PRINCIPAL MANIFOLDS AND NONLINEAR DIMENSIONALITY REDUCTION VIA TANGENT SPACE ALIGNMENT

ZHENYUE ZHANG* AND HONGYUAN ZHA†

Abstract. We present a new algorithm for manifold learning and nonlinear dimensionality reduction. Based on a set of unorganized data points sampled with noise from a parameterized manifold, the local geometry of the manifold is learned by constructing an approximation for the tangent space at each data point, and those tangent spaces are then aligned to give the global coordinates of the data points with respect to the underlying manifold. We also present an error analysis of our algorithm showing that reconstruction errors can be quite small in some cases. We illustrate our algorithm using curves and surfaces both in 2D/3D Euclidean spaces and higher dimensional Euclidean spaces. We also address several theoretical and algorithmic issues for further research and improvements.

Keywords: nonlinear dimensionality reduction, principal manifold, tangent space, subspace alignment, singular value decomposition

AMS subject classifications. 15A18, 15A23, 65F15, 65F50

1. Introduction.

Many high-dimensional data sets in real-world applications can be modeled as sets of points or vectors lying close to a low-dimensional nonlinear manifold. Discovering the structure of the manifold from such a sample of data points represents a very challenging unsupervised learning problem [13]. Example low-dimensional manifolds embedded in high-dimensional input spaces include image vectors representing the same 3D objects under different camera views and lighting conditions. Another example is a set of document vectors in a text corpus dealing with a specific topic. The key observation is that the dimensions of the embedding spaces can be very high (e.g., the number of pixels for each images in the image collection or the number of words or phrases in the vocabulary of the text corpus), the intrinsic dimension of the data points, however, is rather limited due to factors such as physical constraints and linguistic correlations. Traditional dimensionality reduction techniques such as principal component analysis (using eigendecomposition of the sample covariance matrix) and factor analysis usually work well when the data points lie close to a *linear* (affine) subspace in the input space. They, however, tend to fail to detect nonlinear structures in the data points.

Recently, there has been considerable interest in developing efficient algorithms for constructing nonlinear low-dimensional manifolds from sample data points in high-dimensional spaces, emphasizing simple algorithmic implementation and avoiding optimization problems prone to local minima [17, 21]. Two lines of research of manifold learning and nonlinear dimensionality reduction have emerged: one is exemplified by [21, 7] where pairwise *geodesic* distances of the data points with respect to the underlying manifold are estimated, and the classical multi-dimensional scaling is used to project the data points into a low-dimensional space that best preserves the geodesic distances. Another line of research follows the long tradition starting

[*] Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou, 310027, P. R. China. zyzhang@zju.edu.cn. The part work of this author was done while visiting Penn State University and was supported in part by the Special Funds for Major State Basic Research Projects (project G19990328), NSFC (project 60372033), Foundation for University Key Teacher by the Ministry of Education, China, and NSF grants CCR-9901986.

[†] Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, zha@cse.psu.edu. The work of this author was supported in part by NSF grants CCR-9901986.

with self-organizing maps (SOM) [13], principal curves/surfaces [11] and topology-preserving networks [14]. The key idea is that the information about the global structure of a nonlinear manifold can be obtained from a careful analysis of the interactions of the *overlapping* local structures. In particular, the local linear embedding (LLE) method constructs a local geometric structure that is invariant to translations and orthogonal transformations in a neighborhood of each data point and seeks to project the data points into a low-dimensional space that best preserves those local geometries [17, 18]. (A related method using Hessian matrices is presented in [8]).

Our approach draws inspiration from and extends the pioneering work in [17, 18] which opens up new directions in nonlinear manifold learning with many fundamental problems required to be further investigated. Our starting point is not to consider nonlinear dimensionality reduction in isolation as merely constructing a nonlinear projection, but rather to combine it with the process of reconstruction of the nonlinear manifold, and we argue that the two processes interact with each other in a mutually reinforcing way. We address two inter-related objectives of nonlinear structure finding: 1) to construct the so-called principal manifold [11] that goes through “the middle” of the data points; and 2) to find the global coordinate system that characterizes the set of data points in a low-dimensional space. The basic idea of our approach is to use the tangent space in the neighborhood of a data point to represent the local geometry, and then align those tangent spaces to construct the global coordinate system for the nonlinear manifold.

The rest of the paper is organized as follows: in Section 2, we formulate the problem of manifold learning and dimensionality reduction in more precise terms, and illustrate the intricacy of the problem using the linear case as an example. In Section 3, we discuss the issue of learning local geometry using tangent spaces, and in Section 4 we show how to align those tangent spaces in order to learn the global coordinate system of the underlying manifold. Section 5 discusses how to construct the manifold once the global coordinate system is available. We call the new algorithm *local tangent space alignment* (LTSA) algorithm. In Section 6, we present an error analysis of LTSA, especially illustrating the interactions among curvature information embedded in the Hessian matrices, local sampling density and noise level, and the regularity of the Jacobi matrix. In Section 7, we show how the partial eigendecomposition used in global coordinate construction can be efficiently computed. We then present a collection of numerical experiments in Section 8. Section 9 concludes the paper and addresses several theoretical and algorithmic issues for further research and improvements.

NOTATION. We use $\|\cdot\|_2$ to denote the 2-norm of a vector or matrix, and $\|A\|_F = \left(\sum_{i,j} a_{ij}^2\right)^{1/2}$ the Frobenius norm of a matrix. We will also use $\|\cdot\|$ to denote a generic vector or matrix norm.

2. Manifold Learning and Dimensionality Reduction.

The general theory of manifold learning can be cast in the framework of Riemannian geometry, but to avoid unnecessary abstraction, we consider the special case of parameterized manifolds represented as hypersurfaces of arbitrary co-dimension in Euclidean spaces [15].

DEFINITION. Let $d < m$, and Ω open in $\mathcal{R}^d$. Let $f : \Omega \to \mathcal{R}^m$. The set $\mathcal{F} \equiv f(\Omega)$ together with the mapping f is called a *parameterized manifold* of dimension d.

Additionally, to avoid some of the complications discussed in Section 8, we further assume that the manifold is regular, i.e., the Jacobi matrix of f is of full rank at every point of the manifold and the manifold is not self-intersecting. Assume further that we are given a set of data points $x_1, \cdots, x_N$, where $x_i \in \mathcal{R}^m$ are sampled possibly with noise from the manifold, i.e.,

$$x_i = f(\tau_i) + \epsilon_i, \quad i = 1, \ldots, N,$$

where ϵ_i represent noise. We use the term *dimensionality reduction* to mean the estimation of the unknown lower dimensional feature vectors τ_i's from the x_i's, i.e., the x_i's which are data points in $\mathcal{R}^m$ are (nonlinearly) projected to the τ_i's which are points in $\mathcal{R}^d$. Since we assume that $d < m$, we therefore realize the objective of dimensionality reduction of the data points. By *manifold learning* we mean the reconstruction of f from the x_i's, i.e., for an arbitrary test point $\tau \in \Omega \subset \mathcal{R}^d$, we can provide an estimate of $f(\tau)$. These two problems are inter-related, and a solution of one leads to a solution of the other. In some situations, dimensionality reduction can be the means to an end by itself, and it is not necessary to learn the manifold. In this paper, however, we promote the notion that both problems are really the two sides of the same coin, and the best approach is not to consider each in isolation. Before we tackle the algorithmic details, we first want to point out that the key difficulty in manifold learning and nonlinear dimensionality reduction from a sample of data points is that the data points are *unorganized*, i.e., no adjacency relationships among them are known beforehand. Otherwise, the learning problem becomes the well-researched nonlinear regression problem (for a more detailed discussion, see [9] where techniques from computational geometry were used to solve error-free manifold learning problems). To ease discussion, in what follows we will call the high-dimensional space where the data points live the *input* space, and the low-dimensional space into which the data points are projected the *feature* space.

To illustrate the concepts and problems we have introduced, we give a brief review of *linear* manifold learning and *linear* dimensionality reduction, also known in statistics as the principal component analysis (PCA) [11]. We assume that the set of data points are sampled from a d-dimensional affine subspace, ie.,

$$x_i = c + U\tau_i + \epsilon_i, \quad i = 1, \ldots, N,$$

where $c \in \mathcal{R}^m, \tau_i \in \mathcal{R}^d$, and $\epsilon_i \in \mathcal{R}^m$ represent noise. $U \in \mathcal{R}^{m \times d}$ is a matrix forming an orthonormal basis of the affine subspace. Let

$$X = [x_1, \cdots, x_N], \quad T = [\tau_1, \cdots, \tau_N], \quad E = [\epsilon_1, \cdots, \epsilon_N].$$

Then in matrix form, the data-generation model can be written as

$$X = c\,e^T + UT + E,$$

here e is an N-dimensional column vector of all ones. The problem of linear manifold learning is that we seek c, U and T to minimize the reconstruction error, i.e,

$$\min \|E\| = \min_{c,U,T} \quad \|X - (c\,e^T + UT)\|_F,$$

where $\|\cdot\|_F$ stands for the Frobenius norm of a matrix. This problem can be easily solved by singular value decomposition (SVD) based upon the following two observations:

1) The norm of the error matrix E can be reduced by centering the columns of E and hence one can assume that the optimal E has zero mean. This requirement can be fulfilled if c is chosen as the mean of X, i.e, $c = Xe/N \equiv \bar{x}$.

2) The low-rank matrix UT is the optimal rank-d approximation to the centered data matrix $X-\bar{x}e^T$. Hence the the optimal solution is given by the SVD of $X-\bar{x}e^T$,

$$X-\bar{x}e^T = Q\Sigma V^T, \quad P\in\mathcal{R}^{m\times m},\ \Sigma\in\mathcal{R}^{m\times N},\ V\in\mathcal{R}^{N\times N},$$

i.e., $UT = Q_d\Sigma_d V_d^T$, where $\Sigma_d = \operatorname{diag}(\sigma_1,\cdots,\sigma_d)$ with the d largest singular values of $X-\bar{x}e^T$, Q_d and V_d are the matrices of the corresponding left and right singular vectors, respectively. The optimal U^* is then given by Q_d and the learned linear manifold is represented by the linear function

$$f(\tau) = \bar{x} + U^*\tau.$$

In this model, the coordinate matrix T corresponding to the data matrix X is given by

$$T = (U^*)^T(X-\bar{x}e^T) = \operatorname{diag}(\sigma_1,\ldots,\sigma_d)V_d^T. \tag{2.1}$$

Ideally, the dimension d of the learned linear manifold should be chosen such that $\sigma_{d+1} \ll \sigma_d$.

The function f is not unique in the sense that it can be reparameterized, i.e, the coordinate can be replaced by $\tilde{\tau}$ with a global affine transformation $\tau = P\tilde{\tau}$, if we change the basis matrix U^* to U^*P. But it is easy to see that the parameterization given in (2.1) leads to an isometric embedding, i.e.,

$$\|x_i - x_j\|_2 = \|\tau_i - \tau_j\|_2, \quad i,j = 1,\ldots,N.$$

For the linear case we just discussed, the problem of dimensionality reduction is solved by computing the right singular vectors V_d, and this can be done without the help of the linear function f. Similarly, the construction of the linear function f is done by computing U^* which is just the matrix of the d largest left singular vectors of $X-\bar{x}e^T$.

