Locally Linear Embedded Eigenspace Analysis

IFP.TR-LEA.YunFu-Jan.1,2005
Yun Fu *and* **Thomas S. Huang**
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 North Mathews Avenue, Urbana, IL 61801
{yunfu2, huang}@ifp.uiuc.edu

No Institute Given

Abstract

The existing nonlinear local methods for dimensionality reduction yield impressive results in data embedding and manifold visualization. However, they also open up the problem of how to define a unified projection from new data to the embedded subspace constructed by the training samples. Thinking globally and fitting locally, we present a new linear embedding approach, called Locally Embedded Analysis (LEA), for dimensionality reduction and feature selection. LEA can be viewed as the linear approximation of the Locally Linear Embedding (LLE). By solving a linear eigenspace problem in closed-form, LEA automatically learns the local neighborhood characteristic and discovers the compact linear subspace, which optimally preserves the intrinsic manifold structure. Given a new high-dimensional data point, LEA finds the corresponding low-dimensional point in the subspace via the linear projection. This embedding process concentrates the adjacent data points into the same dense cluster, which is effective for discriminant analysis, supervised classification and unsupervised clustering. We test the proposed LEA algorithm on several benchmark databases. Experimental results show that LEA provides better data representation and more efficient dimensionality reduction than the classical linear methods.

1 Introduction

In most computer vision and pattern recognition problems, the large number of sensory inputs, such as images and videos, are often viewed as intrinsically low-dimensional data distributed in a high-dimensional vector space. In order to understand and analyze the multivariate data, we need to reduce the dimensionality and find more compact representations. If the variance of the multivariate data is faithfully represented as a set of parameters, the data can be considered as a set of geometrically related points lying on a smooth low-dimensional manifold [1, 13, 17, 19, 20]. The fundamental issue in dimensionality reduction is how to model the geometry structure of the manifold and produce a faithful embedding for data projection.

During the last few years, a large number of approaches have been proposed for constructing and computing the embedding. We categorize these methods by their linearity. The nonlinear methods such as locally linear embedding (LLE) [13, 17], Laplacian eigenmaps [3], Isomap [19], Hessian LLE (hLLE) [5] and semidefinite programming (SDE) [20], focus on preserving the geodesic distances which reflect the real geometry of the low-dimensional manifold. LLE formulates the manifold learning problem as a neighborhood-preserving embedding, which learns the global structure by exploiting the local symmetries of linear reconstructions. Based on the spectral decomposition of graph Laplacians, Laplacian Eigenmap finds an approximation to the Laplace-Beltrami operator defined on the manifold. Isomap extends the classical MDS [4] by computing the pairwise distances in the geodesic space of the manifold. In the sense of isometry, hLLE estimates the Hessian matrix while SDA considers local angles and distances.

The linear methods, such as principal component analysis (PCA) [6, 10], multidimensional scaling (MDS) [4], locality preserving projections (LPP) [8, 9], and locality pursuit embedding (LPE) [11] are evidently effective in observing the Euclidean structure. PCA finds the embedding that maximizes the projected variance, while MDS preserves the pairwise distances between data points during the embedding. If the metric are Euclidean, MDS is the same as PCA. LPP and LPE are both local approaches. LPP computes the embedding in terms of a "heat kernel" based nearest neighbor graph, which discovers the essential manifold structure by preserving the local information. LPE performs PCA on local nearest neighbor patches to reveal the tangent space structure on manifold.

The nonlinear embedding methods have been successfully applied to some standard data sets and generated satisfying results in dimensionality reduction and manifold visualization. However, these approaches define the input-to-manifold mapping solely based on the training data space. How to project new data into the embedded space is still an open problem for real-word applications. Moreover, it is also unclear how the reverse manifold-to-input mapping is defined [21]. The linear embedding methods are demonstrated to be computationally efficient and suitable for practical applications, such as pattern classification and visual recognition. Since the embedding is approximated to a linear process and in the statistical sense, these methods ignore the geodesic structure of the true manifold. Although linear methods cannot reveal the perfect geometric structure of the nonlinear manifold, one might expect to find a better linear approximation to the nonlinear embedding [6], such as LPP, which is sufficiently effective to deal with practical problems in computer vision and machine learning.

In this paper, we present the locally linear embedded eigenspace analysis, which is abbreviated to *Locally Embedded Analysis (LEA)*, for linear dimensionality reduction and feature selection. Inspired by the neighborhood-preserving property and geometric intuition of LLE, we propose a linear eigenspace-analysis-based embedding strategy, which finds the optimal projection to minimize the local reconstruction error. Our approach builds on an objective function similar to that of LLE but seeks the *linear* input-to-manifold mapping in closed-form. As a result, LEA reveals several novel perspectives in constructing the embedded low-dimensional subspace. In the algorithm, the weight matrix of locally pairwise relation is first constructed by nearest neighbor reconstruction. Then the embedded space is built according to the same weight matrix. In the learned subspace, based on a quadratic form constraint, the eigenspace analysis is applied to compute the optimal basis (eigenvectors) for data projection.

