Chapter

A Survey of Manifold-Based Learning Methods

Xiaoming Huo, Xuelei (Sherry) Ni, and Andrew K. Smith
School of Industrial Engineering, Georgia Institute of Technology

Abstract: We review the ideas, algorithms, and numerical performance of manifold-based machine learning and dimension reduction methods. The representative methods include locally linear embedding (LLE), ISOMAP, Laplacian eigenmaps, Hessian eigenmaps, local tangent space alignment (LTSA), and charting. We describe the insights from these developments, as well as new opportunities for both researchers and practitioners. Potential applications in image and sensor data are illustrated. This chapter is based on an invited survey presentation that was delivered by Huo at the 2004 INFORMS Annual Meeting, which was held in Denver, CO, USA.

***Key words**:* Manifold, statistical learning, nonparametric methods, dimension reduction

1. Introduction

Manifold-based learning is an emerging and promising approach in nonparametric dimension reduction. In this article, we review the state-of-the-art mathematical developments, as well as some interesting applications.

A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in $\Re^n$). A good example of a manifold is the Earth (Figure 1-1). Locally, at each point on the surface of the Earth, we have a 3-D coordinate system: two for location and the last one for the altitude. Globally, it is a 2-D sphere in a 3-D space.

Manifolds offer a powerful framework for dimension reduction. The key idea of dimension reduction is to find the most succinct low dimensional structure that is embedded in a higher dimensional space. Historically, Occam's razor has been used to justify dimension reduction. The key idea of Occam’s razor is to choose the simplest model from a set of equivalent models to explain a given phenomenon. It is easy to see that a manifold gives a dimension reduction. Moreover, if the data are indeed generated according to a manifold, then a manifold-based learning is, in some sense, optimal.

This article is organized as follows. Section 2 surveys existing methods, including principal components analysis (PCA), multidimensional scaling (MDS), generative topological mapping (GTM), locally linear embedding (LLE), ISOMAP, Laplacian eigenmaps, Hessian eigenmaps, and local tangent space alignment (LTSA). Section 3 stresses an important common point among some recent methods: their numerical solutions are based on searching for null spaces under certain situations. We choose LLE and LTSA as our illustrative examples. Such a common point is likely to be the key to unifying the theoretical analysis of many manifold-based methods. Section 4 presents some desirable performance properties of a learning method. Some preliminary thoughts in problem formulations and properties are described. For example, we establish the consistency of LTSA in Section 4.3.2. Section 5 gives some examples and potential applications, including examples of feature extraction in Section 5.1, an example of clustering in Section 5.2, a potential application in image detection in Section 5.3, and an application in sensor localization in Section 5.4. We provide some final thoughts on the future of the field in Section 6. Some additional useful resources are described in the Appendix.

Figure 1-1. An example of a manifold.

Relation to enterprise data mining (DM). This chapter does not directly address the DM in enterprise database. However, it provides powerful nonlinear dimension reduction methods, which are essentially useful in enterprise DM. One possible link is as follows (which is pointed out by an anonymous referee). Sensors are often used to monitor process in a manufacturing enterprise. To inspect the product quality, images of the product are often captured and then processed to detect flaws. The image detection technique in Section 5.3 can potentially be applied. A second possible link is through the object recognition in enterprise. Manifold-based dimension reduction has potential to be applied there. The sensor location problem that is described in Section 5.4 is another potential application in enterprise.

A generic `prescription?' This chapter provides a comprehensive survey on existing manifold learning methods. For readers who are looking for a quick (and possibly dirty) solution, we suggest to experiment with local tangent space alignment (LTSA), which in our experience gives the most satisfactory performance in many cases. There are numerous software packages, which realize LTSA and are available freely on the internet. We refer to the URLs given at the end of this chapter. Scientifically speaking, each problem has to be analyzed before one can decide which method is optimal. Keeping this in mind, one should only take the above as a suggestion (not a rule) – there are always situations under which a method outperforms every other method, as reflected in the following detailed survey.

2. Survey of Existing Methods

We organize our presentation of methodologies into five groups.

Group 1: classical methods, including principal component analysis (PCA). We mention other methods that are related, such as factor analysis and other techniques in multivariate analysis.

Group 2: semi-classical methods, including multidimensional scaling (MDS), as described in Kruskal (1964) and Borg and Groenen (1997).

Group 3: manifold searching methods, including generative topographic mapping (GTM), referring to Bishop, Svensen, and Williams (1998), local linear embedding (LLE), referring to Roweis and Saul (2000), and ISOMAP, referring to Tenenbaum, de Silva, and Langford (2000).

Group 4: methods rooted in continuum spectral theory, including the Laplacian eigenmaps (Belkin and Niyogi, 2001) and Hessian eigenmaps (Donoho and Grimes, 2003), which are based on elegant theory in spectral analysis, and then discretize the results in the continuum to generate numerical approaches.

Group 5: advanced manifold methods, including charting (Brand, 2003) and local tangent space alignment (Zhang and Zha, 2004). These methods are based on global alignment. The key insight in these methods is the realization that the global alignment can be achieved via an eigenvalue computation.

Each group is described in its own subsection below.

2.1. Group 1: Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is one of the most classical methods in dimension reduction. PCA is also known as the Karhunen-Loève transform, or singular value decomposition (SVD). The key idea of PCA is to find the low-dimensional linear subspace which captures the maximum proportion of the variation within the data.

PCA considers the second order statistics of a random vector $\mathbf{X} \in \Re^n$. Let X_1, X_2, …, X_N denote N samples from such a random vector. Let Ω denote the variance-covariance matrix of the random vector $\mathbf{X}$, i.e., $\mathrm{Var}(\mathbf{X}) = \mathrm{E}\left\{[\mathbf{X} - \mathrm{E}(\mathbf{X})][\mathbf{X} - \mathrm{E}(\mathbf{X})]^{\mathrm{T}}\right\} = \Omega$. Assume the symmetric and positive-semidefinite matrix Ω has the following eigen-decomposition:

$$\Omega = UDU^T ,$$

where $U \in \Re^{n \times n}$ is an orthogonal matrix ($U^T U = I_n$), and D is a diagonal matrix,

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}.$$

The diagonal entries of D, $0 \le \lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$, are the ordered eigenvalues of Ω. The columns of U, $U = [U_1, U_2, ..., U_n]$, are the associated eigenvectors. From the following matrix computation, we can see that λ_1, λ_2, …, and λ_k are the variances of the transformed random variables $U_1{}^T\mathbf{X}$, $U_2{}^T\mathbf{X}$, …, and $U_k{}^T\mathbf{X}$:

$$\begin{aligned}\mathrm{Cov}\left([U_1^T\mathbf{X}, U_2^T\mathbf{X}, ..., U_n^T\mathbf{X}]\right) &= \mathrm{Cov}(U^T\mathbf{X}) \\ &= U^T\,\mathrm{Cov}(\mathbf{X})U \\ &= \mathrm{D}.\end{aligned}$$

It is possible to prove that the projection $X \to [U_1, ..., U_k]^T X$ from $\Re^n$ to $\Re^k$ ($k<n$) keeps the greatest possible proportion of the variation in the data. If only the samples are available, the variance-covariance matrix can be estimated as

$$\sum_{i=1}^{N}(X_i - \overline{X})(X_i - \overline{X})^T \text{, where } \overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \,.$$

PCA gives a natural dimension reduction. Consider an extreme case: if all the data lie in a low-dimensional linear subspace of a very high dimensional space, then PCA will find such a linear subspace, because the variations in the directions that are orthogonal to the embedded linear subspace will be equal to zero. An evident disadvantage of PCA is that the embedded subspace has to be linear. For example, if the data are located on a circle in 3-D, PCA will not be able to identify such a structure.

Mathematically speaking, PCA is a problem of finding the largest eigenvalues. We will demonstrate later that many algorithms ultimately lead to a matrix problem that is associated with eigenvalues, including MDS, LLE, Laplacian eigenmaps, and LTSA (Sections 2.2.1, 3.1, and 3.2).

2.2. Group 2: Semi-Classical Method: Multidimensional Scaling (MDS)

MDS is the name of a group of methods that have found a wide range of applications. The key idea is to find a mapping from a high-dimensional space to a low-dimensional space, such that the pairwise distances between the observed points are preserved the best. An intuitive example is to recover the relative positions of cities from the inter-city distances. Imagine that the exact locations (coordinates) of N cities are lost. However, we have the driving distances between pairs of them. These distances form an $N \times N$ matrix. Based on this matrix, MDS can recover a 2-D coordinate system that includes the locations of theses cities, subject to a rigid motion (a combination of rotation, shifting, and reflection), such that the distances among the points on this 2-D plane are close to the driving distances among those cities.

The above in fact gives an example of metric MDS (Torgerson, 1952; Young and Householder, 1938), which is related to nonmetric MDS (Kruskal, 1964; Shepard, 1962) that will be explained later.

For metric MDS, consider some points X_i in a metric space Ω, $X_i \in \Omega$. For $1 \le l \ne m \le N$, let $d(l,m)$ denote the distance between X_l and X_m. We want to find $X_i^{'} \in \Re^k$, $i = 1,2,...,N$, with k a fixed integer, such that the following optimization problem is solved:

$$\min_{X_i^{'} \in \Re^k} \sum_{l \ne m} \left[d(l,m) - d'(l,m) \right]^2 ,$$

where $d'(l,m)$ denotes the distance between $X_l^{'}$ and $X_m^{'}$ in $\Re^k$.

In metric MDS, the numerical values of the inter-distances are to be preserved. Sometimes it makes more sense to preserve the order of these distances. It is even possible that the available distances are ordinal data. In order to map $X_i \in \Omega$ to $X_i^{'} \in \Re^k$, in the case of ordinal data, the following optimization problem is adopted,

$$\min_{X_i^{'}; f} \frac{\sum_{l \ne m} [f(d(l,m)) - d'(l,m)]^2}{\sum_{l \ne m} [d'(l,m)]^2} ,$$

where f is a monotone increasing function. For any fixed set of X_i's, the f is specified. The technical details can be found in Kruskal (1964) and Shepard (1962).

MDS is a very useful tool when the inter-point distances need to be preserved. In most existing MDS algorithms, a linear subspace is still the ultimate result. In ISOMAP, which is a method that will be described later, MDS is applied to geodesic distances, which results in a nonlinear dimension reduction method. We will give more details in Section 2.3.3.

2.2.1. Solving MDS as an Eigenvalue Problem

We present an eigenvalue-based approach to solving the MDS problem approximately. Consider observations X_1, X_2, …, $X_N \in \Re^D$, where N and D are two positive integers. Let X=[X_1, X_2, …, X_N]. Without loss of generality, we assume that the X_i's are centered at the origin, i.e., $X \cdot \mathbf{1}_N^T = O_D$, where $\mathbf{1}_N^T$ is the N-dimensional vector made by all ones, while O_D is the D-dimensional vector made by all zeroes. It is easy to see that

$$d^2(l,m) = \|X_l\|_2^2 + \|X_m\|_2^2 - 2 < X_l, X_m >, \quad \forall\, l, m,$$

where $< X_l, X_m >$ denotes the inner project of two vectors. Let $B = \left(\|X_1\|_2^2, \|X_2\|_2^2, \ldots, \|X_N\|_2^2\right)^T \in \Re^{N\times 1}$. Denote $E = \left(d^2(l,m)\right)_{l,m} \in \Re^{N\times N}$. We have $E = B \cdot 1_N^T + 1_N \cdot B^T - 2X^T X$.

From the above, we can easily verify the following:

$$X^T X = -\frac{1}{2}\left(I - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\right)E\left(I - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\right),$$

where I is an $N \times N$ identity matrix.

To find low-dimensional Y_i, i=1, 2, …, N, $Y_i \in \Re^d$, d<D, such that the matrix $\left(\|Y_l - Y_m\|_2^2\right)_{l,m}$ is a close approximation to E, we can find $Y = [Y_1, \ldots, Y_N] \in \Re^{d\times N}$, such that $Y^T Y$ is close to $X^T X$. Note this approximately solves the original MDS problem, but not exactly. Suppose the eigen-decomposition of matrix $X^T X$ is

$$X^T X = \sum_{i=1}^{N} \lambda_i U_i U_i^T,$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$ are the eigenvalues of $X^T X$ and $U_1, U_2, ..., U_N \in \Re^N$ are the corresponding eigenvectors. We can assign $Y = \text{diag}\left(\sqrt{\lambda_1}, ..., \sqrt{\lambda_d}\right)\left[U_{1,} U_2, ..., U_d\right]^T$.

