Manifold Fitting and Projection via Quadratic Approximation

Anonymous Authors[1]

Abstract

Manifold fitting is an emerging filed with an interaction of statistics and geometry. In this paper, we propose to fit the manifold with a quadratic function defined on the tangent space with a specific form. Compared with the existing linear-approximation methods, the quadratic-approximation approach can fit the unknown manifold with higher precision. Because a more complicated function is adopted in our manifold fitting process, the pulling-back approach turns the linear least-squares problem into a nonlinear quartic minimization problem. By bringing in the auxiliary function, we solve the quartic by repeatedly solving a series of quadratic minimization problems. Numerical experiments demonstrate that our method has a strong recovery capability compared with other current methods.

1. Introduction

The data residing in the high-dimensional ambient space often have some low-dimensional structure, because the principal factors that affect the intrinsic generation process are often very limited. The locations of earthquakes or volcanos, which are caused by crustal movements can be thought of as the points residing on the one-dimensional manifold or the principal flow (Panaretos et al., 2014; Davenport et al., 2010).

With a deep-learning approach (Goodfellow et al., 2014), using a network with fine-tuning parameters, we can convert a very low-dimensional data into a very complicated image that has some specific meaning. Also, we can use a simple sentence to generate a vivid picture via a well-trained network. These phenomena indicate that very large amounts of data in our daily life are low-dimensional.

There are two main approaches to handle the high-dimensional data. The first is to reconstruct the data in the low-dimensional space, such that the reconstructed data in the low-dimensional space also keep the relation (such as distance, geometry, affine transformation etc.) in high-dimensional space. This approach involves a large amount of classical dimension-reduction works, such as LLE (Roweis & Saul, 2000), LTSA (Zhang & Zha, 2004), Isomap (Tenenbaum et al., 2000), and Eigenmap (Belkin & Niyogi, 2003).

The other approach is to try to fit a new smooth manifold by reducing the existing noise in the data. After the fitting process, the true signal will be strengthened and the noise factors will be diminished. Kernel density estimation (Genovese et al., 2014; Ozertem & Erdogmus, 2011) is a very popular tool for manifold fitting, and involve transforming the manifold-fitting problem into a ridge-estimation problem. The ridge is defined as the set of points that satisfy some relation on the gradient and Hessian of the probabilty density function. Overall, the whole manifold-fitting process can be encapsulated by two steps. First, estimate the tangent space, which is often done through the eigenvalue-decomposition of the sample covariance matrix. Second, estimate an attraction force, which is a direction that points to somewhere near the manifold, and is often constructed via the weighted shift-mean approach. By using the subspace-constrained mean-shift algorithm, we can move the outlier onto the manifold to some degree.

Finding a good approximation of the underlying manifold is the first step in our manifold-fitting problem. The second important step is projecting the noisy data onto the fitted manifold obtained in the first step. Here, the term 'project' has two connotations. First, the distance should be minimal, and second, the direction of the projection should be perpendicular to the tangent place of the fitted manifold. A major reason the linear approximation is very widely used is that the form of projection yields a quite simple linear least-squares problem.

1.1. Model Setting

In this paper, we do not concentrate on finding the representation of $\phi(\tau)$ under the noiseless assumption. Instead, we assume we have the observations drawn from some low-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

dimensional manifold and disturbed by some noise, i.e.,

$$x_i = \tilde{x}_i + \epsilon_i, \quad \tilde{x}_i \in \mathcal{M},$$

where ϵ_i is the noise that obeys some distribution, such as the multivariate Gaussian distribution. Since the observations $\{x_i, i = 1 : n\}$ are discretely distributed, the idea of manifold fitting aims at generalizing the discrete data and obtaining a lower-dimensional approximation of the dataset. The manifold fitting approach can be written as a parametric estimation problem under the observations and the constrained model $\mathcal{G}$, i.e.,

$$\theta_* = \arg\min_{\theta} \sum_i \text{Loss}(x_i, \mathcal{G}, \theta), \tag{1}$$

where $\mathcal{G}$ represents the abstract model and θ represents the parameters within the model $\mathcal{G}$. Different models (such as the linear and nonlinear) correspond to different $\mathcal{G}$. When we obtain our best parameter θ_* from (1), we can use the model $\mathcal{G}(\theta_*)$ to refine the outlier x using the projection $P_{\mathcal{G}(\theta_*)}(x)$, via solving the following minimization problem:

$$P_{\mathcal{G}(\theta_*)}(x) = \arg\min_{y \in \mathcal{G}(\theta_*)} \|x - y\|_2.$$

Works such as (Genovese et al., 2014; Ozertem & Erdogmus, 2011) all focus on how to obtain a better affine space to locally approximate the distribution of the data; in other words, in these works $\mathcal{G}(\theta)$ is a linear model. However, as far as we know, the result of the linear-approximation approach relies heavily on the selection of the origin point (the red dot in Figure 1). A good-quality origin selected will improve the projection's recovery ability. However, in most of the cases, the origin can only be selected approximately as the underlying manifold is unknown.

1.2. Manifold Parameterization

For any manifold $\mathcal{M}$ and any point $x_0 \in \mathcal{M}$, there is a corresponding twice differentiable function $\phi_{x_0}(\tau)$:

$$\phi_{x_0}(\tau) : \mathbb{R}^d \to \mathbb{R}^{D-d},$$

such that every point within a local domain $\mathcal{D}_{x_0}(r)$ of $\mathcal{M}$ can be written with a parameterization form of

$$x(\tau) = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\phi_{x_0}(\tau), \tag{2}$$

where the columns of U_{x_0} are the basis on the tangent space and the columns in $U_{x_0}^{\perp}$ are the basis on the normal space.

