x}\|<\epsilon} \mathbf{x}\mathbf{y}^T p(\mathbf{x}) p(\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ = & \beta E\left[\mathbf{x}\mathbf{y}^T | \|\mathbf{y}-\mathbf{x}\| < \epsilon\right] \end{aligned}$$

where β is a normalization factor, $\beta = \int_{\|\mathbf{x}-\mathbf{y}\|<\epsilon} p(\mathbf{x}) p(\mathbf{y}) d\mathbf{x} d\mathbf{y}$. Similarly, we have

$$XDX^T = \sum_{i=1}^{n} D_{ii} \mathbf{x}_i \mathbf{x}_i^T$$

By the definition of D_{ii}, we have

$$\begin{aligned} & \lim_{n\to\infty} \frac{1}{n} D_{ii} = P(\|\mathbf{y} - \mathbf{x}_i\| < \epsilon) \\ = & \int_{\|\mathbf{y}-\mathbf{x}_i\|<\epsilon} p(\mathbf{y}) d\mathbf{y} = h_\epsilon(\mathbf{x}_i) \end{aligned}$$

By the Law of Large Number, we get

$$\begin{aligned} & \lim_{n\to\infty} \frac{1}{n^2} XDX^T \\ = & \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T h_\epsilon(\mathbf{x}_i) \\ = & \int \mathbf{x}\mathbf{x}^T h_\epsilon(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\ = & \int_{\|\mathbf{x}-\mathbf{y}\|<\epsilon} \mathbf{x}\mathbf{x}^T p(\mathbf{x}) p(\mathbf{y}) d\mathbf{x} d\mathbf{y} \\ = & \beta E\left[\mathbf{x}\mathbf{x}^T | \|\mathbf{y}-\mathbf{x}\| < \epsilon\right] \end{aligned}$$

Therefore,

$$\begin{aligned} & \lim_{n\to\infty} \frac{1}{n^2} XLX^T \\ = & \lim_{n\to\infty} \frac{1}{n^2} \left(XDX^T - XSX^T\right) \\ = & \beta E\left[\mathbf{x}\mathbf{x}^T - \mathbf{x}\mathbf{y}^T | \|\mathbf{y}-\mathbf{x}\| < \epsilon\right] \\ = & \frac{\beta}{2} E\left[(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T | \|\mathbf{y}-\mathbf{x}\| < \epsilon\right] \\ = & \beta C_\epsilon \end{aligned}$$

Finally, we get $\lim_{n\to\infty} \frac{1}{n^2\beta} XLX^T = C_\epsilon$. ∎

Thus, the objective function of LPP can be rewritten as follows:

$$\begin{aligned} & \mathbf{w}_{opt} \\ = & \arg\min_{\mathbf{w}} \mathbf{w}^T C_\epsilon \mathbf{w} \\ = & \arg\min_{\mathbf{w}} \mathbf{w}^T E\left[(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T | \|\mathbf{x}-\mathbf{y}\| < \epsilon\right] \mathbf{w} \\ = & \arg\min_{\mathbf{w}} \mathbf{w}^T E\left[\left(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{y}\right)^2 | \|\mathbf{x}-\mathbf{y}\| < \epsilon\right] \mathbf{w} \end{aligned}$$

4. Computational Analysis of LPP

4.1. Connection to Laplacian Eigenmaps

In this subsection, we discuss the computational relationship between LPP and Laplacian Eigenmaps (Belkin & Niyogi, 2001). The eigenvalue problem of LPP scales with the number of dimensions (d), while that of Laplacian Eigenmaps scales with the number of data points (n). The rank of X is no greater than $min(n, d)$. Thus, if $d > n$, 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 4.1 *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:

$$XLX^T\mathbf{w} = \lambda XDX^T\mathbf{w} \tag{4}$$

Let $\mathbf{y} = X^T\mathbf{w}$. Equation (4) can be rewritten as follows:

