Laplacian PCA and Its Applications

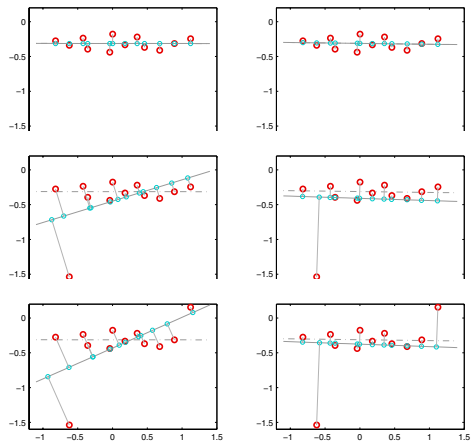
Deli Zhao Zhouchen Lin Xiaoou Tang
Visual Computing Group, Microsoft Research Asia, Beijing, China
{i-dezhao,zhoulin,xitang}@microsoft.com

Abstract

Dimensionality reduction plays a fundamental role in data processing, for which principal component analysis (PCA) is widely used. In this paper, we develop the Laplacian PCA (LPCA) algorithm which is the extension of PCA to a more general form by locally optimizing the weighted scatter. In addition to the simplicity of PCA, the benefits brought by LPCA are twofold: the strong robustness against noise and the weak metric-dependence on sample spaces. The LPCA algorithm is based on the global alignment of locally Gaussian or linear subspaces via an alignment technique borrowed from manifold learning. Based on the coding length of local samples, the weights can be determined to capture the local principal structure of data. We also give the exemplary application of LPCA to manifold learning. Manifold unfolding (non-linear dimensionality reduction) can be performed by the alignment of tangential maps which are linear transformations of tangent coordinates approximated by LPCA. The superiority of LPCA to PCA and kernel PCA is verified by the experiments on face recognition (FRGC version 2 face database) and manifold (Scherk surface) unfolding.

1. Introduction

Principal component analysis (PCA) [13] is widely used in computer vision, pattern recognition, and signal processing. In face recognition, for example, PCA is performed to map samples into a low-dimensional feature space where the new representations are viewed as expressive features [21, 26, 22]. Discriminators like LDA [2, 29], LPP [12], and MFA [31] are performed in the PCA-transformed spaces. In active appearance models (AAM) [8] and 3D morphable models [5], textures and shapes of faces are compressed in the PCA-learnt texture and shape subspaces, respectively. Deformation and matching between faces are performed using these texture and shape features. In manifold learning, tangent spaces of a manifold [14] are presented by the PCA subspaces and tangent coordinates are the PCA features. The representative algorithms in manifold learning like Hessian Eigenmaps [9], local tangent space alignment (LTSA) [33], S-Logmaps [7], and Riemannian normal coordinates (RNC) [15] are all based on tangent coordinates. In addition, K-Means, the classical algorithm for clustering, was proven equivalent to PCA in a relaxed condition [32]. Thus, PCA features can be naturally adopted for clustering. The performance of the algorithms mentioned above is determined by the subspaces and the features yielded by PCA. There are also variants of PCA, such as the probabilistic PCA [24], the kernel PCA (KPCA) [20], the robust PCA [25], the weighted PCA [17], the generalized PCA [27].



(a) (b)

Figure 1. Principal subspaces by PCA (a) and LPCA (b) ($K = 5$). Dotted lines denote the principal directions of unperturbed data (first row). Solid lines denote the principal directions of perturbed data (second and third rows).

However, PCA has some limitations as well. First, PCA is sensitive to noise, meaning that noise samples may incur significant change of principal subspaces. Figure 1 (a) illustrates an example. We can clearly observe the instability of PCA with perturbed sample points. To address this issue, the robust PCA algorithm [25] was proposed but with the sacrifice of the simplicity of PCA. The weighted PCA [17] was developed to perform smoothing on local patches of data in manifold learning. The authors used an iterative approach to compute weights, whose convergence can-

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

not be guaranteed. Besides, the weighted PCA in [17] is performed on local patches of data. The authors did not discuss how to derive the global projection matrix from the locally weighted scatters. Second, in principle, PCA is only reasonable for samples in Euclidean spaces where distances between samples are measured by l_2 norms. For non-Euclidean sample spaces, the scatter of samples cannot be represented by the summation of Euclidean distances. For instance, histogram features are non-Euclidean. Their distances are better measured by the Chi square. Therefore, the principal subspaces of such samples cannot be optimally obtained by the traditional PCA. The KPCA algorithm was designed for extracting principal components of samples whose underlying spaces are non-Euclidean. However, KPCA cannot explicitly produce principal subspaces of samples, which are required in many applications. Besides, KPCA is also sensitive to noise data because its criterion for optimization is intrinsically equivalent to PCA.

In this paper, we aim at enhancing the robustness of PCA and freeing it from the limitation of metrics at the same time with a Laplacian PCA (LPCA) algorithm. Different from the conventional PCA, we first formulate the scatter of samples on local patches of the data by the weighted summation of distances. The local scatter can be expressed in a compact form like the global scatter of the traditional PCA. Furthermore, we formulate a general framework for aligning local scatters to a global one. The framework of alignment is also applicable for methods based on spectral analysis in manifold learning. The optimal principal subspace can be obtained by solving a simple eigen-decomposition problem. Moreover, an efficient approach is provided for computing local LPCA features that are frequently utilized as tangent coordinates in manifold learning. As an application of LPCA, we develop tangential maps of manifolds based on tangential coordinates approximated by local LPCA. Particularly, we locally determine the weights by investigating the reductive coding length [16] of a local data patch, which is the variation of the coding length of a data set by leaving one point out. Hence, the principal structures of the data can be locally captured in this way.