To demonstrate the different behaviors corresponding to the linear- and quadratic-fitting approaches, we provide a toy-fitting case in Figure 1 with different origins. The origin x_0 (the red dot) in the left figure is farther away from the true projection (the black dot) than that in the right

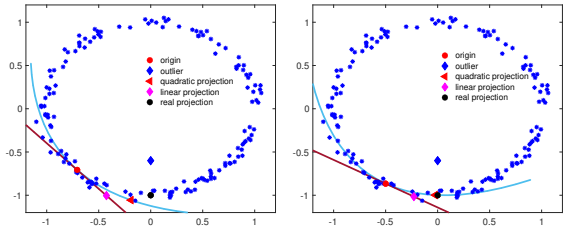

Figure 1. The reliance of the origin point in the process of manifold fitting and projection

figure. This example shows the effects of the different fitting errors, which are $O(\|\tau\|_2^2)$ and $O(\|\tau\|_2^3)$, corresponding to the linear and quadratic form, respectively. From this case, we know that the linear approach relies on a good origin significantly more than the quadratic form does, because the approximation by linear case yields a lower-order error compared with the quadratic case.

In this paper, we convert the manifold-fitting problem into the search for an approximation version of $\phi_{x_0}(\tau)$, through a deep analysis of the characteristics of it. We also show that the dominant term for the Taylor expansion of $\phi_{x_0}(\tau)$ can be written as a quadratic form of a tensor acting on τ; i.e., $\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau, \tau)$, where $\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$ is a third-order tensor with a shape of $d \times d \times (D - d)$.

In addition, we show that, by adopting the dominant term in the Taylor expansion of $\phi_{x_0}(\tau)$, the nonlinear function $\phi_{x_0}(\tau)$ can be locally simplified as a quadratic form of $\mathcal{A}(\tau, \tau)$, where $\mathcal{A}$ is the empirical estimation of $\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$. The unknown parameter $\mathcal{A}$ can be obtained via solving a linear least-squares problem. After obtaining the representation of $x_{\mathcal{A}}(\tau)$, we also develop a projection strategy to refine the outlier point $\bar{x}$ by projecting it onto our fitted manifold $\mathcal{M}_{\mathcal{A}}$. Furthermore, we show that the projection of $\bar{x}$ onto $\mathcal{M}_{\mathcal{A}}$ can be achieved by repeatedly solving a series of linear least-squares problems. The strengths of our method can be summarized as follows:

1. By fitting a function with the quadratic form, the curvature of the manifold is considered in our algorithm. Because of the existence of the curvature, the fitted manifold $\mathcal{M}_{\mathcal{A}}$ can approximate the true manifold within a relatively larger scope.

2. Our algorithm does not rely too much on the accuracy of the estimated origin point, which means a relatively rough estimation of the origin is accessible in our algorithm. Instead of the origin, we just need to get a point (near x) that is also supposed to be next to the true $\mathcal{M}$.

3. The solution of our algorithm has a clear geometric interpretation via the optimized $\hat{\tau}$. When we obtain the local coordinate $\hat{\tau}$, we can think of our projected point $x(\hat{\tau})$ (the

projection onto $\mathcal{M}_{\mathcal{A}}$) as a modification of the origin x_0, through,

$$x(\tau) = U_{\perp}\mathcal{A}(\tau,\tau) + U\tau + x_0.$$

where $U\tau$ is the modification of x_0 in the tangent space and $\mathcal{A}(\tau,\tau)$ is the modification of x_0 in the normal space.

2. Fitting Model

The fitting model has two main requirements. The first is the local fitting of a complicated function by a simple form such as a Taylor expansion. The second is the use of a generalized representation such that the measurement with respect to the observations and the representation is optimal under some criteria.

For a complicated nonlinear function $\phi(\tau)$, we can fit locally via the lower-order Taylor expansion:

$$\begin{aligned}\phi_q(\tau) =& \phi(\tau_0) + \langle \nabla_\tau \phi(\tau)|_{\tau=\tau_0}, \tau - \tau_0\rangle + ... \\ &+ \frac{1}{2}\nabla_\tau\nabla_\tau\phi(\tau)|_{\tau=\tau_0}(\tau-\tau_0,\tau-\tau_0).\end{aligned} \tag{3}$$

1. The linear function $\phi(\tau_0) + \langle \nabla_\tau \phi(\tau)|_{\tau=\tau_0}, \tau - \tau_0\rangle$ can be thought of as a locally fitting model of $\phi(\tau)$ at $\tau = \tau_0$ with an error of order $O(\|\tau - \tau_0\|_2^2)$.

2. The quadratic function (3) can be thought of as a locally fitting model of $\phi(\tau)$ at $\tau = \tau_0$ with an error of order $O(\|\tau - \tau_0\|_2^3)$. In this paper, we solve the manifold-fitting problem via the local quadratic approximation under the noisy observations. To simplify the problem, we first assume: ?

2.1. Quadratic surface in 3-D space

A manifold is a topological space that locally resembles Euclidean space. Except for a few special cases, we cannot have global representation to parameterize the manifold. As a result, we try to approximate the manifold locally within an interested area. First, we provide a simple demo as an illustration.

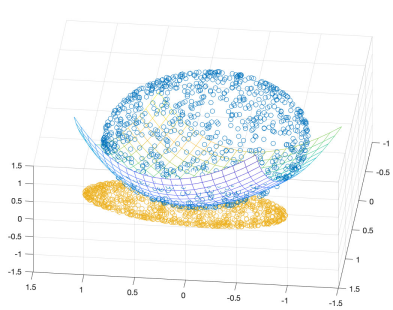

Figure 2. Fitting a manifold with the elliptical paraboloid surface

Example 1. The data, consisting of blue circles evenly distributed on a 2-D sphere as shown in (2), is a toy example of a manifold. Fixing a point $x_0 \in \mathbb{S}^2$, we can have a tangent plane. The yellow circles are the projection of the data onto the tangent plane. The continuous elliptical paraboloid surface is the structure we want. Figure 2 clearly shows that the mesh surface is a better structure to approximate $\mathbb{S}^2$, compared with the tangent plane. Here, the parameterized function of the quadratic form $x_A(\tau) : \mathbb{R}^2 \to \mathbb{R}^3$ yields: $x_A(\tau) = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\tau^T A\tau$, where A is a matrix of size 2×2 that is used to control the shape and direction of the paraboloid surface. The eigenvalues of A represent the curvature or flatness in the corresponding eigenvector direction.