LEA exhibits several attractive properties for dimensionality reduction and feature selection. **(a)** *LEA is based on an exact closed-form solution.* **(b)** *LEA can be viewed as a linear approximation of LLE.* Using the objective function of LLE, LEA formulates the embedding problem as that of linear eigenspace analysis. The advantage is that, given a new high-dimensional data point, LEA automatically finds the corresponding low-dimensional point on manifold via the linear projection. **(c)** *LEA is a reasonable data representation method.* Consider the neighbors of a particular point lying in a certain subspace of the entire data space. This subspace has the property of the *Linear Object Class Model*, on which any prototype data can be represented by the linear combination of the others. LEA imposes this important property into the embedded space. **(d)** *LEA has the discriminant property of both supervised classification and unsupervised clustering.* According to the pattern classification theory [6], the classifier should be sufficiently smooth, that is, if two data points are adequately close to each other, they share the same label. Considering the nearest-neighbor based classification or small sample issues, we believe that the local structure is more important than between-class distribution, because the training data is often inadequate in revealing the faithful statistical class information. However, LEA directly holds the local geometry to concentrate the nearest data points into compact clusters for pattern discriminant. We have demonstrated that LEA outperforms both PCA and LDA [10] in some visual recognition cases. **(e)** *LEA has latent relation with LPP in the sense of linear subspace analysis.* Although these two methods both try to preserve the local structure for linear embedding, they have absolutely different objective function, mathematical derivation, and geometric sense. The main advantage of LEA is that it automatically constructs the weight matrix in a closed-form solution by learning the intrinsic neighborhood structure, while LPP constructs the Laplacian graph using a "heat kernel" function which needs to tune a kernel parameter in practice.

In the following sections, we first review the LLE algorithm in Sec.2. We then present the locally linear embedded eigenspace analysis (LEA) in Sec.3 and discuss the properties of LEA in Sec.4. The experimental results are shown in Sec.5. We finally conclude the paper in Sec.6.

2 Locally Linear Embedding (LLE)

The original Locally Linear Embedding (LLE) algorithm [13] includes two stages [15]: linear nearest neighbor reconstruction and locally structure preserved embedding.

2.1 Linear Nearest Neighbor Reconstruction

Consider a high dimensional data set with n elements is denoted by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathcal{R}^D$ and $i = 1, 2, \cdots, n$. In LLE, original data lie on a nonlinear manifold that can be approximated with a local linear model. The geometric relationship among the high dimensional data is described by the linear nearest-neighbor reconstruction. It means that each data point is represented by the linear combination of its k nearest neighbors.

Denote the set of $\mathbf{x}_i$'s k nearest neighbors as $\mathcal{X}_N^{(i)} = \{\mathbf{x}_{N(1)}^{(i)}, \mathbf{x}_{N(2)}^{(i)}, \cdots, \mathbf{x}_{N(k)}^{(i)}\}$, where $\mathbf{x}_{N(j)}^{(i)} \in \mathcal{R}^D$ and $j = 1, 2, \cdots, k$. The reconstruction error is calculated by the following cost function

$$\varepsilon_D = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_j^{(i)} \mathbf{x}_{N(j)}^{(i)} \right\|^2 \tag{1}$$

where $\mathcal{W}_K^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \cdots, w_k^{(i)}\}$ is the reconstruction weight vector, subject to the constrain $\sum_{j=1}^{k} w_j^{(i)} = 1$ and $w_j^{(i)} = 0$ when $\mathbf{x}_{N(j)}^{(i)} \notin \mathcal{X}_N^{(i)}$. The weights are calculated by solving $\mathcal{W}_K^{(i)} = \arg\min(\varepsilon_D)$.

2.2 Locally Structure Preserved Embedding

In the embedding stage, the representations of those intrinsic *characteristics* (weights) are preserved to maintain the nonlinear manifold geometry in the low-dimensional space. A low-dimensional data set is denoted by $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$, where $\mathbf{y}_i$ corresponds to $\mathbf{x}_i$ and $\mathbf{y}_i \in \mathcal{R}^d$ $(d < D)$. With the same weights $\mathcal{W}_K^{(i)}$ solved in Sec.2.1, we define the embedding cost function as

$$\varepsilon_d = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{k} w_j^{(i)} \mathbf{y}_{N(j)}^{(i)} \right\|^2 \tag{2}$$

The reconstruction error is minimized by solving an eigenvalue problem, subject to the constrains $\sum_{i=1}^{n} \mathbf{y}_i = 0$ and $\mathbf{Y}\mathbf{Y}^T = n\mathbf{I}$, where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$ and $\mathbf{I}$ is the $d \times d$ identity matrix.

3 Locally Linear Embedded Eigenspace Analysis

3.1 Why Not Nonlinear Projection?

Using a nonlinear projection, the nonlinear methods effectively reflect the intrinsic geometric structure of the high-dimensional data in a visible low-dimensional subspace. However, in real-world problems, especially the clustering and visual recognition, these methods have two essential limitations: (1) The nonlinear projection is defined only on the training data space. For each novel input datum, the entire embedding procedure, including the global nearest-neighbor search and the weight matrix calculation, has to be repeated. (2) The nonlinear methods only provide the forward input-to-manifold mapping, but not a reversible parametric mapping from the low-dimensional space to the high-dimensional space. To overcome these limitations, we propose the *Locally Linear Embedded Eigenspace Analysis*, in which the embedding is approximated as a nonparametric linear space projection that optimally preserves the local neighborhood structure.

3.2 Linear Space Projection

Linear Projection [6, 8, 10, 11] is particularly attractive because it is in closed-form solution and computationally inexpensive. For the problems of excessive dimensionality, linear projection maps the high-dimensional data into a lower dimensional subspace using eigenspace analysis. Suppose there is a one-to-one mapping between $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$. The space projection is defined by a transformation $P : \mathcal{R}^D \rightarrow \mathcal{R}^d$. The D-by-d projection matrix is denoted by $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_d]$ which satisfies $\mathbf{y}_i = \mathbf{P}^T\mathbf{x}_i$. The D-to-d projection can be written as a single matrix equation

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} \tag{3}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are respectively viewed as columns of D-by-n matrix $\mathbf{X}$ and d-by-n matrix $\mathbf{Y}$. When $d = 1$, the original D-dimensional data set is projected onto a line. We obtain the D-to-1 projection

$$\mathbf{y} = \mathbf{p}^T\mathbf{X} \tag{4}$$

The vector $\mathbf{p} = (p_1, p_2, \cdots, p_n)^T$ is the line to project, while $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$ is the vector of the coordinates, where $y_i = \mathbf{p}^T\mathbf{x}_i$.