Experiments are performed on face recognition and manifold unfolding to test LPCA. Face recognition is conducted on a subset of FRGC version 2 [18]. Three representative discriminators, LDA, LPP, and MFA, are performed on LPCA and PCA expressive features. The results indicate that the recognition performance of three discriminators based on LPCA is consistently better than that based on PCA. Besides, we perform dimensionality reduction on LBP non-Euclidean features using LPCA, PCA, and KPCA. LPCA shows significant superiority to PCA and KPCA. For manifold learning, we introduce Scherk surface [4] as a new example for manifold unfolding. LPCA-based tangential maps yields the faithful embeddings with or without noise.

$\mathbf{I}$	The identity matrix.
$\mathbf{e}$	The all-one column vector.
$\mathbf{H}$	$\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^T$ is a centering matrix.
$\mathcal{R}^D$	The D-dimensional Euclidean space.
$\mathbf{x}_i$	The i-th sample in $\mathcal{R}^D$, $i = 1, \ldots, n$.
$\mathcal{S}^x$	$\mathcal{S}^x = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.
$\mathbf{X}$	$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$.
$\bar{\mathbf{x}}$	$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}\mathbf{e}$ is the center of sample points.
$\mathbf{x}_{i_k}$	The k-th nearest neighbor of $\mathbf{x}_i$, $k = 1, \ldots, K$.
$\mathcal{S}_i^x$	$\mathcal{S}_i^x = \{\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_K}\}$, $i_0 = i$.
$\mathbf{X}_i$	$\mathbf{X}_i = [\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_K}]$.
$\mathbf{y}_i$	The representation of $\mathbf{x}_i$ in $\mathcal{R}^d$, $d < D$.
tr	The trace of a matrix.
det	The determinant of a matrix.
$\mathbf{X}^T$	The transpose of $\mathbf{X}$.
$\mathbf{W}_i$	$\mathbf{W}_i = diag(w_{i_0}, \ldots, w_{i_K})$ is a diagonal matrix.
$\mathbf{H}_w$	$\mathbf{H}_w = \mathbf{I} - \frac{\mathbf{W}_i\mathbf{e}\mathbf{e}^T}{\mathbf{e}^T\mathbf{W}_i\mathbf{e}}$, a weighted centering matrix.

Table 1. Notations

However, PCA-based tangential maps fails for noisy manifolds.

2. Laplacian PCA

The criterion of LPCA is to maximize the local scatter of data instead of the global one pursued by PCA. The scatter is the summation of *weighted* distances between low dimensional representations of original samples and their means. Like PCA, we aim at finding a global projection matrix $\mathbf{U}$ such that

$$\mathbf{y} = \mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}}), \tag{1}$$

where $\mathbf{U}$ is of size D by d. In the matrix form, we can write $\mathbf{Y} = \mathbf{U}^T(\mathbf{X}\mathbf{H})$[1]. In the following sub-sections, we present the formulations of performing local LPCA, the alignment of local LPCA, the global LPCA, and the efficient computation of local LPCA features. The notations used in this paper are listed in Table 1.

2.1. Local LPCA

For non-Gaussian or manifold-valued data, we usually deal with it from local patches because non-Gaussian data can be viewed locally Gaussian and a curved manifold can be locally viewed Euclidean [14]. Particularly, Gaussian distribution is the theoretical base of many statistical operations [11], and tangent spaces and tangent coordinates are the fundamental descriptors of a manifold. So, we begin with local LPCA.

Specifically, let α_i denote the local scatter on the i-th neighborhood $\mathcal{S}_i^y$. It is defined as

$$\alpha_i = \sum_{k=0}^{K} w_{i_k} \|\mathbf{y}_{i_k} - \bar{\mathbf{y}}_i\|^2_{\mathcal{R}^d}, \tag{2}$$

[1]The size of $\mathbf{I}$ and the length of $\mathbf{e}$ are easily known from the contexts.

where w_{i_k} is the related weight and $\bar{\mathbf{y}}_i$ is the geometric centroid of $\mathcal{S}_i^y$, i.e., $\bar{\mathbf{y}}_i = \frac{\mathbf{Y}_i\mathbf{W}_i\mathbf{e}}{\mathbf{e}^T\mathbf{W}_i\mathbf{e}}$. We will present the definition of w_{i_k} in Section 4. The distance between $\mathbf{y}_{i_k}$ and $\bar{\mathbf{y}}_i$ are measured by the l_2 norm $\| \bullet \|_{\mathcal{R}^d}$. Rewriting (2) yields

$$\alpha_i = \sum_{k=0}^{K} w_{i_k} tr\left((\mathbf{y}_{i_k} - \bar{\mathbf{y}}_i)(\mathbf{y}_{i_k} - \bar{\mathbf{y}}_i)^T\right) \tag{3}$$

$$= tr(\mathbf{Y}_i\mathbf{W}_i\mathbf{Y}_i^T) - \frac{tr(\mathbf{Y}_i\mathbf{W}_i\mathbf{e}\mathbf{e}^T\mathbf{W}_i\mathbf{Y}_i^T)}{\mathbf{e}^T\mathbf{W}_i\mathbf{e}}. \tag{4}$$

Thus we obtain

$$\alpha_i = tr(\mathbf{Y}_i\mathbf{L}_i\mathbf{Y}_i^T), \tag{5}$$

where

$$\mathbf{L}_i = \mathbf{W}_i - \frac{\mathbf{W}_i\mathbf{e}\mathbf{e}^T\mathbf{W}_i}{\mathbf{e}^T\mathbf{W}_i\mathbf{e}} \tag{6}$$

is called the local Laplacian scatter matrix. For $\mathbf{Y}_i$, we have

$$\mathbf{Y}_i = \mathbf{U}_i^T(\mathbf{X}_i - \bar{\mathbf{x}}_i\mathbf{e}^T) = \mathbf{U}_i^T(\mathbf{X}_i\mathbf{H}_w), \tag{7}$$

where $\bar{\mathbf{x}}_i$ is the geometric centroid of $\mathcal{S}_i^x$. Plugging (7) into (5) gives

$$\alpha_i = tr(\mathbf{U}_i^T\mathbf{X}_i\mathbf{H}_w\mathbf{L}_i\mathbf{H}_w^T\mathbf{X}_i^T\mathbf{U}_i). \tag{8}$$