We can verify that $Y^T Y$ is the best approximation to $X^T X$.

2.3. Group 3: Manifold Searching Methods

In this group, we review generative topological mapping (GTM), locally linear embedding (LLE), and ISOMAP.

2.3.1. Generative Topological Mapping (GTM)

Generative topological mapping (GTM) is an inspiring nonlinear dimension reduction method. Compared to the methods that will be introduced later, GTM does not contain the same sophisticated numerical approaches. But its formulation highlights some key components in modern dimension reduction.

Let *x* be a point in a latent space and *t* be a point in the data space. Let $t_1, t_2, ..., t_N$ denote the observed points (realizations of *t*). Point t_i is generated according to the following:

- First of all, there is a quantity x_i associated with t_i in the latent space. Note that the x_i 's are not observable. The latent space has a much lower dimension than the data space does.
- There is a mapping, $x \rightarrow y(x, W)$, from the latent space to the data space. This mapping is continuously differentiable and has full column rank in its Jacobian. Notation *W* denotes the parameters of this mapping. In fact, one can assume that the

images *y(x,W)* for all *x* form a low-dimensional manifold in the data space.

- Suppose that the observation t_i is generated according to the model

  $$t_i = y(x_i;W) + \varepsilon_i, \quad i\text{=1, 2, …, N},$$

  where ε_i satisfies a multivariate normal distribution with zero mean and variance-covariance matrix β .

Thus, GTM assumes the existence of an implicit manifold. There are unknown parameters *W* and β . The latent variables x_i exist, but are also unknown.

By assuming a special distribution for the x_i 's and placing the problem in a Bayesian model estimation framework, the authors of GTM introduced an expectation-maximization (EM) based method to estimate the above model (Bishop, Svensen, and Williams, 1998). The dimension reduction is achieved by finding a maximum a posteriori (MAP) estimate.

GTM considers a prior *p(x)* for the x_i 's. This prior is a sum of a finite number of Dirac functions, i.e.,

$$p(x) = \sum_{i=1}^{k} \delta(x - \bar{x}_i),$$

where $\bar{x}_1$, $\bar{x}_2$, …, $\bar{x}_k$ are *k* given points in the latent space. According to the previous way of generating t_i, there is a probability density function for *t*: $p(t \mid x;W,\beta)$. The density function on the data space is simply

$$p(t \mid W,\beta) = \int p(t \mid x,W,\beta) p(x)\,\mathrm{d}\,x .$$

Given that *p(x)* is a sum of *k* Dirac functions, we have

$$p(t \mid W,\beta) = \sum_{i=1}^{k} p(t \mid \bar{x}_i,W,\beta) .$$

The principle of maximum likelihood estimation (MLE) is to find *W* and β such that the log-likelihood function,

$$\sum_{j=1}^{N} \ln p(t_j \mid W, \beta),$$

is maximized. The authors of GTM (Bishop, Svensen, and Williams, 1998) proposed an EM approach to estimate W and β. Here we omit some of the technical details regarding how to choose the functional classes in the nonlinear mapping.

The numerical solution of GTM is based on a strong assumption on the prior. The application of the EM algorithm seems *ad-hoc*. It is also hard to justify the performance of GTM. As a matter of fact, GTM can only be established in some special cases, like clustering, as an alternative to self-organizing map (SOM). However, the probabilistic model is consistent with other models in data analysis.

2.3.2. Locally Linear Embedding (LLE)

Locally linear embedding (LLE) and ISOMAP comprise a new generation of dimension reduction methods. They have been successfully applied to both synthetic and "real" data sets. We review the LLE in this section, and ISOMAP in the next.

Again, we consider a data space with a very high dimension *D*. Let $\vec{X}_i$, *i*=1, 2, …, *N*, be *N* vectors in such a data space. LLE starts with finding the *k* nearest neighbors (based on the Euclidean distance) for each vector $\vec{X}_i$, $1 \le i \le N$. Let N_i denote the indices of the *k* nearest neighbors of the vector $\vec{X}_i$. LLE finds the optimal local convex combinations of the *k*-nearest neighbors to represent each original vector. It is equivalent to minimizing the objective

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_{j \in N_i} W_{ij} \vec{X}_j \right|^2 ,$$

where $\sum_j W_{ij} = 1$. It can be shown that the above can be solved as a least-squares problem.

Next, LLE considers a projection space. A projection space plays a role similar to that of the latent space in GTM. Let $\vec{Y}_i$ be the projection of $\vec{X}_i$ in the projection space. The projection space has a dimension much smaller than *D*. The projections $\vec{Y}_i$ are chosen such that the following objective function is minimized:

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_{j \in N_i} W_{ij} \vec{Y}_j \right|^2 .$$

Note that the above is equivalent to finding a lower dimensional representation, such that the local convex representations are preserved. It can be shown that with some additional conditions, which make the problem well defined, the minimization task can be accomplished by solving a sparse $N \times N$ eigenvector problem. More specifically, the *d* eigenvectors associated with the *d* smallest non-zero eigenvalues provide an ordered set of orthogonal coordinates centered on the origin.

We summarize the LLE algorithm in Table 1. The LLE authors suggest that *k-b* trees can be used to compute the *k*-nearest neighbors efficiently (Friedman, Bentley, and Finkel, 1977). The sparse eigenvector problem can be solved by fast algorithms as well, e.g., Bai, Demmel, Dongarra, et al. (2000).

Table 1. LLE Algorithm.

LLE Algorithm
1. Compute the *k* nearest neighbors of each point $\vec{X}_i$.
2. Compute the weights W_{ij} of a convex combination of the *k* nearest neighbors that best represent the point $\vec{X}_i$.
3. Find a low-dimensional projection $\vec{Y}_i$ such that the above local representations are best preserved.

Note that unlike GTM, LLE does not have a probabilistic model imposed on the data. In fact, the authors of LLE predicted the integration of probabilistic models in their future research.

One disadvantage of LLE is that it implicitly assumes that the manifold is convex. The methods that will be described later can overcome such a disadvantage.

2.3.3. ISOMAP

ISOMAP is another nonlinear dimension reduction method. It can be viewed as an extension of metric MDS, by replacing the Euclidean distance with another type of distance.

ISOMAP works as follows. Consider N points, $\vec{X}_i$, i=1, 2, …, N, in the data space. First of all, for each data point $\vec{X}_i$, consider its neighbors. There are two possibilities:

- k-nearest neighbors of each point $\vec{X}_i$; or
- an ε-neighborhood, which includes all the points that are no more than ε-distance away from $\vec{X}_i$.

Let N_i denote the index set of the points that are the neighbors of $\vec{X}_i$. We construct a graph, in which each $\vec{X}_i$ is a vertex, and two vertices are connected if and only if $i \in N_j$ or $j \in N_i$. Define the distance between two points, $\vec{X}_i$ and $\vec{X}_j$, to be the sum of the arc lengths of the shortest chain connecting $\vec{X}_i$ and $\vec{X}_j$. The shortest chain can be computed via dynamic programming (e.g., Dijkstra, 1959). The above is called a graphical distance. The geodesic distance between two points on a manifold is the length of the shortest curve that is on the manifold and connects the two points. Bernstein, de Silva, Langford, and Tenenbaum (2000) show that the graphical distance is in some sense a good substitute for the geodesic distance. Note that a graphical distance is computable from data, while the geodesic distance is not computable. A low dimensional projection is then generated by calling a metric MDS.

2.4. Group 4: Methods from Spectral Theory

Both Laplacian eigenmaps (Belkin and Niyogi, 2001) and Hessian eigenmaps (Donoho and Grimes, 2004) are motivated by spectral theory in the continuum. The numerical approaches are discretizations of the continuum theory.

2.4.1. Laplacian Eigenmaps

Laplacian eigenmaps are proposed in Belkin and Niyogi (2001). This work establishes both a unified approach to dimension reduction and a new connection to spectral theory. Laplacian eigenmaps are the predecessor of the next method -- Hessian eigenmaps, which overcome the convexity limitation.

We first describe the Laplacian eigenmap for discrete data. Its relevant theorem in the continuum will follow. Again, we consider *N* points, $\vec{X}_i$, *i*=1, 2, …, *N*, in the *D*-dimensional data space. For each point $\vec{X}_i$, $1 \le i \le N$, suppose a neighbor set N_i is computed. A graph identical with the graph in ISOMAP can be defined. For any pair of connected points $\vec{X}_i$ and $\vec{X}_j$, we define a weight function

$$W_{ij} = \exp\left\{ -\frac{1}{t} \left\| \vec{X}_i - \vec{X}_j \right\|_2^2 \right\}.$$

Let *D* denote a diagonal matrix such that $D_{ii} = \sum_j W_{ji}$. Let *W* denote the symmetric matrix with entries W_{ij}, $1 \le i, j \le N$. Finally, let *L* denote the matrix *L*=*D*-*W*. Consider the solutions to the problem:

$$Lf = \lambda Df, \tag{2-1}$$

where $f \in \Re^N$. Let f_0, f_1, …, f_{k-1} be the solution vectors with corresponding eigenvalues $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{k-1}$; i.e.,

$$Lf_0 = \lambda_0 Df_0,$$
$$Lf_1 = \lambda_1 Df_1,$$
$$\vdots$$
$$Lf_{k-1} = \lambda_{k-1} Df_{k-1}.$$

The eigenvectors associated with zeros eigenvectors is left out and the next m eigenvectors are used for the embedding in an m-dimensional Euclidean space

$$\vec{X}_i \to (f_1(i), f_2(i), ..., f_m(i)).$$

An intuitive justification for solving the eigenvalue and eigenvector problem (2-1) is to consider minimizing the objective,

$$\sum_{i,j} (y_i - y_j)^2 W_{ij}, \qquad (2\text{-}2)$$

where $y = (y_1, y_2, ..., y_N)$ consists of N maps from a point to $\Re$. It is shown in Belkin and Niyogi (2001) that (2-2) is equivalent to finding

$$\begin{array}{ll} \text{argmin} & y^T Ly, \\ \text{subject to} & y^T Dy = 1. \end{array}$$

Minimizing the objective in (2-2) is equivalent to finding an optimal embedding. By generalizing it to an embedding in $\Re^m$, we have the described eigenvector and eigenvalue problem. We refer the reader to Belkin and Niyogi (2001) for the details.

The above approach uses the Laplacian of a graph, which is analogous to the Laplace Beltrami operator on manifolds. Chung (1997) serves as a good reference. Let M be a smooth, compact, m-dimensional Riemannian manifold. Let f be a map from the manifold to $\Re$. Assume that $f : M \to \Re$ is twice differentiable. Belkin and Niyogi (2001) explain how

$$\int_f L(f) f$$

serves as the weighted sum in (2-1). Suppose ∇f is the gradient of f and *L(f)* is the Laplace Beltrami operator. It is known that the $\hat{f}$, which minimizes $\int_M \|\nabla f\|^2$, is an eigenvector of the Laplace Beltrami operator.

The spectrum of L on a compact manifold M is known to be discrete. The rest of the dimension reduction is identical with the approach in the discrete case.

The connection between spectral theory and dimension reduction, which is established in Laplacian eigenmaps, is very inspiring.

2.4.2. Hessian Eigenmaps

In all the aforementioned methods, it is required that the embedded manifold is sampled on a convex region. Hessian eigenmaps, as proposed by Donoho and Grimes (2004), relax the convexity condition.

We explain the motivation of Hessian eigenmaps (HLLE) in the continuum. Recall that in Laplacian eigenmaps, the following functional $H_1(f)$ is considered:

$$H_1(f) = \int_M L(f) f \, .$$

In Hessian eigenmaps, the above functional is replaced with

$$H_2(f) = \int_M \left\| H_f(m) \right\|_F^2 \mathrm{d}m \, ,$$

where $H_f(m)$ is the Hessian of the function *f*. $\|\cdot\|_F^2$ denotes the square of the Frobenius norm of a matrix. Donoho and Grimes prove that by minimizing $H_2(f)$, the convexity condition in the previous approaches can be relaxed.