3.3 Computing Weight Matrix in Closed-Form

We solve the LEA projection by first calculating the weight matrix in closed-form [17, 19]. Since the weights $\mathcal{W}_K^{(i)}$ are constrained to sum up to 1, Eq. (1) can be rewritten as

$$\begin{aligned} \varepsilon_D &= \sum_{i=1}^{n} \Big\| \sum_{j=1}^{k} w_j^{(i)} \big(\mathbf{x}_i - \mathbf{x}_{N(j)}^{(i)}\big) \Big\|^2 \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{k} w_j^{(i)} w_l^{(i)} G_{jl}^{(i)} \end{aligned} \tag{5}$$

Define the k-by-k local Gram matrix $\mathbf{G}_i$ for $\mathbf{x}_i$ as $\mathbf{G}_i[j, l] = (\mathbf{x}_i - \mathbf{x}_{N(j)}^{(i)})^T(\mathbf{x}_i - \mathbf{x}_{N(l)}^{(i)})$, that is $\mathbf{G}_i = (\mathbf{x}_i\mathbf{1}^T - \mathbf{X}^{(i)})^T(\mathbf{x}_i\mathbf{1}^T - \mathbf{X}^{(i)})$, where the k-by-1 column vector $\mathbf{1}$ consists of ones and the columns of the D-by-k matrix $\mathbf{X}^{(i)}$ contains $\mathbf{x}_i$'s k

nearest neighbors. Setting k-by-1 column vector $\mathbf{w}_i = (w_1^{(i)}, w_2^{(i)}, \cdots, w_k^{(i)})^T$, we rewrite Eq. (5) as

$$\varepsilon_D = \sum_{i=1}^{n} \mathbf{w}_i^T \mathbf{G}_i \mathbf{w}_i \tag{6}$$

Given the constraint on $\mathbf{w}_i$ (Sec. 2.1), we solve the least squares problem and obtain the closed-form solution

$$\mathbf{w}_i = \frac{\mathbf{G}_i^{-1}\mathbf{1}}{\mathbf{1}^T \mathbf{G}_i^{-1}\mathbf{1}} \tag{7}$$

Define the n-by-n sparse *Weight Matrix* $\mathbf{W}$, where $\mathbf{W}[i, N(j)] = \mathbf{w}_i(j) = w_j^{(i)}$, and the other elements of $\mathbf{W}$ are 0. In general, $\mathbf{W}$ is a sparse matrix consisting of neighborhood characteristic of the original space.

3.4 Learning a Locally Linear Embedded Subspace

In this section, we derive the closed-form solution for learning the locally linear embedded subspace in two cases. These two formulations, subject to different constraints, corresponds to the supervised D-to-d embedding and unsupervised D-to-1 embedding, respectively.

Solution for Supervised D-to-d Embedding In the supervised embedding, suppose that we know the desired low-dimensional degree d to project. The problem of embedding is that of calculating the mapping from the D-dimensional space to the d-dimensional subspace. To obtain the closed-form solution to the linear projection in Eq.(3), we rearrange Eq.(2) as follows according to $\sum_{j=1}^{k} w_j^{(i)} = 1$

$$\begin{aligned}\varepsilon_d &= \sum_{i=1}^{n} \Big\| \sum_{j=1}^{k} w_j^{(i)} \big(\mathbf{y}_i - \mathbf{y}_{N(j)}^{(i)}\big) \Big\|^2 \\ &= \sum_{i=1}^{n} \Big\| \sum_{j=1}^{k} w_j^{(i)} \big(\mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_{N(j)}^{(i)}\big) \Big\|^2 \\ &= \mathtt{Tr}\big[\mathbf{P}^T\mathbf{X}\big(\mathbf{I}-\mathbf{W}\big)^T\big(\mathbf{I}-\mathbf{W}\big)\mathbf{X}^T\mathbf{P}\big]\end{aligned} \tag{8}$$

where $\mathtt{Tr}$ is the trace operator on matrices.

Subject to the constraint $\mathbf{P}^T(\mathbf{X}\mathbf{X}^T)\mathbf{P} = n\mathbf{I}$, we obtain the following eigenvalue problem by Lagrange optimization.

$$\mathbf{X}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{X}^T\mathbf{P} = \mathit{\Lambda}\mathbf{X}\mathbf{X}^T\mathbf{P} \tag{9}$$

where $\mathit{\Lambda}$ is the diagonal Lagrange multiplier matrix. The columns of $\mathbf{P}$ are the smallest d eigenvectors of matrix $(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{X}^T$ after discarding the bottom eigenvector which is the mean of $\mathbf{P}^T\mathbf{X}$.

Solution for Unsupervised D-to-1 Embedding In the unsupervised embedding, suppose that we do not know the desired low-dimensional degree d to project. We formulate the embedding problem as a D-to-1 projection, that is, the mapping from D-dimensional space to each axis of the low-dimensional space is optimal. In geometrical sense, the high-dimensional data are linearly projected onto an optimal line preserving the underlying neighborhood structure. Define the n-by-n diagonal matrix $\mathbf{D}$, where $\mathbf{D}[i,i] = \sum_{j=1}^{n} \mathbf{W}[i,j]$. According to Eq.(4), we substitute each y_i with $\mathbf{p}^T\mathbf{x}_i$. Then Eq.(2) is written as

$$\varepsilon_1 = \sum_{i=1}^{n} \| \sum_{j=1}^{k} w_j^{(i)} \big(y_i - y_{N(j)}^{(i)}\big) \|^2$$