It is not hard for one to check that $\mathbf{H}_w\mathbf{L}_i\mathbf{H}_w^T = \mathbf{L}_i$. So, we get the final expression of the local scatter

$$\alpha_i = tr(\mathbf{U}_i^T\mathbf{S}_l^i\mathbf{U}_i), \tag{9}$$

where $\mathbf{S}_l^i = \mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T$ is the local scatter matrix of $\mathcal{S}_i^x$. Imposing the orthogonality constraint on $\mathbf{U}_i$, we arrive at the following maximization problem

$$\begin{cases} \underset{\mathbf{U}_i}{\operatorname{argmax}}\ \alpha_i = \underset{\mathbf{U}_i}{\operatorname{argmax}}\ tr(\mathbf{U}_i^T\mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T\mathbf{U}_i), \\ \text{s.t.} \quad \mathbf{U}_i^T\mathbf{U}_i = \mathbf{I}. \end{cases} \tag{10}$$

$\mathbf{U}_i$ is essentially the principal column space of $\mathbf{S}_l^i$, i.e., the space spanned by the eigenvectors associated with the d largest eigenvalues of $\mathbf{S}_l^i$. We will present the efficient method for the computation in Section 2.4.

2.2. Alignment of Local Geometry

If we aim at deriving the global $\mathbf{Y}$ or the global projection $\mathbf{U}$, then global analysis can be performed on the alignment of localities. For the traditional approach, the Gaussian mixing model (GMM) [11], along with the EM scheme, is usually applied to fulfill this task (probabilistic PCA for instance). For spectral methods however, there has a simple approach. Here, we present a unified framework of alignment for spectral methods, by which the optimal solution in closed form can be obtained by eigen-analysis.

In general, the following form of optimization like (5) is involved on local patches

$$\underset{\mathbf{Y}_i}{\operatorname{argmax}}\ tr(\mathbf{Y}_i\mathbf{L}_i\mathbf{Y}_i^T), \tag{11}$$

where $\mathbf{L}_i$ is the local Laplacian scatter matrix[2]. For each $\mathcal{S}_i^y$, we have $\mathcal{S}_i^y \subset \mathcal{S}^y$, meaning that $\{\mathbf{y}_{i_0}, \mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_K}\}$ are always selected from $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. What is more, the selection labels are known from the process of nearest neighbors searching. Thus, we can write $\mathbf{Y}_i = \mathbf{Y}\mathbf{S}_i$, where $\mathbf{S}_i$ is the n by $(K+1)$ binary selection matrix associated with $\mathcal{S}_i^y$. Let $I_i = \{i_0, i_1, \ldots, i_K\}$ denote the label set. It is not hard to know that the structure of $\mathbf{S}_i$ can be expressed by

$$(\mathbf{S}_i)_{pq} = \begin{cases} 1 & \text{if } p = i_{q-1} \\ 0 & \text{otherwise} \end{cases}, \ i_{q-1} \in I_i, \ q = 1, \ldots, K+1, \tag{12}$$

meaning that $(\mathbf{S}_i)_{pq} = 1$ if the q-th vector in $\mathbf{Y}_i$ is the p-th vector in $\mathbf{Y}$. Then rewriting (11) gives

$$\arg\max_{\mathbf{Y}}\ tr(\mathbf{Y}\mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T\mathbf{Y}^T). \tag{13}$$

For each $\mathcal{S}_i^y$, such maximization must be performed. So, we have the following problem

$$\underset{\mathbf{Y}}{\operatorname{argmax}} \sum_{i=1}^{n} tr(\mathbf{Y}\mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T\mathbf{Y}^T) = \arg\max_{\mathbf{Y}}\ tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T), \tag{14}$$

where $\mathbf{L} = \sum_{i=1}^{n} \mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T$ is called the global Laplacian scatter matrix. The expression of $\mathbf{L}$ implies that, initialized by a zero matrix of the same size, $\mathbf{L}$ can be obtained by the update $\mathbf{L}(I_i, I_i) \leftarrow \mathbf{L}(I_i, I_i) + \mathbf{L}_i, i = 1, \ldots, n$.

The alignment technique presented here is hiddenly contained in [9], formulated (a little different from ours) in [6] and [33], and applied in [34, 37, 36]. Therefore, it is capable of aligning general local geometry matrices in manifold learning as well [35].

2.3. LPCA

For LPCA, our goal is to derive a global projection matrix. To this end, we need to plug $\mathbf{Y} = \mathbf{U}^T(\mathbf{X}\mathbf{H})$ in (14) to derive the expression of the global scatter when the global Laplacian scatter matrix is ready. Thus we obtain the following maximization problem

$$\begin{cases} \underset{\mathbf{U}}{\operatorname{argmax}}\ tr(\mathbf{U}^T\mathbf{X}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}^T\mathbf{U}), \\ \text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{cases} \tag{15}$$

Similar to the optimization in (10), $\mathbf{U}$ can be achieved by the eigen-decomposition of $\mathbf{X}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}^T$.

2.4. Efficient Computation

For real data however, the dimension of $\mathbf{x}_i$ is large. So it is computationally expensive to compute $\mathbf{U}_i$ in (10) by the eigen-decomposition of $\mathbf{S}_l^i$. However, the computation of local $\mathbf{Y}_i$ in (10) can be significantly simplified via SVD [10] in the case of $K \ll D$.

[2] In fact, $\mathbf{L}_i$ can be an arbitrary matrix that embodies the geometry of data on the local patch.

