Manifold Fitting and Projection via Quadratic Approximation

Anonymous Authors[1]

Abstract

In this paper, we propose to fit the manifold by 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 precisions. Because more complicated function is adopted in our manifold fitting process, the pulling back approach turns from the linear least square problem into a nonlinear quartic minimization problem. By bringing in the auxiliary function, we solve the quartic by repeatedly solving a series of quadric minimization problems. Numerical experiments demonstrate that our method has a very strong recovery capability compared with the current works.

1. Introduction

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

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

To handle the high-dimensional data, there are mainly two approaches. First, reconstruct the data in the low-dimensional space, such that, the reconstructed data in the low-dimensional space also keeps the relation (such as distance, geometry, affine transformation etc) in high dimensional space. This approach containing lots of classical dimension reduction works, such as LLE (Roweis & Saul, 2000), LTSA (Zhang & Zha, 2004), Isomap (Tenenbaum et al., 2000), Eigenmap (Belkin & Niyogi, 2003).

The other approach try to fit a new smooth manifold by reducing the noise existing 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 the manifold fitting by transforming the manifold fitting problem into the ridge estimation problem. The ridge is defined as the set of points which satisfy some relation on gradient and Hessian of the density function. Overall, all the manifold fitting process can be considered with two steps. First, estimate the tangent space which is often implemented through the eigenvalue-decomposition of the covariance matrix. Second, estimate an attraction force which is a direction which points to somewhere near the manifold, this direction often constructed via the weighted shift-mean approach. By using the subspace constraint mean shift algorithm, we can move the outlier onto the manifold to some degree.

The approximation error for the local affine space will yield an approximate error of $\|\tau\|_2^2$, where τ is the local coordinate. In other words, when we restrict the error subjects to the condition $\|\tau\|_2^2 \leq \epsilon$, the maximum radius of the valid region is with the order of $\epsilon^{1/2}$. In this paper, we propose to approximate the manifold via the quadric function, by adopting this strategy, the approximation error will become $\|\tau\|_2^3$. With the same precision requirement ϵ, we have the maximum radius of our interested region is of the order $\epsilon^{1/3}$. Normally, ϵ is some scalar less than 1, in this way, the quadratic approximation is a better approximation compared with the tangent affine plane.That is to say, to achieve the same precision, we will need much less times of fitting process.

A good approximation of the underlying manifold is the first step in our manifold fitting problem. The second important step needs to project the noisy data onto the fitted manifold which we obtained in the first step. Here, the term ‘project’ has two meanings. First, the distance should be minimum,

[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.

second, the direction of the projection should be perpendicular with the tangent place of the fitted manifold. A major reason for the linear approximation to be very widely used is the form of projection yields a very simple linear least square problem.

1.1. Fitting Model

In this paper, we are not concentrating on finding the representation of $\phi(\tau)$ under the noiseless assumption. Instead, we assume to have the observations drawn from some low-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 which obeys some distribution, such as the multi-dimensional gaussian distribution.

Since the observations $\{x_i, i = 1 : n\}$ are discrete 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 linear or 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:

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

The works such as (Genovese et al., 2014; Ozertem & Erdogmus, 2011) all focus on how to get a better affine space to locally approximate the distribution of the data, i.e, in their works, $\mathcal{G}(\theta)$ is a linear model. However, at far as we know, the result of linear approximation approach relies heavily on the selection of the origin point (the red dot in Figure 1). An origin selected with good quality will surly improve the ability of recovery of the projection. 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

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

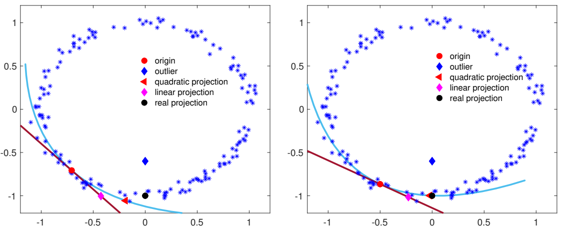

Figure 1. The reliance of the origin point in the process of manifold fitting and projection

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.