$$= \sum_{i=1}^{n} \| \sum_{j=1}^{k} w_j^{(i)} \big(\mathbf{p}^T\mathbf{x}_i - \mathbf{p}^T\mathbf{x}_{N(j)}^{(i)}\big) \|^2$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{k} w_j^{(i)} w_l^{(i)} \big(\mathbf{p}^T\mathbf{x}_i - \mathbf{p}^T\mathbf{x}_{N(j)}^{(i)}\big)\big(\mathbf{p}^T\mathbf{x}_i - \mathbf{p}^T\mathbf{x}_{N(l)}^{(i)}\big)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{k} w_j^{(i)} w_l^{(i)} \big(\mathbf{p}^T\mathbf{x}_i\mathbf{p}^T\mathbf{x}_i + \mathbf{p}^T\mathbf{x}_{N(j)}^{(i)}\mathbf{p}^T\mathbf{x}_{N(l)}^{(i)}$$

$$-\mathbf{p}^T\mathbf{x}_i\mathbf{p}^T\mathbf{x}_{N(j)}^{(i)} - \mathbf{p}^T\mathbf{x}_i\mathbf{p}^T\mathbf{x}_{N(l)}^{(i)}\big)$$

According to the definition of the weight matrix in Sec.3.3, we obtain the following cost function

$$\varepsilon_1 = \mathbf{p}^T\mathbf{X}\big(\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W} - \mathbf{D}^T\mathbf{W} - \mathbf{W}^T\mathbf{D}\big)\mathbf{X}^T\mathbf{p} \tag{10}$$

Recall that each diagonal value $\mathbf{D}[i,i]$ of matrix $\mathbf{D}$ corresponds to the weights summation for the k nearest neighbors of a particular data point. Moreover, since the nonsymmetric rows and columns of sparse matrix $\mathbf{W}$ have very few overlappings, it follows that $\mathbf{W}^T\mathbf{W}$ is almost a diagonal matrix. Therefore, the matrix $\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W}$ indicates the local distribution around y_i because a large value in the diagonal means a short distance between neighbors. To solve the least squares problem, we minimize the objective function subject to the constraint $\mathbf{p}^T\mathbf{X}\big(\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W}\big)\mathbf{X}^T\mathbf{p} = 1$. It is straightforward to obtain the projection vector $\mathbf{p}$ by solving the eigenvalue problem

$$\mathbf{X}\big(\mathbf{D} - \mathbf{W}\big)^T\big(\mathbf{D} - \mathbf{W}\big)\mathbf{X}^T\mathbf{p} = \lambda\mathbf{X}\big(\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W}\big)\mathbf{X}^T\mathbf{p} \tag{11}$$

Obviously, matrices $\mathbf{X}\big(\mathbf{D}-\mathbf{W}\big)^T\big(\mathbf{D}-\mathbf{W}\big)\mathbf{X}^T$ and $\mathbf{X}\big(\mathbf{D}^T\mathbf{D}+\mathbf{W}^T\mathbf{W}\big)\mathbf{X}^T$ are both positive semidefinite and symmetric. Subject to the constraint $\sum_{j=1}^{k} w_j^{(i)} = 1$ and $\mathbf{D} = \mathbf{I}$, Eq.(11) can be rewritten as $\mathbf{X}\big(\mathbf{I} - \mathbf{W}\big)^T\big(\mathbf{I} - \mathbf{W}\big)\mathbf{X}^T\mathbf{p} = \lambda\mathbf{X}\big(\mathbf{I} + \mathbf{W}^T\mathbf{W}\big)\mathbf{X}^T\mathbf{p}$, which is similar to the D-to-d solution when $\mathbf{W} \approx \mathbf{I}$ and $d = 1$.

3.5 Supervised Locally Embedded Analysis

In supervised learning [14], each sample pattern in the training set is associated with a category label or cost. The task of a classifier is to distinguish the patterns with various labels and to recognize new patterns with unknown labels. LEA is a local method which is suitable for the supervised locally embedded analysis in some cases. The strict constraint imposed on the supervised LEA is that the k nearest neighbors $\{\mathbf{x}_{N(1)}^{(i)}, \mathbf{x}_{N(2)}^{(i)}, \cdots, \mathbf{x}_{N(k)}^{(i)}\}$ are chosen in the same class of $\mathbf{x}_i$. Therefore, if we have n_c samples for each class, it follows that $1 \leq k \leq (n_c - 1)$ and $n_c \geq 2$.

4 Properties of LEA

1. **Linear Input-to-Manifold Projection**. Since LEA holds the objective functions of LLE, it maintains the locally linearity in the D-to-d embedding. One of the basic properties of LEA is the unified input-to-manifold projection $\mathbf{Y} = \mathbf{P}^T\mathbf{X}$, which is undefined in LLE or other nonlinear methods. Given a novel D-dimensional data point, LEA can directly find the corresponding d-dimensional point in the subspace via the linear projection. Furthermore, LEA provides the possible reverse transformation by pseudoinverse operation, that is $\mathbf{X} = (\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{Y}$ under the constraint that $\mathbf{P}\mathbf{P}^T$ is nonsingular.

2. **Locally Linear Embedded Discriminant Analysis**. LEA has the discriminant property in both supervised pattern classification and unsupervised clustering. Based on the objective function, we can see that LEA finds the underlying embedding which holds the neighborhood reconstruction characteristic. Each embedded low-dimensional data point is located in a particular compact cluster containing all of its k nearest neighbors coming from the original space. Theoretically, if we assume $\mathbf{W}$ is symmetric and $\mathbf{W}^T\mathbf{W} \approx \mathbf{I}$ in supervised learning case, Eq.(9) can be rewritten as