For the local Laplacian scatter matrix $\mathbf{L}_i$ and the global Laplacian scatter matrix $\mathbf{L}$, it is easy for one to verify that $\mathbf{L}_i\mathbf{e} = \mathbf{0}$ and $\mathbf{L}\mathbf{e} = \mathbf{0}$, implying that they have zero eigenvalues and the corresponding eigenvectors are the all-one vectors. Thus, we can say that $\mathbf{Y}_i$ in (10) and $\mathbf{Y}$ in (15) are all centered at the origin. For $\mathbf{L}_i$, it is not hard for one to check that $\mathbf{L}_i = \tilde{\mathbf{L}}_i\tilde{\mathbf{L}}_i^T$, where

$$\tilde{\mathbf{L}}_i = \mathbf{H}_w\mathbf{W}_i^{\frac{1}{2}}. \tag{16}$$

Then the local scatter matrix $\mathbf{S}_l^i$ can be rewritten as $\mathbf{S}_l^i = \mathbf{X}_i\tilde{\mathbf{L}}_i(\mathbf{X}_i\tilde{\mathbf{L}}_i)^T$, and we have the following theorem:

Theorem 1. *Let the d-truncated SVD of the tall-skinny matrix $\mathbf{X}_i\tilde{\mathbf{L}}_i$ be $\mathbf{X}_i\tilde{\mathbf{L}}_i = \mathbf{P}_i\mathbf{D}_i\mathbf{Q}_i^T$. Then the left singular matrix $\mathbf{P}_i$ is the local projection matrix $\mathbf{U}_i$, and the local coordinates $\mathbf{Y}_i$ is $\mathbf{Y}_i = (\mathbf{Q}_i\mathbf{D}_i)^T\mathbf{W}_i^{-\frac{1}{2}}$.*

By Theorem 1, the computational complexity of $\mathbf{U}_i$ and $\mathbf{Y}_i$ is reduced from $O(D^3)$ to $O(DK^2)$. Such speedup is critical for computing tangent coordinates in manifold learning.

3. Applications to Manifold Learning

In many cases, the data set $\mathcal{S}^x$ is manifold-valued. The low-dimensional representations can be obtained by nonlinear embeddings of original points. Here, we formulate tangential maps between manifolds to fulfill such tasks. The tangent spaces and the tangent coordinates of a manifold are approximated by local LPCA.

3.1. Tangential Maps

For a d-dimensional Riemannian manifold $\mathcal{M}^d$, its tangent space at each point is isomorphic to the Euclidean space $\mathcal{R}^d$ [14]. Thus linear transformations are allowable between tangent spaces of $\mathcal{M}^d$ and $\mathcal{R}^d$. Given a set of points $\mathcal{S}^x$ sampled from $\mathcal{M}^d$, the parameterization of $\mathcal{M}^d$ can be performed by tangential maps, where $\mathbf{x}_i$ is viewed as the natural coordinate representation in the ambient space $\mathcal{R}^D$ in which $\mathcal{M}^d$ is embedded.

With little abuse of notations, we let $\tilde{\mathcal{S}}_i^y = \{\tilde{\mathbf{y}}_{i_0}, \tilde{\mathbf{y}}_{i_1}, \ldots, \tilde{\mathbf{y}}_{i_K}\}$ denote the low-dimensional representation yielded by the local LPCA of $\mathcal{S}_i^x$, where an extra constraint $d < K$ should be imposed. The global representation $\mathcal{S}_i^y$ is obtained via the following linear transformation of $\tilde{\mathcal{S}}_i^y$:

$$\mathbf{Y}_i\mathbf{H}_w = \mathbf{A}_i\tilde{\mathbf{Y}}_i + \mathbf{E}_i, \tag{17}$$

where $\mathbf{E}_i$ is the error matrix and $\mathbf{A}_i$ is the Jacobian matrix of size $(K+1)$ by $(K+1)$ to be determined. Here, $\mathbf{Y}_i$ is centerized by $\mathbf{H}_w$ because the center of $\tilde{\mathcal{S}}_i^y$ lies at the origin. To derive the optimal $\mathbf{Y}_i$, we need to minimize $\mathbf{E}_i$, thus giving

$$\arg\min_{\mathbf{Y}_i} \|\mathbf{E}_i\|^2 = \arg\min_{\mathbf{Y}_i} \|\mathbf{Y}_i\mathbf{H}_w - \mathbf{A}_i\tilde{\mathbf{Y}}_i\|^2. \tag{18}$$

For the Jacobian matrix, we have $\mathbf{A}_i = \mathbf{Y}_i\mathbf{H}_w\tilde{\mathbf{Y}}_i^\dagger$, where $\dagger$ denotes the Moore-Penrose inverse [10] of a matrix. Plugging it in (18) and expanding the norm yields

$$\arg\min_{\mathbf{Y}_i} tr(\mathbf{Y}_i\mathbf{Z}_i\mathbf{Y}_i^T), \tag{19}$$

where

$$\mathbf{Z}_i = \mathbf{H}_w(\mathbf{I} - \tilde{\mathbf{Y}}_i^\dagger\tilde{\mathbf{Y}}_i)(\mathbf{I} - \tilde{\mathbf{Y}}_i^\dagger\tilde{\mathbf{Y}}_i)^T\mathbf{H}_w^T. \tag{20}$$

What we really need is the global representation $\mathbf{Y}$ instead of local $\mathbf{Y}_i$. So, the alignment technique is needed to align local representations to be a global one, which has been presented in Section 2.2.

To make the optimization presented here well-posed, we need a constraint on $\mathbf{Y}$. Let it be $\mathbf{Y}\mathbf{Y}^T = \mathbf{I}$. Putting everything together, we get a well-posed and easily solvable minimization problem

$$\begin{cases} \operatorname{argmin}_{\mathbf{Y}} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T), \\ \text{s.t.} \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}, \end{cases} \tag{21}$$

where $\mathbf{L} = \sum_{i=1}^n \mathbf{S}_i\mathbf{Z}_i\mathbf{S}_i^T$. Again, the optimization can be solved by the spectral decomposition of $\mathbf{L}$: the d-column matrix $\mathbf{Y}^T$ corresponds to the d eigenvectors associated with the d smallest nonzeros eigenvalues of $\mathbf{L}$. Thus, we complete a general framework of tangential maps.