$$XL\mathbf{y} = \lambda XD\mathbf{y} \tag{5}$$

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

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{6}$$

which is just the eigenvalue problem of Laplacian Eigenmaps. ∎

Table 1. Performance comparisons on Reuters-21578 corpus*

	Accuracy						
k	Kmeans	PCA(best)	PCA	LPP(best)	LPP	LE	NMF-NCW
2	0.871	0.913	0.864	0.963	0.923	0.923	0.925
3	0.775	0.815	0.768	0.884	0.816	0.816	0.807
4	0.732	0.773	0.715	0.843	0.793	0.793	0.787
5	0.671	0.704	0.654	0.780	0.737	0.737	0.735
6	0.655	0.683	0.642	0.760	0.719	0.719	0.722
7	0.623	0.651	0.610	0.724	0.694	0.694	0.689
8	0.582	0.617	0.572	0.693	0.650	0.650	0.662
9	0.553	0.587	0.549	0.661	0.625	0.625	0.623
10	0.545	0.573	0.540	0.646	0.615	0.615	0.616
ave.	0.667	0.702	0.657	0.773	0.730	0.730	0.730
	Mutual Information						
k	Kmeans	PCA(best)	PCA	LPP(best)	LPP	LE	NMF-NCW
2	0.600	0.666	0.569	0.793	0.697	0.697	0.705
3	0.567	0.594	0.536	0.660	0.601	0.601	0.600
4	0.598	0.621	0.573	0.671	0.635	0.635	0.634
5	0.563	0.567	0.538	0.633	0.603	0.603	0.587
6	0.579	0.587	0.552	0.636	0.615	0.615	0.603
7	0.573	0.572	0.547	0.629	0.617	0.617	0.600
8	0.556	0.557	0.530	0.615	0.587	0.587	0.583
9	0.549	0.545	0.532	0.605	0.586	0.586	0.560
10	0.552	0.549	0.528	0.607	0.586	0.586	0.561
ave.	0.571	0.584	0.545	0.650	0.614	0.614	0.604

**PCA* denotes the clustering result obtained by PCA with k-1 dimensions and *PCA(best)* denotes the best result obtained by PCA at the optimal dimension.

In many real world applications such as information retrieval, the dimensionality of the data space is typically much larger than the number of data points. In such a case, LPP and Laplacian Eigenmaps will have the same embedding result if these data vectors are linearly independent.

4.2. Connection to Principal Component Analysis

LPP is essentially obtained from a graph model. In the original algorithm, a nearest neighbor graph is constructed to discover the *local* manifold structure (He & Niyogi, 2003). Intuitively, LPP with a complete graph should discover the *global* structure. In this subsection, we present a theoretical analysis on the relationship between LPP and PCA. Specifically, we show that LPP with a complete inner product graph is similar to PCA. Without loss of generality, we assume that the data features are uncorrelated (the covariance matrix is of full rank), otherwise we can apply Singular Value Decomposition first. Also, we assume that the data points have a zero mean. Thus, the matrix XX^T is the covariance matrix.

Suppose the weight on an edge linking $\mathbf{x}_i$ and $\mathbf{x}_j$ is set to their inner product $\mathbf{x}_i^T\mathbf{x}_j$. Thus, the weight matrix S of the complete graph can be written as X^TX. The generalized minimum eigenvalue problem of LPP can be written as follows:

$$\begin{aligned} & XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \\ \Rightarrow \quad & X(D-S)X^T\mathbf{a} = \lambda XDX^T\mathbf{a} \\ \Rightarrow \quad & XSX^T\mathbf{a} = (1-\lambda)XDX^T\mathbf{a} \\ \Rightarrow \quad & XX^TXX^T\mathbf{a} = (1-\lambda)XDX^T\mathbf{a} \end{aligned} \tag{7}$$

Since the diagonal matrix D is close to the identity matrix, $XDX^T \approx XX^T$, the *minimum* eigenvalues of equation (7) correspond to the *maximum* eigenvalues of the following equation:

$$XX^TXX^T\mathbf{a} = \lambda XX^T\mathbf{a}$$

Since XX^T is of full rank, we get:

$$XX^T\mathbf{a} = \lambda\mathbf{a}$$

which is just the eigenvalue problem of PCA. The above analysis shows that LPP with a complete inner product graph is similar to PCA. Both of them

discover the global structure. The only difference is that there is a diagonal matrix D in LPP which measures the local density around $\mathbf{x}_i$, while in PCA, every data point is equally treated.