In other words, corresponding to x_0 and $\mathcal{M}$, there is some radius r, such that:

$$\mathcal{M} \cap \mathcal{D}_{x_0}(r) = \{y | y = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\phi_{x_0}(\tau), \tau \in \mathbb{R}^d\} \cap \mathcal{D}_{x_0}(r). \tag{3}$$

where $\mathcal{D}_{x_0}(r) = \{x | \|x - x_0\| \leq r\}$, $\phi_{x_0}(\tau) = [\phi_{x_0}^{d+1}(\tau), ..., \phi_{x_0}^{D}(\tau)]^T$ is the function defined on the tangent space, the range of $\phi_{x_0}(\tau)$ represents the coordinate in the normal space and τ is the local coordinate in the tangent space.

To demonstrate the difference behaviors corresponding to the linear and quadratic fitting approaches, we give 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. This example shows the effect of the different fitting errors, which is $O(\|\tau\|_2^2)$ and $O(\|\tau\|_2^3)$ corresponding to the linear and quadratic forms, respectively. From this case, we know that the linear approach relies on a good origin heavily than the quadratic form because the approximation by linear case yields a lower-order error comparing with the quadratic case.

In this paper, we transfer our manifold fitting problem into finding an approximation version of $\phi_{x_0}(\tau)$ through a deep analysis on the characteristic of it. We also show the dominant term for 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 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 parameters $\mathcal{A}$ can be obtained via solving a linear least square 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 the projection of $\bar{x}$ onto $\mathcal{M}_{\mathcal{A}}$ can be achieved by repeatedly solving a series of linear least square problems. The strengths of our method can be summarized as:

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

2. Our algorithm does not rely on the accuracy of the estimated origin point too much, 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 (nearby x) which 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 got 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

There are two meanings of the fitting model. The first refers to locally fitting a complicated function by a simple form such as Taylor expansion. The second indicates we use a generalized representation such that the measurement with respect to the observations and the representation is optimal under some criteria.

For a complicate nonlinear function $\phi(\tau)$, we can fit locally via the lower order Taylor expansion such as

$$\begin{aligned}\phi_q(\tau) =& \phi(\tau_0) + \langle \nabla_\tau \phi(\tau)|_{\tau=\tau_0}, \tau - \tau_0\rangle + ... \\ &+ \frac{1}{2}\nabla_\tau\nabla_\tau\phi(\tau)|_{\tau=\tau_0}(\tau-\tau_0,\tau-\tau_0).\end{aligned} \tag{4}$$

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

2. The quadratic function (4) can be thought as a locally fitting model of $\phi(\tau)$ at $\tau = \tau_0$ with the 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 show a simple demo for our motivation.

Example 1. The data of blue circles evenly distributed on a 2-D sphere as shown in (2) is a toy example of manifold. Fixing a point $x_0 \in \mathbb{S}^2$, we can have a tangent place. The yellow circles are the projection of the data onto the tangent plane. The continuous elliptical paraboloid surface is the structure what we wanted.

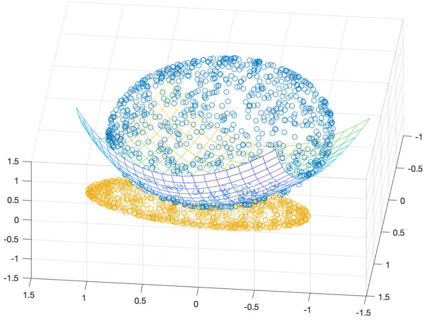

Figure 2. Fitting a manifold with the elliptical paraboloid surface

Obviously, from Figure 2 we can see 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 matrix of size 2×2 which is used to control the shape and the direction of the paraboloid surface. The eigenvalues of A represents the curvature or flatness in the corresponding eigenvector direction.

Remark. In 3-D space, obviously, when the paraboloid surface is at one side of the tangent plane, the matrix A is positive definite or negative definite.

2.2. Quadratic surface in high-dimensional space

Instead of fitting the manifold by 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 $\mathcal{M}$ locally at x_0 can be written as the range of function $x(\tau)$

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

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 (5), we know that there is a one-to-one correspondence between $\phi_{x_0}(\tau)$ and $x(\tau)$ as:

$$\phi_{x_0}(\tau) = {U_{x_0}^{\perp}}^T(x(\tau) - x_0) \tag{6}$$

From (6), we know 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.

3, $U_{x_0}^{\perp}$ is the orthogonal directions at x_0 which can also be obtained from the local PCA.