The case for nonlinear manifolds is more complicated. In general, the global nonlinear structure will have to come from local linear analysis and alignment [17, 20]. In [17], local linear structure of the sample data points are extracted by representing each point x_i as a weighted linear combination of its neighbors, and the local weight vectors are preserved as much as possible in the feature space in order to obtain a global coordinate system. In [20], a linear alignment strategy was proposed for aligning a general set of local linear structures. The type of local geometric information we propose to use is the tangent space at a given point which is constructed from a neighborhood of the given point (independent of our work, Brand in [5] also proposed a similar approach using tangent spaces formulated using probabilistic modeling terms). The tangent space at each sample point provides a low-dimensional linear approximation of the local geometric structure of the nonlinear manifold. What we want to preserve are the local coordinates of the data points in the neighborhood with respect to the tangent space. Those local tangent coordinates will be aligned in the low-dimensional space by different local affine transformations to obtain a global coordinate system. Our alignment method is similar in spirit to that proposed in [20] (see also related alignment approaches in [23, 24]). In the next section we will discuss preservation of local geometry using the tangent spaces and their global alignment which will then be applied to data points sampled with noise in Section 4.

3. Tangent Space and Its Global Alignment. We assume that $\mathcal{F} = f(\Omega)$ is a parameterized manifold with $f:\Omega\subset\mathcal{R}^d\to\mathcal{R}^m$ as defined in section 2. The objective as we mentioned before for nonlinear dimensionality reduction is to reconstruct the underlying coordinate τ from the corresponding function value $f(\tau)$ without explicitly constructing f. Assume that the function f is smooth enough, using first-order Taylor expansion at a *fixed* τ, we have that for a neighbor $\bar{\tau}$ of the fixed τ,

$$f(\bar{\tau}) = f(\tau) + J_f(\tau) \cdot (\bar{\tau} - \tau) + O(\|\bar{\tau} - \tau\|^2), \tag{3.2}$$

where $J_f(\tau) \in \mathcal{R}^{m \times d}$ is the Jacobi matrix of f at τ. If we write the m components of $f(\tau)$ as ($\tau = [\tau_1, \dots, \tau_d]^T$),

$$f(\tau) = \begin{bmatrix} f_1(\tau) \\ \vdots \\ f_m(\tau) \end{bmatrix}, \quad \text{then} \quad J_f(\tau) = \begin{bmatrix} \partial f_1/\partial\tau_1 & \cdots & \partial f_1/\partial\tau_d \\ \vdots & \vdots & \vdots \\ \partial f_m/\partial\tau_1 & \cdots & \partial f_m/\partial\tau_d \end{bmatrix}.$$

The tangent space $\mathcal{T}_\tau$ of f at τ is spanned by the d column vectors of $J_f(\tau)$ and is therefore of dimension at most d, i.e., $\mathcal{T}_\tau = \mathrm{span}(J_f(\tau))$. The vector $\bar{\tau} - \tau$ gives the local coordinate of the first-order approximation of $f(\bar{\tau})$ corresponding to the affine subspace $f(\tau) + \mathcal{T}_\tau$. Without knowing the function f, we can not explicitly compute the Jacobi matrix $J_f(\tau)$. However, the tangent space at $f(\tau)$ can be approximated using points in a neighbor set, say $\{f(\bar{\tau}) \,|\, \bar{\tau} \in \Omega_\tau\}$, of $f(\tau)$ in the high-dimensional input space, where Ω_τ denotes a neighbor set of τ in the low-dimensional feature space. Let Q_τ be an orthonormal basis matrix of $\mathcal{T}_\tau$ and $\theta^*_\tau(\bar{\tau})$ the local coordinate of $\bar{\tau}$ corresponding to the basis Q_τ. Then

$$J_f(\tau)(\bar{\tau} - \tau) = Q_\tau \theta^*_\tau(\bar{\tau}).$$

The local coordinate $\theta^*_\tau(\bar{\tau})$ also depends on the local *centroid* τ. Our purpose is to retrieve the global coordinate τ using the local coordinates θ^*_τ by a certain global alignment technique proposed below. To this end, denote $P_\tau = Q_\tau^T J_f(\tau)$. We have

$$\theta^*_\tau(\bar{\tau}) = Q_\tau^T J_f(\tau)(\bar{\tau} - \tau) \equiv P_\tau(\bar{\tau} - \tau).$$

The unknown matrix P_τ represents a local affine transformation from the global τ-coordinate-system to the local θ^*_τ-coordinate-system. Note that $\theta^*_\tau(\bar{\tau})$ is the local representation of the *linear part* of $f(\bar{\tau}) - f(\tau)$. It can be approximated by $\theta(\bar{\tau})$ which is defined to be the local coordinate of the *orthogonal projection* of $f(\bar{\tau}) - f(\tau)$ onto $\mathcal{T}_\tau$,

$$\theta(\bar{\tau}) \equiv Q_\tau^T(f(\bar{\tau}) - f(\tau)) = \theta^*_\tau(\bar{\tau}) + O(\|\bar{\tau} - \tau\|^2). \tag{3.3}$$

($\theta(\bar{\tau})$ also depends on τ.) Ignoring the second-order term, the global coordinate τ satisfies

$$\int_\Omega \Big(\int_{\Omega(\tau)} \|P_\tau(\bar{\tau} - \tau) - \theta(\bar{\tau})\|^2 d\bar{\tau} \Big/ \int_{\Omega(\tau)} d\bar{\tau} \Big) d\tau \approx 0.$$

Therefore, a natural way to approximate the global coordinate is to find a global coordinate τ and a local affine transformation P_τ that minimize the error function

$$\int_\Omega \Big(\int_{\Omega(\tau)} \|P_\tau(\bar{\tau} - \tau) - \theta(\bar{\tau})\|^2 d\bar{\tau} \Big/ \int_{\Omega(\tau)} d\bar{\tau} \Big) d\tau \tag{3.4}$$

over all possible P_τ. The above idea leads to an *optimal embedding* using *nonlinear* alignment for the manifold learning and dimensionality reduction problems. This idea will be picked up partially at the end of section 4, and a full discussion will be given in a forthcoming paper

This paper will focus on a *linear* alignment approach that can be devised as follows. Since $J_f(\tau)$ is of full column rank, the matrix P_τ should be non-singular and we can represent τ in terms of θ_τ with the inverse matrix L_τ of P_τ,

$$\bar\tau - \tau \approx P_\tau^{-1}\theta(\bar\tau) = L_\tau\theta(\bar\tau).$$

The above equation shows that the affine transformation L_τ should align this local coordinate θ to the *global* coordinate τ for $f(\tau)$. Naturally we should seek to find a global coordinate τ and a local affine transformation L_τ to minimize

$$\int_\Omega \Big(\int_{\Omega(\tau)} \|\bar\tau - \tau - L_\tau\theta(\bar\tau)\| d\bar\tau \Big/ \int_{\Omega(\tau)} d\bar\tau \Big) d\tau \tag{3.5}$$

over all possible nonsingular L_τ.

Under certain conditions, we can have a clearer idea of the global minimizer of (3.5): let f be a locally isometric mapping i.e., $J_f(\tau)^T J_f(\tau) = I$, and Ω is connected, then P_τ is orthogonal. So we have $\bar\tau - \tau = P_\tau^T\theta_\tau^*(\bar\tau)$. By (3.3),

$$\|\bar\tau - \tau - P_\tau^T\theta(\bar\tau)\| = O(\|\bar\tau - \tau\|^2).$$

If the Hessian of f is bonded over Ω, then there is a constant C determined by the Hessian of f such that $\|\bar\tau - \tau - P_\tau^T\theta(\bar\tau)\| \le C\|\bar\tau - \tau\|^2$, and

$$\begin{aligned} & \int_\Omega \Big(\int_{\Omega(\tau)} \|\bar\tau - \tau - P_\tau^T\theta(\bar\tau)\| d\bar\tau \Big/ \int_{\Omega(\tau)} d\bar\tau \Big) d\tau \\ \le & \int_\Omega \Big(\int_{\Omega(\tau)} C\|\bar\tau - \tau\|^2 d\bar\tau \Big/ \int_{\Omega(\tau)} d\bar\tau \Big) d\tau \\ \le & \int_\Omega Cr^2(\Omega_\tau) d\tau \le C|\Omega| \sup_\tau r^2(\Omega_\tau), \end{aligned}$$

where $r(\Omega_\tau) = \sup_{\bar\tau\in\Omega_\tau} \|\bar\tau - \tau\|$ denotes the radius of the neighbor set Ω_τ, and $|\Omega| = \int_\Omega d\tau$ denotes the size of Ω. It implies that $\int_\Omega \Big(\int_{\Omega(\tau)} \|\bar\tau - \tau - P_\tau^T\theta(\bar\tau)\| d\bar\tau \Big/ \int_{\Omega(\tau)} d\bar\tau \Big) d\tau \to 0$ as $\sup_\tau r(\Omega_\tau) \to 0$. Therefore we proved the following

THEOREM 3.1. *Assume the $f : \Omega \subset \mathcal{R}^d \to \mathcal{R}^m$ is locally isometric and Ω is connected and bounded. Denote $\rho = \sup_\tau r(\Omega_\tau)$. Then*

$$\lim_{\rho\to 0} \min_{f(\Omega)=\mathcal{M},\, L_\tau} \int_\Omega \Big(\int_{\Omega(\tau)} \|\bar\tau - \tau - L_\tau\theta(\bar\tau)\| d\bar\tau \Big/ \int_{\Omega(\tau)} d\bar\tau \Big) d\tau = 0.$$

The above theorem shows that under the condition $\sup_\tau r(\Omega_\tau) \to 0$ the minimization problem (3.5) achieves its global minimum of zero if f is locally isometric.

As we will see later, the linear approach in (3.5) is more readily amendable to computation than the nonlinear one in (3.4). The discrete version of optimization problem can be solved by the computation of an eigenvalue problem. Obviously, if the manifold $\mathcal{F}$ is not *regular*, i.e., the Jacobi matrix J_f is not of full column rank at some points $\tau \in \Omega$, then the linear approach may result in local geometric distortion. In this case, the two minimization problems (3.4) and (3.5) may lead to quite different solutions.

As discussed in the linear case, the low-dimensional feature vector τ is not uniquely determined by the manifold $\mathcal{F}$. We can reparameterize $\mathcal{F}$ using $f(g(\tau))$ where $g(\cdot)$ is a smooth 1-to-1 onto mapping of Ω to itself. As we will see in the next section, the actual parameterization of $\mathcal{F}$ computed is closely related to the normalization conditions imposed on the low-dimensional coordinates.