2.2. Quadratic surface in high-dimensional space

Instead of fitting the manifold via the tangent plane, we bring a unknown function $\phi_{x_0}(\tau)$ from the tangent space to the normal space

$$\phi_{x_0}(\tau) : \mathbb{R}^d \to \mathbb{R}^{D-d}.$$

Using $\phi_{x_0}(\tau)$, we know any manifold $\mathcal{M}$ that is local at x_0 can be written as the range of function $x(\tau)$:

$$x(\tau) = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\phi_{x_0}(\tau), \tag{4}$$

where $\phi_{x_0}(\tau) = [\phi_{x_0}^{d+1}(\tau), ..., \phi_{x_0}^{D}(\tau)]^T$ is the function from tangent space to the normal space. τ is the local coordinate in the tangent space. From (4), we know that there is a one-to-one correspondence between $\phi_{x_0}(\tau)$ and $x(\tau)$:

$$\phi_{x_0}(\tau) = {U_{x_0}^{\perp}}^T(x(\tau) - x_0) \tag{5}$$

From (5), we know that, if we are given the structure of $\phi_{x_0}(\tau)$ with some parameters, we can have the representation of $\phi_{x_0}(\tau)$, for the following reasons:

1. x_0 is any point on the manifold we are interested in.

2. U_{x_0} is the basis of principal space at x_0, which can be obtained from the local PCA. $U_{x_0}^{\perp}$ is the orthogonal direction at x_0, which can also be obtained from the local PCA.

3. The only remaining part is $\phi_{x_0}(\tau)$.

Note that, to implement the local PCA, the local covariance matrix at x_0 is defined as $C(x_0) = \sum_i K_h(x_0, x_i)(x_i - x_0)(x_i - x_0)^T$. The basis of the estimated tangent space can be obtained from the eigenvalue decomposition:

$$C(x_0) = [U_d, U_{D-d}]\begin{bmatrix}\Lambda_1 & 0\\ 0 & \Lambda_2\end{bmatrix}[U_d, U_{D-d}]^T. \tag{6}$$

From the decomposition in (6), we can set the basis on the tangent plane as $U_{x_0} = U_d$, and the basis on the normal space as $U_{x_0}^{\perp} = U_{D-d}$. In the local domain $\mathbb{R}_{x_0}^D(r) \cap \mathcal{M}$, $\phi_x(\tau)$ can be approximated by the polynomial of order two.

Thus, we will get

$$\begin{aligned}\phi_{x_0}(\tau) =&\phi_{x_0}(0) + \nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau)+ \\ &+ \frac{1}{2}\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau,\tau) + O(\|\tau\|_2^3),\end{aligned} \tag{7}$$

where $\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$ stands for the second fundamental form, which is a three-order tensor of size $d \times d \times (D-d)$. Acting on τ twice will result in a vector of size $D-d$. The linear gradient operator $\nabla\phi_{x_0}(\tau)|_{\tau=0} \in \mathbb{R}^{(D-d)\times d}$ acting on τ will result in a vector in the normal space of size $D-d$.

Recalling (7), we know $\mathcal{M}$ can be parameterized with the remainder term:

$$x(\tau) = x_0 + U_{x_0}\tau + \frac{1}{2}U_{x_0}^{\perp}\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau,\tau) + O(\|\tau\|_2^3)$$

For notational convenience, we let the tensor be denoted by $\mathcal{A} = \frac{1}{2}\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$, and define a function $x_{\mathcal{A}}(\tau)$ related to $\mathcal{A}$ as

$$x_{\mathcal{A}}(\tau) = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\mathcal{A}_{\phi_{x_0}}(\tau,\tau). \tag{8}$$

From the range $x_{\mathcal{A}}(\tau)$ in (8), we have a new manifold $\mathcal{M}_{\mathcal{A}}$ derived from $x_{\mathcal{A}}(\tau)$:

$$\mathcal{M}_{\mathcal{A}} = \{z | z = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\mathcal{A}_{\phi_{x_0}}(\tau,\tau), \tau \in \mathbb{R}^d\} \tag{9}$$

Because $x_{\mathcal{A}}(\tau) - x(\tau) = O(\|\tau\|_2^3)$, we know $\mathcal{M}_{\mathcal{A}}$ approximates $\mathcal{M}$ well when the scale of τ_i is relatively small.

In most of the manifold-fitting cases, both he true manifold $\mathcal{M}$ and the function $\phi_{x_0}(\tau)$ are unknown, and as a result it is impossible to obtain the second-order parameter $\mathcal{A}_{x_0}$ through differentiating the multivariate function $\phi_{x_0}(\tau)$. However, although we cannot obtain the accurate $\phi_{x_0}(\tau)$, we can estimate it by using the data (observations) that are supposed to be drawn from some low-dimensional manifold. In this case, we need to estimate the $\mathcal{A}_{\phi_{x_0}}$ from the observations to obtain $\hat{\mathcal{A}}$, and as a result the range $\mathcal{M}_{\hat{\mathcal{A}}}$ of $x_{\hat{\mathcal{A}}}(\tau)$ can be assumed to be a local smoothed approximation of $\mathcal{M}$ around x_0 to some degree. From now on, we will abbreviate $\hat{\mathcal{A}}$ as $\mathcal{A}$, to stand for the estimated second-order parameter.