Donoho and Grimes (2004) then propose a discrete algorithm, which is based on a discrete approximation to the Hessian on a manifold.

2.5. Group 5: Methods Based on Global Alignment

We review the local tangent space alignment (LTSA) method that is proposed in Zhang and Zha (2004). There is another similar method, charting (Brand, 2003), which is not as well-developed mathematically.

The following derivation can be divided into two stages. In the first stage, a local parametrization is established for each data point. In the second stage, a global alignment is computed. Suppose that the *i*th observation is generated according to $x_i = f(\theta_i) + \varepsilon_i$, where θ_i is a natural parameter of x_i, and the ε_i's are random and i.i.d. Let $x_{i,j}$ denote the *j*th nearest neighbor of x_i. Similarly, we have $x_{i,j} = f(\theta_{i,j}) + \varepsilon_{i,j}$. We assume that $\theta_{i,j} \approx \theta_i$, because they are neighbors. Assume *f* is smooth enough so that

$$\begin{aligned} x_{i,j} - x_i &= f(\theta_{i,j}) - f(\theta_i) + \varepsilon_{i,j} - \varepsilon_i \\ &= \mathrm{g} f(\theta_i)(\theta_{i,j} - \theta_i) + \mathrm{O}(\left\|\theta_{i,j} - \theta_i\right\|^2) + \varepsilon_{i,j} - \varepsilon_i. \end{aligned}$$

Here $\mathrm{g} f(\theta_i)$ is the gradient of function *f* whose variable is θ_i. The above is merely a Taylor expansion. Let $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,k}] - x_i \mathbf{1}_k^T \in \Re^{D\times k}$, where $x_{i,1}, ..., x_{i,k}$ are the *k* nearest neighbors of x_i, $\mathbf{1}_k^T = (1,1,...,1)^T \in \Re^k$. Let $L_i = \mathrm{g} f(\theta_i) \in \Re^{D\times k}$. Let α_i, $\alpha_{i,1}$, …, $\alpha_{i,k}$ denote the temporary local parameterizations of observations x_i, $x_{i,1}$, …, $x_{i,k}$. Similarly, let $A_i = [\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,k}] - \alpha_i \mathbf{1}_k^T$. If the ε_i's satisfy a multivariate normal distribution with zero mean and constant variance, and if the second order term $\left\|\alpha_{i,j} - \alpha_i\right\|_2^2$ is negligible, the local parameterization and the tangent space can be computed by solving the following optimization problem:

$$\min_{L_i, A_i} \left\| X_i - L_i A_i \right\|_F^2 .$$

Note that in order to make the solution well defined, we impose the constraint $L_i^T L_i = I_d$. The above is solved via a singular value decomposition (SVD). L_i is made by the singular vectors that are associated with the d largest singular values of X_i. A_i is also computable, and is the only quantity that will be conveyed to the next stage.

In the second stage, a global parameterization that is locally identical to A_i up to a rigid transform is computed. Let

$\Theta_i = [\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,k}] - \theta_i \mathbf{1}_k^T$.

Let $T_i \in \Re^{d \times d}$ be an orthogonal matrix. We solve

$$\min_{\text{all } \theta_i\text{'s}, T_i\text{'s}} \sum_{i=1}^{N} \left\| \Theta_i - T_i A_i \right\|_F^2 .$$

By following a derivation in Zhang and Zha (2004), it is possible to show that the problem eventually becomes that of finding the *2*nd to the *(k+1)*st smallest eigenvalues and eigenvectors of an $N \times N$ matrix. Due to space limitations, the specific form of this matrix is omitted.

3. Unification via Null-Space Methods

We have presented a large set of methods, all having the flavor of finding the embedded geometric structure, i.e., a manifold. Different methods are based on different ideas. It seems like each method should be analyzed individually in order to determine its performance. In this section, we will demonstrate that many of them eventually become null-space searching algorithms. (Recall that null-spaces are spanned by the solutions of a system of linear equations corresponding to a predetermined matrix.) Hence, if we can characterize the behavior of null-spaces under uncertainty, we can provide a unified analysis of these methods. We show that LLE and LTSA are null space-based methods in Section 3.1 and 3.2, respectively. We describe the matrices that are used in these methods as a way to compare them on a common ground.

3.1. LLE as a Null-space-based Method

The content of this subsection extends the description in Section 2.3.2. Recall that LLE contains two steps. In the first step, a linear representation of each observation (point) based on its *k*-nearest-neighbors is computed. The second step computes a low-dimensional representation that best preserves these local linear representations.

The first step is achieved by solving the following problem:

$$\min_{\substack{\omega\in\Re^k \\ \omega^T\mathbf{1}_k=1}} \left\|X_i - M_i\omega\right\|_2^2,$$

where $X_i \in \Re^D$, $i = 1,2,...,N$, are the observed points, $M_i =$ $[X_{i1}, X_{i2}, \ldots, X_{ik}]$ is formed by taking the *k* nearest neighbors of X_i as its columns, and $\mathbf{1}_k \in \Re^k$ is an all one vector.. It is shown in an online introduction of LLE (Saul and Roweis, 2001) that the above is equivalent to solving

$$\min_{\omega^T\mathbf{1}_k=1} \omega^T\left(X_i\mathbf{1}_k^T - M_i\right)^T\left(X_i\mathbf{1}_k^T - M_i\right)\omega .$$

Let $\Omega_i = \left(X_i\mathbf{1}_k^T - M_i\right)^T\left(X_i\mathbf{1}_k^T - M_i\right)$. Using a Lagrange multiplier approach, one can show that

$$\omega_i = {\Omega_i^{-1}\mathbf{1}_k} \big/ {\mathbf{1}_k^T\Omega_i^{-1}\mathbf{1}_k},$$

provided that Ω_i is invertible.

As demonstrated in the original LLE paper, the second step can be achieved by solving

$$\min_{\substack{Y\in\Re^{d\times N} \\ YY^T=I_d}} \sum_i \left\|Y_i - N_i\omega_i\right\|_2^2, \qquad (3\text{-}1)$$

where *d*<*D*, $Y = [Y_1, Y_2, ..., Y_N]$, matrix $N_i = [Y_{i1}, Y_{i2}, ..., Y_{ik}]$, which is made by *k* Y_i's that correspond to the *k* nearest neighbors of X_i. I_d is the *d*-by-*d* identity matrix. The above objective function can be rewritten as

$$\text{obj(LLE)} = \sum_i \left\| Y(e_i - S_i\omega_i) \right\|_2^2 ,$$

where e_i is an N-dimensional column vector taking one at the ith position and zeros elsewhere, S_i is the selection matrix associated with the k nearest neighbors of X_i, and ω_i is computed in the first step. Moreover, we have

$$\text{obj(LLE)} = \sum_i (e_i - S_i\omega_i)^T Y^T Y (e_i - S_i\omega_i) .$$

Minimizing the above objective function with the constraints in (3-1) is equivalent to finding the eigenvectors associated with the *2*nd to the *(d+1)*st smallest eigenvalues of the matrix

$$\begin{aligned}\text{M(LLE)} &= \sum_i (e_i - S_i\omega_i)(e_i - S_i\omega_i)^T \\ &= I_N - \sum_i S_i\omega_i e_i^T - \sum_i e_i \omega_i^T S_i^T + \sum_i S_i\omega_i\omega_i^T S_i^T .\end{aligned}$$

Let

$$\begin{aligned}W &= [S_1\omega_1, S_2\omega_2, \ldots, S_N\omega_N] \\ &= [S_1\, S_2 \cdots S_N]_{N\times kN} \begin{bmatrix} \omega_1 & & & \\ & \omega_2 & & \\ & & \ddots & \\ & & & \omega_N \end{bmatrix}_{kN\times N} .\end{aligned}$$

We can simplify M(LLE) as

$$M(LLE) = (I_N - W)(I_N - W)^T .$$

Note that M(LLE) is an $N \times N$ symmetric matrix. Because $\mathbf{1}_k^T \omega_i = 1$, $\forall i$, it is evident that the all one vector $\mathbf{1}_N$ belongs to the null space of matrix M(LLE). The choice of the second to the *(d+1)*st smallest eigenvalues is to exclude such a special case.

3.2. LTSA as a Null-space-based Method

We review LTSA, emphasizing that LTSA is another null-space method, and compare it with LLE. Recall LTSA includes two steps: local parameterization and global alignment.

In the local parameterization step, the following is solved.

$$\min_{\substack{\Theta_i \in \Re^{d\times k} \\ Q^T Q = I_d}} \left\| X_i \overline{P} - Q\Theta_i \right\|_2^2 ,$$

where $X_i \in \Re^{D\times k}$ is a matrix whose columns are the k nearest neighbors of the ith point including the ith point, $\overline{P} = (I_k - \mathbf{1}_k \mathbf{1}_k^T / k)$, which is a projection matrix projecting $\Re^k$ to a k-1 dimensional linear subspace that is orthogonal to the all one vector $\mathbf{1}_k \in \Re^k$, $Q \in \Re^{D\times d}$ satisfies $Q^T Q = I_d$, and we assume $d<$ min(D, k). Let $X_i \overline{P} = \sum_i \lambda_i u_i v_i^T$ be the singular value decomposition of matrix $X_i \overline{P}$, where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_{\min(D,k)} \ge 0$, column vectors $u_i \in \Re^D$ are the left singular vectors, and column vectors $v_i \in \Re^k$ are the right singular vectors. Zhang and Zha (2004) demonstrate that the solutions are $Q = [u_1, u_2, \ldots, u_d]$ and

$$\begin{aligned} \Theta_i &= Q^T X_i \overline{P} \\ &= \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d) \begin{bmatrix} v_1^T \\ \vdots \\ v_d^T \end{bmatrix}. \end{aligned} \qquad (3\text{-}2)$$

In the global alignment, Zhang and Zha (2004) show that the optimal low-dimensional representation is given by the eigenvectors associated with the d+1 smallest eigenvalues of the matrix

$$\mathrm{M(LTSA)} = SWW^T S^T ,$$

excluding the zero eigenvalue associated with a constant-valued eigenvector. A detailed explanation can be found in Zhang and Zha

(2004). Here $S=[S_1,S_2,...,S_N]$, where S_i is a selection matrix associated with X_i that is defined in the foregoing subsection (Section 3.1). Moreover,

$W=\mathrm{diag}(W_1,W_2,...,W_n)$,

where $W_i=\overline{P}(I_k-\Theta_i^+\Theta_i)$, and Θ_i^+ is the generalized inverse of matrix Θ_i.

Recalling (3-2), we have

$$W_i=\overline{P}\left(I_k-[v_1,\cdots,v_d]\begin{bmatrix}v_1^T\\ \vdots\\ v_d^T\end{bmatrix}\right).$$

Letting $P_i=W_iW_i^T$, we have

$$P_i=\overline{P}\left(I_k-[v_1,\cdots,v_d]\begin{bmatrix}v_1^T\\ \vdots\\ v_d^T\end{bmatrix}\right)\overline{P},$$

which is a projection matrix that projects to a $\min(D,k)-d-1$ dimensional subspace of $\Re^k$. The subspace is spanned by the right singular vectors of $X_i\overline{P}$ associated with the $\min(D,k)-d$ smallest singular values and is orthogonal to vector $\mathbf{1}_k$. It is easy to see that

$$\mathrm{M(LTSA)}=SBB^TS^T, \qquad (3\text{-}3)$$

where $B=\mathrm{diag}(P_1,P_2,...,P_N)$.

Once again, LTSA is a null-space problem.

3.3. Comparison between LTSA and LLE

Recall M(LLE)=$(I-W)(I-W)^T$, which is formally different from M(LTSA). If we want to write M(LLE) in a format that is similar to the expression of M(LTSA), we can take

$$I_n = [S_1, S_2, \ldots, S_N] \begin{bmatrix} S_1^T \\ \vdots \\ S_N^T \end{bmatrix} \mathrm{diag}(c_1^{-1}, c_2^{-1}, \ldots, c_N^{-1}),$$

where c_i is the number of times that point $\vec{X}_i$ is included in a *k* nearest neighbor set. One can verify that

$$\mathrm{M(LLE)} = STT^{\mathrm{T}}S^T,$$

where

$$T = \begin{bmatrix} S_1^T \\ \vdots \\ S_N^T \end{bmatrix} \mathrm{diag}(c_1^{-1}, c_2^{-1}, \ldots, c_N^{-1}) - \begin{bmatrix} \omega_1 & & & \\ & \omega_2 & & \\ & & \ddots & \\ & & & \omega_N \end{bmatrix}.$$

Comparing with (3-3), we find that TT^T is no longer a block diagonal matrix. Such a difference between LTSA and LLE may lead to different performance. The detailed analysis is left as a future research topic.