4. Feature Extraction through Alignment. Now we consider how to construct the global coordinates and local affine transformation when we are given a data set $X = [x_1, \ldots, x_N]$ sampled with noise from an underlying nonlinear manifold,

$$x_i = f(\tau_i) + \epsilon_i, \quad i = 1, \ldots, N,$$

where $\tau_i \in \mathcal{R}^d$, $x_i \in \mathcal{R}^m$ with $d < m$.[1] For each x_i, let $X_i = [x_{i_1}, \ldots, x_{i_k}]$ be a matrix consisting of its k-nearest neighbors including x_i, say in terms of the Euclidean distance.[2] Consider computing the best d-dimensional affine subspace approximation for the data points in X_i,

$$\min_{x,\Theta,Q} \sum_{j=1}^{k} \left\| x_{i_j} - (x + Q\theta_j) \right\|_2^2 = \min_{x,\Theta,Q} \left\| X_i - (xe^T + Q\Theta) \right\|_2^2,$$

where Q is of d columns and is orthonormal, and $\Theta = [\theta_1, \ldots, \theta_k]$. As discussed in Section 2, the optimal x is given by $\bar{x}_i$, the mean of all the x_{i_j}'s and the optimal Q is given by Q_i, the matrix of d left singular vectors of $X_i(I - ee^T/k)$ corresponding to its d largest singular values, and Θ is given by Θ_i defined as

$$\Theta_i = Q_i^T X_i(I - \frac{1}{k}ee^T) = [\theta_1^{(i)}, \cdots, \theta_k^{(i)}], \quad \theta_j^{(i)} = Q_i^T(x_{i_j} - \bar{x}_i). \tag{4.6}$$

Therefore we have

$$x_{i_j} = \bar{x}_i + Q_i\theta_j^{(i)} + \xi_j^{(i)}, \tag{4.7}$$

where $\xi_j^{(i)} = (I - Q_iQ_i^T)(x_{i_j} - \bar{x}_i)$ denotes the reconstruction error.

We now consider constructing the global coordinates $\tau_i, i = 1, \ldots, N$, in the low-dimensional feature space based on the local coordinates $\theta_j^{(i)}$ which represents the local geometry. Specifically, we want τ_{i_j} to satisfy the following set of equations, i.e., the global coordinates should respect the local geometry determined by the $\theta_j^{(i)}$,

$$\tau_{i_j} = \bar{\tau}_i + L_i\theta_j^{(i)} + \epsilon_j^{(i)}, \quad j = 1, \ldots, k, \;\; i = 1, \ldots, N, \tag{4.8}$$

where $\bar{\tau}_i$ is the mean of τ_{i_j}'s, L_i is a local affine transformation matrix that needs to be determined,[3] and $\epsilon_j^{(i)}$ the local reconstruction error. Denoting $T_i = [\tau_{i_1}, \ldots, \tau_{i_k}]$

[1] We are being vague about the distributions of τ_i and ϵ_i. An asymptotic convergence analysis of our proposed algorithm in the style of [2] even in the isometric case will involve analysis of the interactions of the distributions in the feature space, the curvature of the manifold, and distributions in the input space.

[2] Discovering the local geometry using a homogeneous Euclidean metric as we do here is not necessarily the best approach: for one thing, k needs to be chosen adaptively to reflect the local geometry and sampling density; A case can also be made to select the neighbors that lie close to a linear subspace, see the discussion towards the end of Section 8.

[3] It seems to be more natural to require that L_i be orthogonal since Q_i in (4.7) is orthonormal. This restriction leads to a more complicated optimization problem which will be addressed in the Appendix.

and $E_i = [\epsilon_1^{(i)}, \cdots, \epsilon_k^{(i)}]$, we have

$$T_i = \frac{1}{k} T_i ee^T + L_i \Theta_i + E_i,$$

and the local reconstruction error matrix E_i has the form

$$E_i = T_i(I - \frac{1}{k} ee^T) - L_i \Theta_i. \tag{4.9}$$

To preserve as much of the *local* geometry in the low-dimensional feature space, we seek to find τ_i and L_i to minimize the reconstruction errors $\epsilon_j^{(i)}$, i.e.,

$$\sum_i \|E_i\|_2^2 \equiv \sum_i \|T_i(I - \frac{1}{k} ee^T) - L_i \Theta_i\|_2^2 = \min.^4 \tag{4.10}$$

Obviously, the optimal alignment matrix L_i that minimizes the local reconstruction error $\|E_i\|_F$ for a fixed T_i, is given by

$$L_i = T_i(I - \frac{1}{k} ee^T)\Theta_i^+ = T_i \Theta_i^+, \text{ and therefore } E_i = T_i(I - \frac{1}{k} ee^T)(I - \Theta_i^+ \Theta_i),$$

where Θ_i^+ is the Moore-Penrose generalized inverse of Θ_i.

Let $T = [\tau_1, \ldots, \tau_N]$ and S_i be the 0-1 selection matrix such that $TS_i = T_i$,. We then need to find T to minimize the overall reconstruction error

$$\sum_i \|E_i\|_F^2 = \|TSW\|_F^2,$$

where $S = [S_1, \cdots, S_N]$, and $W = \operatorname{diag}(W_1, \cdots, W_N)$ with

$$W_i = (I - \frac{1}{k} ee^T)(I - \Theta_i^+ \Theta_i). \tag{4.11}$$

To uniquely determine T, we will impose the constraints $TT^T = I_d$, it turns out that the vector e of all ones is an eigenvector of

$$B \equiv SWW^T S^T \tag{4.12}$$

corresponding to a zero eigenvalue. Therefore the optimal T is given by the d eigenvectors of B corresponding to the 2nd to d+1st smallest eigenvalues. We call the above algorithm, *Local Tangent Space Alignment* (LTSA).

Several numerical computation issues still need to be considered. We will present the relevant discussions and the details of our LTSA algorithm in section 7. As a preview of the effectiveness of LTSA, we plot in Figure 1 the computed coordinates τ_i vs. the centered arc-length coordinates τ_i^* for some 1D manifold parameterized by the arc-length, i.e., $x = f(\tau)$, τ is the arc-length of the 1D manifold. We also assume the data points are sampled without noise.

Remark. The minimization problem (4.10) needs certain constraints (i.e., normalization conditions) to be well-posed, otherwise, one can just choose both T_i and L_i to be zero. However, there are more than one way to impose the normalization

[4] If k is chosen adaptively and is dependent on i, i.e., $k = k_i$, it seems to be more appropriate to minimize the weighted error $\sum_i \frac{1}{k_i} \|E_i\|_2^2$.

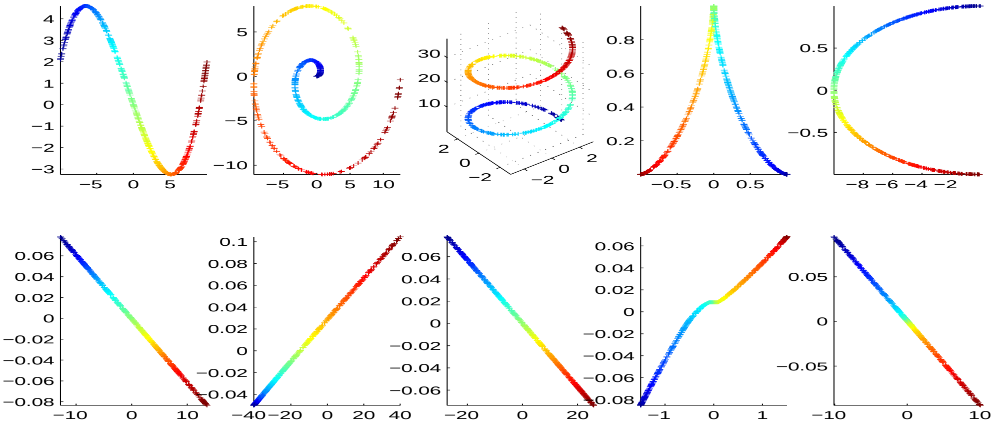


Fig. 1. *Sampled data points with no noise from various 1-D manifolds (top) and computed coordinates τ_i vs. centered arc-length coordinates τ_i^* (bottom).*

conditions. The one we have selected, i.e., $TT^T = I_d$, is just one of the possibilities. To illustrate the issue we look the following minimization problem,

$$\min_{X,\,Y} \|X - YA\|_F.$$

The approach we have taken amounts to substitute $Y = XA^+$, and minimize $\|X(I - A^+A)\|_F$ with the normalization condition $XX^T = I$. However,

$$\|X - YA\|_F = \left\| [X, Y] \begin{bmatrix} I \\ -A \end{bmatrix} \right\|_F,$$

and we can minimize the above by imposing the normalization condition $[X, Y][X, Y]^T = I$. This nonuniqueness issue is closely related to to nonuniqueness of the parameterization of the nonlinear manifold $f(\tau)$, which can be reparameterized as $f(\tau(\eta))$ with a 1-to-1 mapping $\tau(\eta)$.

Remark. We now briefly discuss the *nonlinear* alignment idea mentioned in (3.4). In particular, in a neighborhood $X_i = [x_{i_1}, \dots, x_{i_k}]$ of a data point x_i, by first-order Taylor expansion, we have

$$X_i(I - \frac{1}{k}ee^T) \approx J_f(x_i)T_i(I - \frac{1}{k}ee^T),$$

where $J_f(x_i)$ is the Jacobi matrix of f at x_i. Let S_i be the neighborhood selection matrix as defined before, we seek to find a sequence of $J_i \in \mathcal{R}^{m\times d}$ and $T \in \mathcal{R}^{d\times N}$ to minimize

$$E_X(J, T) \equiv \sum_{i=1}^{N} \|(X - J_iT)S_i(I - \frac{1}{k}ee^T)\|_F^2,$$

where $J = \{J_1, \dots, J_N\}$. Or similarly, we seek to find a sequence of $P_i \in \mathcal{R}^{d\times d}$ and $T \in \mathcal{R}^{d\times N}$ to minimize

$$E_\Theta(P, T) \equiv \sum_{i=1}^{N} \|(\Theta_i - P_iT_i)(I - \frac{1}{k}ee^T)\|_F^2.$$

($P = \{P_1, \ldots, P_N\}$.) The LTSA algorithm can be considered as an approach to find an approximate solution to the above minimization problem. We can, however, seek to find the optimal solution of $E_\Theta(P, T)$ using an *alternating* least squares approach: fix P minimize with respect to T, and fix T minimize with respect to P, and so on. As an initial value to start the alternating least squares, we can use the T obtained from the LTSA algorithm. The details of the algorithm will be presented in a separate paper.

5. Constructing Principal Manifolds. Once the global coordinates τ_i are computed for each of the data points x_i, we can apply some non-parametric regression methods such as local polynomial regression to $\{(\tau_i, x_i)\}_{i=1}^N$ to construct the principal manifold underlying the set of points x_i. Here each of the component functions $f_j(\tau), j = 1, \ldots, m$ can be constructed separately; for example, we have used the simple `loess` function [22] in some of our experiments for generating the principal manifolds.

In general, when the low-dimensional coordinates τ_i are available, we can construct a mapping from τ-space (feature space) to x-space (input space) as follows.

1. For each fixed τ, let τ_i be the nearest neighbor (i.e., $\|\tau - \tau_i\| \le \|\tau - \tau_j\|$, for $j \neq i$) Define

$$\theta = L_i^{-1}(\tau - \bar{\tau}_i),$$

where $\bar{\tau}_i$ be the mean of the feature vectors in a neighbor to which τ_i belong.