4, The only remaining part is to $\phi_{x_0}(\tau)$.

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 as

$$C(x_0) = [U_d, U_{D-d}] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_d, U_{D-d}]^T. \tag{7}$$

From the decomposition in (7), 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

$$\begin{aligned}\phi_{x_0}(\tau) =& \phi_{x_0}(0) + \nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau) + \\ &+ \frac{1}{2}\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}(\tau,\tau) + O(\|\tau\|_2^3),\end{aligned} \tag{8}$$

where $\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$ stands for the second fundamental form which is a three order tensor with the 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 with the size of $D-d$.

Recalling that (8), 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 denote the tensor as $\mathcal{A} = \frac{1}{2}\nabla\nabla\phi_{x_0}(\tau)|_{\tau=0}$ and define a function $x_{\mathcal{A}}(\tau)$ related with $\mathcal{A}$ as

$$x_{\mathcal{A}}(\tau) = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\mathcal{A}_{\phi_{x_0}}(\tau,\tau). \tag{9}$$

From the range $x_{\mathcal{A}}(\tau)$ of (9), we have a new manifold $\mathcal{M}_{\mathcal{A}}$ derived from $x_{\mathcal{A}}(\tau)$ as

$$\mathcal{M}_{\mathcal{A}} = \{z | z = x_0 + U_{x_0}\tau + U_{x_0}^{\perp}\mathcal{A}_{\phi_{x_0}}(\tau,\tau), \tau \in \mathbb{R}^d\} \tag{10}$$

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

Remark. In most of the manifold fitting cases, both of the true manifold $\mathcal{M}$ and the function $\phi_{x_0}(\tau)$ are unknown, as a result, it is impossible for us to get the second order parameter $\mathcal{A}_{x_0}$ through differentiating the multivariable function $\phi_{x_0}(\tau)$.

Instead of getting the accurate $\phi_{x_0}(\tau)$, we can estimated it by using the data (observations) which is 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 $\hat{\mathcal{A}}$, as a result, the range $\mathcal{M}_{\hat{\mathcal{A}}}$ of $x_{\hat{\mathcal{A}}}(\tau)$ can be assumed 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 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 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 use x_0 as the origin of the coordinate. By projecting onto the tangent space, we have the local coordinate has the closed-form solution as

$$\tau_i = U_{x_0}^T(x_i - x_0). \tag{11}$$

When having $\{\tau_i, U_{x_0}, x_0\}$, we can determine the tensor $\mathcal{A}$ by a least square problem. The global coordinate x_i has a second order approximation as

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

We should find a tensor $\mathcal{A}$ such that the remainder is a higher-order item. Substitute (11) into (12), 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

$$ {U_{x_0}^{\perp}}^T(x_i - x_0) - \mathcal{A}(\tau_i,\tau_i) = o(\|\tau_i\|_2^2). \tag{13}$$

Noticing that (13) is a vector form, split (13) into each dimension, e.g, for the k-th dimension in the normal space, we have

$$\begin{aligned}(u_{x_0}^k)^T(x_i - x_0) =& \mathcal{A}_{\cdot\cdot k}(\tau_i,\tau_i) + o(\|\tau_i\|_2^2) \\ =& \tau_i^T S^k \tau_i + o(\|\tau_i\|_2^2).\end{aligned} \tag{14}$$

In (14), 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 yields 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 make the vectorized result into the matrix form. Also, we show the quadratic form equals to the inner product of two vectors as

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

2.4. Closed-form of S^k by Vectorization

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

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

where the diagonal elements multiplied by a scalar t, which will bring us convenience for our following notations. When $t = 1$, the vector is constructed by picking the upper-triangle elements of A including the diagonal ones, i.e.,

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

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

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

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

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

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

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

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 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 denote $\theta_k = \text{vech}(S^k, 1/2)$ and $g_i = \text{vech}(\tau_i\tau_i^T, 1)$. Using the vector notations, the equation in (14) can be converted as:

$$g_i^T \theta_k - \frac{1}{2}(u_{x_0}^k)^T (x_i - x_0) = o(\|\tau_i\|_2^2). \tag{15}$$

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 (15), 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. Denote the matrix G and the vector ℓ_k as