4. Principles Guiding the Methodological Developments

4.1. Sufficient Dimension Reduction

We review the general principle of dimension reduction. We start with the concept of *sufficiency* in classical mathematical statistics. Let $x \in \Re^D$ denote an observation. Imagine another quantity $\theta \in \Re^d$, which is an implicit (simpler) representation of *x*. For example, θ could be a parameter in classical mathematical statistics. Let $p(x, \theta)$ denote their joint distribution. The parameter θ can be thought as the meaningful part of *x*. If there exists a function of *x*, denoted as $\phi(x)$, such that

$p(x,\theta) = p_1(\phi(x),\theta) \cdot p_2(x)$, then $\phi(x)$ is a sufficient statistic of θ. Here $p_1(\cdot)$ and $p_2(\cdot)$ are two functions. We assume that θ resides on one (or a few) simple manifold(s), and $p_1(\phi(x),\theta)$ is approximately $p_3(\theta)$, a distribution of θ, if and only if $\phi(x)$ is close to θ. It is easy to see that when the previous factorization holds, the conditional probability $p(x \mid \phi(x))$ does not depend on θ. We say that $\phi(x)$ is an *ideal* dimension reduction of *x*. The idealness is based on the fact that this data description takes the simplest possible form.

The above describes an abstract principle. A lot of specifications are needed to make it concrete. There are many existing works in dimension reduction, both for supervised learning (Globerson and Tishby, 2003; Fukumizu et al., 2004) and unsupervised learning. We described an unsupervised learning framework. We will describe a manifold-based dimension reduction framework with assumptions on the conditional distribution of $x \mid \phi(x)$.

4.2. Desired Statistical Properties

There are more criteria that are commonly adopted in evaluating the fundamental performance of dimension reduction algorithms. Note that nearly all of them take an asymptotic perspective (i.e., assuming the sample size *n* goes to ∞).

4.2.1. Consistency

For any estimate, the first requirement typically is statistical consistency. In our case, assume that each time course x_i is a combination of a structural component $f(\tau_i)$ and i.i.d. random errors ε_i, where $i = 1, 2, ..., n$, and τ_i is a natural parameterization of a compact manifold, or a concatenation of several compact manifolds. Let *x* denote all the available data: $x = \{x_1, ..., x_n\}$. The estimated parameter value at point x_i

is denoted by $\hat{\phi}_n(x_i;x)$. An estimate $\hat{\phi}_n$ is consistent if and only if the following holds:

$$\hat{\phi}_n(x_i;x) \Rightarrow T(\tau_i), \qquad \text{as } n \to \infty,$$

where T is a one-to-one rigid transform. In words, a consistent estimate gives the theoretically true estimate when the sample size goes to infinity.

4.2.2. Rate of Convergence

There could be many estimates that are statistically consistent. The rate of convergence is a quantity to further evaluate them. Let $\text{std}(\cdot)$ denote the standard deviation of an estimate. Let $f_1(n) \clubsuit f_2(n)$ denote that $\lim_{n\to\infty} f_1(n)/f_2(n) = \text{constant}$. There exists a constant $\rho > 0$ such that

$$\text{std}(\hat{\phi}_n) \clubsuit n^{-\rho}.$$

When $\rho = 1/2$, $\hat{\phi}_n$ is $\sqrt{n}$-consistent. If $-\rho$ achieves the smallest possible value, the optimal rate of convergence is achieved. The optimal rate of convergence can be computed via Fisher information -- a well-established technique in statistics.

4.2.3. Exhaustiveness

We hope to have $\hat{\phi}_n(x_i;x) \Rightarrow T(\tau_i)$. It is possible that $\hat{\phi}_n(x_i;x)$ converges to a function (not invertible) of $T(\tau_i)$. On the other hand, it might be possible that $T(\tau_i)$ is a function of the limit of $\hat{\phi}_n(x_i;x)$. In both cases, estimate $\hat{\phi}_n$ does not converge to the true natural parameterization. When $\hat{\phi}_n(x_i;x)$ converges exactly to a $T(\tau_i)$, the estimate $\hat{\phi}_n$ is called *exhaustive*. This concept has been developed in statistics, such as searching for *central subspaces* in regression. See the Introduction of Li et al. (2004) for more related information. Examining whether a manifold learning algorithm leads to an exhaustive estimate is a future task.

4.2.4. Robustness

The last requirement is robustness -- namely, if the data are generated according to the model $x_i = f(\tau_i) + \varepsilon_i$, except for a small proportion of them, one should still expect that a *robust* manifold learning algorithm will recover the embedded structure *f*. The threshold of the proportion that can mislead a manifold learning algorithm is called the *breakdown point* of this method. This is an indicator of the robustness of a learning algorithm. Calculating the robustness properties of some manifold learning algorithms will be a future task.

4.3. Initial Results

4.3.1. Formulation and Related Open Questions

We propose a framework to analyze the consistency of a dimension reduction method, especially for those methods that are intended to learn an embedded manifold. The solution to this problem and the technical details will appear in a future publication. We propose this framework to illustrate the necessary components for a theoretical analysis.

We consider a compact subset Ω in the Euclidean space $\Re^d$, $\Omega \subset \Re^d$. Let μ_1 denote a probability measure on Ω. We assume $\mu_1(x) > 0$, $\forall x \in \Omega$, i.e., μ_1 is always positive. We assume that there is an isometric mapping $f : \Omega \to \Re^D$, where $d < D$, and $f \in C^2$, i.e., *f* has continuous (partial) derivatives. It is easy to see that $f(\Omega)$ is a manifold in $\Re^D$ with intrinsic dimension *d*. More specifically, $f(\Omega)$ is a chart, and *x* (as in $f(x)$) is a parameterization of this manifold.

Now we consider a sample version. Assume points X_1, X_2, …, X_N are i.i.d. sampled from Ω according to μ_1. Because *f* is an isometric mapping, we have $\| X_i - X_j \|_E = \mathrm{d}(f(X_i), f(X_j))$, where $\| X_i - X_j \|_E$ is the Euclidean distance between points X_i and X_j, and $\mathrm{d}(f(X_i), f(X_j))$ is the geodesic distance on the manifold between points $f(X_i)$ and $f(X_j)$. We can consider the following questions:

Question 1: Given the observed points $Y_i = f(X_i)$, $i = 1, 2, ..., N$, as $N \to \infty$, can we use a manifold learning method to recover the X_i's up to a rigid motion?

If we consider sampling noise, we may ask the following question:

Question 2: Given the observed points $Y_i = f(X_i) + \varepsilon_i$, $i = 1,2,...,N$, where $\varepsilon_i \overset{i.i.d.}{\sim} \mu_2$, as $N \to \infty$, what are the necessary and sufficient conditions on μ_2, under which a manifold learning algorithm will recover the X_i's up to a rigid motion?

Moreover, in the above setting, we can consider the rate of convergence to the true parameterization as $N \to \infty$.

Our formulation is different from the consistency that has been addressed by the authors of ISOMAP (Tenenbaum, de Silva, and Langford, 2000). They show that as the sample density goes to zero, the graphical distance converges to the geodesic distance. It follows that a subsequent application of MDS will recover the true parameterization (i.e., the true values of X_i). Their approach is different from a traditional way of data analysis.

Laplacian and Hessian eigenmaps in some sense address the problem of consistency. Both Laplacian eigenmaps and Hessian eigenmaps are discrete approximations of the algorithms that have proven consistency in the continuum. Given that a discrete algorithm converges to the continuum version asymptotically, they will have the same property. It is easy to see that this approach cannot provide an analysis of the rate of convergence.

Comprehensive error analysis is given in Zhang and Zha (2004) regarding LTSA. Their pioneering work is very inspiring to us. However, their analysis focuses on an upper bound, which is equivalent to a worst-case study. Our formulation can lead to a more statistical analysis, which we believe in many situations is more meaningful than the worst case study.

4.3.2. Consistency of LTSA

In this section, we establish the consistency of the LTSA algorithm under some mild conditions. The purpose of doing so is to demonstrate some key ingredients in the theoretical analysis.

Recall that Ω is a subset of the feature space $\Re^d$. The function f maps Ω into the data space $\Re^D$, with $d < D$, i.e., $f : \Omega \to \Re^D$. When f satisfies some regularity conditions, the range $f(\Omega)$ forms a manifold. We assume that Ω is bounded, which is formalized in the following:

Condition 1: The domain Ω is bounded, i.e., $|\Omega| < \infty$, where $|\Omega|$ is the Lebesgue measure of Ω in $\Re^d$.

The following notation is needed later. For $x_0 \in \Omega$, an ε-neighborhood of x_0, denoted by $N_\varepsilon(x_0)$, is defined as

$$N_\varepsilon(x_0) = \{x : x \in \Omega, \|x - x_0\|_2 < \varepsilon\}.$$

A function $f : \Omega \to \Re^D$ can be written as

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_D \end{bmatrix}_{D\times 1},$$

where each $f_i(x) = f_i(x_1, x_2, \ldots, x_d)$ is a real-valued function of d variables. The Jacobian of f at the point x_0 $(x_0 \in \Omega)$ is

$$\mathrm{J}\, f(x_0)=\begin{pmatrix} \frac{\partial f_1(x_0)}{\partial x_1} & \cdots & \frac{\partial f_1(x_0)}{\partial x_d} \\ \frac{\partial f_2(x_0)}{\partial x_1} & \cdots & \frac{\partial f_2(x_0)}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_D(x_0)}{\partial x_1} & \cdots & \frac{\partial f_D(x_0)}{\partial x_d} \end{pmatrix}_{D\times d}.$$

The Hessian of $f_i, 1 \le i \le D$, is

$$\{\mathrm{H}\, f_i(x_0)\}_{s,t} = \frac{\partial^2 f_i(x_0)}{\partial x_s \partial x_t},\ 1 \le s,\ t \le d$$
.

Another regularity condition on *f* is the assumption that its Hessians are bounded:

Condition 2: There exists a constant C_1 such that for any $1 \le s$, $t \le d$, $1 \le i \le D$, and $x_0 \in \Omega$, we have $\left|\{\mathrm{H}\, f_i(x_0)\}_{s,t}\right| < C_1$.

The next condition assumes that the mapping *f* is *locally isometric.*

Condition 3: For any $x_0 \in \Omega$ and $\bar{x}_0 \in N_\varepsilon(x_0)$, $\|\bar{x}_0 - x\| \to 0$ implies that

$$\|f(\bar{x}_0) - f(x_0)\|_2 = \|\bar{x}_0 - x_0\|_2 + O\left(\|\bar{x}_0 - x_0\|_2^2\right).$$

Recall $O(x)$ is a quantity that has the same asymptotic order as x when x goes to the positive infinity.