2. Back in the input space, we define

$$x = \bar{x}_i + Q_i\theta$$

Let us define by $g:\ \tau \rightarrow x$ the resulted mapping,

$$g(\tau) = \bar{x}_i + Q_i L_i^{-1}(\tau - \bar{\tau}_i). \tag{5.13}$$

To distinguish the computed coordinates τ_i from the generating ones, in the rest of this paper, we denote by τ_i^* the generating coordinate, i.e.,

$$x_i = f(\tau_i^*) + \epsilon_i^*. \tag{5.14}$$

Obviously, the errors of the reconstructed manifold represented by g depend on the sample errors ϵ_i^*, the tangent subspace reconstruction errors $\xi_j^{(i)}$, and the alignment errors $\epsilon_j^{(i)}$. The following result show that this dependence is linear.

THEOREM 5.1. *Let* $\epsilon_i^* = x_i - f(\tau_i^*)$, $\xi_j^{(i)} = (I - Q_i Q_i^T)(x_i - \bar{x}_i)$, *and* $\epsilon_i = \tau_i - \bar{\tau}_i - L_i Q_i^T(x_i - \bar{x}_i)$. *Then*

$$\|g(\tau_i) - f(\tau_i^*)\|_2 \le \|\epsilon_i^*\|_2 + \|\xi_i\|_2 + \|L_i^{-1}\epsilon_i\|_2.$$

Proof. Substituting $L_i^{-1}(\tau_i - \bar{\tau}_i) = L_i^{-1}\epsilon_i + Q_i^T(x_i - \bar{x}_i)$ into (5.13) gives

$$\begin{aligned} g(\tau_i) &= \bar{x}_i + Q_i L_i^{-1}(\tau_i - \bar{\tau}_i) \\ &= \bar{x}_i + Q_i Q_i^T(x_i - \bar{x}_i) + Q_i L_i^{-1}\epsilon_1^{(i)}. \end{aligned}$$

Because $Q_i Q_i^T(x_i - \bar{x}_i) = x_i - \bar{x}_i - \xi_j^{(i)}$, we obtain that

$$\begin{aligned} g(\tau_i) &= x_i - \xi_j^{(i)} + Q_i L_i^{-1}\epsilon_1^{(i)} \\ &= f(\tau_i^*) + \epsilon_i^* - \xi_j^{(i)} + Q_i L_i^{-1}\epsilon_1^{(i)}. \end{aligned}$$

Therefore we have

$$\|g(\tau_i) - f(\tau_i^*)\|_2 \le \|\epsilon_i^*\|_2 + \|\xi_i\|_2 + \|L_i^{-1}\epsilon_i\|_2,$$

completing the proof. □

In the next section, we will give a detail error analysis to estimate the errors of alignment and tangent subspace approximation in terms of the noise and the geometric properties of the generating function f and the density of the generating coordinates τ_i^*. Note that ϵ_i is the first column of E_i and ξ_i the first column of $(I - Q_iQ_i^T)X_i(I - \frac{1}{k}ee^T)$.

6. Error Analysis. As mentioned in the previous section, we assume that that the data points are generated by

$$x_i = f(\tau_i^*) + \epsilon_i^*, \quad i = 1, \dots, N.$$

For each x_i, let $X_i = [x_{i_1}, \dots, x_{i_k}]$ be a matrix consisting of its k-nearest neighbors including x_i in terms of the Euclidean distance. Similar to E_i defined in (4.9), we denote by E_i^* the corresponding local noise matrix, $E_i^* = [\epsilon_{i_1}^*, \dots, \epsilon_{i_k}^*]$. The low-dimensional embedding coordinate matrix computed by the LTSA algorithm is denoted by $T = [\tau_1, \dots, \tau_N]$. We first present a result that bounds $\|E_i\|$ in terms of $\|E_i^*\|$.

THEOREM 6.1. *Assume $T^* = [\tau_1^*, \dots, \tau_N^*]$ satisfies $(T^*)^T T^* = U_d$. Let $\bar{\tau}_i$ be the mean of $\tau_{i_1}, \cdots, \tau_{i_k}$, Denote $P_i = Q_i^T J_f(\bar{\tau}_i^*)$ and $H_{f_\ell}(\bar{\tau}_i^*)$ the Hessian matrix of the ℓ-th component function of f. If the P_i's are nonsingular, then*

$$\|E_i\|_F \le \|P_i^{-1}\|_F(\delta_i + \|E_i^*\|_F),$$

where δ_i is defined by

$$\delta_i^2 = \sum_{\ell=1}^{m}\sum_{j=1}^{k} \|H_{f_\ell}(\bar{\tau}_i^*)\|_2^2 \, \|\tau_{i_j}^* - \bar{\tau}_i^*\|_2^4$$

Furthermore, if each neighborhood is of size $O(\eta)$. Then $\|E\| \le \|P_i^{-1}\|_F\|E^*\| + O(\eta^2)$

Proof. First by definition (4.9), we have

$$E_i = T_i(I - \frac{1}{k}ee^T) - L_i\Theta_i = (T_i - L_iQ_i^T X_i)(I - \frac{1}{k}ee^T). \tag{6.15}$$

To represent X_i in terms of the Jacobi matrix of f, we assume that f is smooth enough and use Taylor expansion at $\bar{\tau}_i^*$, the mean of the k neighbors of τ_i^*,

$$x_{i_j} = f(\bar{\tau}_i^*) + J_i(\tau_{i_j}^* - \bar{\tau}_i^*) + \delta_j^{(i)} + \epsilon_{i_j},$$

where $J_i = J_f(\bar{\tau}_i^*)$ and $\delta_j^{(i)}$ represents the remainder term beyond the first-order expansion, in particular, its ℓ-th components can be approximately written as (using second order approximation),

$$\delta_{\ell,j}^{(i)} \approx \frac{1}{2}(\tau_{i_j}^* - \bar{\tau}_i^*)^T H_{f_\ell}(\bar{\tau}_i^*)(\tau_{i_j}^* - \bar{\tau}_i^*)$$

with the Hessian matrix $H_{f_\ell}(\bar{\tau}_i^*)$ of the ℓ-th component function f_ℓ of f at $\bar{\tau}_i^*$. We have in matrix form,

$$X_i = f(\bar{\tau}_i^*)e^T + J_iT_i^*(I - \frac{1}{k}ee^T) + \Delta_i + E_i^*$$

with $\Delta_i = [\delta_1^{(i)}, \cdots, \delta_k^{(i)}]$. Multiplying by the centering matrix $I - \frac{1}{k}ee^T$ gives

$$X_i(I - \frac{1}{k}ee^T) = (J_iT_i^* + \Delta_i + E_i^*)(I - \frac{1}{k}ee^T). \tag{6.16}$$

Substituting (6.16) into (6.15) and denoting $P_i = Q_i^T J_i$, we obtain that

$$E_i = (T_i - L_iP_iT_i^* - L_iQ_i^T(\Delta_i + E_i^*))(I - \frac{1}{k}ee^T) \tag{6.17}$$

For any $\tilde{T}$ satisfying the orthogonal condition $\tilde{T}\tilde{T}^T = I_d$ and any $\tilde{L}_i$, we also have the similar expression of (6.17) for $\tilde{T}_i$ and $\tilde{L}_i$. Note that T and L_i, $i = 1, \cdots, N$, minimize the overall reconstruction error, $\|E\|_F \le \|\tilde{E}\|_F$. Setting $\tilde{T} = T^*$ and $\tilde{L}_i = P_i^{-1}$, we obtain the upper bound

$$\|E_i\|_F \le \|P_i^{-1}\|_2(\|\Delta_i\|_F + \|E_i^*\|_F).$$

We estimate the norm $\|\Delta_i\|_F$ by ignoring the higher order terms, and obtain that

$$\|\Delta_i\|_F^2 \le \sum_{\ell=1}^{m}\sum_{j=1}^{k} \|H_{f_\ell}(\bar{\tau}_i^*)\|_2^2 \, \|\tau_{i_j}^* - \bar{\tau}_i^*\|_2^4 = \delta^2,$$

completing the proof. □

The non-singularity of the matrix P_i requires that the Jacobi matrix J_i be of full column rank and the two subspaces span(J_i) and the d largest left singular vector space span(Q_i) are not orthogonal to each other. We now give a quantitative measurement of the non-singularity of P_i.

Theorem 6.2. *Let $\sigma_d(\tilde{J}_i)$ be the d-th singular value of $\tilde{J}_i \equiv J_iT_i^*(I - \frac{1}{k}ee^T)$, and denote $\alpha_i = 4(\|E_i^*\|_F + \delta_i)$ with δ_i defined in Theorem 6.1. Then*

$$\|P_i^{-1}\|_F \le (1 + \alpha_i^2)^{1/2}\|J_i\|_F.$$

Proof. The proof is simple. Let $\tilde{J}_i = U_J\Sigma_JV_J^T$ be the SVD of the matrix $\tilde{J}_i$. By (6.16) and perturbation bounds for singular subspaces [10, Theorem 8.6.5], the singular vector matrix Q_i can be expressed as

$$Q_i = (U_J + U_J^{\perp}H)(I + H_i^TH_i)^{-1/2} \tag{6.18}$$

with

$$\|H_i\|_F \le \frac{4}{\sigma_d(\tilde{J}_i)}\Big(\|E^*\|_F + \|\Delta_i\|_F)\Big) \le \alpha_i,$$

where $\sigma_d(\tilde{J}_i)$ is the d-largest singular value of $\tilde{J}_i$ On the other hand, from the SVD of $\tilde{J}_i$, we have $J_iT_i^*V_J = U_J\Sigma_J$, which gives

$$J_i = U_J\Sigma_J\big(T_i^*V_J\big)^{-1}.$$

It follows that

$$P_i = Q_i^TJ_i = (I + H_i^TH_i)^{-1/2}\Sigma_J\big(T_i^*V_J\big)^{-1} = (I + H_i^TH_i)^{-1/2}U_J^TJ_i.$$

Therefore we have

$$\|P_i^{-1}\|_F \le (1 + \|H_i\|_F^2)^{1/2}\|J_i^+\|_F,$$

completing the proof. □

The degree of non-singularity of J_i is determined by the curvature of the manifold. On the other hand, the rotation of the singular subspace is mainly affected by the noise values ϵ_j and the neighborhood structure of x_i's The above error bounds clearly show that reconstruction accuracy might suffer if the manifold underlying the data set has singular or near-singular points. This phenomenon will be illustrated in the numerical examples in section 8. Finally, we give an error upper for the tangent subspace approximation.

THEOREM 6.3. *Let* $\mathrm{cond}(\tilde{J}_i) = \sigma_1(\tilde{J}_i)/\sigma_d(\tilde{J}_i)$ *be the spectrum condition number of the d-column matrix* $\tilde{J}_i$ *Then*

$$\|(I - Q_iQ_i^T)X(I - \frac{1}{k}ee^T)\|_F \le \Big(1 + 4(1+\alpha_i^2)\mathrm{cond}(\tilde{J}_i)\Big)\,\alpha_i/4.$$

Proof. By (6.16), we write

$$(I - Q_iQ_i^T)X(I - \frac{1}{k}ee^T) = (I - Q_iQ_i^T)\tilde{J}_i + \tilde{\Delta}_i,$$

with $\|\tilde{\Delta}_i\|_F \le \|E_i^*\|_F + \delta_i$ To estimate $\|(I - Q_iQ_i^T)\tilde{J}_i\|_F$, we use the expression (6.18) to obtain

$$\begin{aligned}(I - Q_iQ_i^T)\tilde{J}_i &= U\left(\begin{pmatrix} I \\ O \end{pmatrix} - \begin{pmatrix} I \\ H_i \end{pmatrix}(I + H_i^TH_i)^{-1}\right)\Sigma_J V_J^T \\ &= U\begin{pmatrix} H_i^T \\ -I \end{pmatrix} H_i(I + H_i^TH_i)^{-1}\Sigma_J V_J^T.\end{aligned}$$

Taking norms gives that

$$\|(I - Q_iQ_i^T)\tilde{J}_i\|_F \le (1 + \|H_i\|_2^2)\|H_i\|_F\|\tilde{J}_i\|_2 \le 4(1+\alpha_i^2)(\|E_i^*\|_F + \delta_i)\mathrm{cond}(\tilde{J}_i).$$

The result required follows. □

The above results show that the accurate determination of the tangent space is dependent on several factors: curvature information embedded in the Hessian matrices, local sampling density and noise level, and the regularity of the Jacobi matrix.