3.2. LPCA Based on Tangential Maps

In general, the principal subspace of data set $\mathcal{S}_i^x$ are employed as the approximation of the tangent space tangent to the point $\mathbf{x}_i$. Thus, more robust approximation of the tangent space can provide better results of manifold unfolding. For LPCA however, we can obtain $\mathbf{Z}_i$ without the explicit computation of $\tilde{\mathbf{Y}}_i^\dagger$ by the following theorem:

Theorem 2. $\mathbf{Z}_i = \mathbf{H}_w(\mathbf{I} - \tilde{\mathbf{Q}}_i(\tilde{\mathbf{Q}}_i^T\tilde{\mathbf{Q}}_i)^{-1}\tilde{\mathbf{Q}}_i^T)\mathbf{H}_w^T$, *where* $\tilde{\mathbf{Q}}_i = \mathbf{W}_i^{-\frac{1}{2}}\mathbf{Q}_i$.

The inverse of $\tilde{\mathbf{Q}}_i^T\tilde{\mathbf{Q}}_i$ can be efficiently handled because $\tilde{\mathbf{Q}}_i^T\tilde{\mathbf{Q}}_i$ is of size d by d. The computation of $\mathbf{Z}_i$ is efficient by noting that $\mathbf{H}_w$ is a rank-one modification of $\mathbf{I}$ and $\mathbf{W}_i$ is diagonal. Zhang and Zha [33] first developed the LTSA algorithm based on tangential maps to unfold manifolds, where tangent spaces and tangent coordinates are derived by PCA. For LTSA, we have the following observation:

Proposition 1. *LPCA-based tangential maps coincide with the LTSA algorithm if $\mathbf{W}_i = \mathbf{I}$.*

Therefore, the framework formulated here is the generalization of Zhang and Zha's LTSA [33].

4. Definition of Weights

For traditional methods [17, 12], weights are determined by exponentials of Euclidean distances or its analogues. We

will show that such pairwise distance based dissimilarities cannot capture the principal structure of data robustly. So, we introduce the reductive coding length as a new dissimilarity that is compatible with the intrinsic structure of data.

4.1. Reductive Coding Length

The coding length [16] $L(\mathcal{S}_i^x)$[3] of a vector-valued set $\mathcal{S}_i^x$ is the intrinsic structural characterization of the set. We notice that if a point $\mathbf{x}_{i_k}$ complies with the structure of $\mathcal{S}_i^x$, then removing $\mathbf{x}_{i_k}$ from $\mathcal{S}_i^x$ will not affect the structure much. In contrast, if the point $\mathbf{x}_{i_k}$ is an outlier or a noise point, then removing $\mathbf{x}_{i_k}$ from $\mathcal{S}_i^x$ will change the structure significantly. This motivates us to define the variation of coding length as the structural descriptor between $\mathbf{x}_{i_k}$ and $\mathcal{S}_i^x$. The reductive variation of $L(\mathcal{S}_i^x)$ with and without $\mathbf{x}_{i_k}$ is defined as

$$\delta L_{i_k} = \left| L(\mathcal{S}_i^x) - L\left(\mathcal{S}_i^x \setminus \{x_{i_k}\}\right) \right|, \qquad k = 0, 1, \ldots, K, \tag{22}$$

where $|\bullet|$ denotes the absolute value of a scalar. Thus, the weight w_{i_k} in (2) can be defined as

$$w_{i_k} = \exp\left(-\frac{(\delta L_{i_k} - \delta \bar{L}_i)^2}{2\sigma_i^2} \right), \tag{23}$$

where $\delta \bar{L}_i$ and σ_i are the mean and the standard deviation of $\{\delta L_{i_0}, \ldots, \delta L_{i_K}\}$, respectively.

In fact, the reductive coding length is a kind of contextual distances. One can refer to [36] for more details.

4.2. Coding Length vs. Traditional Distance

We compare the difference between reductive coding length and the traditional pairwise distance by a toy example.

From Figure 2 (a), we observe that, using reductive coding length, the perturbed point (bottom) is slightly weighted whereas the five points that are consistent to the principal structure are heavily weighted. As shown in Figure 2 (c), the local principal direction (solid line) learnt by LPCA based on reductive coding length is highly consistent with the global principal structure (dotted line).

In contrast, as shown in Figure 2 (b), it seems promising that the perturbed point is very lightly weighted. However, the two significant points (pointed by two arrows) that are important to the principal structure are also lightly weighted. Thus, the local principal direction is mainly governed by the three central points. As a result, the principal direction (dotted line in Figure 2 (d)) learnt by LPCA based on pairwise Euclidean distance cannot capture the principal structure of the data. Note that, based on reductive coding length, the two significant points are most heavily weighted (Figure 2 (a)).

[3]The definition of it is presented in Appendix.

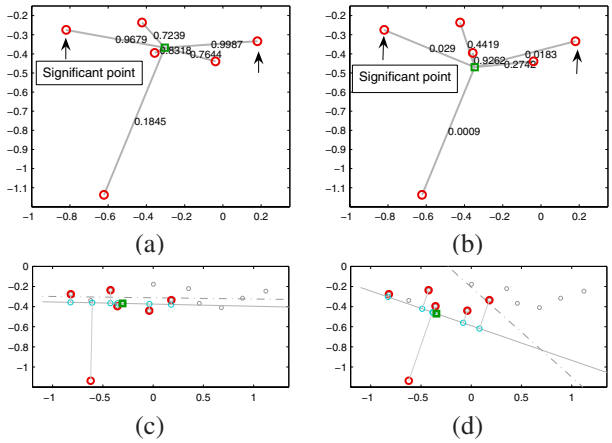


Figure 2. Illustrations of reductive coding length vs. pairwise Euclidean distance on one of the local patches (red circle markers) of the toy data. (a) and (b) illustrate the weights computed by reductive coding length and pairwise Euclidean distance, respectively. In (b), the green square marker denotes the geometric center instead of physical centroid. (c) and (d) illustrate the local principal directions (solid lines) learnt by LPCA based on reductive coding length and pairwise Euclidean distance, respectively.