The following argument demonstrates that when *f* is locally isometric, its Jacobian $\mathrm{J}\, f(x_0)$ has to be orthonormal for every $x_0 \in \Omega$. To see this, we consider the Taylor expansion at the point x_0. For $\bar{x}_0 \in N_\varepsilon(x_0)$, we have

$$f(\bar{x}_0) = f(x_0) + \mathrm{J}\, f(x_0)(\bar{x}_0 - x_0) + O\left(\|\bar{x}_0 - x_0\|_2^2\right).$$

If *f* is locally isometric, we have

$$\|\bar{x}_0 - x_0\|_2 = \|f(\bar{x}_0 - f(x_0)\| = \|\mathrm{J} f(x_0)(\bar{x}_0 - x_0)\| .$$

The above is true for any $\bar{x}_0 \in N_\varepsilon(x_0)$. Hence $\mathrm{J} f(x_0)$ is made by a subset of columns of an orthogonal matrix, i.e., $\mathrm{J} f(x_0)$ is orthonormal. Mathematically, we can write

$$[\mathrm{J} f(x_0)]^T [\mathrm{J} f(x_0)] = I_d .$$

In LTSA, it is assumed that the *k* nearest neighbors in the data space correspond to the *k* nearest neighbors in the feature space. The following introduces a sufficient condition for this neighbor-preserving property. Consider points $X_1, X_2, ..., X_N$ that are sampled in Ω. Their images in the data space are $f(X_1), f(X_2), \ldots, f(X_N)$. For each $f(X_t)$, $1 \le t \le N$, let $f(X_{t,1}), f(X_{t,2}), \ldots, f(X_{t,k})$ denote the *k* nearest neighbors of $f(X_t)$ in $\Re^D$. The following is a neighbor-preserving condition:

Condition 4: For any $\delta > 0$, there exist integers $N(\delta)$ and $K(\delta)$ such that for any *t*, $1 \le t \le N$, $X_{t,j} \in N_\delta(X_t), j = 1, 2, \ldots, k = K(\delta)$.

In fact, the reader may verify that if f^{-1} exists and is absolutely continuous, and if the distribution of random points is dense everywhere on $f(\Omega)$, then Condition 4 holds.

Under Conditions 1, 2, 3, and 4, we show that the LTSA algorithm provides a consistent estimate. Recall that LTSA solves the following optimization problem:

$$\min_{\substack{X_t, L(X_t) \\ 1 \le t \le n}} \frac{1}{N} \sum_{t=1}^{N} \frac{1}{k} \sum_{j=1}^{k} \left\| X_{t,j} - X_t - L(X_t)[f(X_{t,j}) - f(X_t)] \right\|_2^2 ,$$

where $L(X_t)$ is a $d \times D$ orthonormal matrix, i.e., $L(X_t)[L(X_t)]^T = I_d$. Recall that $X_t, X_{t,j} \in \Re^d$. Note that the objective function, which is also the objective function in LTSA, is nonnegative. Under conditions 1, 2, 3, and 4, we will show that by taking the original parameterization of the manifold, the above objective goes to zero, which is the smallest possible value of the objective function. Moreover, considering the local solution, for $1 \le t \le N$, we have

$$\left\| X_{t,j} - X_t - L(X_t)[f(X_{t,j}) - f(X_t)] \right\|_2^2 \approx 0 .$$

We can see that the solution is unique up to a rigid motion, i.e., $X_t^{'} = UX_t + V$ is another solution if and only if U is a $d \times d$ orthogonal matrix and V is a d-dimensional vector. Combining the above two, the consistency of LTSA is proved.

We now show that the value of the objective function of LTSA goes to zero under the above four conditions. Recall that for $1 \le t \le N$ and $1 \le j \le k$, we have

$$\begin{aligned} &\left\| f(X_{t,j}) - f(X_t) - Jf(X_t)(X_{t,j} - X_t) \right\|_2 \\ &\le \sqrt{D}\frac{1}{2}C_1 d^2 \left\| X_{t,j} - X_t \right\|_2^2 \le \frac{1}{2}C_1\sqrt{D}d^2\delta^2 . \end{aligned}$$

The above is derived directly from the Taylor expansion at the X_t. Moreover, we have

$$\begin{aligned} &\min_{L(X_t)} \left\| X_{t,j} - X_t - L(X_t)[f(X_{t,j}) - f(X_t)] \right\| \\ &\le \left\| X_{t,j} - X_t - [Jf(X_t)]^T [f(X_{t,j}) - f(X_t)] \right\| \\ &\le \frac{1}{2}C_1\sqrt{D}d^2\delta^2 . \end{aligned}$$

From the above, it is easy to see that the value of the objective function of LTSA is less than or equal to $C_2 \times \delta^2$, where C_2 is a constant. In fact, we can take $C_2 = 1/2\, C_1\sqrt{D}d^2$. When $\delta \to 0$, the objective of LTSA converges to zero. From all of the above, we have established the consistency of LTSA.

5. Examples and Potential Applications

5.1. Successes of Manifold Based Methods on Synthetic Data

We give some numerical examples to demonstrate the effectiveness of manifold learning approaches.

5.1.1. Examples of LTSA Recovering Implicit Parameterization

The following examples show that LTSA can successfully recover hidden low dimensional parameterization from high dimensional data sets. In Figure 5-1 (a, top), data points are sampled from a 1-D curve in a 2-D (or 3-D) space. For each curve, starting from one end of it, its distance to any point on the curve gives a natural parameterization. Obviously, these data sets are intrinsically one-dimensional. In Figure 5-1 (a, bottom), the recovered parameter values are plotted against the true distance parameter values (mentioned above). When the recovered values are consistent with the true parameterization, the bottom figures should be diagonals (i.e., $y = x$ or $y = -x$). Such a pattern is clearly observed.

We would also like to see how LTSA behaves with noise. In Figure 5-1 (b, top), data are sampled with noise around 1-D curves. In Figure 5-1 (b, bottom), we see that LTSA still reliably recovers the implicit parameterization, because of the observable diagonal patterns. More real-world applications can be found in Zhang and Zha (2004).

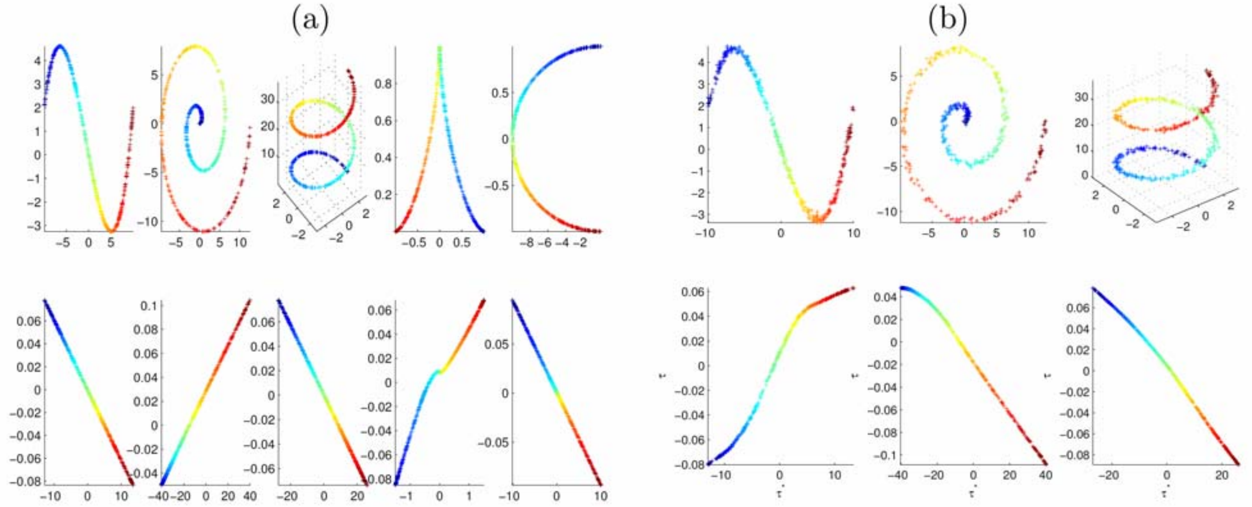


Figure 5-1. Examples of LTSA recovering the intrinsic parameters from (a) noiseless and (b) noisy data.

5.1.2. Example of Locally Linear Projection (LLP) in Denoising

An LLP (Huo, 2003; Huo and Chen, 2002) can be applied to extract the local low-dimensional structure. In the first step, neighbor observations are identified. In the second step, singular value decomposition (SVD) or principal components analysis (PCA) is used to estimate the local linear subspace. Finally, the observation is projected into this subspace. An illustration of LLP in 2-D with local dimension 1 (i.e., linear) and 15 nearest neighbors is provided in Figure 5-2. A detailed description of the algorithm is given in the following.

```
ALGORITHM: LLP
for each observation $y_i$, $i = 1,2,3,...,N$,
    a) Find the K-nearest neighbors of $y_i$. The neighboring points are
       denoted by $\tilde{y}_1$, $\tilde{y}_2$, …, $\tilde{y}_K$.
Use PCA or SVD to identify the linear subspace that contains most of the
information in the vectors $\tilde{y}_1$, $\tilde{y}_2$, …, $\tilde{y}_K$. Suppose the linear subspace
is $\mathrm{A}_i$, and let $P_{\mathrm{A}_i}(x)$ denote the projection of a vector $x$ into this
subspace.
    b) Let $k_0$ denote the assumed dimension of the embedded
       manifold. Then subspace $\mathrm{A}_i$ can be viewed as a linear subspace
       spanned by the vectors associated with the first $k_0$ singular
       values.
    c) Project $y_i$ into the linear subspace $\mathrm{A}_i$ and let $\hat{y}_i$ denote this
       projection: $\hat{y}_i = P_{\mathrm{A}_i}(x)$.
end.
```

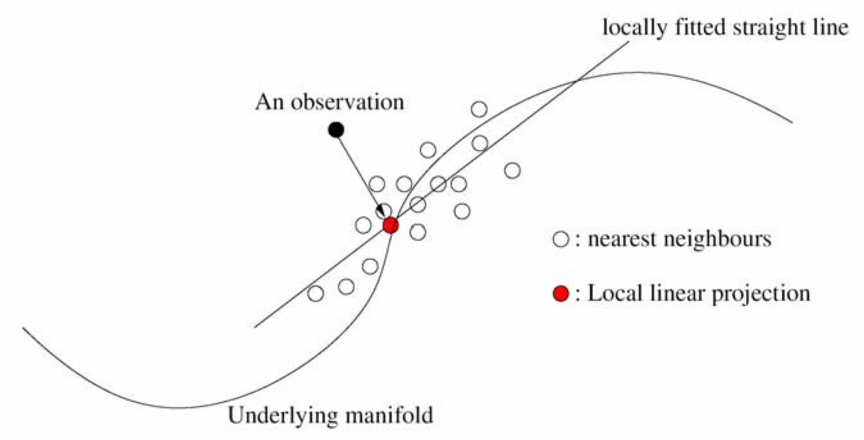


Figure 5-2. An illustration of Local Linear Projection in a 2-D space with local dimension 1 and 15 nearest neighbors.

In Figure 5-3 a denoising example via LLP is provided. The noisy data are presented in the left panel, while the denoised data are presented in the right panel. It is clear that the LLP reveals the true underlying structure in the data set.

5.2. Curve Clustering

Clustering is an important technique in data processing. We consider a data set containing $N = 512$ time series. Each series has dimension $p = 64$. The time series are generated according to the following rule:

$$y_i(t) = \sin\left(\frac{2\pi t}{64} + I(i)\frac{\pi}{2}\right) + \frac{1}{2}\varepsilon_{i,t}, \qquad i = 1,2,...,512; \quad t = 1,2,...,64,$$

where $\varepsilon_{i,t} \sim N(0,1)$ and the function $I(\cdot)$ is defined as

$$I(i)=\begin{cases}0, & \text{if } 1\le i\le 128, & \text{type - I signal,}\\ 1, & \text{if } 129\le i\le 256, & \text{type - II signal,}\\ 2, & \text{if } 257\le i\le 384, & \text{type - III signal,}\\ 3, & \text{if } 385\le i\le 512, & \text{type - IV signal.}\end{cases}$$

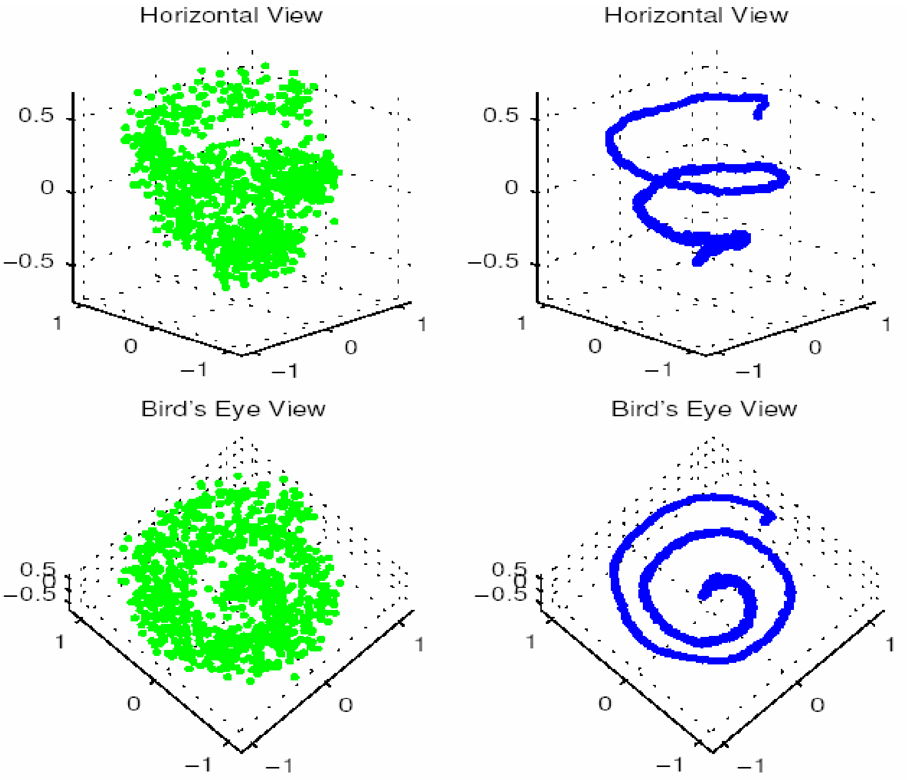


Figure 5-3. Denoising via LLP.