7. Numerical Computation Issues.

The major computational cost of LTSA involves the computation of the smallest eigenvectors of the symmetric positive semi-defined matrix B defined in (4.12). B in general will be quite sparse because of the local nature of the construction of the neighborhoods [18]. Algorithms for computing a subset of the eigenvectors for large and/or sparse matrices are based on computing projections of B onto a sequence of Krylov subspaces of the form

$$K_p(B, v_0) = \mathrm{span}\{v_0, Bv_0, B^2v_0, \ldots, B^{p-1}v_0\},$$

for some initial vectors v_0 [10]. Hence the computation of matrix-vector multiplications Bx need to be done efficiently. Because of the special nature of B, Bx can be computed neighborhood by neighborhood without explicitly forming B,

$$Bx = S_1W_1W_1^TS_1^Tx + \cdots + S_NW_NW_N^TS_N^Tx,$$

where as defined in (4.11),

$$W_i = (I - \frac{1}{k}ee^T)(I - \Theta_i^+\Theta_i).$$

Each term in the above summation only involves the x_i's in one neighborhood.

The matrix $\Theta_i^+\Theta_i$ in the right factor of W_i is the orthogonal projector onto the subspace spanned by the columns of Θ_i^T. It can be easily obtained by computing the d largest right singular vectors $g_1, \ldots, g_d$ of Θ_i [10]. By the definition (4.6) of Θ_i, $g_1, \ldots, g_d$ are also the d largest right singular vectors of $X(I - \frac{1}{k}ee^T)$, or equivalently the unit eigenvectors corresponding to the d largest eigenvalues of $(I - \frac{1}{k}ee^T)X^TX(I - \frac{1}{k}ee^T)$. It is easy to verify that with $G_i = [e/\sqrt{k}, g_1, \ldots, g_d]$

$$W_i = I - G_iG_i^T.$$

spanned by the rows of Θ_i and $e^T/\sqrt{k}$. Therefore the matrix-vector product $y = S_1W_1W_1^TS_1^Tx$ can be easily computed as follows: denote by $I_i = \{i_1, \cdots, i_k\}$ the set of indices for the k nearest-neighbors of x_i, then $y_j = 0$ for $j \notin I_i$ and

$$y(I_i) = T_i(I - G_iG_i^T)T_i^Tx(I_i).$$

Here $y(I_i) = [y_{i_1}, \cdots, y_{i_k}]^T$ denotes the section of y determined by the neighborhood index set I_i.

If one likes to compute the d smallest eigenvectors that are orthogonal to e by applying some eigen-solver, the matrix B should be constructed first. The matrix B can be computed by partially (locally) summing as following

$$B(I_i, I_i) \;\leftarrow\; B(I_i, I_i) + I - G_iG_i^T, \quad i = 1, \cdots, N \tag{7.19}$$

with initial $B = 0$. Clearly the computation cost is linear with respect to N.

Now we are ready to present our LTSA algorithm.

Algorithm LTSA (Local Tangent Space Alignment). Given N m-dimensional points sampled possibly with noise from an underlying d-dimensional manifold, this algorithm produces N d-dimensional coordinates $T \in \mathcal{R}^{d\times N}$ for the manifold constructed from k local nearest neighbors.

```
Step 1. [Extracting local information.] For each $i = 1, \cdots, N$,
        1.1 Determine $k$ nearest neighbors $x_{i_j}$ of $x_i$, $j = 1, \ldots, k$.
        1.2 Compute the $d$ largest unit eigenvectors $g_1, \ldots, g_d$ of the
            correlation matrix $(X_i - \bar{x}_ie^T)^T(X_i - \bar{x}_ie^T)$, and set
                $G_i = [e/\sqrt{k}, g_1, \ldots, g_d]$.
Step 2. [Constructing alignment matrix.] Form the matrix $B$ by locally
        summing (7.19) if a direct eigen-solver will be used. Otherwise
        implement a routine that computes matrix-vector multiplication
        $Bu$ for an arbitrary vector $u$.
Step 3. [Aligning global coordinates.] Compute the $d+1$ smallest eigen-
        vectors of $B$ and pick up the eigenvector matrix $[u_2, \cdots, u_{d+1}]$
        corresponding to the 2nd to $d+1$st smallest eigenvalues, and set
        $T = [u_2, \cdots, u_{d+1}]^T$.
```

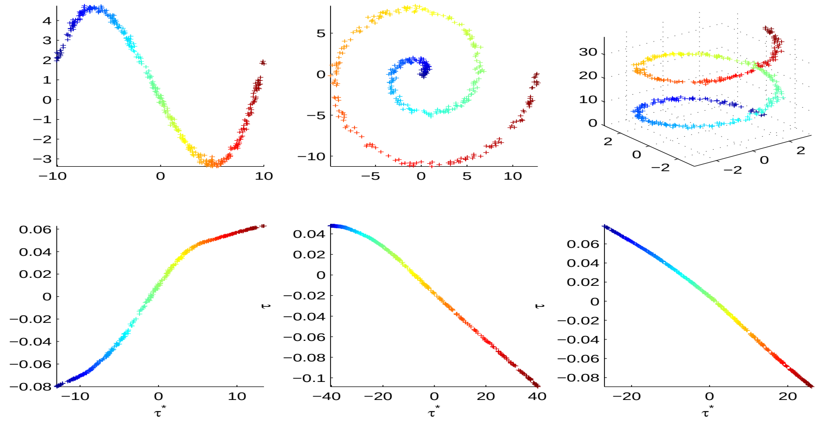


FIG. 2. *Sampled data points with noise from various 1-D manifolds (top) and computed coordinates τ_i vs. centered arc-length coordinates τ_i^* (bottom).*

REMARK. We should point out that the first step of the algorithm involves k-nearest-neighbor search which can be very expensive especially when N is large. Spatial indexing structures such as K-d trees have been used for efficient computation of the neighbors [3]. There are also several methods proposed for computing approximate nearest neighbors, see [4] for a review.

8. Experimental Results.

In this section, we present several numerical examples to illustrate the performance of the LTSA algorithm. The test data sets include curves in 2D/3D Euclidean spaces and surfaces in 3D Euclidean spaces. Especially, we take a closer look at the effects of singular points of a manifold and the interaction of noise levels and sample density. To show that our algorithm can also handle data points in high-dimensional spaces, we also consider curves and surfaces in Euclidean spaces with dimension equal to 100 and a face image data set For some of the benchmark data sets in [17], we also compare the projection results of LTSA and LLE.

First we test LTSA for 1D manifolds (curves) in both 2D and 3D. Each set of sample points is generated from a function $x = g(t)$ as follows: A uniformly sampled coordinates $t_1, \cdots, t_N$ in a fixed interval, say $[\alpha, \beta]$, are generated. Then we reparameterize the manifold as $x = f(\tau)$ with the centered arc length $\tau = \tau(t)$ defined as,

$$\tau(t) = \int_{t_0}^{t} \|J_g(t)\|_2 dt$$

for a constant $t_0 \in [\alpha, \beta]$ such that $\tau(\beta) = -\tau(\alpha)$, here $J_g(t)$ is the Jacobian of g at the point t. We add Gaussian noise to obtain the data set $\{x_i\}$ as,

$$x_i = g(t_i) + \eta\,\mathtt{randn}(m, 1),$$

where $m = 2, 3$ is the dimension of the input space, `randn` is Matlab's standard Gaussian distribution, and η controls the level of noise. Denoting $\tau_i^* = \tau(t_i)$, the sample points can be represented as

$$x_i = f(\tau_i^*) + \eta\,\mathtt{randn}(m, 1).$$

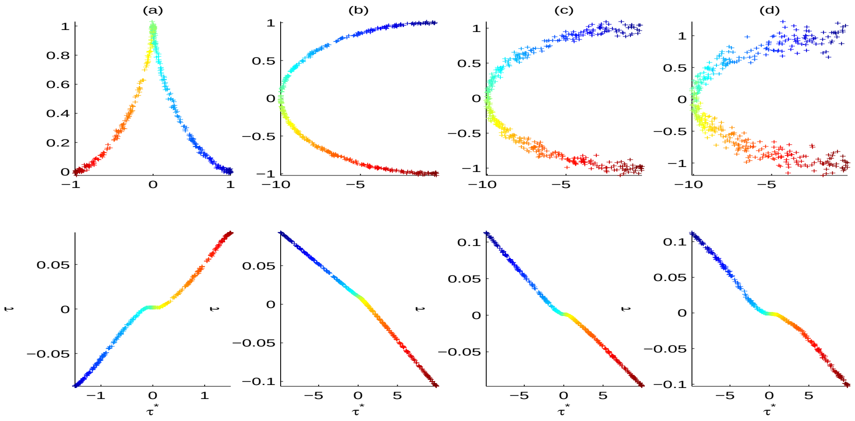


FIG. 3. *1-D manifolds with singular points (top) and computed coordinates τ_i vs. centered arc-length coordinates τ_i^* (bottom).*

Notice that, $\{\tau_i^*\}$ is not necessarily uniformly distributed in the corresponding τ-interval.

In the top row of Figure 2 from left to right, we plot the color-coded sample data of $N = 400$ points corresponding to the following three generating functions,

$$\begin{array}{rclll} g(t) & = & (10t,\ 10t^3+2t^2-10t)^T, & t\in[-1,\ 1], & \eta=0.1,\\ g(t) & = & (t\cos(t),\ t\sin(t))^T, & t\in[0,\ 4\pi], & \eta=0.2,\\ g(t) & = & (3\cos(t),\ 3\sin(t),\ 3t)^T, & t\in[0,\ 4\pi], & \eta=0.1. \end{array}$$

In the bottom row, we plot the centered arc length τ_i^* vs the computed τ_i by LTSA with $k = 15$ for each function. Ideally, the (τ_i^*, τ_i) should form a straight line LTSA recovers the true arc-length coordinates.