2.3. Fitting Model

In a real case, when we have samples from the manifold, we need to determine $U_{x_0}, \mathcal{A}_{\phi_{x_0}}(0)$ by knowing $x_0, \{x_i\}$. Using local principal analysis, we know that for each observation x_i, there is a local coordinate $\tau_i = \tau(x_i, x_0, U_{x_0})$ in the tangent space, when we x_0 is used as the origin of the coordinate. By projecting onto the tangent space, we have the local coordinate, and the closed-form solution is

$$\tau_i = U_{x_0}^T(x_i - x_0). \tag{10}$$

When $\{\tau_i, U_{x_0}, x_0\}$, we can determine the tensor $\mathcal{A}$ via a least-squares problem. The global coordinate x_i has a second-order approximation:

$$x_i - (x_0 + U_{x_0}\tau_i + U_{x_0}^{\perp}\mathcal{A}(\tau_i,\tau_i)) = o(\|\tau_i\|_2^2). \tag{11}$$

We should find a tensor $\mathcal{A}$ such that the remainder is a higher-order item. Substituting (10) into (11), we have

$$\begin{aligned}&x_i - (x_0 + U_{x_0}\tau_i + U_{x_0}^{\perp}\mathcal{A}(\tau_i,\tau_i)) \\ =&U_{x_0}^{\perp}({U_{x_0}^{\perp}}^T(x_i - x_0) - \mathcal{A}(\tau_i,\tau_i)) = o(\|\tau_i\|_2^2),\end{aligned}$$

which is equivalent to

$${U_{x_0}^{\perp}}^T(x_i - x_0) - \mathcal{A}(\tau_i,\tau_i) = o(\|\tau_i\|_2^2). \tag{12}$$

As (12) is a vector form, split (12) into each dimension: for instance, for the k-th dimension in the normal space, we have

$$\begin{aligned}(u_{x_0}^k)^T(x_i - x_0) =&\mathcal{A}_{\cdot\cdot k}(\tau_i,\tau_i) + o(\|\tau_i\|_2^2) \\ =&\tau_i^T S^k \tau_i + o(\|\tau_i\|_2^2).\end{aligned} \tag{13}$$

In (13), because of the symmetric position of τ_i, we know that the best-fitted S^k is symmetric; i.e., each slice of $\mathcal{A}_{\cdot\cdot k}$ is a symmetric matrix. Therefore, the unknown parameters in S^k yield a total number of $d(d+1)/2$.

In the following section, we will show how to vectorize each slice of $\mathcal{A}$ and how to convert the vectorized result into the matrix form. We also show that the quadratic form equals the inner product of two vectors:

$$\tau_i^T S^k \tau_i = 2\mathrm{vech}(\tau_i\tau_i^T, 1)^T \mathrm{vech}(S^k, 1/2).$$

2.4. Closed-form of S^k by Vectorization

For any symmetric matrix A, we know $A_{ij} = A_{ji}$. Therefore, we need only $d(d+1)/2$ elements with the corresponding order to restore it. As a result, we can vectorize only the upper-triangle elements in the matrix A as a vector:

$$\mathrm{vech}(A,t)_{\frac{(2d-i)(i-1)}{2}+j-i+1} = \begin{cases} A_{ij}, & j > i \\ tA_{ij}, & j = i, \end{cases}$$

where the diagonal elements are multiplied by a scalar t, which will make the succeeding notations more convenient. When $t = 1$, the vector is constructed by picking the upper-triangle elements of A including the diagonal ones, i.e.,

$$\mathrm{vech}(A,1) = [A_{11}, A_{12}, ...A_{1d}, ...A_{dd}]^T.$$

When $t = 1/2$, the vector is constructed by picking the upper-triangle elements of A, and half of the diagonal elements, i.e.,

$$\mathrm{vech}(A,1/2) = [A_{11}/2, A_{12}, ...A_{1d}, ...A_{dd}/2]^T.$$

Note that we can easily recover the matrix A from the vector $\text{vech}(A, 1/2)$ via

$$A = \text{Mat}(\text{vech}(A, 1/2)) + \text{Mat}^T(\text{vech}(A, 1/2))$$

where $\text{Mat}(y)$ is an operator constructed by realigning the elements in y into an upper-triangle matrix $\text{Mat}(y)$ such that the i, j-th elements are equal to

$$\text{Mat}(y)_{i,j} = \begin{cases} y_{(2d-i)(i-1)/2+j-i+1}, & j \geq i \\ 0, & j < i \end{cases}$$

With the above notations, the quadratic form $x^T Ax$ can be written in the vectorized version as

$$x^T Ax = 2\text{vech}(xx^T, 1)^T \text{vech}(A, 1/2),$$

where $\text{vech}(xx^T, 1)$ is a vectorization of the symmetric matrix xx^T including the diagonal ones.

Because of the symmetric nature of the matrix $\tau_i \tau_i^T$ and S^k, using the above notations, it can be easily verified that

$$\tau_i^T S^k \tau_i = \langle \tau_i \tau_i^T, S^k \rangle = 2\text{vech}(\tau_i \tau_i^T, 1)^T \text{vech}(S^k, 1/2).$$

For notational convenience, we let $\theta_k = \text{vech}(S^k, 1/2)$ and $g_i = \text{vech}(\tau_i \tau_i^T, 1)$. Using the vector notations, the equation in (13) can be converted as follows:

$$g_i^T \theta_k - \frac{1}{2}(u_{x_0}^k)^T (x_i - x_0) = o(\|\tau_i\|_2^2). \tag{14}$$

For notational convenience, we use z_i^k to stand for the local coordinate in the k-th normal dimension corresponding to x_i, i.e.,

$$z_i^k = \frac{1}{2}(u_{x_0}^k)^T (x_i - x_0).$$

From (14), we know that $g_i^T \theta_k - z_i^k = o(\|\tau_i\|_2^2)$. To determine the $d(d+1)/2$ parameters in θ_k, we need $d(d+1)/2$ linear independent equations. In other words, we need at least $d(d+1)/2$ samples to construct $d(d+1)/2$ linear independent $\{g_i, i = 1, ..., d(d+1)/2\}$.