In words, there are *4* trigonometric time series with different phases. One quarter of these time series belong to each type. Figure 5-4 provides an illustration of all the time series. Each plot contains *128* time series belonging to one of the four types. The result of LLP-based denoising is shown in Figure 5-5. Note that the information on how the time series are generated is not used in applying LLP. One can observe that the LLP recovers the underlying patterns of this set of data.

5.3. Image Detection

We now consider the detection of inhomogeneous regions in a homogeneous background (e.g., textures). The underlying assumption is that the samples from the homogeneous background reside on an underlying manifold, while the samples that intersect with the embedded object (i.e., the inhomogeneous region) are 'away' from this manifold. The empirical distance from each sample to the manifold is a quantity to determine the likelihood of a sample's overlapping with an embedded object. This result can consequently be integrated with the 'Significance Run Algorithm' to predict the presence of the embedded structures. A 'local projection' algorithm is designed to estimate the distances between the samples and the manifold. Simulation results for the features embedded in the textural images show promise. This work can be extended to a formal theoretical framework for underlying feature detection. It is particularly well-suited to textural images.

We consider detecting objects in a homogeneous background. The *objects* are the regions within which the distributional properties of these image pixels are different from those in the rest of the image. Two example cases are given in Figure 5-6 and 5-7. In each case, there is a textural image, a trigonometric-function-shaped slim region with contents different from the texture, and a combination of both of them. The detection problem is (1) to determine the presence of an object region, and furthermore (2) to infer the location and the shape of the object region.

This problem is a fundamental one in many applications, such as target recognition, satellite image processing, and so on.

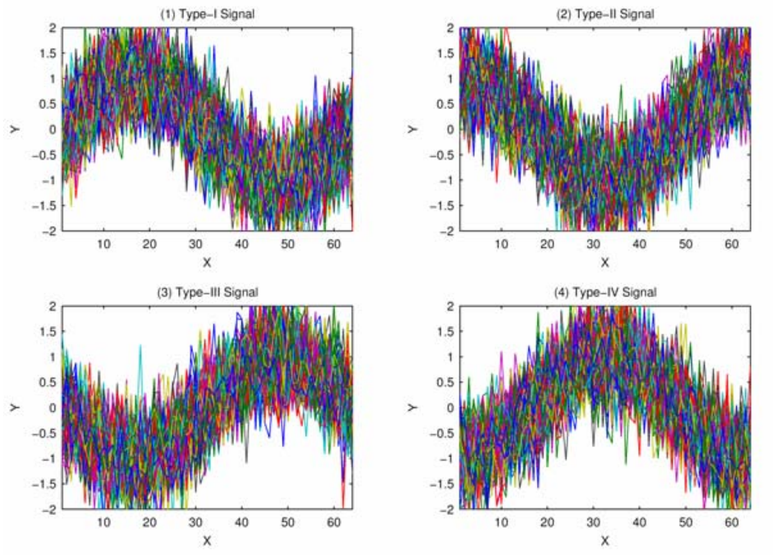


Figure 5-4. Noisy Time Series Data Set.

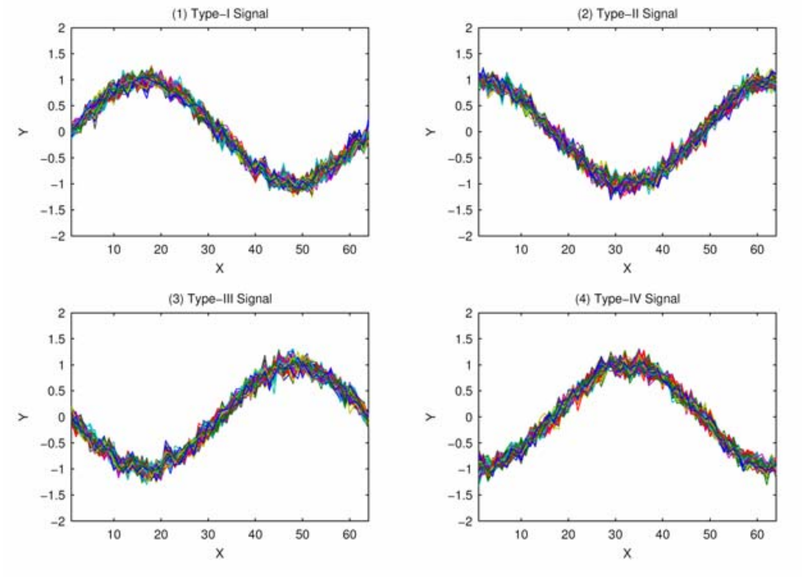
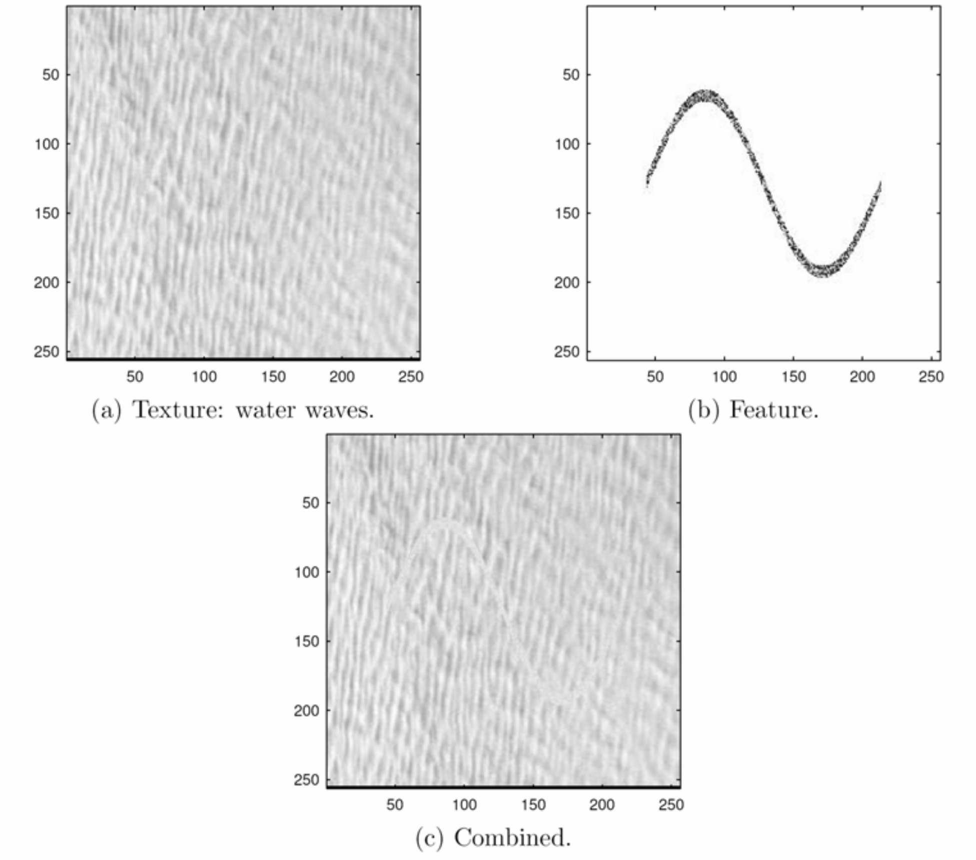


Figure 5-5. Denoised Time Series via LLP.



(a) Texture: water waves.

(b) Feature.

(c) Combined.

Figure 5-6. Example of an object (shaped like a trigonometric function, with its own textural distribution, as depicted in (b)) that is embedded in a textural image ((a)). Panel (c) is a combination: (c)=(a)+(b).

We explore the following idea: (1) the background makes the majority of an image, while an object region is the 'minority'; (2) In addition, the majority of the images (from the homogeneous background), if appropriately sampled, are located on a low-dimensional manifold; (3) The samples that overlap with the embedded region are 'far' from the manifold. Given that the above three conjectures are true, the distance from a sampled patch to the underlying manifold gives the probability that the sample overlaps with the embedded object. If all the *high probability samples* are relatively concentrated, then one has evidence for the presence of an embedded object; otherwise there may not be an embedded object. An illustration of an underlying manifold for samples (e.g., patches) from a homogeneous background is given in Figure 5-8.

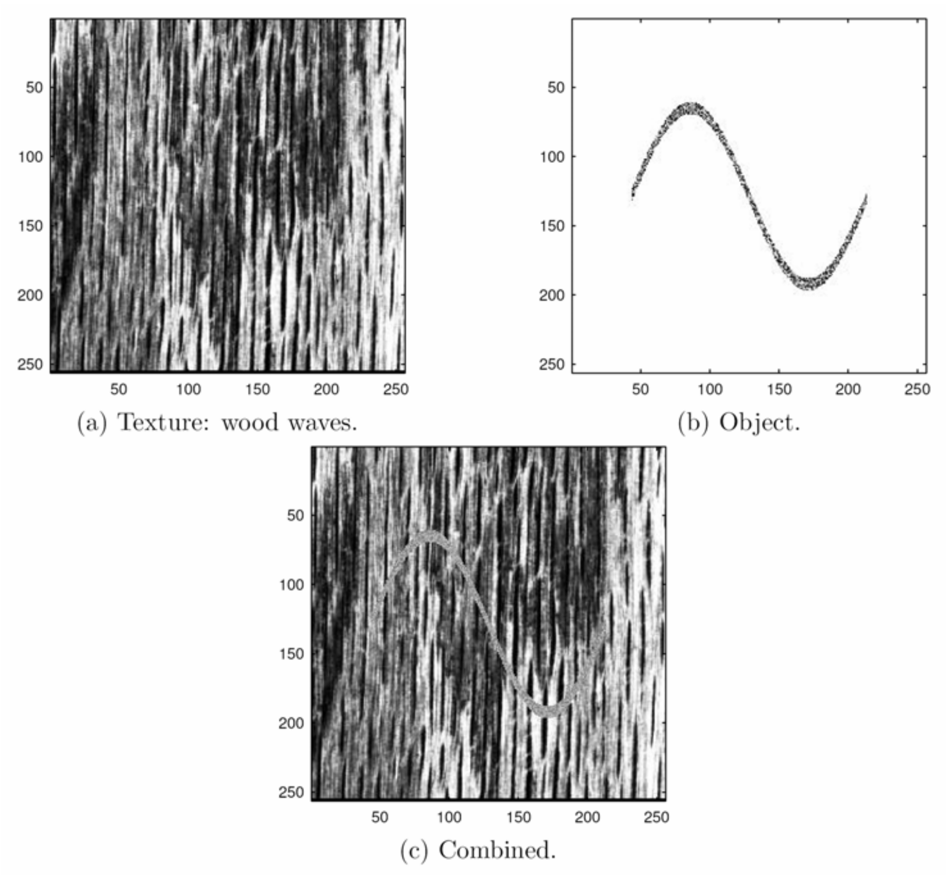


Figure 5-7. Another example of an embedded object.