As we have shown in the error analysis in section 6, it will be difficult to align the locale tangent information Θ_i if some of the P_i's defined in Section 3 are close to be singular. One effect of this is that the computed coordinates τ_i and its neighbors may be compressed together. To clearly demonstrate this phenomenon, we consider the following generating function,

$$g(t) = [\cos^3(t),\ \sin^3(t)]^T, \quad t\in[0,\ \pi].$$

The Jacobi matrix (now a single vector since $d = 1$) given by

$$J_g(t) = 1.5\sin(2t)[-\cos(t),\ \sin(t)]^T$$

is equal to zero at $t = \pi/2$. Note that for the arc-length-parameterized form $f(\tau)$, if $t \neq \pi/2$,

$$J_f(\tau(t)) = J_g(t)/\|J_g(t)\|_2 = \mathrm{sign}(\sin(2t))[-\cos(t),\ \sin(t)]^T.$$

Obviously near the singular point $t = \pi/2$, it is difficult to approximate the tangent space in the presence of noise. In that case the θ-vector Θ_i defined in (4.6) will be computed poorly. Usually the corresponding Θ_i will be small which also results in small τ_i and the neighbors of τ_i will also be small. In the first column of Figure 3, we plot the computed results for this 1D curve. We see clearly near $t = \pi/2$ the

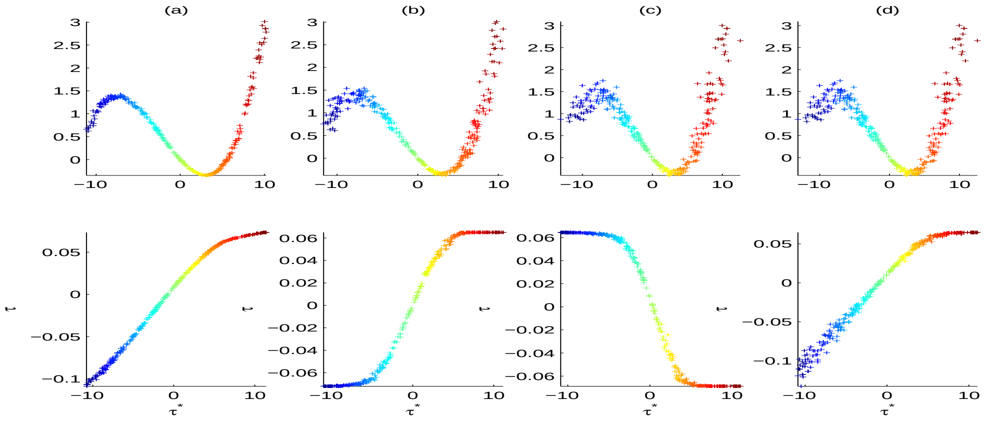


FIG. 4. *1-D manifolds with different noise levels (top) and computed coordinates τ_i vs. centered arc-length τ_i^* (bottom).*

computed τ_i's become very small, all compressed to a small interval around zero. In the right three columns of Figure 3, we examine another 1D curve generated by

$$g(t) = [10\cos(t),\ \sin(t)]^T, \quad t \in [\pi/2,\ 3\pi/2]$$

with different noise levels η,

$$x_i = g(t_i) + \eta\epsilon_i,$$

where ϵ_i are standard Gaussian, and $\eta = 0.01, 0.05, 0.1$, respectively. ($N = 400$, $k = 15$ for both examples.) We notice that similar phenomenon also occurs near the point $t = \pi$ where the *curvature* of the curve is relatively large, the computed τ_i's also become very small and cluster around zero, especially in the presence of large noise

Next we look at the issues of the interaction of sampling density and noise levels. If there are large noises around $f(\tau_i)$ relative to the sampling density near $f(\tau_i)$, the resulting centered local data matrix $X_i(I - \frac{1}{k}ee^T)$ will not be able to provide a good tangent space approximation, i.e, $X_i(I - \frac{1}{k}ee^T)$ will have singular values σ_d and σ_{d+1} that are close to each other. This will result in a nearly singular matrix $P_i = Q_i^T J_i$, and when plotting τ_i^* against τ_i, we will see the phenomenon of the computed coordinates τ_i getting compressed, similar to the case when the generating function $g(\tau)$ has singular and/or near-singular points. However, in this case, the result can usually be improved by increasing the number of neighbors used for producing the shifted matrix $X_i(I - \frac{1}{k}ee^T)$. In Figure 4, we plot the computed results for the generating function

$$g(t) = [10t,\, 3t^3 + 2t^2 - 2t]^T, \quad t \in [-1.1,\, 1].$$

The data set is generated by adding noise in a multiplicative fashion,

$$x_i = g(t_i)(1 + \eta\epsilon_i)$$

with standard Gaussian ϵ_i. The first three columns in Figure 4 correspond to the noise levels $\eta = 0.01$, $\eta = 0.03$, and $\eta = 0.05$, respectively. For the three data sets, we use the same number of neighbors, $k = 10$. The quality of the computed τ_i's can

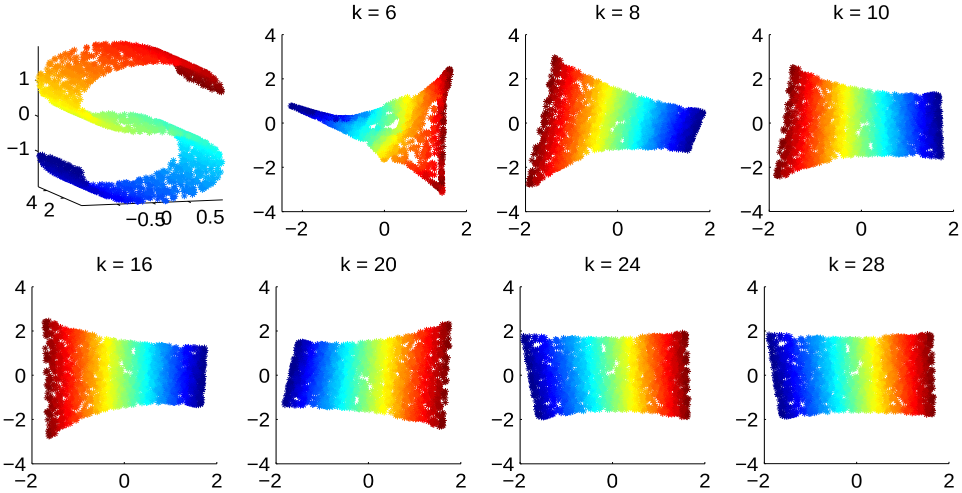


FIG. 5. *S-curve (top left) and computed 2D coordinates by LLE with various neighborhood size k.*

be improved if we increase the number of neighbors as shown on the column (d) in Figure 4 with $k = 20$ used.

As we have shown in Figure 4 (column (d)), different neighborhood size k will produce different embedding results. In general, k should be chosen to match the sampling density, noise level and the curvature at each data points so as to extract an accurate tangent space. Too few neighbors used may result a rank-deficient tangent space and leads to over-fitting, while too large a neighborhood will introduce too much bias and the computed tangent space will not match the local geometry well. It is therefore worthy of considering variable number of neighbors that are adaptively chosen at each data point. Fortunately, LTSA seems to be less sensitive to the choice of k than LLE does as will be demonstrated below.

We applied both LTSA and LLE to the S-curve data set uniformly sampled without noise generated as follows,

```
t = (3*rand(1,N)-1)*pi;
s = 5*rand(1,N);
X = [cos(t); s; (sin(t)-1).*sign(pi/2-t)];
```

with $N = 2000$ and different number of neighbors. For $d = 2$, and k which is chosen from $k = 6$ to $k = 30$. There are little geometric deformations in the coordinates generated by LTSA, see Figure 6. In Figure 5, we plot the results for LLE, the deformations (stretching and compression) in the generated coordinates are quite prominent. Similar results are plotted for the swissroll data set [21],

```
t = (3*pi/2)*(1+2*rand(1,N));
s = 21*rand(1,N);
X = [t.*cos(t); s; t.*sin(t)];
```

with $N = 2000$ in Figure 7 (LLE) and Figure 8 (LTSA). Both of these two surfaces have zero Gaussian curvature, and therefore they can be flattened without any

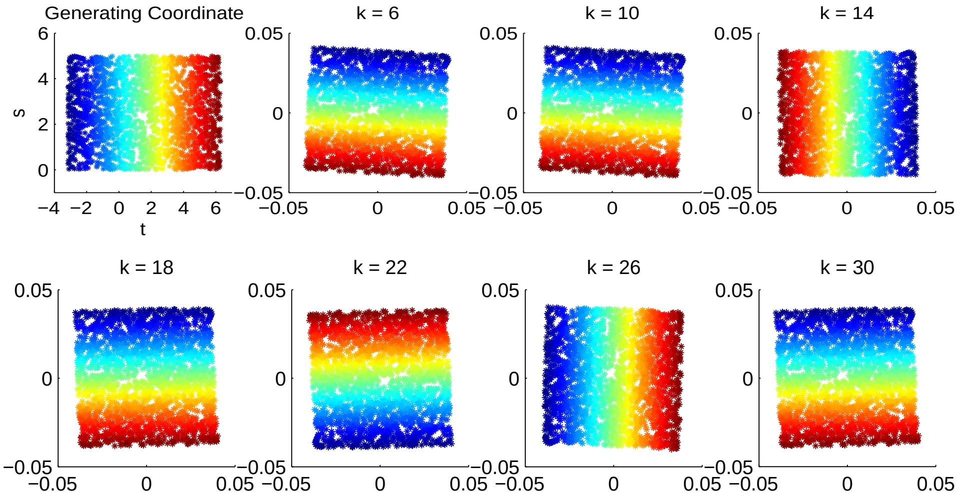


Fig. 6. *Generating coordinates of the S-curve (left) and computed coordinates by LTSA with various neighborhood size k.*

geometric deformation, i.e., the two surfaces are *isometric* to a 2D plane.

We now apply LTSA to a 2D manifold embedded in a 100 dimensional space. The data points are generated as follows. First we generate $N = 5000$ 3D points,

$$y_i = (t_i,\, s_i,\, h(t_i, s_i))^T + 0.01\eta_i$$

with t_i and s_i uniformly distributed in the interval $[-1,\ 1]$, the η_i's are standard Gaussian. The $h(t,s)$ is a peak function defined by

$$h(t,s) = 0.3(1-t)^2 e^{-t^2-(s+1)^2} - (0.2t - t^3 - s^5)e^{-t^2-s^2} - 0.1e^{-(t+1)^2-s^2}.$$

This function is plotted in the left of Figure 9. Then we linearly transform the 3D points to 100D points; two kinds of such 100D data points x_i^Q and x_i^H are generated as follows.

$$x_i^Q = Qy_i, \quad x_i^H = Hy_i,$$

where $Q \in \mathcal{R}^{100\times 3}$ is a random orthonormal matrix resulting in an orthogonal transformation and $H \in \mathcal{R}^{100\times 3}$ a matrix with its singular values uniformly distributed in $(0,\,1)$ resulting in an affine transformation. Figure 9 plots the coordinates for x_i^Q (middle) and x_i^H (right).