$$\begin{aligned} 2\mathbf{X}\big(\mathbf{I} - \mathbf{W}\big)\mathbf{X}^T\mathbf{P} &\doteq \Lambda\mathbf{X}\mathbf{I}\mathbf{X}^T\mathbf{P} \\ \big(2\Lambda^{-1} - \mathbf{I}\big)\mathbf{X}\big(\mathbf{I} - \mathbf{W}\big)\mathbf{X}^T\mathbf{P} &\doteq \mathbf{X}\big(\mathbf{I} - \big(\mathbf{I} - \mathbf{W}\big)\big)\mathbf{X}^T\mathbf{P} \\ \widetilde{\Lambda}\mathbf{S}_W\mathbf{P} &\doteq \mathbf{S}_B\mathbf{P} \end{aligned} \tag{12}$$

where $\widetilde{\Lambda} = \big(2\Lambda^{-1} - \mathbf{I}\big)$. The $\mathbf{S}_W = \mathbf{X}\big(\mathbf{I} - \mathbf{W}\big)\mathbf{X}^T$ and $\mathbf{S}_B = \mathbf{X}\mathbf{W}\mathbf{X}^T$ are the within-class scatter matrix and the between-class scatter matrix, respectively, if the total mean of the training data is 0 [11]. Note that Eq.(12) is exactly the generalized eigenvalue problem of *Linear Discriminant Analysis* [6] which seeks the transformation matrix $\mathbf{P}$ by maximizing the ratio of the between-class scatter to the within-class scatter. Hence, the objective function of LEA potentially contains the discriminant property.

3. **Manifold Visualization and Clustering**. We have seen that LEA seeks the optimal linear projection to preserve the neighborhood characteristic that is suitable for discriminant analysis in supervised learning. On the other hand,

LEA can also visualize an approximated low-dimensional manifold to solve unsupervised problems, such as clustering. Since LEA focuses on the local structure, the display of the manifold is often very rough. In order to smooth the approximated manifold, we rearrange Eq.(7) by inserting a regularization parameter r [15], where $\mathbf{G}_i^{-1} = (\mathbf{G}_i + (r \cdot \sum_{j=1}^{k} \lambda_G^{(j)}) \cdot \mathbf{I})^{-1}$ and $\lambda_G^{(j)}$ is the eigenvalue of matrix $\mathbf{G}$.

4. **Relating LEA with Laplacian Eigenmaps**. The mathematical derivation of solving the D-to-1 linear embedding is related to that of the Laplacian eigenmaps (or LPP). The differences between the two embedding processes lie in their different objective functions and computation of the weight matrix. We can relate LEA with Laplacian eigenmaps by

$$\varepsilon_1 = \sum_{i=1}^{n} \sum_{j=1}^{k} (w_j^{(i)})^2 (y_i - y_{N(j)}^{(i)})^2 +$$

$$\sum_{i=1}^{n} \sum_{u=1}^{k} \sum_{v \neq u}^{k} w_u^{(i)} w_v^{(i)} (y_i - y_{N(u)}^{(i)})(y_i - y_{N(v)}^{(i)}) \quad (13)$$

The first term of Eq. (13) is similar to the objective function of Laplacian eigenmaps except for the square of the weights. Note that Laplacian eigenmaps chooses weights as a "heat kernel" rather than the intrinsic reconstruction weights in LEA. The second term of Eq.(13) reflects the covariance information between neighbors which represents the local structure better. More generally, if we assume $\mathbf{W}$ is symmetric and $\mathbf{W}^T\mathbf{W} \approx \mathbf{D}$ in the case of unsupervised embedding, Eq. (11) can be rewritten as

$$\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{p} \doteq \lambda \mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{p} \quad (14)$$

where $\mathbf{D} = \mathbf{I}$. Eq. (14) is the generalized eigenvalue problem of LPP when the nearest-neighbor graph $\mathbf{S}$ is subject to the constraint $\sum_{j=1}^{n} \mathbf{S}[i,j] = \sum_{i=1}^{n} \mathbf{S}[i,j] = 1$. Moreover, LEA automatically learns and preserves the locally reconstruction weights in the embedding. Unlike LPP, LEA does not need to tune parameter for setting the weight matrix.

5 Experimental Results

5.1 Benchmark Data Sets

We demonstrate the properties of LEA using six benchmark data sets. **(1) Rotating Teapot** [20]. The 400 RGB color images, in a resolution of 76×101, were taken by viewing the 360 degrees of a rotating teapot. **(2) Frey Faces** [13]. Brendan Frey's 1965 grayscale face images taken from sequential frames of a video. The images, in a resolution of 28×20, show variations in face expression and view rotation. **(3) ORL Face Database** [16]. This database contains 40 subjects with 10 grayscale face images for each. The images, in a resolution of

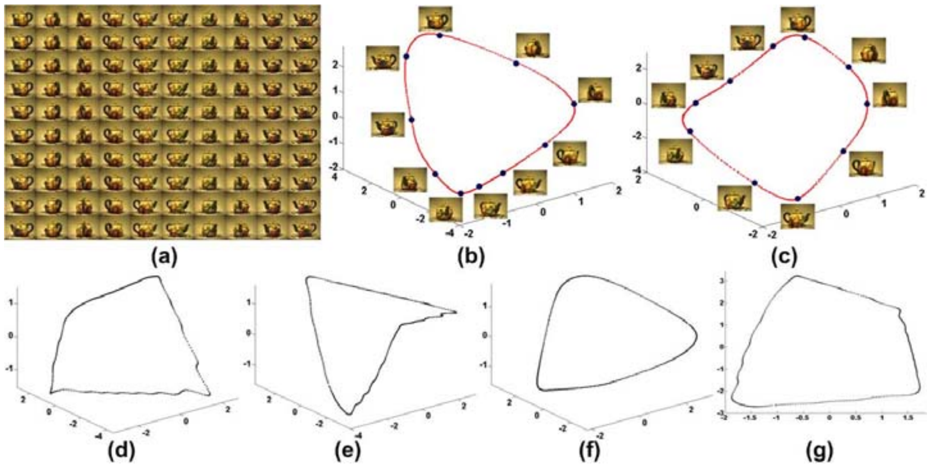

Fig. 1. Linear embedding and subspace projection of 400 rotating teapot images. The number of nearest neighbors is k=6. (a) Original images. (b) Manifold curve of 390 embedded data points with 10 novel projections. The regularization parameter of LEA is r=0.01. (c) is the same as (b) except for r=0.1. (d) Manifold curve of 400 data points using LLE with r=0.001. (e)(g) The 3D and 2D manifold curves of 400 data points using LEA with r=0.001. (f) is the same as (e) except for r=0.01.