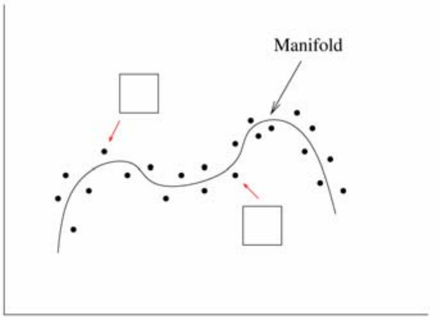


Figure 5-8. Illustration of an underlying manifold. Each square represents a sample patch from the image.

A previously developed framework named *significance run algorithm* (Arias-Castro, Donoho, and Huo, 2003; Huo, Chen, and Donoho, 2003a, b) can be used to process the patterns of the high probability samples. The distance from a sample to an underlying manifold can be estimated by LLP. Simulations demonstrate the effectiveness of this approach, which will be shown in Section 5.3.5.

The rest of this subsection is organized as follows. In Section 5.3.1, the formulation of the problem is given. In Section 5.3.2, the distance to a manifold is defined. Section 5.3.3 describes the Significance Run Algorithm (SRA). In Section 5.3.4, some issues in parameter estimation are discussed. In Section 5.3.5, we present the simulation results. Some conclusions are presented in Section 5.3.6.

5.3.1. Formulation

For an $N \times N$ image, let y_i, $i \in \mathrm{I}$, denote all of the 8 by 8 sampled patches with two diagonal corners being *(*4*a*+1,4*b*+1*)* and *(*4*a*+8,4*b*+8), where $0 \le a, b \le (N-8)/4$. The patch size 8×8 is chosen for computational convenience. We assume that if patch y_i is sampled in the background, then

$$y_i = f(t_i) + \varepsilon_i, \qquad i \in \mathrm{I},$$

where $f(\cdot)$ is a locally smooth function that determines the underlying manifold, the t_i's denote the underlying parameters for the manifold, and the ε_i's are random errors.

5.3.2. Distance to Manifold

For any patch y_i, the distance from this patch to its original image on the manifold $f(t_i)$ is

$$\|y_i - f(t_i)\|_2 .$$

As explained earlier, this distance measures how likely the patch is in the background. The larger the above distance is, the less likely this patch is on the background.

An illustration of the distance from a patch to the manifold is given in Figure 5-9. Note that the function $f(\cdot)$ is not available.

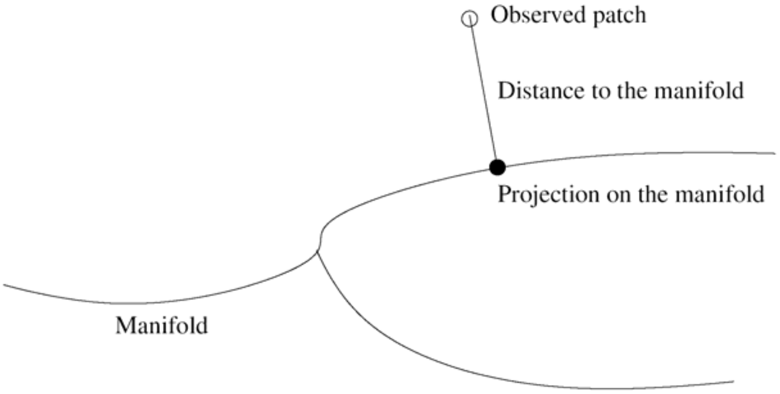


Figure 5-9. Illustration of the distance from an observed patch to the manifold.

The distance between y_i, $i \in \mathrm{I}$ and $f(t_i)$ can be estimated by $\|y_i - \hat{y}_i\|_2$, as described in Section 5.1.2.

5.3.3. SRA: Significance Run Algorithm

Even though the distance to a manifold can be estimated, it still remains unclear when the distance is *significantly* large. Instead of studying the distribution of the distances themselves, we study their spatial patterns by using SRA, which was introduced in Arias-Castro, Donoho, and Huo (2003), and was later used in Huo, Chen, and Donoho (2003a) and Huo, Chen, and Donoho (2003b).

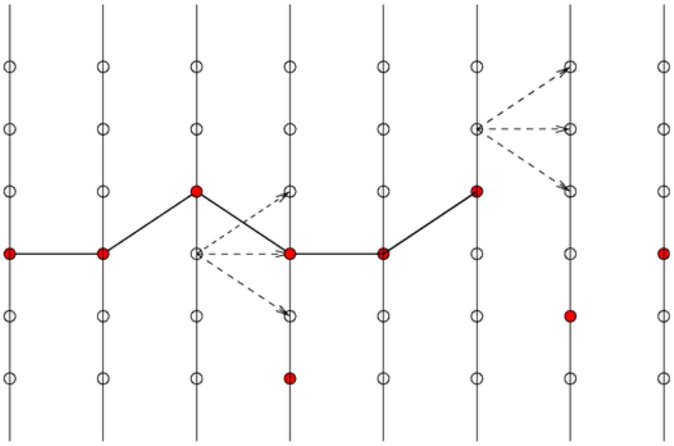


Figure 5-10. An illustration of a Significance Graph and a Significance Run.

A summary of SRA is as follows. Each patch is associated with a node. Because patches are equally spaced, they form a table as in Figure 5-10. (See detailed interpretation on this figure in Chen and Huo (2006).)

There is an edge between two nodes if and only if the corresponding patches are spatially connected. A node is significant if and only if the corresponding distance $\|y_i - \hat{y}_i\|_2$ is above a prescribed threshold (denoted by T_1). A *significance run* is a chain of the connected significant nodes. The length of the longest significance run is the test statistic: an embedded object is claimed to be present if and only if this length is above a constant (denoted by T_2). It has been shown (e.g., Arias-Castro, Donoho, and Huo (2003); Huo, Chen, and Donoho (2003b)) that SRA leads to a powerful test.

Note that both T_1 and T_2 can be determined numerically. T_1 can be a given percentile of the empirical estimates of the distances: $\|y_i - \hat{y}_i\|_2$, and T_2 can be derived from simulations.

5.3.4. Parameter Estimation

In LLP, one needs to specify the number of the nearest neighbors and the local dimension. This can be done by studying the empirical distribution of the distances and the total residual sum of squares.

5.3.4.1. Number of nearest neighbors

An illustration of the percentiles of the distances to the nearest neighbors is given in Figure 5-11. We choose *50* nearest neighbors, because it is approximately a kink point in this figure. It is possible to choose the number of the nearest neighbors by studying the distances to the nearest neighbors. Here we do not pursue this problem further.

5.3.4.2. Local Dimension

The problem of estimating the local dimension has been analyzed in Roweis and Saul (2000) and Tenenbaum, de Silva, and Langford (2000). There are follow-up works in this line. Due to space limitations, we omit the details. Figure 5-12 gives the plot of the residual sum of squares $\sum_{i\in I}\|y_i - \hat{y}_i\|_2^2$ versus the local dimension (as k_0 in the LLP). An approximate kink point is at $k_0 = 15$, which is our choice of the local dimension in the simulations.

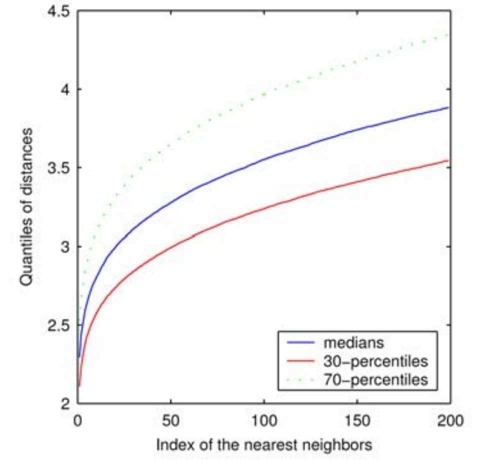


Figure 5-11. Percentiles of the distances from the nearest neighbors.

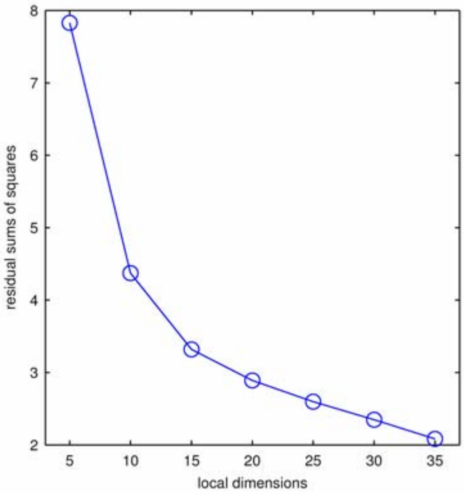


Figure 5-12. Residual sum of squares versus local dimension.

5.3.5. Simulations

We apply the above approach to the two figures in Figure 5-6 (c) and Figure 5-7 (c). The positions of the significant patches are displayed in Figure 5-13 (for the water image) and in Figure 5-14 (for the wood image), respectively. In both cases, the constant T_1 is chosen to be the 95th percentile of the squared distances: $\|y_i - \hat{y}_i\|_2^2$, $\forall i \in \mathrm{I}$. Obviously, the significant patches are concentrated around the embedded object, which is the trigonometric shape. Hence SRA will unveil the presence of the object.

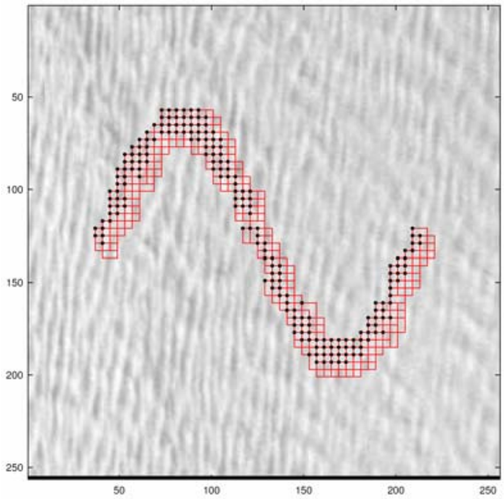


Figure 5-13. Pattern of significant patches for the water image. Northwestern corners of the significant patches are marked by dark dots.

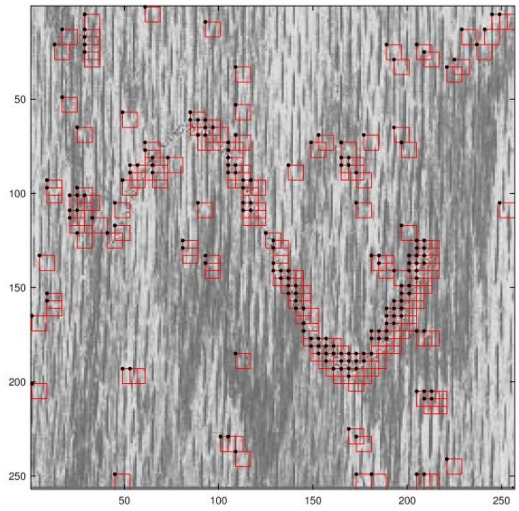


Figure 5-14. Pattern of significant patches for the wood image. Northwestern corners of the significant patches are again marked by dark dots.

For comparison, Figure 5-15 gives the patterns of significant patches when there is no embedded object.

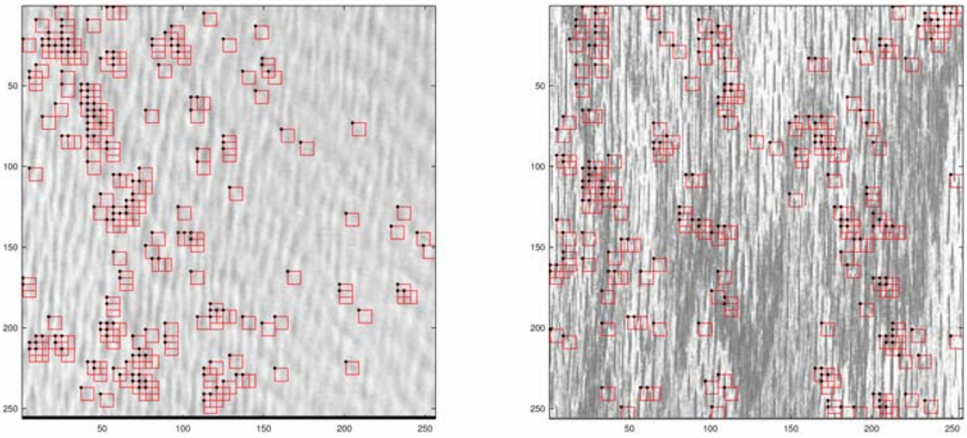


Figure 5-15. Pattern of significant patches for water and wood images when there is *no* embedded object.