5. Experimental Evaluation

In this section, we evaluate the applicability of LPP, PCA, and Laplacian Eigenmaps for document clustering.

5.1. Data Preparation

Two document data sets were used in our experiments, i.e. Reuters-21578 and TDT2. Reuters contains 21578 documents in 135 categories. Documents that appear in two or more categories were removed. We selected the largest 30 categories. Finally, it left us with 8067 documents in 30 categories. Each document is represented as a term-frequency vector. We simply removed the stop words. Each document vector is normalized so that it has unit norm. No further preprocessing was done.

The TDT2 document data set [1]consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newsviews, 2 radio programs and 2 televisions. It consists of 11201 on-topic documents which are classified into 96 semantic classes. Those documents appearing in two or more classes were removed, and we selected the largest 50 categories, thus leaving us with 9394 documents of 30 categories.

5.2. Experimental Design

In this work, we compared the following five document clustering methods:

- K-means (K-means)
- LPP+K-means (LPP)
- PCA+K-means (PCA)
- Non-negative Matrix Factorization based clustering (NMF-NCW, (Xu et al., 2003))
- Laplacian Eigenmaps+K-means (LE)

The weighted Non-negative Matrix Factorization based document clustering algorithm (NMF-NCW, (Xu et al., 2003)) is a recently proposed algorithm, which has shown to be very effective in document clustering. Please see (Xu et al., 2003) for details. In the LPP and Laplacian Eigenmap algorithms, one needs to build a ϵ neighborhood graph (Eqn. 3). However, in real applications, it is difficult to choose a optimal ϵ. In our experiments, we build a p nearest neighbor graph as follows:

$$S_{ij} = \begin{cases} \mathbf{x}_i^T \mathbf{x}_j, & \text{if } \mathbf{x}_i \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_j, \text{ or } \mathbf{x}_j \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The parameter p was set to 15. For each method, $k(= 2, 3, \cdots, 10)$ document classes were randomly selected from the data corpus. Clustering is then performed on these k document classes. The clustering result is evaluated by comparing the obtained label of each document 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 (Xu et al., 2003). 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:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{9}$$

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 (Lovasz & Plummer, 1986).

Given two sets of data clusters C, C', their mutual information metric $MI(C, C')$ is defined as:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \tag{10}$$

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

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \tag{11}$$

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 set of clusters are identical, and $\overline{MI} = 0$ if the two sets are independent.

[1]Topic Detection and Tracking corpus at http://www.nist.gov/speech/tests/tdt/tdt98/index.htm

Table 2. Performance comparisons on TDT2 corpus*

	Accuracy						
k	Kmeans	PCA(best)	PCA	LPP(best)	LPP	LE	NMF-NCW
2	0.989	0.992	0.977	0.998	0.998	0.998	0.985
3	0.974	0.985	0.944	0.996	0.996	0.996	0.953
4	0.959	0.970	0.894	0.996	0.996	0.996	0.964
5	0.948	0.961	0.914	0.993	0.993	0.993	0.980
6	0.945	0.954	0.879	0.993	0.992	0.992	0.932
7	0.883	0.903	0.849	0.990	0.988	0.987	0.921
8	0.874	0.890	0.829	0.989	0.987	0.988	0.908
9	0.852	0.870	0.810	0.987	0.983	0.984	0.895
10	0.835	0.850	0.786	0.982	0.979	0.978	0.898
ave.	0.918	0.931	0.876	0.992	0.990	0.990	0.937
	Mutual Information						
k	Kmeans	PCA(best)	PCA	LPP(best)	LPP	LE	NMF-NCW
2	0.962	0.965	0.925	0.981	0.981	0.981	0.939
3	0.946	0.962	0.894	0.977	0.976	0.976	0.924
4	0.932	0.942	0.856	0.979	0.979	0.979	0.951
5	0.935	0.942	0.892	0.975	0.973	0.973	0.965
6	0.936	0.939	0.878	0.975	0.974	0.974	0.923
7	0.884	0.892	0.849	0.969	0.968	0.966	0.915
8	0.889	0.895	0.841	0.970	0.967	0.967	0.911
9	0.875	0.878	0.831	0.970	0.966	0.967	0.905
10	0.865	0.869	0.813	0.962	0.959	0.958	0.897
ave.	0.914	0.920	0.864	0.973	0.971	0.97	0.926