Suppose we have m samples on the manifold. Then, the matrix G and the vector ℓ_k can be expressed as follows:

$$G = [g_1, ..., g_m]^T, \quad \ell_k = [z_1^k, ..., z_m^k]^T.$$

2.5. Sample-Related Weights

Using the samples to estimate θ_k in (14), it should be noted that we want a locally fitting model, which means we want $\mathcal{M}_{\mathcal{A}}$ to be defined as the range of $x_{\mathcal{A}}(\tau)$:

$$\mathcal{M}_{\mathcal{A}} = \{x : x_{\mathcal{A}}(\tau) = U_\perp \mathcal{A}(\tau, \tau) + U\tau + x_0, \tau \in \mathbb{R}^d\}, \tag{15}$$

to fit $\mathcal{M}$ well when $\|\tau\|_2$ is relatively small. To achieve this, the points that reside near x_0 should have a larger weight. Then, by using a nonlinear kernel function $K_h(\cdot)$, the optimal minimization problem with respect to the k-th dimension becomes

$$\begin{aligned} &\min_{\theta_k} \sum_{i=1}^m K_h(x_i - x_0)\{g_i^T \theta_k - \frac{1}{2}(u_{x_0}^k)^T (x_i - x_0)\}^2 \\ &= \min_{\theta_k} \|W_h^{1/2}(G\theta_k - \ell_k)\|_2^2, \end{aligned} \tag{16}$$

where W_h is a diagonal matrix and the i-th diagonal element is $\{W_h\}_{ii} = K_h(x_i - x)$.

3. Manifold Fitting via Nonlinear Projection

In the previous section, we show how to obtain the parameterized manifold $x_{\mathcal{A}}$ by taking the second-order term $\mathcal{A}$ into consideration via the linear-least-squares problem. In this section, we show how to implement a nonlinear projection on our fitted manifold. The manifold-fitting problem can be viewed as projecting data x onto a locally approximated structure. The existing manifold-fitting methods all consider this local structure a linear affine plane (parameter with τ):

$$x(\tau) = U\tau + x_0.$$

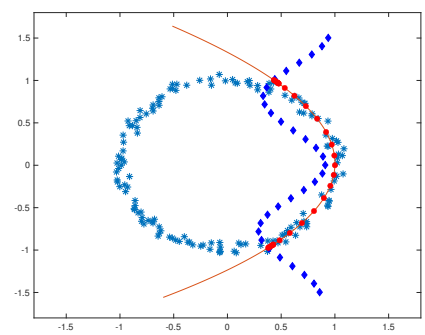

Figure 3. Demo for the projection via solving the nonlinear least-squares problem. The blue stars '*' are the discrete samples that are assumed to be approximately sampled from some manifold. The paraboloid curve is our fitted manifold with the second-order term taken into consideration. The blue diamonds '$\diamond$' are the outliers, which can be of any type or distribution. In this demo, we let the outliers be samples from the sin curve. The red dots '$\circ$' are the projections of '$\diamond$' onto the paraboloid curve (the fitted manifold). In this section, we provide the algorithm to find the projection of outliers onto the fitted second-order function.

Different fitting approaches differ in terms of the construction of U and x_0. In our work, we can have a fitted paraboloid (quadratic) surface, with the following form:

$$x_{\mathcal{A}}(\tau) = U_\perp \mathcal{A}(\tau, \tau) + U\tau + x_0, \tag{17}$$

where x_0 is the origin (center for the local coordinate) and $\mathcal{A}(\tau, \tau)$ represents the second fundamental form, which is

curvature-function mapping from the tangent space to the normal space.

3.1. Projection onto Manifold

Suppose we have a locally fitted manifold whose parameterized form is written as $x(\tau) : \mathbb{R}^d \to \mathbb{R}^D$. Since the range of the function $x_{\mathcal{A}}(\tau)$ generates a paraboloid manifold, we express it as

$$\mathcal{M}_{\mathcal{A}} = \{y : y = x_{\mathcal{A}}(\tau), \tau \in \mathbb{R}^d\}.$$

Given any $z \notin \mathcal{M}_{\mathcal{A}}$, we define the projection $P_{\mathcal{M}_{\mathcal{A}}}(z)$ of z onto $\mathcal{M}_{\mathcal{A}}$ as

$$P_{\mathcal{M}_{\mathcal{A}}}(z) = \arg\min_{w \in \mathcal{M}_{\mathcal{A}}} \|w - z\|_2. \tag{18}$$

Using the function $x_{\mathcal{A}}(\tau)$, we know the points on $\mathcal{M}_{\mathcal{A}}$ and the Euclidean space R^d are one-to-one. As a result, instead of having to find w in (18), we just need to find τ:

$$\hat{\tau} = \arg\min_{\tau \in R^d} \|z - x_{\mathcal{A}}(\tau)\|_2^2. \tag{19}$$

Finally, using the explicit form $x_{\mathcal{A}}(\tau)$, we know the projection yields

$$P_{\mathcal{M}_{\mathcal{A}}}(z) = x_{\mathcal{A}}(\hat{\tau}) = x(\arg\min_{\tau \in R^d} \|z - x_{\mathcal{A}}(\tau)\|_2^2).$$

Overall, the projection onto $\mathcal{M}_{\mathcal{A}}$ consists of the following two steps:

1. Solve the nonlinear problem. Given z, find τ such that

$$\hat{\tau} = \min_{\tau} \|z - (U_{\perp}\mathcal{A}(\tau, \tau) + U\tau + x_0)\|_2^2. \tag{20}$$

2. Construct the coordinate in the ambient space:

$$\hat{z} = U_{\perp}\mathcal{A}(\hat{\tau}, \hat{\tau}) + U\hat{\tau} + x_0.$$

To solve the nonlinear projection problem (20), we can simplify it as

$$\begin{aligned}
&\|z - (U_{\perp}\mathcal{A}(\tau, \tau) + U\tau + x_0)\|_2^2 \\
=&\|P_U(z - x_0) - U\tau\|_2^2 + \|P_{U_{\perp}}(z - x_0) - (U_{\perp}\mathcal{A}(\tau, \tau))\|_2^2 \\
=&\|U(U^T(z - x_0) - \tau)\|_2^2 + \|U_{\perp}(U_{\perp}^T(z - x_0) - \mathcal{A}(\tau, \tau))\|_2^2 \\
=&\|U^T(z - x_0) - \tau\|_2^2 + \|U_{\perp}^T(z - x_0) - \mathcal{A}(\tau, \tau)\|_2^2.
\end{aligned} \tag{21}$$

The optimization problem in (21) is nonlinear: the highest-order term corresponding to τ is quartic. As far as we know, there is no closed-form solution to directly minimize (21). The difficulty originates from the relatively higher order. If we can decrease the order with respect to τ to 2, we will have a closed form. In the next section, we show how to obtain the optimum τ via solving a series of quartic optimization problems.