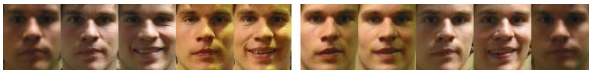


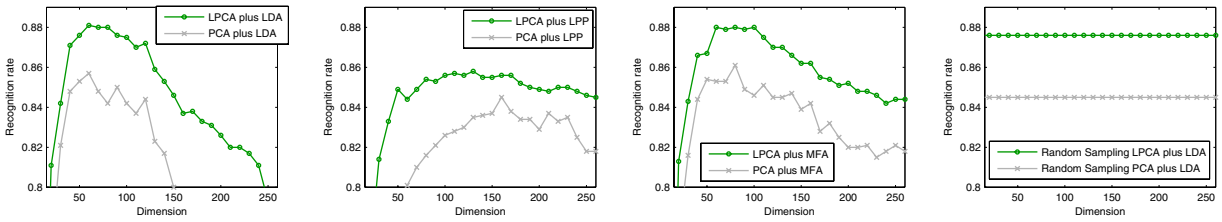
Figure 3. Facial images of one subject for our experiment in FRGC version 2. The first five facial images are in the gallery set and others in the probe set.

5. Experiment

We perform experiments on face recognition and manifold unfolding to compare the performance of our LPCA algorithm to that of existing related methods.

5.1. Face Recognition

Face database.We perform face recognition on a subset of facial data in FRGC version 2 [18]. The query set for the experiment 4 in this database consists of single uncontrolled still images which contain the variations of illumination, expression, time, and blurring. There are 8014 images of 466 subjects in the set. However, there are only two facial images available for some persons. So, we select a subset for our experiments. First, we search all images of each person in the set and take the first 10 facial images if the number of facial images is not less than 10. Thus we get 3160 facial images of 316 subjects. Then we divide the 316 subjects into three subsets. First, the first 200 subjects are used as the gallery set and the probe set, and the remaining 116 subjects are exploited as the training set. Second, we take the first five facial images of each person in the first 200 subjects as the gallery set and the remaining five images as the probe set. Therefore, the set of persons for training is disjoint with that of persons in the gallery and for the probe.



(a) (b) (c) (d)

Figure 4. The performance of LPCA and PCA as expressive feature extractors for face recognition. We reduce the dimensions of original facial images to be 290. Thus LPCA and PCA preserve 95% power and 98.8% power, respectively. Here, the power is defined as the ratio of the summation of eigenvalues corresponding to applied eigen-vectors to the trace of the scatter matrix. (a) LDA. (b) LPP ($K = 2$). (c) MFA ($k_1 = 2, k_2 = 20$). (d) Random sampling subspace LDA, with four hundred eigen-vectors computed. As in [30], we take the first 50 eigen-vectors as the base, and randomly sample another 100 eigen-vectors. Twenty LDA classifiers are designed.

We align the facial images according to the positions of eyes and mouths. Then each facial image is cropped to a size of 64×72. Figure 3 shows ten images of one subject. The nearest neighbor classifier is adopted. For the experiments in this subsection, $K = 2$ for the LPCA method.

Note that the experiments in this section are *not* to achieve the high performance of recognition. Rather, the goal is to compare the performance of LPCA as the same role where PCA or KPCA may be applied.

Dimensionality reduction as expressive features. For the development of discriminators in face recognition, PCA plays an important role. These discriminators solve generalized eigen-decomposition problems like $\mathbf{Au} = \lambda\mathbf{Bu}$. Due to the small sample size problem, the matrix $\mathbf{B}$ is usually singular, which leads to the difficulty of computation. So, dimensionality reduction is first performed by PCA to extract expressive features [21]. Then these discriminators are performed in the PCA-transformed space. Here we perform both PCA and LPCA to extract expressive features. And three representive discriminators LDA [2], LPP [12], and MFA [31] are applied for extracting discriminative features on these two kinds of expressive features, respectively. As shown in Figure 4 (a), (b), and (c), the recognition rates of these three discriminators based on LPCA are consistently higher than those based on PCA. Another application of PCA subspaces in face recognition is to the random sampling strategy [28, 30]. Figure 4 (d) shows that the discriminative power of LDA based on the random sampling of LPCA subspaces is superior to that based on PCA subspaces. These results verify that robust expressive subspaces can significantly improve the recognition rates of discriminators.

Dimensionality reduction on non-Euclidean features. In image-based recognition, visual features are sometimes extracted as expressive ones. However, the dimension of a visual feature vector is usually high, which leads to the load of storage of features and the consumption of time in computation. To reduce these loads, dimensionality reduction is necessary. The LBP algorithm is a newly emerging approach which is proven superior in un-supervised visual feature extraction [1]. The LBP features are based on histograms. Thus the LBP feature space is non-Euclidean. A distance measure in such a space is often chosen as the Chi square, defined as $\chi^2(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{D} \frac{(x_i^s - x_j^s)^2}{x_i^s + x_j^s}$, where x_i^s is the s-th component of $\mathbf{x}_i$. In this experiment, we compare the results of dimensionality reduction on LBP features by PCA, KPCA, and LPCA.