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

2.5. Sample Related Weights

Using the samples to estimate θ_k in (15), it should be noted that we want a locally fitting model, which means we want $\mathcal{M}_\mathcal{A}$ be defined as the range of $x_\mathcal{A}(\tau)$

$$\mathcal{M}_\mathcal{A} = \{x : x_\mathcal{A}(\tau) = U_\perp \mathcal{A}(\tau, \tau) + U\tau + x_0, \tau \in \mathbb{R}^d\}, \tag{16}$$

to fit $\mathcal{M}$ well when $\|\tau\|_2$ is relatively small. To achieve this goal, the points which reside nearby 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

$$\begin{aligned} &\min_{\theta_k} \sum_{i=1}^m K_h(x_i - x_0)\{g_i^T \theta_k - \frac{1}{2}(u_{x_0}^k)^T (x_i - x_0)\}^2 \\ &= \min_{\theta_k} \|W_h^{1/2}(G\theta_k - \ell_k)\|_2^2, \end{aligned} \tag{17}$$

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 former 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 square problem. In this section, we show how to implement a nonlinear projection onto 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 as a linear affine plane (parameter with τ), such as

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

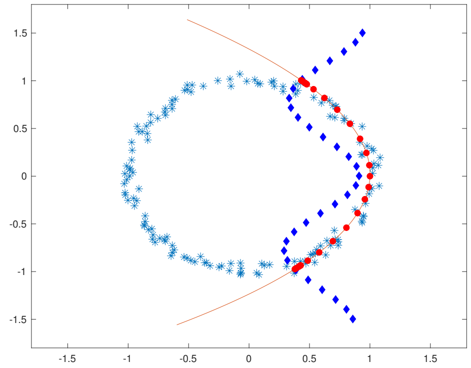

Figure 3. Demo for the projection via solving the nonlinear least problem. The blue stars '*' are the discrete samples which are assumed to be approximately sampled from some manifold. The paraboloid curve is our fitted manifold with the second order term taken into consideration. The blue diamonds '$\diamond$'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

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

$$x_{\mathcal{A}}(\tau) = U_{\perp}\mathcal{A}(\tau, \tau) + U\tau + x_0, \tag{18}$$

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 which has the parameterized form 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 denote 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

$$P_{\mathcal{M}_{\mathcal{A}}}(z) = \arg\min_{w \in \mathcal{M}_{\mathcal{A}}} \|w - z\|_2. \tag{19}$$

Using the function $x_{\mathcal{A}}(\tau)$, we know the points on $\mathcal{M}_{\mathcal{A}}$ and the Euclidean space R^d is one-to-one. As a result, instead of find w, in (19), we just need to find τ

$$\hat{\tau} = \arg\min_{\tau \in R^d} \|z - x_{\mathcal{A}}(\tau)\|_2^2. \tag{20}$$

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 as
$$\hat{\tau} = \min_{\tau} \|z - (U_{\perp}\mathcal{A}(\tau, \tau) + U\tau + x_0)\|_2^2. \tag{21}$$

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 (21), we can simplify it as

$$\begin{aligned}
&\|z - (U_{\perp}\mathcal{A}(\tau, \tau) + U\tau + x_0)\|_2^2 \\
=&\|P_U(z - x_0) - U\tau\|_2^2 + \|P_{U_{\perp}}(z - x_0) - (U_{\perp}\mathcal{A}(\tau, \tau))\|_2^2 \\
=&\|U(U^T(z - x_0) - \tau)\|_2^2 + \|U_{\perp}(U_{\perp}^T(z - x_0) - \mathcal{A}(\tau, \tau))\|_2^2 \\
=&\|U^T(z - x_0) - \tau\|_2^2 + \|U_{\perp}^T(z - x_0) - \mathcal{A}(\tau, \tau)\|_2^2.
\end{aligned} \tag{22}$$

The optimization problem in (22) is nonlinear, with the highest order term corresponding to τ is quartic. As far as we know, there is no closed-form solution to directly minimize (22). 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 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. Firstly, for notational convenience, let's denote