**PCA* denotes the clustering result obtained by PCA with k-1 dimensions and *PCA(best)* denotes the best result obtained by PCA at the optimal dimension.

5.3. Clustering Results

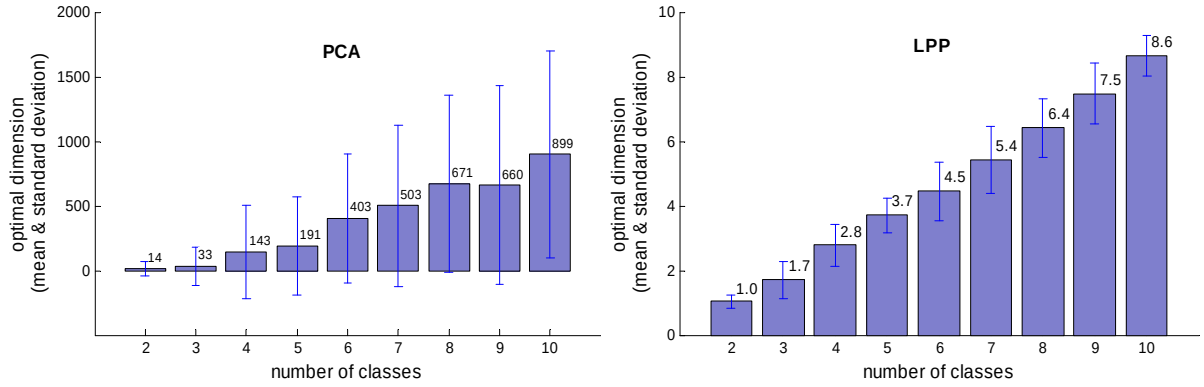
Table 1 and 2 shows the experimental results. The evaluation was conducted with different number of clusters, ranging from two to ten. For each given cluster number k, 50 tests were conducted on the randomly chosen clusters, and the average performance was computed over these 50 tests. For each single test, K-means algorithm was applied 10 times with different initializations and the best result was recorded. As can be seen, the performance of LPP is much better than that of PCA, and close to that of Laplacian Eigenmaps. Also, LPP based document clustering algorithm performed slightly better than NMF based document clustering algorithm (Xu et al., 2003).

When applying dimensionality reduction algorithms for document analysis, how to estimate the optimal dimension is a key problem. In spectral clustering (Belkin & Niyogi, 2001; Shi & Malik, 2000; Ng et al., 2001), the dimension of the subspace can be set to the number of clusters. For PCA and LPP based clustering algorithms, in generally their performance varies with the number of dimensions. In Figure 1, we show the optimal dimensions obtained by LPP and PCA, as well as the standard deviations. As can be seen, the optimal dimension obtained by LPP is very close to $k-1$, while it is difficult for PCA to estimate the optimal dimension. Also, the standard deviation of the optimal dimension for PCA is much bigger than that for LPP. This indicates that dimensionality estimation for LPP is much more stable.