92×112, show all frontal and slight tilt of the head. The images are cropped and resized to 32×39. **(4) YALE Face Database** [2]. This database contains 165 grayscale images of 15 individuals. The 11 images for each subject, in a resolution of 320×243, were taken in the conditions of center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. The images are cropped and resized to 32×32. **(5) UMIST Face Database** [7]. This database contains 564 grayscale face images for 20 subjects. Each subject has 19 to 36 images, in a resolution of 220×220, in various angles from left profile to frontal view. The images are cropped and resized to 28×34. **(6) IFP-Y (internal) Face Database**. This is an internal face database which contains 3520 grayscale images taken from video sequences for 22 subjects, with 160 images for each. Each cropped image has a resolution of 40×40, with large variations in facial expression, illumination, pose and occlusion.

5.2 Results

Linear Embedding and Space Projection. In this experiment, we project the 400 teapot rotating images of data set (1) into a three-dimensional space using LEA. Figure 1 shows the results of the embedding from image space to the 3D manifold space with k=6 nearest neighbors. In (b)(c), we select 390 for training. The other 10 were used for testing. The figures show that the 10 new data are projected precisely onto the manifold curve constructed by the training data. In (b)(e) (f)(g), we use different values of the regularization parameter ($r = 0.1, 0.01$, and 0.001) to smooth the manifold curves. According to the results of LLE in (d), LEA maintains the main structures of the nonlinear manifold, such as the turning points and the point sequence.

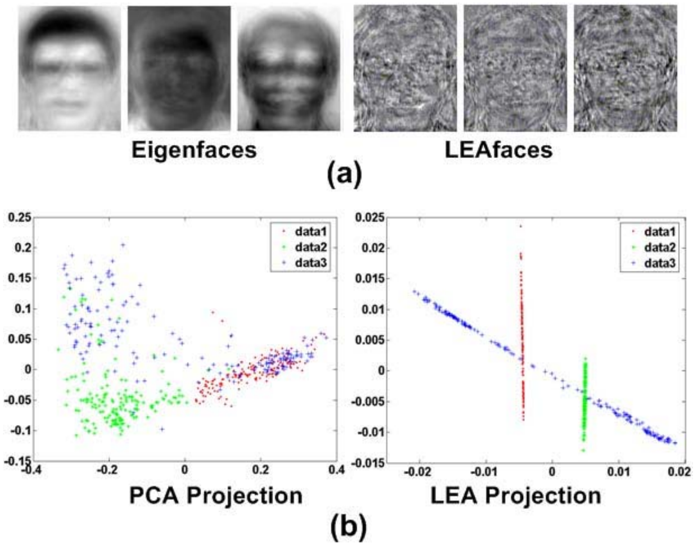

Fig. 2. Feature selection and dimensionality reduction. (a) Eigenfaces vs. LEAfaces on ORL face database. (b) PCA vs. LEA. Two-dimensional projection for 3 IFP-Y (internal) face subjects with 160 images for each.

Feature Selection and Dimensionality Reduction. In this experiment, we use data set (3) and (6) to visualize the data representation of supervised LEA and compare it with PCA. Figure 2-(a) shows the results of Eigenfaces vs. LEAfaces. We first reduce the dimensionality of the 400-ORL-face image space and represent the 3 top eigenvectors as Eigenfaces. Then in the PCA subspace we apply the supervised LEA, with k=9 nearest neighbors, and select the 3 lowest eigenvectors as LEAfaces, which span the entire LEA subspace. The LEAfaces, which contain more sparse local textures like Laplacianfaces [9], are quite different from Eigenfaces. Figure 2-(b) shows the data projection results of PCA and supervised LEA. We first arbitrarily choose 3 subjects with 160 images for each in the IFP-Y(internal) face database, and then project each image into both two-dimensional PCA subspace and LEA subspace (k=159). We find that LEA concentrates the data points of the same class densely along a line, which is more suitable for discriminant analysis, comparing to the sparse global distribution in PCA subspace.

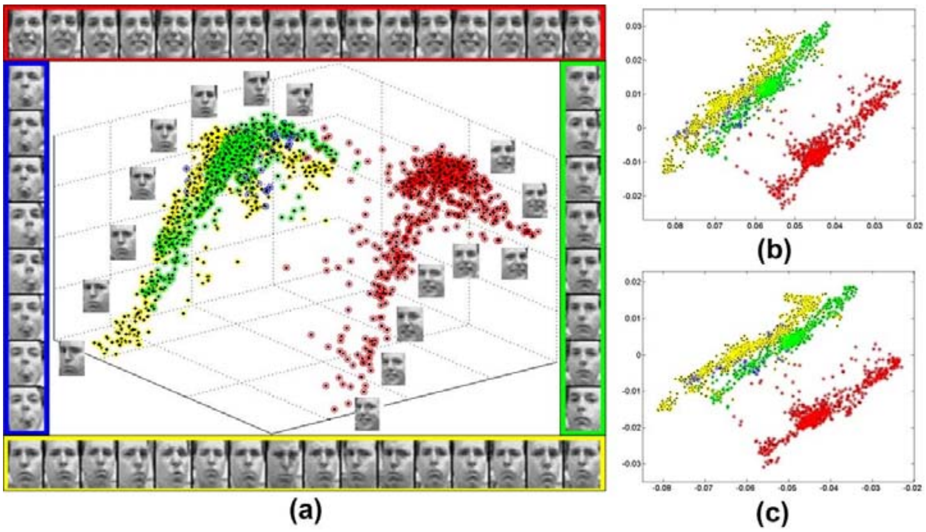

Fig. 3. The manifold visualization of 1965 Frey's face images by LEA using k=6 nearest neighbors. (a) Three-dimensional linear embedding and unsupervised clustering. The original images are divided into 4 emotion categories: happy (red), sad (yellow), licking (blue), and neutral (green). (b) Two-dimensional linear embedding and unsupervised clustering. (c) is the same as (b) except for r=0.1.