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

where the bases U^T and $U_{\perp}^T$ can be obtained from solving the locally principal component analysis problem (LPCA) 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 (22) can be converted into:

$$f(\tau) = \|s - \tau\|_2^2 + \|c - \mathcal{A}(\tau, \tau)\|_2^2. \tag{23}$$

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 fixing τ_1 and τ_2, we could derive a sequence of $\{\tau_n, n = 1, 2..., \}$ in (24) by

$$\begin{aligned}\tau_{n+1} &= \arg\min_{\tau} g(\tau_n, \tau)\\ &= \arg\min_{\tau} \frac{1}{2}\|s - \tau_n\|_2^2 + \frac{1}{2}\|s - \tau\|_2^2 + \|c - \mathcal{A}(\tau_n)\tau\|_2^2.\end{aligned} \tag{24}$$

For notational convenience, define $\phi(\tau_n)$ as the optimal minimum solution of (24), 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 by $\tau_{n+1} = \phi(\tau_n)$ and set the convergence point τ^* as the coordinate for the true projection.

4. Algorithm: Projection by Repeated Nonlinear Least Square

Because the underlying manifold is unknown, in real computational cases, we cannot find a point $x_0 \in \mathcal{M}$ which 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 inaccurate of x_0 is also acceptable. With the process of $\bar{x}$ approaching $\mathcal{M}$, 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 as:

$$x_0 = \frac{1}{\sum_{i \in \mathcal{N}_k(\bar{x})} \phi_h(\bar{x}, x_i)} \sum_{i \in \mathcal{N}_k(\bar{x})} \phi_h(\bar{x}, x_i) x_i, \tag{25}$$

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

Algorithm 1 Fitting Algorithm:

1. For outlier $\bar{x}$, compute the shift mean x_0 from (25) 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, get 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 as $G = [g_1, ..., g_m]^T$, where each column consists $g_i = \mathrm{vech}(\tau_i\tau_i^T, 1)$.
3. Solve the manifold fitting problem for each dimension in the normal space, e.g, solve the weighted least square problem in the k-th dimension of the normal space via minimizing:
$$\begin{aligned}&\min_{\theta_k} \sum_{i=1}^{m} K_h(x_i - \bar{x})\{g_i^T\theta_k - (u_{x_0}^k)^T(x_i - x_0)/2\}^2\\ =&\min_{\theta_k} \|W_h^{1/2}(G\theta_k - \ell_k/2)\|_2^2,\end{aligned}$$
where the m (number of samples in the neighborhood) dimensional vector ℓ_k equals to $[u_k^T(x_1 - x_0), u_k^T(x_2 - x_0), ..., u_k^T(x_n - x_0)]$.
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 $\mathrm{Mat}(\theta_k)$. Afterwards, 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 $\mathcal{A}_{\cdot\cdot k} = S_k$. Here, we get a parametrized manifold $\mathcal{M}_{\mathcal{A}}$ to fit the complicated $\mathcal{M}$ locally, where $\mathcal{M}_{\mathcal{A}}$ yields a form as
$$x_{\mathcal{A}}(\tau) = U_d^{\perp}\mathcal{A}(\tau, \tau) + U_d\tau + x_0.$$