For estimating the dimension of the manifold, we consider examining the distribution of the singular values of the data matrix X_i consisting of the data points in the neighborhood of each data point x_i. (The reader is also referred to [6, 12, 18] for alternative approaches). If the manifold is of dimension d, then X_i will be close to a rank-d matrix. We illustrate this point below. The data points are x_i^Q of the 2D peak manifold embedded into the 100D space. For each local data matrix X_i, let $\sigma_{j,i}$ be the j-the singular value of the centered matrix $X_i(I - \frac{1}{k}ee^T)$. Define the ratios

$$\rho_i^{(j)} = \frac{\sigma_{j+1,i}}{\sigma_{j,i}}.$$

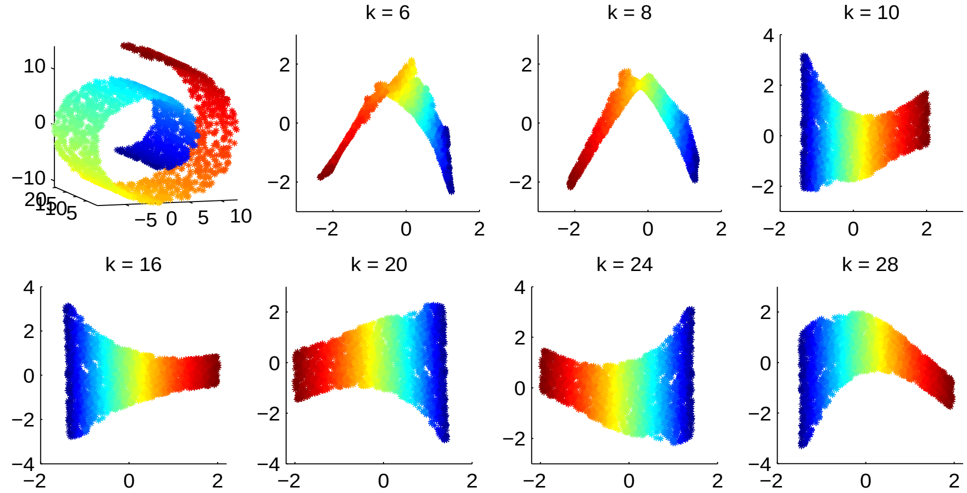


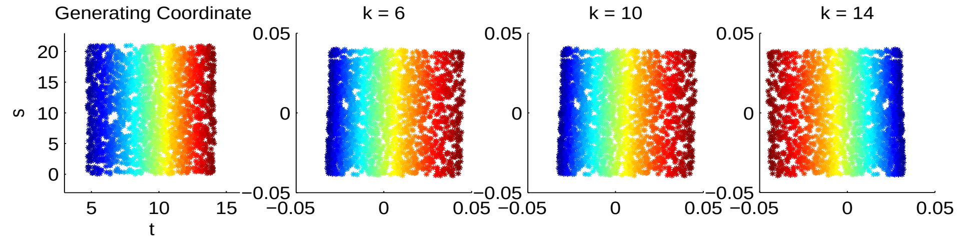
Fig. 7. *Computed 2D coordinates of the swissroll by LLE with various neighborhood size k.*

In Figure 10, we plot the ratios $\rho_i^{(1)}$ and $\rho_i^{(2)}$. It clearly shows the feature space should be 2-dimensional.

Next, we discuss the issue of how to use the global coordinates τ_i's as a means for clustering the data points x_i's. The situation is illustrated by Figure 11. The data set consists of three bivariate Gaussians with covariance matrices $0.2I_2$ and mean vectors located at $[1, 1], [1, -1], [-1, 0]$. There are 100 sample points from each Gaussian. The thick curve on the right panel represents the principal curve computed by LTSA and the thin curve by LLE. It is seen that the thick curve goes through each of the Gaussians in turn, and the corresponding global coordinates (plotted in the middle panel) clearly separate the three Gaussians. LLE did not perform as well, mixing two of the Gaussians.

The selection of the set of points to estimate the tangent space is very crucial to the success of the algorithm. Ideally, we want this set of points to be close to the tangent space. However, with noise and/or at the points where the curvature of the manifold is large, this is not an easy task. One line of ideas is to do some preprocessing of the data points to construct some *restricted* local neighborhoods. For example, one can first compute the minimum Euclidean spanning tree for the data set, and restrict the neighbors of each point to those that are linked by the branches of the spanning tree. This idea has been applied in self-organizing map [13]. Another idea is to use iterative-refinement, combining the computed τ_i's with the x_i's for neighborhood construction in another round of nonlinear projection. The rationale is that τ_i's as the computed global coordinates of the nonlinear manifold may give a better measure of the local geometry. An example using iterative-refinement is shown in Figure 12, the data points are sampled from the left half of a very flat ellipse (the long axis is the x-axis), one iterative-refinement gave a much better result.

Finally, we look at the results of applying LTSA to the face image data set [21]. The data set consists of a sequence of 698 64-by-64 pixel images of a face rendered under various pose and lighting conditions. Each image is converted to an $m = 4096$ dimensional image vector. We apply LTSA with $k = 12$ neighbors and $d = 2$. The constructed coordinates are plotted in the middle of Figure 13 We



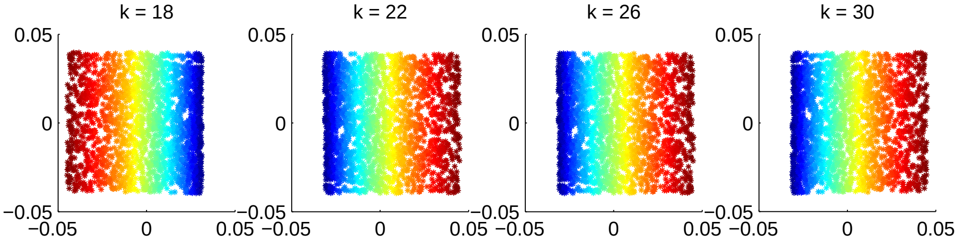


FIG. 8. *Computed coordinates of the swissroll by LTSA with various neighborhood size k.*

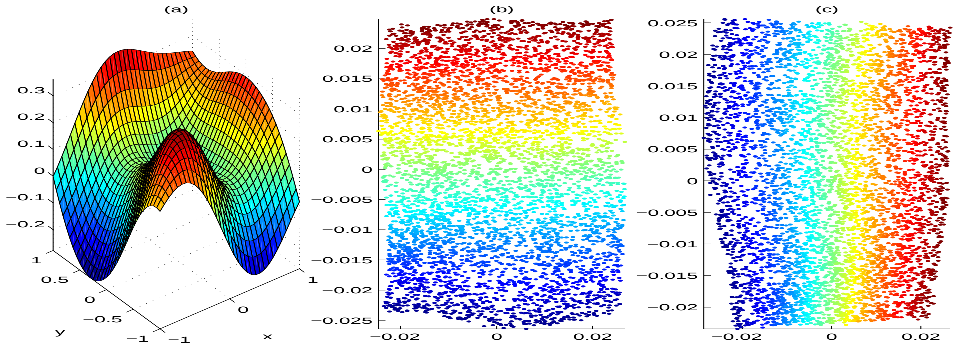


FIG. 9. *2D manifold in a 100D space generated by 3D peak function: (a) 3D peak curve, (b) coordinates for orthogonal transformed manifold, (c) coordinates for affine transformed manifold.*

also extracted four paths along the boundaries of the set of the 2D coordinates, and display the corresponding images along each path. It can be seen that the computed 2D coordinates do capture very well the pose and lighting variations in a continuous way.

9. Conclusions and Further Work. In this paper, we proposed a new algorithm (LTSA) for nonlinear manifold learning and nonlinear dimensionality reduction. The key techniques we used are the construction of approximations of tangent spaces to represent local geometry of the manifold and the global alignment of the tangent spaces to obtain the global coordinate system. We provide some error analysis to exhibit the interplay of approximation accuracy, sampling density, noise level and curvature structure of the manifold. In the following, we list several issues that deserve further investigation.

1. To better understand the properties of algorithms such as LTSA (and similarly LEE and Isomap) we need to investigate the issue of *optimal embedding*, i.e., dimensionality reduction will result in certain amount of geometric distortion, but we seek

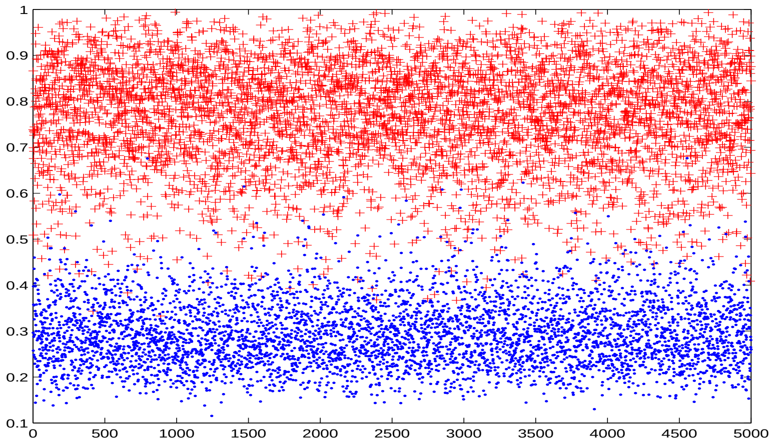


FIG. 10. *Singular value ratios* $\rho_i^{(1)}$ *(+-dots) and* $\rho_i^{(2)}$ *(·-dots).*

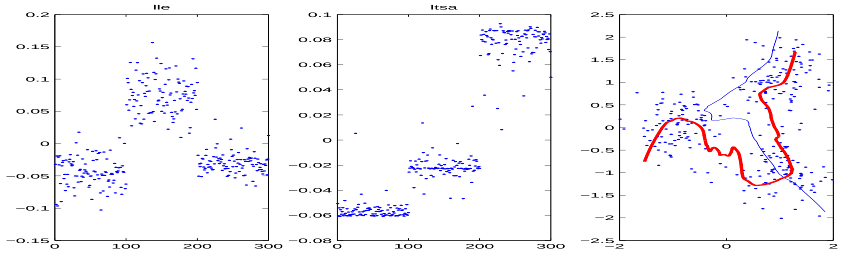


FIG. 11. *(left) Global coordinates by LLE, (middle) global coordinates by LTSA, (right) Three Gaussian data with principal curves*

to minimize this distortion under certain criterion. This line of ideas will explore the nonlinear alignment approach we discussed in section 3.

2. To make LTSA (similarly LLE) more robust against noise, we need to resolve cases where several of the smallest eigenvalues of B defined by (4.12) are roughly the same magnitude. This problem can be clearly seen when the manifold itself consists of several disjoint components. If this is the case, one needs to apply LTSA to each of the disjoint components. That is equivalent to break the matrix B into several diagonal blocks and compute several smallest eigenvectors of each block. However, with noise, the situation becomes far more complicated; several eigenvectors corresponding to near-zero eigenvalues can mix together. The information for the global coordinates seems to be contained in the eigen-subspace, but how to unscramble the eigenvectors to extract the global coordinate information needs more careful analysis of the eigenvector matrix of B and various models of the noise. Some preliminary results on this problem have been presented in [16]. Another approach, as was pointed out by one of the referees, is to partition the k-nearest-neighbor graph into several pieces, and then apply LTSA on each piece. This in some sense is equivalent to finding a block diagonal approximation of the matrix B which can be computed using a spectral clustering approach such as [19, 25].