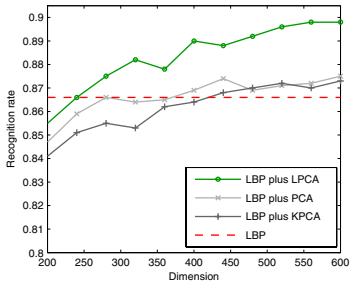
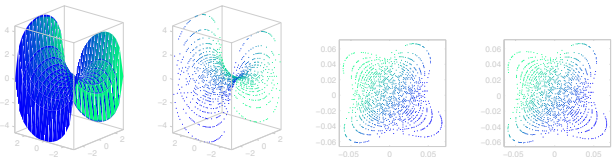


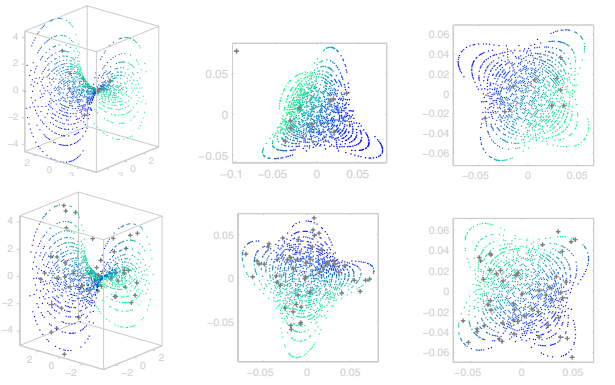
Figure 5. The performance of dimensionality reduction on LBP features. The recognition rate of LBP is the baseline.

We perform LBP on each facial image and then subdivide each facial image by 7×7 grids. Histograms with 59 bins are performed on each sub-block. An LBP feature vector is obtained by concatenating the feature vectors on sub-blocks. Here we use 58 uniform patterns for LBP and each uniform pattern accounts for one bin. The remaining 198 binary patterns are all put in another bin, resulting in a 59-bin histogram. So, the number of tuples in a LBP feature vector is $59 \times (7 \times 7) = 2891$. The $(8, 2)$ LBP is adopted. Namely, the number of circular neighbors for each pixel is 8 and the radius of the circle is 2. The above settings are consistent with that in [1]. As shown in Figure 5, with 250-dimensional features, the recognition rate of LBP plus LPCA is higher than that of LBP, which means that the dimensionality reduction is effective. In comparison, PCA and KPCA needs higher dimensional features. Overall, the performance of dimensionality reduction of LPCA is significantly better than that of PCA and KPCA.



(a) (b) (c) (d)

Figure 6. Scherk surface and its faithful Embeddings in the least squares sense. (a) Scherk surface ($b = 1$). (b) Randomly sampled points (without noise points). (c) and (d) are embeddings yielded by PCA-based tangential maps (LTSA) and LPCA-based tangential maps, respectively.



(a) (b) (c)

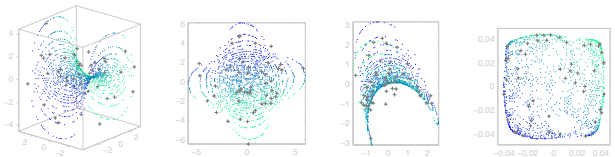
Figure 7. Noisy Scherk surface unfolding by tangential maps. (a) Randomly sampled points with noise points shown as crosses. (b) PCA-based tangential maps (LTSA). (c) LPCA-based tangential maps.

5.2. Manifold learning

The following experiments mainly examine the capability of LPCA and PCA on approximating tangent spaces and tangent coordinates of a manifold via tangential maps.

The manifold we use here is Scherk surface (Figure 6 (a)) which is a classical minimal surface, formulated as [4] $f(x, y) = \frac{1}{b} \ln \frac{\cos(bx)}{\sin(by)}$, where $-\frac{\pi}{2} < bx < \frac{\pi}{2}$, $-\frac{\pi}{2} < by < \frac{\pi}{2}$, and b is a positive constant. The minimal surface is a kind of zero mean curvature surface. Therefore, Scherk surface cannot be isometrically parameterized by an open subset in the two-dimensional Euclidean space. The faithful embeddings are only obtainable in the least squares sense. So, it is more challenging to unfold Scherk surface than Swiss roll [23] (zero Gaussian curvature surface) that is widely used for experiments in manifold learning. In addition, the randomly sampled points on Scherk's surface, we think, well simulate the real-world distribution of random samples: dense close to the center and sparse close to the boundary like noncurved Gaussian normal distribution [11]. These are the motivations that we use this surface for testing.

For each trial, 1200 points (including noise points) are randomly sampled from the surface, shown in Figure 6 (b). For all trials in this subsection, $K = 15$ for all involved methods. As shown in Figure 6 (c) and (d), both



(a) (b) (c) (d)

Figure 8. Noisy Scherk surface unfolding by existing representative methods. (a) Randomly sampled points with noise points shown as crosses. (b) Isomap. (c) LLE. (d) Laplacian Eigenmaps.

PCA-based tangential maps (LTSA) and LPCA-based tangential maps result in the faithful embeddings. However, PCA-based tangential maps distort the embeddings (Figure 7 (b)) when noise points appear. We can clearly see that PCA-based tangential maps is very sensitive to noise because very few noise point can change the result of the algorithm. In constrast, LPCA-based tangential maps show its robustness against noise (Figure 7 (c)). These results imply that LPCA can yield faithful tangent spaces that are less affected by noise to the manifold. From Figure 8, we can see that LLE [19] and Laplacian Eigenmaps [3] produce unsatisfactory embeddings. Among the three existing representative algorithms in Figure 8, only the Isomap [23] algorithm works for the surface[4].

6. Conclusion

The sensitivity to noise and the incompletence to non-Euclidean samples are two major problems of the traditional PCA. To address these two issues, we propose a novel algorithm, named Laplacian PCA. LPCA is an extension of PCA by optimizing the locally weighted scatters instead of the single global non-weighted scatter in PCA. The principal subspace is learnt by the alignment of local optimizations. A general alignment technique is formulated. Based on the coding length in information theory, we present a new approach to determining weights. As an application, we formulate the tangential maps in manifold learning via LPCA, which can be exploited for non-linear dimensionality reduction. The experiments are performed on face recognition and manifold unfolding, which testify to the superiority of LPCA to PCA and other variants of PCA, like KPCA.