3.2. Quadratic Approximation

In this section, we solve the quartic optimization problem by repeatedly implementing the quadratic approximation. First, for notational convenience, we let

$$s = U^T(z - x_0) \in \mathbb{R}^d, \quad c = U_{\perp}^T(z - x_0) \in \mathbb{R}^{D-d},$$

where the bases U^T and $U_{\perp}^T$ can be obtained by solving the local principal component analysis (LPCA) problem centered at x_0. Consequently, s, c can be obtained from U^T, $U_{\perp}^T$, and x_0. With these notations, the problem of (21) can be converted into

$$f(\tau) = \|s - \tau\|_2^2 + \|c - \mathcal{A}(\tau, \tau)\|_2^2. \tag{22}$$

Note that $\mathcal{A}(\tau, \tau)$ is the quadratic form with respect to τ, and $\|c - \mathcal{A}(\tau, \tau)\|_2^2$ is the quartic term. Even though $f(\tau)$ is a polynomial function with the input as a vector τ, it is still difficult to obtain the closed form for τ. Based on this observation, we reduce the order of $f(\tau)$ by bringing in the auxiliary function $g(\tau_1, \tau_2)$, which is defined as

$$g(\tau_1, \tau_2) = \frac{1}{2}\|s - \tau_1\|_2^2 + \frac{1}{2}\|s - \tau_2\|_2^2 + \|c - \mathcal{A}(\tau_1, \tau_2)\|_2^2.$$

Because of the symmetric property of $g(\tau_1, \tau_2)$ with respect to τ_1 and τ_2, we have

$$g(\tau_1, \tau_2) = g(\tau_2, \tau_1).$$

When fixing τ_1, the quadratic function $g(\tau_1, \tau)$ approximates $f(\tau)$ with a small error when τ approaches τ_1. For the extreme case, when $\tau = \tau_1$, we have $g(\tau_1, \tau_1) = f(\tau_1)$. By minimizing via alternating the fixing of τ_1 and τ_2, we can derive the sequence $\{\tau_n, n = 1, 2..., \}$ in (23) by

$$\begin{aligned}
\tau_{n+1} &= \arg\min_{\tau} g(\tau_n, \tau) \\
&= \arg\min_{\tau} \frac{1}{2}\|s - \tau_n\|_2^2 + \frac{1}{2}\|s - \tau\|_2^2 + \|c - \mathcal{A}(\tau_n)\tau\|_2^2.
\end{aligned} \tag{23}$$

For notational convenience, we define $\phi(\tau_n)$ as the optimal minimum solution of (23), i.e.,

$$\phi(\tau_n) = (2\mathcal{A}(\tau_n)\mathcal{A}(\tau_n)^T + I_d)^{-1}(2\mathcal{A}(\tau_n)c + s).$$

Using $\phi(\tau_n)$, we can define a vector sequence $\{\tau_n, n = 1, 2..., \}$ via the fixed-point iteration with $\tau_{n+1} = \phi(\tau_n)$, and set the convergence point τ^* as the coordinate for the true projection.

4. Algorithm: Projection by Repeated Nonlinear Least Squares

Because the underlying manifold is unknown, in real computational cases, we cannot find a point $x_0 \in \mathcal{M}$ that is also next to our interested outlier $\bar{x}$. To solve this problem, we

use an iteration method to find x_0. When $\bar{x}$ is far away from $\mathcal{M}$, the inaccuracy of x_0 is also acceptable. However, as $\bar{x}$ approaches $\mathcal{M}$, we need the accuracy of x_0 to be improved. Since the observations $\{x_i, i = 1, ..., n\}$ are drawn from $\mathcal{M}$ with noise, for the outlier $\bar{x}$, we can select x_0 as the weighted-mean:

$$x_0 = \frac{1}{\sum_{i \in \mathcal{N}_k(\bar{x})} \phi_h(\bar{x}, x_i)} \sum_{i \in \mathcal{N}_k(\bar{x})} \phi_h(\bar{x}, x_i) x_i, \quad (24)$$

where $\phi_h(\bar{x}, x_i) = K_h(x_i, \bar{x})$ is the kernel weight, $\mathcal{N}_k$ controls the neighbor size, and h is the bandwidth parameter, affecting the bias and smoothness.

Algorithm 1 Fitting Algorithm:

1. For outlier $\bar{x}$, compute the shift mean x_0 from (24) and use x_0 as the origin of our local coordinate to implement our fitting and projection process.
2. Given x_0, neighborhood size k, and bandwidth parameters h, obtaining the local coordinate $\{\tau_i, \iota_i\}$ for each x_i by applying the eigen-decomposition method. $\tau_i = U_d^T(x_i - x_0)$. Using τ_i, construct matrix G from $\{\tau_i\}$ such that $G = [g_1, ..., g_m]^T$, where each column consists of $g_i = \mathrm{vech}(\tau_i \tau_i^T, 1)$.
3. Solve the manifold-fitting problem for each dimension in the normal space; e.g., for the k-th dimension,

$$\min_{\theta_k} \sum_{i=1}^{m} K_h(x_i - \bar{x}) \{g_i^T \theta_k - (u_{x_0}^k)^T (x_i - x_0)/2\}^2$$
$$= \min_{\theta_k} \|W_h^{1/2}(G\theta_k - \ell_k/2)\|_2^2,$$