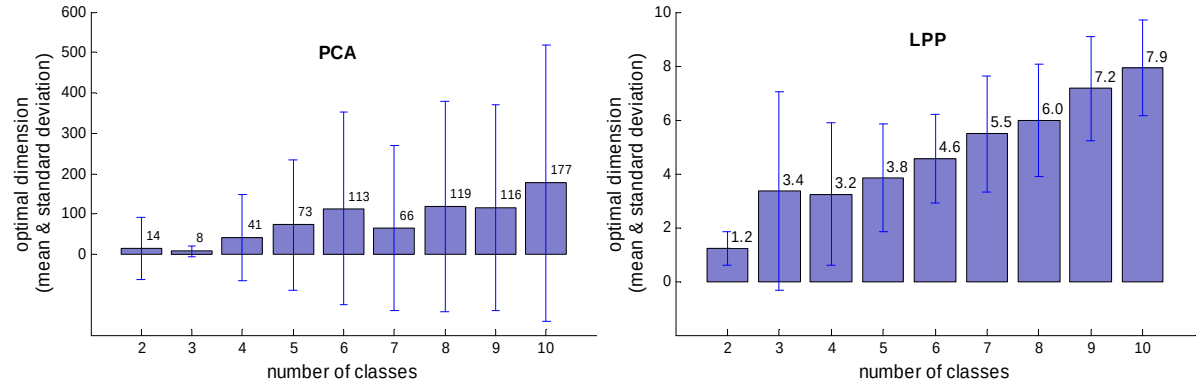
Moreover, it can be seen that the performance of LPP is very close to that of Laplacian Eigenmap. Actually in our experiments, for 312 out of 450 (50×9) tests on Reuters corpus and 430 out of 450 (50×9) tests on TDT2 corpus, the data matrix X is full rank square matrix, thus the clustering results using LPP are identical to those using Laplacian Eigenmaps according to Proposition 4.1.

5.4. Local VS. Global

In LPP based clustering, one needs to set the number of nearest neighbors, i.e. the value of p, which defines the "locality". In Figure 2, we show the relationship between the clustering performance and the value of



(a) TDT2 Corpus



(b) Reuters Corpus

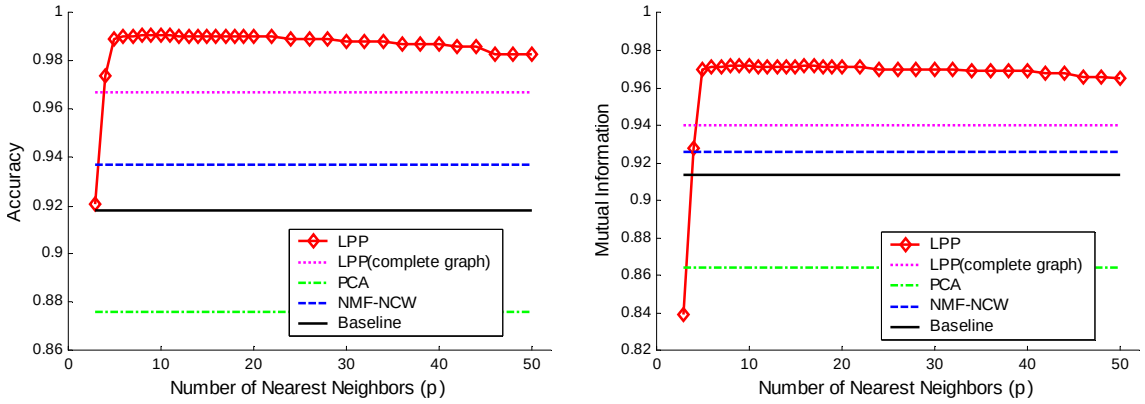
Figure 1. In generally, the performance of PCA and LPP based clustering algorithms varies with the dimensionality of the reduced subspace. The left plot shows the dimensionality of the subspace obtained by PCA in which the best clustering performance is obtained, and the right plot shows the optimal dimensionality obtained by LPP. As can be seen, the optimal dimensionality of LPP is very close to $k-1$, where k is the number of clusters. Therefore, if k is given, the optimal dimensionality of LPP can be accurately estimated. However, for PCA based clustering algorithm, there seems no relationship between the number of clusters and the optimal dimensionality. Each bar shows the average of 50 test runs, and the error bar indicates the standard deviation. The standard deviation indicates that LPP is less sensitive to dimensionality than PCA.

p. For LPP and PCA, clustering were performed in $k-1$ dimensional subspace. Here, the clustering performance is the average over 2∼10 classes. The value of p varies from 3 to 40. As can be seen, the performance of LPP clustering reaches its peak when p is 6 in TDT2 corpus and 15 in Reuters21578 corpus. After than, as p increases, the performance decreases. We also show the result of LPP with a complete graph (p is taken to be infinity). This experiment shows that the local structure is more important than the global structure as to discovering the semantic structure of the document space.