Manifold Visualization and Clustering. We evaluate the unsupervised clustering property of LEA using the entire data set (2). The 1965 face images are projected into three-dimensional LEA subspace using k=6 nearest neighbors. Figure 3-(a) shows the visualized manifold. We then label the original data with 4 emotional expressions: happy (red), sad (yellow), licking (blue), and neutral (green). It is obvious that the data belonging to the same class congregate in a certain region of the space. In general, The cluster corresponding to positive emotion (happy) is distinctly separated from the other clusters. The cluster corresponding to neutral faces is located in the middle of positive and negative emotion clusters. In each cluster, the data points corresponding to head rotation are distributed orderly along the manifold surface. Moreover, we see that the 2D embedding in (b) reveals the same cluster distribution. After smoothing the manifold with regularization parameter r=0.1, we obtain the results in Figure 3-(c). The clusters are even more compact and distinct from each other.

Visual Recognition. We are interested in the application of LEA in face recognition [18, 2, 12]. The four data sets (3)(4)(5)(6) are adopted for the experiments. Figure 4-(a) shows some cropped image examples of the ORL, YALE, UMIST and IFP-Y(internal) database. We choose Euclidean distance and the nearest neighbor classifier in all the recognition experiments. First, we assume that the gallery set of each experiment is the same as the training set. On ORL, YALE and UMIST, respectively, we randomly select 3, 6 and 25% images of each

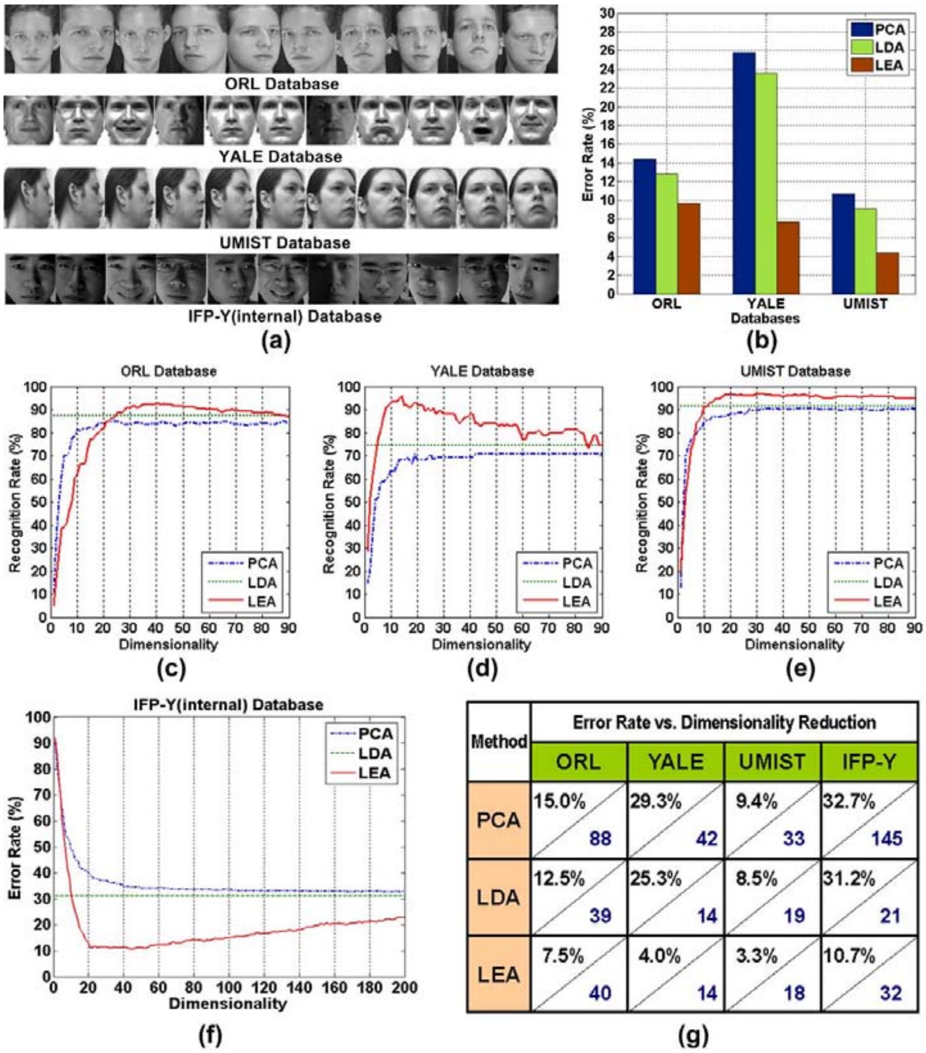

Fig. 4. Face recognition on ORL, YALE, UMIST and IFP-Y (internal) databases. (a) The cropped example images. (b) Recognition accuracy of PCA, LDA and LEA. The average error rates of 100 times of recognition test on the three databases with random data set partitions. (c)(d)(e) Recognition rate vs. dimensionality reduction on the three databases. (f) Error rate vs. dimensionality reduction. (g) Evaluate the generalization of LEA for face recognition on IFP-Y (internal) database.

subject for training and the rest of 7, 5 and 75% for testing. The training data of the three database are 120, 90 and 145, respectively, while the testing data are 280, 75 and 419. Then test the recognition accuracy of PCA, LDA and LEA on each case of dimensionality reduction. To generalize the performance, we repeat the data set partition for 100 times on each database. Figure 4-(b) shows the average of the 100 lowest error rates for each method on each database. The average error rates of (PCA, LDA, LEA) on ORL, YALE and UMIST are respectively (14.4%, 12.8%, 9.7%), (25.8%, 23.6%, 7.7%) and (10.7%, 9.1%, 4.3%). The results show that LEA outperforms both PCA and LDA. Comparing with LPP's recognition error rate (11.3%) [9] on YALE database, LEA has better result(7.7%) for the same data set partition.