Acknowledgement

The authors would like to thank Yi Ma and John Wright for discussion, and Wei Liu for his comments.

Appendix

Coding Length. For the set of vectors $\mathcal{S}_i^x$, the total number of bits needed to code $\mathcal{S}_i^x$ is [16]

[4] Actually, there is an outlier-detection procedure in the published Matlab codes of Isomap, which is one of reasons why Isomap is robust.

$$L(\mathcal{S}_i^x) = \frac{K+1+n}{2}\log\det\left(\mathbf{I} + \frac{n}{\varepsilon^2(K+1)}\mathbf{X}_i\mathbf{H}\mathbf{X}_i^T\right) + \frac{n}{2}\log\left(1 + \frac{\bar{\mathbf{x}}_i^T\bar{\mathbf{x}}_i}{\varepsilon^2}\right), \quad (24)$$

where ε is the allowable distortion. In fact, the computation can be considerably reduced by the commutativity of determinant

$$\det(\mathbf{I} + \mathbf{X}_i\mathbf{H}\mathbf{X}_i^T) = \det(\mathbf{I} + \mathbf{H}\mathbf{X}_i^T\mathbf{X}_i\mathbf{H}) \quad (25)$$

in the case of $K+1 \ll n$. It is worth noting that $\mathbf{X}_i^T\mathbf{X}_i$ in (25) will be a kernel matrix if the kernel trick is exploited. The allowable distortion ε in $L(\mathcal{S}_i^x)$ is a free parameter. In all our experiments, we empirically choose $\varepsilon = (\frac{10n}{K})^{\frac{1}{2}}$.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face decription with local binary patterns: application to face recognition. *IEEE Tran. on PAMI*, 28(12):2037–2041, 2006. 6

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Tran. on PAMI*, 19(1):711–720, 1997. 1, 6

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003. 7

[4] M. Berger and B. Gostiaux. Differential Geometry: Manifolds, Curves, and Surfaces. *Springer-Verlag*, 1988. 2, 7

[5] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Tran. on PAMI*, 25(9):1063–1074, 2003. 1

[6] M. Brand. Charting a manifold. *NIPS*, 15, 2003. 3

[7] A. Brun, C. Westin, M. Herberthsson, and H. Knutsson. Sample logmaps: intrinsic processing of empirical manifold data. *Proceedings of the Symposium on Image Analysis*, 2006. 1

[8] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Tran. on PAMI*, 23(6):681–685, 2001. 1

[9] D. Donoho and C. Grimes. Hessian eigenmaps: locally linear embedding techniques for high dimensional data. *PNAS*, 100:5591–5596, 2003. 1, 3

[10] G. Golub and C. Van. Matrix Computations. *The Johns Hopkins University Press*, 1996. 3, 4

[11] T. Hastie. The Elements of Statistical Learning. *Springer*, 2001. 2, 3, 7

[12] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using Laplacianfaces. *IEEE Tran. on PAMI*, 27(3):328–340, 2005. 1, 4, 6

[13] T. Jollie. Principal Component Analysis. *Springer-Verlag, New York*. 1

[14] J. Lee. Riemannian Manifolds: An Introduction to Curvature. *Springer-Verlag*, 2003. 1, 2, 4

[15] T. Lin, H. Zha, and S. Lee. Riemannian manifold learning for nonlinear dimensionality reduction. *ECCV*, pages 44–55, 2006. 1

[16] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Tran. on PAMI*, 2007. 2, 5, 7

[17] J. Park, Z. Zhang, H. Zha, and R. Kasturi. Local smoothing for manifold learning. *CVPR*, 2004. 1, 2, 4

[18] P. Philips, P. Flynn, T. Scruggs, and K. Bowyer. Overview of the face recognition grand challenge. *CVPR*, 2005. 2, 5

[19] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000. 7

[20] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 1

[21] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Tran. on PAMI*, 18(8):831–836, 1996. 1, 6

[22] X. Tang and X. Wang. Face sketch recognition. *IEEE Tran. on Circuits and Systems for Video Technology*, 14(1):50–57, 2004. 1

[23] J. Tenenbaum, V. D. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000. 7

[24] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 2001. 1

[25] D. Torre and M. Black. Robust principal component analysis for computer vision. *ICCV*, pages 362–369, 2001. 1

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

[27] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis. *IEEE Tran. on PAMI*, 27:1–15, 2005. 1

[28] X. Wang and X. Tang. Random sampling LDA for face recognition. *International Conf. on Computer Vision and Pattern Recognition*, pages 259–265, 2004. 6

[29] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, 2004. 1

[30] X. Wang and X. Tang. Random sampling for subspace face recognition. *International Journal of Computer Vision*, 70(1):91–104, 2006. 6

[31] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Tran. on PAMI*, 29(1):40–51, 2007. 1, 6

[32] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for K-means clustering. *NIPS*, 2001. 1

[33] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction by local tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338, 2004. 1, 3, 4

[34] D. Zhao. Formulating LLE using alignment technique. *Pattern Recognition*, 39:2233–2235, 2006. 3

[35] D. Zhao. Numerical geomtry of data manifolds. *Shanghai Jiao Tong University*, Master Thesis, Jan. 2006. 3

[36] D. Zhao, Z. Lin, and X. Tang. Contextual distance for data perception. In *International Conference on Computer Vision*, 2007. 3, 5

[37] D. Zhao, Z. Lin, R. Xiao, and X. Tang. Linear Laplacian discrimination for feature extraction. In *International Conf. on Computer Vision and Pattern Recognition*, 2007. 3