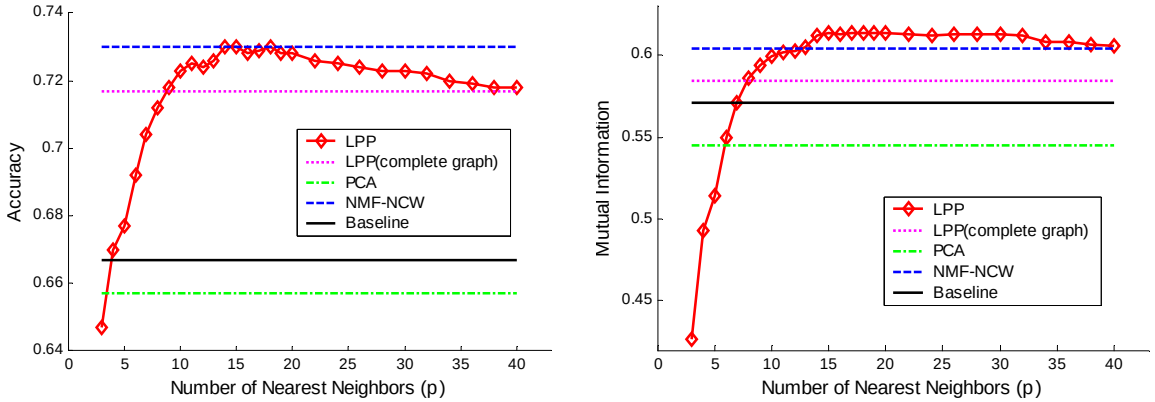
6. Conclusion

In this paper, we have presented statistical and computational analysis of the Locality Preserving Projections algorithm. We define an ϵ covariance matrix. When ϵ tends to infinity, the ϵ covariance matrix becomes the standard covariance matrix which is used in PCA. When ϵ is extremely small, the ϵ covariance matrix captures the local covariance and is used in LPP. We also show that the matrix XLX^T converges to the ϵ covariance matrix as the number of data points tends to infinity.

Computational analysis of LPP and Laplacian Eigenmaps shows that they can give the same result when the number of dimensions is larger than the number of data points and the data vectors are linearly independent. We have also shown that LPP with a complete inner product graph model is similar to Principal Component Analysis. In such a case, both of LPP and PCA discover the global structure of the data space.



(a) TDT2 Corpus



(b) Reuters Corpus

Figure 2. Graph model - Local vs. Global

References

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14*.

Bregler, C., & Omohundro, S. (1995). Nonlinear image interpolation using manifold learning. *Advances in Neural Information Processing Systems, 7*.

Chung, F. R. K. (1997). *Spectral graph theory*, vol. 92 of *Regional Conference Series in Mathematics*.

He, X., Cai, D., Liu, H., & Ma, W.-Y. (2004). Locality preserving indexing for document representation. *Proceedings of ACM SIGIR Conference on Information Retrieval*.

He, X., & Niyogi, P. (2003). Locality preserving projections. *Advances in Neural Information Processing Systems, 16*.

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H.-J. (2005). Face recognition using laplacianfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 27*, 328–340.

Lovasz, L., & Plummer, M. (1986). *Matching theory*. North Holland, Budapest: Akadémiai Kiadó.

Ng, A. Y., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14*.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*, 2323–2326.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 22*, 888–905.

Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*, 2319–2323.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. *Proceedings of ACM SIGIR Conference on Information Retrieval*.