Another interesting result is the efficient dimensionality reduction of LEA for visual recognition. Figure 4-(c)(d)(e) show LEA's recognition rate comparing with PCA and LEA. Figure 4-(f) shows the error rate vs. dimensionality reduction. The best reduced dimensionality of (PCA, LDA, LEA) on ORL, YALE and UMIST are respectively (88, 39, 40), (42, 14, 14) and (33, 19, 18). We see that the reduced dimensionality of LEA is close to that of LDA which is much lower than that of PCA.

We choose IFP-Y (internal) Face Database to evaluate the generalization of LEA for visual recognition. We first make sure that the gallery set, training set and testing set are all different. The 20, 10 and 130 images of each subject are randomly selected for training, gallery and testing, respectively. For all 20 subjects, we have 440, 220 and 2860 images for each data sets. Figure 4-(g) shows the recognition rate of (PCA, LDA, LEA) on IFP-Y (internal) Database. The lowest error rates and the reduced dimensionality are (32.7%, 31.2%, 10.7%) and (145, 21, 32), see Figure 4-(f). For the same reduced dimensionality of 21, the error rates are (39.4%, 31.2%, 11.2%). These results indicate that LEA still outperforms both PCA and LDA in generalized recognition problems with efficient dimensionality reduction.

6 Conclusions

We have presented the *Locally Embedded Analysis* for dimensionality reduction and feature selection. Using the objective function of LLE, LEA solves the input-to-manifold embedding by a linear projection. LEA is a method aiming at seeking the optimal linear projection for subspace analysis. By solving a linear eigenspace problem in closed-form, LEA automatically learns the local neighborhood relation and discovers the compact subspace, which optimally preserves the intrinsic structure of the manifold in the sense of linearity. The properties of LEA are suitable for practical applications, such as the pattern representation, dimensionality reduction, unsupervised clustering and supervised visual recognition, which have been demonstrated by the experiments on the six benchmark databases.

References

1. M. Brand, "Charting a Manifold," *Proc. of NIPS'02*, 2002.
2. P.N. Bellhumer, J. Hespanha and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. PAMI*, vol. 17, no. 7, pp. 711-720, 1997.

3. M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373-1396, 2003.
4. T.F. Cox and M.A.A. Cox, *Multidimensional Scaling* (2nd ed.), Chapman and Hall, 2001.
5. D.L. Donoho and C.E. Grimes, "Hessian Eigenmaps: Locally Linear Embedding Techniques for High-dimensional Data," *Proc. of the National Academy of Arts and Sciences*, vol. 100, pp. 5591-5596, 2003.
6. R.O. Duda, E.H. Peter and G.S. David, *Pattern Classification* (2nd ed.), Wiley Interscience, 2000.
7. D.B. Graham and N.M. Allinson, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition," *Face Recognition: From Theory to Applications*, NATO ASI Series F, Computer & Systems Sciences, vol.163, pp. 446-456, 1998.
8. X.F. He and P. Niyogi, "Locality Preserving Projections," *Proc. of NIPS'03*, 2003.
9. X.F. He, S.C. Yan, Y.X. Hu, P. Niyogi and H.J. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. on PAMI*, vol. 27, no. 3, pp. 328-340, 2005.
10. A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. on PAMI*, vol. 23, no. 2, pp. 228-233, 2001.
11. W.L. Min, K. Lu and X.F. He, "Locality Pursuit Embedding," *Pattern Recoginition*, vol. 37, no. 4, pp. 781-788, 2004.
12. B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Trans. on PAMI*, vol. 19, no. 7, pp. 696-710, 1997.
13. S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, pp. 2323-2326, 2000.
14. D. de Ridder, M. Loog and M.J.T. Reinders, "Local Fisher Embedding," *IEEE Conf. ICPR'04*, vol. 2, pp. 295-298, 2004.
15. D. de Ridder and R.P.W. Duin, *Locally Linear Embedding for Classification*, TR Series, No. PH-2002-01, PR Group, IS&T Dept., Delft Uni. of Tech., Netherlands, 2002.
16. F. Samaria and A.C. Harter, "Parameterisation of a Stochastic Model for Human Face Identification," *IEEE Workshop on Applications of Computer Vision*, pp. 138-142, 1994.
17. L.K. Saul and S.T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds," *J. of Machine Learning Research*, vol. 4, pp. 119-155, 2003.
18. M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces," *IEEE Conf. on CVPR*, pp. 586-591, 1991.
19. J.B. Tenenbaum, V.de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, pp. 2319-2323, 2000.
20. K.Q. Weinberger, L.K. Saul, "Unsupervised Learning of Image Manifolds by Semidefinite Programming," *IEEE Conf. on CVPR'04*, vol. 2, pp. 988-995, 2004.
21. C.S. Zhang, J. Wang, N.Y. Zhao and D. Zhang, "Reconstruction and Analysis of Multi-pose Face Images Based on Nonlinear Dimensionality Reduction," *Pattern Recoginition*, vol. 37, no. 2, pp. 325-336, 2004.