3. We also plan to investigate the case where the manifold can be self-intersecting and the case where several manifolds possibly intersect each other. In the first case, we need to build several tangent spaces at some sample points. In the second case,

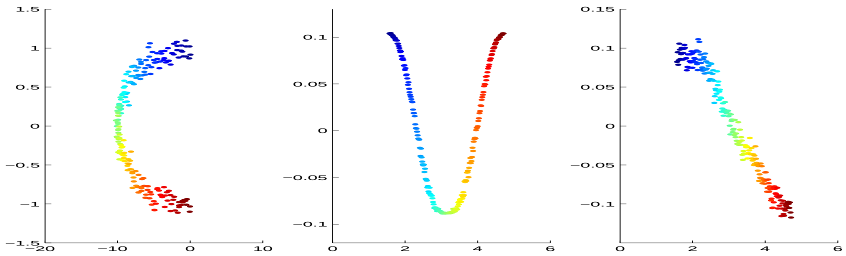


Fig. 12. *(left) Half ellipse data set, (middle) the global coordinates for the half ellipse data set, (right) the global coordinates after one iterative-refinement*

we need to find a way to group the samples to each of the manifold they belong. This problem can be formulated as a mixture of manifolds model, and an EM-style algorithm coupled with LTSA can be used for estimation.

4. From a statistical point of view, it is also of great interest to investigate more precise formulation of the error model and the associated consistency issues and convergence rate as the sample size goes to infinity. The learnability of the nonlinear manifold also depends on the sampling density of the data points. Some of the results in non-parametric regression and statistical learning theory will be helpful to pursue this line of research.

5. In section 8, we briefly looked at the issue of estimating the dimension d of the manifold using the singular value distribution of the k-nearest-neighbor set. More sophisticated algorithms have been proposed: Kegl considered an approach using packing numbers [12], and Costa and Hero considered asymptotically consistent estimates of d using geodesic minimal spanning trees [6]. Certainly more research is needed for this problem

Acknowledgement. We thank the referees for many thoughtful comments and suggestions which greatly improve the quality and presentation of the paper. In particular, suggestions and discussions from one referee regarding the connection between the Hessian-LLE in [8] and LTSA leads to the derivation of Theorem 3.1.

REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimension reduction and data representation. *Neural Computation*, 15:1373-1396., 2003.

[2] M. Berstein, V de Silva, J. Langford and J. Tenenbaum. Graph approximations to geodesics on embedded manifolds. `http://isomap.stanford.edu/BdSLT.pdf`, 2000.

[3] J. Bentley and J. Friedman. Data structures for range searching. *ACM Computing Surveys*, 11:397-409, 1979.

[4] S Berrani, L Amsaleg and P. Gros. Approximate k-nearest-neighbor searches: a new algorithm with probabilistic control of the precision. Technical Report, RR-4675, INRIA, 2002.

[5] M. Brand. Charting a manifold. *Advances in Neural Information Processing Systems* 15, MIT Press, 2003.

[6] J. Costa and A. Hero. Manifold learning with geodesic minimal spanning trees. Submitted to *IEEE Transactions on Signal Processing*, 2003.

[7] D. Donoho and C. Grimes. Image Manifolds which are Isometric to Euclidean Space. To appear in *Journal Mathematical Imaging and Vision.*

[8] D. Donoho and C. Grimes. Hessian Eigenmaps: new tools for nonlinear dimensionality reduction. *Proceedings of National Academy of Science*, 5591-5596, 2003.

[9] D. Freedman. Efficient simplicial reconstructions of manifolds from their samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1349 -1357, 2002.

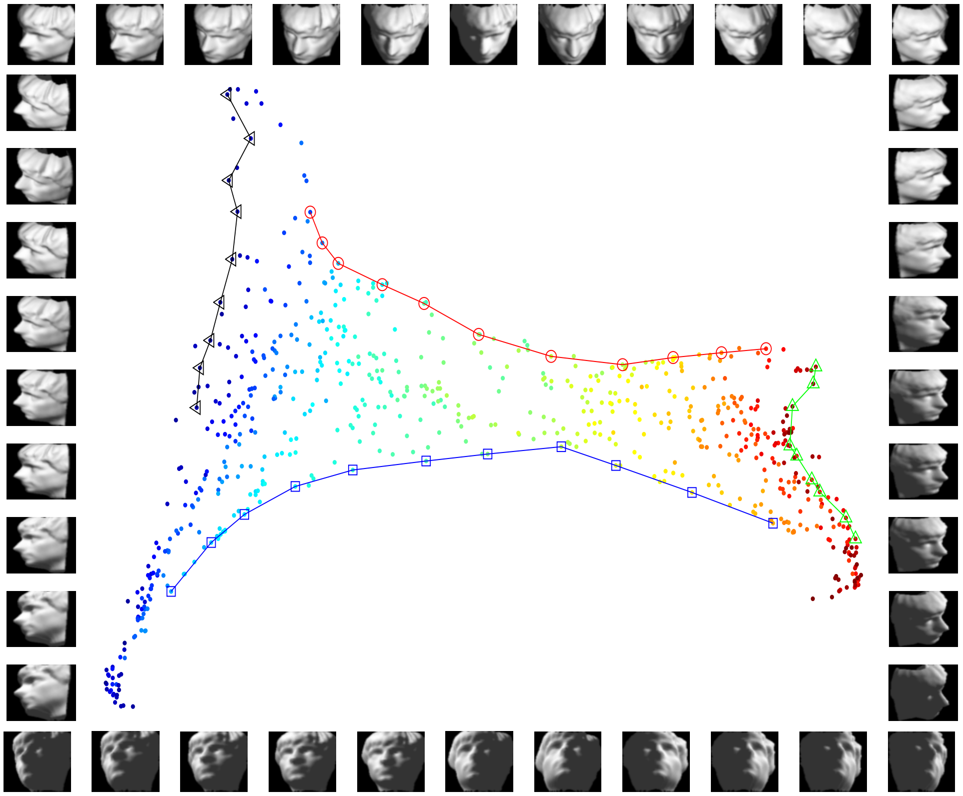


FIG. 13. *Coordinates computed by Algorithm LTSA with $k = 12$ neighbors (middle) and images corresponding to the points on the bound lines (top, bottom,left, and right)*

[10] G. H. Golub and C. F Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, 3nd edition, 1996.

[11] T. Hastie and W Stuetzle. Principal curves. *J. Am. Statistical Assoc.*, 84: 502–516, 1988.

[12] B. Kegl. Intrinsic dimension estimation using packing numbers. *Advances in Neural Information Processing Systems*, 15, MIT Press, 2003.

[13] T. Kohonen. *Self-organizing Maps.* Springer-Verlag, 3rd Edition, 2000.

[14] T Martinetz and K. Schulten. Topology representing networks *Neural Networks*, 7: 507–523, 1994.

[15] J. Munkres. *Analysis on Manifold* Addison Wesley, Redwood City, CA, 1990.

[16] M. Polito and P. Perona. Grouping and Dimensionality reduction by Locally Linear Embedding. *Advances in Neural Information Processing Systems* 14, MIT Press 2002.

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

[18] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of nonlinear manifolds. *Journal of Machine Learning Research*, 4:119-155, 2003.

[19] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:888-905, 2000.

[20] Y. Teh and S. Roweis Automatic Alignment of Hidden Representations. *Advances in Neural Information Processing Systems*, 15, MIT Press, 2003.

[21] J. Tenenbaum, V De Silva and J. Langford A global geometric framework for nonlinear dimension reduction. *Science*, 290:2319–2323, 2000

[22] WN. Venables and B.D. Ripley. Modern Applied Statistics with S-plus. Springer-verlag, 1999

[23] J Verbeek, N. Vlassis and B. Krose. Procrustes analysis to coordinate mixtures of probabilistic principal component analyzers Technical report, Computer Science Institute, University of Amsterdam, The Netherlands, IAS-UVA-02-01, 2002.

[24] J. Verbeek, N. Vlassis and B. Krose. Coordinating principal component analyzers. *ICANN 2002*: 914-919, 2002.

[25] H. Zha, C. Ding, M. Gu, X. He and H. Simon. Spectral Relaxation for K-means Clustering. *Advances in Neural Information Processing Systems*, 14, MIT Press, 2002.

A. Alignment Preserving Orthogonality. Notice that the coordinates in the Θ_i's in (4.6) are obtained with respect to an orthonormal basis, therefore it seems quite natural to preserve this orthogonality in the low-dimensional feature space as well. To this end, we consider *orthogonal* transformation for alignment, i.e., we restrict the matrices L_i's in (4.8) to be orthogonal,

$$\tau_{i_j} = \bar{\tau}_i + G_i\theta_j^{(i)} + \epsilon_j^{(i)}, \quad G_i^T G_i = I_d, \tag{1.20}$$

and minimize the alignment error

$$\|E\|_F^2 \equiv \min_{T,\,\{G_i\}} \sum_i \|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2. \tag{1.21}$$

Note that here we do not impose the orthogonality constraints on T. Clearly for any fixed $T = T^*$, we have

$$\|E\|_F^2 \le \min_{\{G_i\}} \sum_i \|T_i^*(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2,$$

giving an upper bound for the minimum. To obtain a tighter upper bound on $\|E\|_F^2$, we need the following lemma [10].

Lemma A.1. *Let $AB^T = U\Sigma V^T$ be the SVD of AB^T with*

$$\Sigma = \operatorname{diag}(\sigma_1(AB^T), \cdots, \sigma_d(AB^T)).$$

Then the optimal orthogonal matrix G that minimizes $\|A - GB\|_F$ is given by $G = UV^T$. Furthermore,

$$\min_{G^TG=I} \|A - GB\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\sum_j \sigma_j(AB^T).$$

We now define $\hat{T}_i^* \equiv T_i^*(I - \frac{1}{k}ee^T)$. Similar to (6.17), we have

$$\hat{T}_i^* - G_i\Theta_i = \hat{T}_i^* - G_iP_i\hat{T}_i^* - Q_i^TE_i^*(I - \frac{1}{k}ee^T) + O(\|\hat{T}_i^*\|^2). \tag{1.22}$$

By LemmaA.1 and the inequalities $\sigma_i(AB) \ge \sigma_i(A)/\sigma_{\min}(B)$, we have

$$\begin{aligned}\min_{G_i^TG_i=I_d} \|\hat{T}_i^* - G_iP_i\hat{T}_i^*\|_F^2 &\le \|\hat{T}_i^*\|_F^2 + \|P_i\hat{T}_i^*\|_F^2 - 2\sum_j \sigma_j(\hat{T}_i^*(P_i\hat{T}_i^*)^T)\\ &\le \big(1 + \sigma_{\max}^2(P_i) - 2\sigma_{\min}^2(P_i)\big)\|\hat{T}_i^*\|_F^2.\end{aligned}$$

Therefore we obtain the following upper bound

$$\|E\|_F \le \sum_i \big(1 + \sigma_{\max}^2(P_i) - 2\sigma_{\min}^2(P_i)\big)\|\hat{T}_i^*\|_F^2 + O\Big(\big(\sum_i \|\hat{T}_i^*\|_F^4\big)^{1/2}\Big).$$

The optimization problem (1.21) can be solved iteratively alternating between the following two steps.

1. For fixed $T_i, i = 1, \ldots, N$, minimize $\|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|$ to obtain an optimal G_i. By Lemma A.1, G_i is given by $G_i = U_iV_i^T$, where $T_i\Theta_i^T = U_i\Sigma_iV_i^T$.
2. For fixed $G_i, i = 1, \ldots, N$, solve the optimization problem

$$\min_T \sum_i \|T_i(I - \frac{1}{k}ee^T) - G_i\Theta_i\|_F^2$$

to obtain a new T_i. This is a LS problem.

It converges monotonically.