where the m (number of samples in the neighborhood) dimensional vector ℓ_k equals $[u_k^T(x_1 - x_0), u_k^T(x_2 - x_0), ..., u_k^T(x_n - x_0)]$.
4. For each $k = 1, ..., D - d$, transform the vector θ_k into the matrix form by putting the elements of θ_k onto the upper-triangle positions to obtain $\mathrm{Mat}(\theta_k)$. Subsequently, the symmetric S_k is obtained via $S_k = \mathrm{Mat}(\theta_k) + \mathrm{Mat}(\theta_k)^T$. By aligning each slice of S_k, we obtain the tensor $\mathcal{A}$, such that $\mathcal{A}_{..k} = S_k$. Here, we get a parametrized manifold $\mathcal{M}_\mathcal{A}$ to fit the complicated $\mathcal{M}$ locally, where $\mathcal{M}_\mathcal{A}$ yields the following form:

$$x_\mathcal{A}(\tau) = U_d^\perp \mathcal{A}(\tau, \tau) + U_d \tau + x_0.$$

Algorithm 2 Projection Algorithm:

1. For an outlier $\bar{x}$, set $\tau_0 = U_d^T(\bar{x} - x_0)$ and apply the fixed-point iteration, to get the convergence point τ^* of the sequence $\{\tau_n\}$. The fixed-point iteration is

$$\phi(\tau_n) = (2\mathcal{A}(\tau_n)\mathcal{A}(\tau_n)^T + I_d)^{-1}(2\mathcal{A}(\tau_n)c + s).$$

2. Put τ^* onto the fitted function to obtain the point $\hat{x}$, which is the projection of $\bar{x}$ onto $\mathcal{M}_\mathcal{A}$:

$$\hat{x} = P_{\mathcal{M}_\mathcal{A}}(x) = U_\perp \mathcal{A}(\tau^*, \tau^*) + U\tau^* + x_0.$$

Our Algorithm 3 consists of steps repeatedly implementing the fitting and projection procedures. The fitting procedures intend to find the parameter of the tensor $\mathcal{A}$ from the observations, which represents the curvature of the local manifold in each of the dimensions for the normal space. When the fitting process is finished, we get the closed-form representation of $x_\mathcal{A}$. The projection procedures are intended to find the projection of $\bar{x}$ on $\mathcal{M}_\mathcal{A}$ (or the nearest point from $\bar{x}$ to $\mathcal{M}_\mathcal{A}$).

Algorithm 3 Iterative Fitting and Projection Algorithm:

Input: Outliers $\{\bar{x}_k\}$, data $\{x_i\}$, parameter h.
Output: Projected result $\hat{x}$.
for each outlier $\bar{x} \in \{\bar{x}_k\}$ **do**
  Set $\hat{x}' = \bar{x}$.
  repeat
    Implement *Algorithm* 1 to get $x_\mathcal{A}$.
    Implement *Algorithm* 2 to get the projection $\hat{x}$.
  until $\|\hat{x}' - \hat{x}\|_2 \leq \epsilon$, **otherwise:** Set $\hat{x}' = \hat{x}$
end for

5. Simulation

To evaluate the performance of different results, we define the criteria $c(\hat{\mathcal{M}})$, which represents the percentage of improvement of the corresponding algorithm

$$c(\hat{\mathcal{M}}) = 1 - \frac{d(\hat{\mathcal{M}}, \mathcal{M})}{d(\mathcal{D}, \mathcal{M})},$$

where $\mathcal{D}$ stands for the set corresponding to '$\diamond$' which is the outlier we want to pull towards the underlining $\mathcal{M}$ and $\hat{\mathcal{M}}$ stands for the set corresponding to '$\circ$' which is the result of different methods. The distance of $d(\mathcal{D}, \mathcal{M})$ is defined as:

$$d(\mathcal{D}, \mathcal{M}) = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \|x_i - \tilde{x}_i\|_2 = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \sigma_2 \|\epsilon_i\|_2.$$

By replacing $\mathcal{D}$ with $\hat{\mathcal{M}}$, we can similarly obtain $d(\hat{\mathcal{M}}, \mathcal{M})$.

5.1. Data-Recovery Capability

In this section, we construct some artificial manifolds, e.g., the circle and the Swiss-roll, which can be written as a parameterization form: $\begin{cases} x = \phi(t) \\ y = \psi(t) \end{cases}$. Thus, we have the tangent space spanned by the vector $(\phi'(t), \psi'(t))$ and the normal space spanned by the vector $(\psi'(t), -\phi'(t))$. The curvature at t can be calculated as $\kappa(t) = \frac{\phi'(t)\psi''(t)-\phi''(t)\psi'(t)}{((\phi'(t))^2+\psi'(t)^2)^{3/2}}$. In particular, when the curve is parameterized as $(x, f(x))$, the curvature yields quite a simple form: $\kappa(x) = \frac{|f''(x)|}{(1+f'(x))^{3/2}}$. Next, we provide three examples and show the performance of our algorithm. The first example, a 2-D circle, is a curve with constant curvature, and the other two examples are curves with varying curvatures. The Swiss-roll has a curvature monotonously decreasing with the parameter θ.

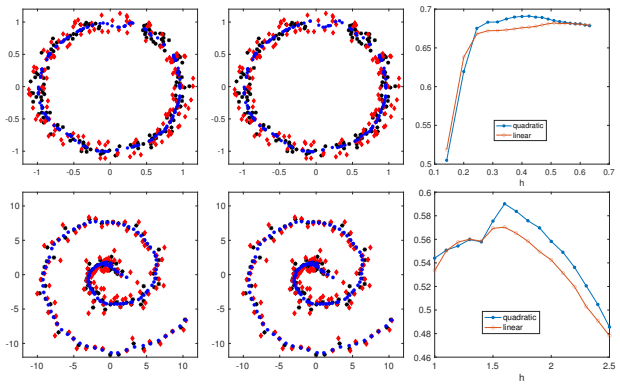

Figure 4. Comparison between nonlinear projection and linear projection. Left: nonlinear approach. Right: linear approach.