5.3.6. Discussion

By modifying the structure of the significance graph, the above approach can be applied to more general objects, e.g., instead of graphs, one can consider curves, or even non-filamentary objects. We leave this as a future research topic.

If the background is non-homogeneous, which is true in many cases, the above approach will fail. The proposed framework can be used to derive a general theory on when an embedded object is detectable, and when it is not. This will be another topic for future research.

5.4. Applications in Localization of Sensor Networks

One area in which manifold-based learning methods can be applied is sensor positioning in wireless networks. This type of application is of interest in, for example, military surveillance. We typically assume that there are a large number of sensors randomly deployed over an area. Each sensor contains a simple radio transmitter, and from this we know the pairwise distances between the sensors. Based on this information, we would like to compute the relative positions of all the sensors. Furthermore, we may know the true global positions of a few sensors (called "anchor nodes"), and based on this we may wish to compute the global positions of all the sensors. An example of the situation, in which we may need to compute the global positions, is given in Figure 5-16.

The solution to the first problem depends on whether we have *all* the pairwise distances available. Of course this may or may not be true in practice. If all the distances are available, the method is known as classical multidimensional scaling (MDS), which as mentioned earlier is a variation on the idea of principal components. Let $T=[t_{ij}]_{n\times 2}$ be the matrix of the true locations of the set of n sensor nodes in the *2*-dimensional Euclidean space, and let $d_{ij}(T)$ denote the true distance between sensors *i* and *j*. We assume that we know the true distances d_{ij}. Then the classical MDS algorithm is as follows:

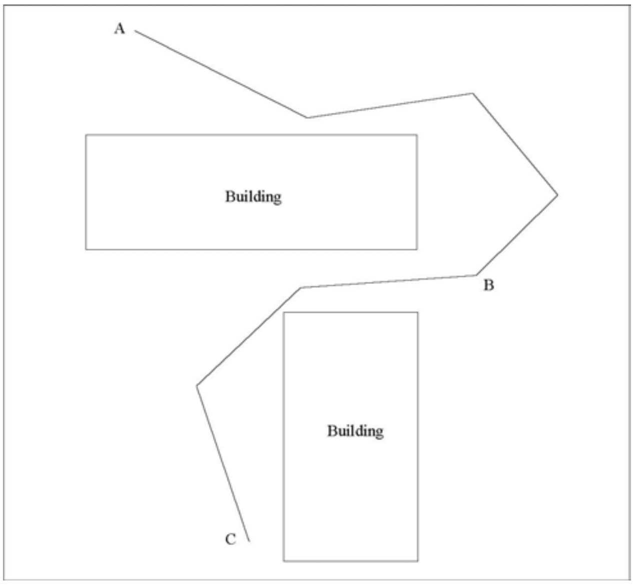


Figure 5-16. Illustration of the sensor localization problem.

- Compute the matrix of squared distances D^2, where $D = [d_{ij}]_{n \times n}$.
- Compute the matrix *J* with $J = I - \frac{\mathbf{1}\mathbf{1}^T}{n}$, where $\mathbf{1} = (1,1,\ldots,1)$.
- Apply double centering to this matrix: $H = -\frac{1}{2} J D^2 J$.
- Compute the eigen-decomposition $H = UVU^T$.
- To recover the solution in *i* dimensions, the coordinate matrix is $X = U_i V_i^{\frac{1}{2}}$, where U_i is formed by the first *i* columns of *U*, and

V_i is the diagonal matrix containing the largest i eigenvalues of H.

If there are missing distances, we can use a more complicated iterative MDS optimization algorithm to minimize the sum of residual errors of our estimated positions. Such a solution has been presented in Ji and Zha (2004).

The second case is more interesting. Recall that the relative positions of the sensors are assumed to be known, and we wish to compute the global positions based on some knowledge of the exact positions of a few sensors. Intuitively, since the *relative* positions that are computed will be unaltered under rigid motions, the problem is to find the optimal isometric mapping of the local positions to match the known global positions of the anchors. In this sense it can be thought of as a variant of the Local Tangent Space Alignment (LTSA) idea presented above. For simplicity we assume that our measured pairwise distances are all equal to the corresponding true distances to ensure that a solution exists. As it turns out, we need to know the exact global positions of at least 3 anchor nodes in order to have a feasible problem. The requirement that we need at least 3 anchor nodes is also intuitively explained by viewing the optimal isometry as 3 separate functions – a shifting, a rotation, and a reflection. Then the first anchor node can be thought of as determining the optimal shift, the second determines the optimal rotation, and the third determines the reflection.

6. Conclusion

We have given a broad survey of manifold-based learning methods, emphasizing their mathematical formulations. By doing so, we hope to give new insight into the similarities between the various methods, and their underlying unified theoretical framework, which we believe will be the focus of future research in this area. It is our hope that this article will attract more researchers to work in this area and stimulate a new direction for work in the theoretical analysis of manifold-based methods and related applied problems.

Appendix: Some related and useful URLs

The following websites provided useful information while this chapter was written.

- MSU: http://www.cse.msu.edu/~lawhiu/manifold/
- MIT: http://www.ai.mit.edu/courses/6.899/doneClasses.html
- UBC: http://www.cs.ubc.ca/~mwill/dimreduct.htm
- Penn: http://www.seas.upenn.edu/%7Ekilianw/workpage/drg/
- Fudan, China: http://www.iipl.fudan.edu.cn/people/zhangjp/literatures/MLF/INDEX.HTM

References

Abdullaev, Y. G. and Posner, M. I. (1998). Event-related brain potential imaging of semantic encoding during processing single words. *NeuroImage*, **7**, 1-13.

Arias-Castro E., Donoho, D. L., and Huo, X. (2006). Adaptive multiscale detection of filamentary structures embedded in a background of uniform random points. *Annals of Statistics*, **34**(1), 326-349, February.

Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., and van der Vorst, H. (2000). Templates for the solution of algebraic eigenvalue oroblems: a practical guide. Society for Industrial and Applied Mathematics, Philadelphia, U.S.A.

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In Dietterich, T. G., Becker, S., and Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems, **14**, 585-591.

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, **15**(6), 1373-1396.

Bernstein, M., de Silva, V., Langford, J. C., and Tenenbaum, J. B. (2000). Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University, Stanford, December.

Bishop, C. M., Svensen, M., and Williams, C. K. I. (1998). GTM: The generative topographic mapping. *Neural Computation*, **10**(1), 215-234.

Borg, I. and Groenen, P. (1997). *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, New York, NY, U.S.A.

Brand, M. (2003). Charting a manifold. In Proceedings, Neural Information Processing Systems, Volume 15. Mitsubishi Electric Research Lab: MIT

Press. TR-2003-13 March 2003, http://www.merl.com, Presented at NIPS-15, December 2002.

Chen, J. and Huo, X. (2004a). Sparse representations for multiple measurement vectors (MMV) in an over-complete dictionary. ICASSP 2005, Philadelphia, PA, U.S.A.

Chen, J. and Huo, X. (2004b). Theoretical results about finding the sparsest representations of multiple measurement vectors (MMV) in an over-complete dictionary, using ℓ_1-norm minimization and greedy algorithms. To appear in IEEE Trans on Signal Processing. URL: http://www.isye.gatech.edu/~xiaoming/publication/pdfs/mmv101204.pdf.

Chen, J. and Huo, X. (2006). Distribution of the length of the longest significance run on a Bernoulli net, and its applications. *Journal of the American Statistical Association*, **101**(473), 321-331, March.

Costa, J. A., Patwari, N., and Hero, A. O. (2004). Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. Submitted to ACM Trans. on Sensor Networks, June.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerical Mathematics*, **1**, 269-271.

Donoho, D. L. and Grimes, C. E. (2003). Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences*; **100**, 5591-5596.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, **3**(3), 290-226.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, **5**, 73-79.

Globerson, A. and Tishby, N. (2003). Suffient dimensionality reduction. Journal of Machine Learning Research, **3**, 1307-1331.

Haase, A. (1990). Snapshot flash MRI: Application to T1, T2, and chemical shift imaging. *Magn. Reson. Med*. **13**, 77-89.

Hero, A. O., Costa, J., and Ma, B. (2003). Convergence rates of minimal graphs with random vertices. Submitted to IEEE Trans. on Information Theory, March.

Huo, X. (2003). A geodesic distance and local smoothing based clustering algorithm to utilize embedded geometric structures in high dimensional noisy data. In *SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data and its Applications*, San Francisco, CA. May.

Huo, X. and Chen, J. (2002). Local linear projection (LLP). In First IEEE Workshop on Genomic Signal Processing and Statistics (GENSIPS), Raleigh, NC. http://www.gensips.gatech.edu/proceedings/, October.

Huo, X., Chen, J., and Donoho, D. L. (2003a). Multiscale detection of filamentary features in image data. In *SPIE Wavelet-X*, San Diego, CA. August.

Huo, X., Chen, J., and Donoho D. L. (2003b). Multiscale significance run: Realizing the 'most powerful' detection in noisy images. *Asilomar Conference on Signals, Systems, and Computers*. November.

Huo, X. and Ni, X. (2004a). Computational and statistical perspectives on the importance of phase information in signal reconstruction. Submitted to a journal.

Huo, X. and Ni, X. (2004b). Counting the number of convex sets in a digital image. Submitted to a journal.

Ji, X. and Zha, H. (2004). Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. *Proceedings of IEEE INFOCOM*, pp. 2652-2661.

Kohonen, T. ((1995, 1997,) 2001). *Self-organizing maps* (3rd edition Ed.). Springer-Verlag, New York, NY, U.S.A.

Kruskal, J. B. (1964). Multidimensal scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, **29**, 1-27.

Li, B., Zha, H., and Chiaromonte, F. (2004). Contour regression: a general approach to dimension reduction. Annals of Statistics. To appear.

Petersen, S. E., Fox, P. T., Posner, M. I., Mintun, M., and Raichle, M. E. (1990). Positron emission tomographic studies of the processing of single words. *J. Cognitive Neurosci.* **1**(2), 154-170.

Petersen, S. E., Fox, P. T., Posner, M. I., Mintun, M., and Raichle, M. E. (1988). Positron emission tomographic studies of the cortical anatomy of single word processing. *Nature*, **331**, 585-589.

Raichle, M. E., Fiez, J. A., Videen, T. O., MacLeod, A. -M. K., Pardo, J. V., Fox, P. T., and Petersen, S. E. (1994). Practice-related changes in human brain functional anatomy during non-motor learning. *Cereb Cortex*, **4**, 8-26.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**, 2323-2326.

Saul, L. K. and Roweis, S. T. (2001). An introduction to Locally Linear Embedding. URL: http://www.cs.toronto.edu/~roweis/lle/publications.html.

Shepard, R. N. (1962) The analysis of proximities: multidimensional scaling with an unknown distance function: I & II. *Psychometrika*, **27**, 125-140 & 219-246.

Snyder, A. Z., Abdullaev, Y. G., Posner, M. I., and Raichle, M. E. (1995). Scalp electrical potentials reflect regional cerebral blood flow responses during processing of written words. *Proc. Natl. Acad. Sci.*, USA **92**, 1689-1693.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci.*, USA **96**(6), 2907-2912.

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

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, **17**, 401-419.

Wu, Y. N., Zhu, S. C., and Liu, X. W. (2000). Equivalence of Julesz ensemble and FRAME models. *International Journal of Computer Vision*, **38**(3), 247-265, July.

Young, G. and Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, **3**, 19-22.

Yuille, A. L., Coughlan, J. M., Wu, Y. N., and Zhu, S. C. (2001). Order parameter for detecting target curves in images: how does high level knowledge helps? *International Journal of Computer Vision*, **41**(1/2), 9-33.

Zhang, Z. and Zha, H. (2004). Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Scientific Computing,* **26**(1), 313-338.