It is difficult to fit the data drawn from a manifold which is assumed to have varying curvature using a linear model or a tangent plane. For x in an area that leads to a small $\kappa(\theta)$, the manifold can be approximated by a low-dimensional affine space (a flat plane). However, in an area that has a large curvature $\kappa(\theta)$, such as $\theta = \pi/2$ in our triangle curve case, it is impossible to approximate $y = \sin(\theta)$ with a flat structure. Therefore, it is necessary to use a more complicated model to approximate.

Note that our nonlinear second-order fitting function $x_{\mathcal{A}}$ is a generalization of the linear function x_ℓ. When the second-order parameter $\mathcal{A}$ equals zero, the derived manifold $\mathcal{M}_{\mathcal{A}}$ will degenerate into $\mathcal{M}_\ell$. By learning $\mathcal{A}$, we automatically consider the curvature information hidden in the manifold. Since our nonlinear second-order fitting model $\mathcal{M}_{\mathcal{A}}$ is more complicated by virtue of having more parameters than $\mathcal{M}_\ell$, we need to use more data to solve our model. Solving the least-squares problem with too little data will lead to a phenomenon called 'overfitting'. When the overfitting phenomenon occurs, the model will learn not only from the true signal of the underlining manifold, but also from the noise factors that will distort the model itself.

In our fitting model, the number of points used is controlled by the bandwidth parameter h. Because the kernel weight decreases quickly with the radius, the points reside far from our area of interest, and hence their contribution can ignored. As a result, our model works well with a relatively large h, which can be seen in the rightmost partition of Figure (4). In Figure (4), we randomly sample the black stars '*' as $x_i = \tilde{x}_i + \sigma_1\epsilon_i$ and the red diamonds '$\diamond$' as $x_i = \tilde{x}_i + \sigma_2\epsilon_i$, where $\tilde{x}_i$ is on the underlying $\mathcal{M}$ and ϵ_i is in the normal space of $\mathcal{M}$ at $\tilde{x}_i$. In our experimental setting, we have $\sigma_1 = 0.2$ and $\sigma_2 = 0.5$ and the length of ϵ_i obeys the normal distribution of $N(0, 1)$. The blue dots '$\circ$' represent the result, which is the projection onto the fitted structure.

The leftmost figure shows the result obtained from the projection onto the nonlinear $\mathcal{M}_{\mathcal{A}}$, and the middle figure shows the result obtained from the projection onto $\mathcal{M}_\ell$. The rightmost figure shows that, with the increase of the bandwidth h, the nonlinear projection perform better (under the measurement of $c(\hat{\mathcal{M}})$) in terms of data recovery than the linear projection.

5.2. Real Data

COIL-20 (S. A. Nene & Murase, 1996) is a classical manifold image dataset where the photos are taken by rotating an object at a constant speed. As a consequence, the photos of the same object can be considered to be drawn from a one-dimensional manifold and disturbed by some noise, with the intrinsic variable being the rotated angle. To speed up the algorithm, eigen-face is a suitable dimension-reduction method for images. Let Img denote the reshape-operation by transforming a vector of size $m * n$ into a matrix of shape $m \times n$. We thus assume the images are generated as $I = \mathrm{Img}(U_p(\tilde{x} + \epsilon_1)) + \epsilon_2, \tilde{x} \in \mathcal{M}$, where $\tilde{x}$ is assumed to be drawn from a d dimensional manifold. The length of the vector $\tilde{x}$ is p, which indicates that the dimension of the ambient space is p. The columns of U_p span a p-dimensional subspace in the $m * n$ dimensional Euclidean space. Since the images $\{I_1, ..., I_n\}$ of the same object share the same U_p, we can recover U_p from the image set through the eigen-decomposition operation. Then, from the assumption, we know that the p dimensional ambient space, with the coordinates $\{v_i = U_p^T \mathrm{vec}(I_i), i = 1, ..., k\}$, of the images under the basis U_p lies approximately on the d dimensional manifold. Therefore, we only need to recover a d dimensional manifold in the ambient space of dimension p.

With the insertion of the original photo, we can recover a new photo, which can be considered a sample from the underlying smooth manifold. In Figure 5, the odd rows are the inserted original photos and the even rows are the photos

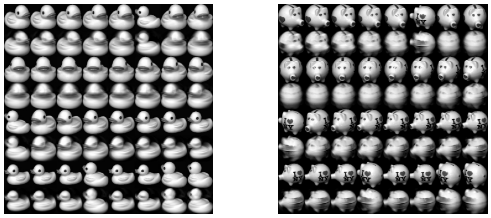

Figure 5. The performance of a nonlinear manifold with Coil20 dataset

produced by our algorithm. Comparing the original and produced photos, we can see that the sudden-change areas of the original photos have been smoothed, and the rotating trend has become more apparent than before.

References

Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Davenport, M. A., Hegde, C., Duarte, M. F., and Baraniuk, R. G. Joint manifolds for data fusion. *IEEE Transactions on Image Processing*, 19(10):2580–2594, 2010.

Genovese, C. R., Perone-Pacifico, M., Verdinelli, I., Wasserman, L., et al. Nonparametric ridge estimation. *The Annals of Statistics*, 42(4):1511–1545, 2014.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Ozertem, U. and Erdogmus, D. Locally defined principal curves and surfaces. *Journal of Machine learning research*, 12(Apr):1249–1286, 2011.

Panaretos, V. M., Pham, T., and Yao, Z. Principal flows. *Journal of the American Statistical Association*, 109(505): 424–436, 2014.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326, 2000.

S. A. Nene, S. K. N. and Murase, H. Columbia object image library (coil-20). *Technical Report CUCS-005-96*, 1996.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Zhang, Z. and Zha, H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338, 2004.