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

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 the Algorithm 1 to get $x_{\mathcal{A}}$.
        Implement the Algorithm 2 to get the projection $\hat{x}$.
    until $\|\hat{x}' - \hat{x}\|_2 \le \epsilon$, otherwise: Set $\hat{x}' = \hat{x}$
end for
```

Our Algorithm 3 consists of steps by 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 locally manifold in each of the dimensions for the normal space. When the fitting process finishes, we get the closed-form representation of $x_{\mathcal{A}}$. The projection procedures intend to find the projection of $\bar{x}$ on $\mathcal{M}_{\mathcal{A}}$ (or the nearest point from $\bar{x}$ to $\mathcal{M}_{\mathcal{A}}$).

5. Simulation

In this section, we compare our nonlinear manifold fitting approach with the existing manifold fitting methods on various occasions. We consider the manifolds having constant curvature and varying curvature. The numerical results show our model can handle all the cases by leading a very promising result.

5.1. Data Recovery Capability

In this section, we construct some artificial manifolds e.g triangle cure, circle and the swiss-roll, which can be written as a parameterization form as $\begin{cases} x = \phi(t) \\ y = \psi(t) \end{cases}$. Thus, we have the tangent space is spanned by the vector $(\phi'(t), \psi'(t))$ and the normal space is spanned by the vector $(\psi'(t), -\phi'(t))$. The curvature at t can be calculated as

$$\kappa(t) = \frac{\phi'(t)\psi''(t) - \phi''(t)\psi'(t)}{((\phi'(t))^2 + \psi'(t)^2)^{3/2}}. \tag{26}$$

Especially, when the curve parameterized as $(x, f(x))$, the curvature yields quite a simple form as $\kappa(x) = \frac{|f''(x)|}{(1+f'(x))^{3/2}}$. Next, we give three examples and show the performance of our algorithm.

Example 2. By substituting the derivatives of the 2-D circle's parametric equation into (26). we know that the curvature of the circle $\begin{cases} x = \cos(\theta) \\ y = \sin(\theta) \end{cases}$ is the constant, such that $\kappa(\theta) = 1$ everywhere.

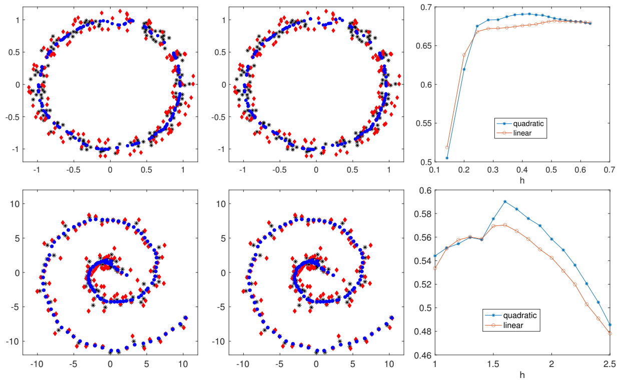

Figure 4. Comparison between nonlinear projection and linear projection. Left: nonlinear approach. Right: linear approach

Example 3. For the swiss-roll in the 2-D space, the parametric equation is $\begin{cases} x = \theta\cos(\theta) \\ y = \theta\sin(\theta) \end{cases}$. Thus, we have the first derivatives and the second derivatives:

$$\begin{aligned} x'(\theta) &= \cos(\theta) - \theta\sin(\theta), \quad y'(\theta) = \sin\theta + \theta\cos\theta \\ x''(\theta) &= -2\sin\theta - \theta\cos\theta, \quad y''(\theta) = 2\cos(\theta) - \theta\sin(\theta). \end{aligned} \tag{27}$$

Substitute (27) into the equation of curvature, we have $\kappa(\theta) = \frac{2+\theta^2}{(1+\theta^2)^{3/2}}$ and $\kappa'(\theta) = \frac{-\theta(\theta^2+4)}{(1+\theta^2)^{5/2}}$. For any $\theta \in \mathbb{R}^+$, the curvature decreases with the increasing of θ.

Remark. The first 2-D circle example is a curve with constant curvature and the remaining two examples are curves with varying curvature. The swiss-roll is an example which has the curvature monotonously decreasing with the parameter θ.

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. Because for x in an area which leads $\kappa(\theta)$ small, the manifold can be approximated by a low-dimensional affine space (a flat plane). However, in a area which has the large curvature $\kappa(\theta)$, such as $\theta = \pi/2$ in our triangle curve case, it is impossible to approximate $y = \sin(\theta)$ by a flat structure. Therefore, it is quit necessary to use a more complicated model to approximate.

Noting that, our nonlinear second-order fitting function $x_{\mathcal{A}}$ is a generalization of the linear function x_ℓ. When the second order parameter $\mathcal{A}$ equal to zero, the derived manifold $\mathcal{M}_{\mathcal{A}}$ will degenerate into $\mathcal{M}_\ell$. By learning $\mathcal{A}$, we automatically considering the curvature information hidden in the manifold.

Since our nonlinear second-order fitting model $\mathcal{M}_{\mathcal{A}}$ is more complicated by having more parameters compared with $\mathcal{M}_\ell$, we need to use more data to solve our model. Solving the least square problem with too few data will lead to a phenomenon called 'overfitting'. When the overfitting phenomenon occurs, the model will not only learn from the true signal of the underlining manifold, but also the noise factors which will distort our model.

In our fitting model, the number of points used is controlled by the bandwidth parameter h. Because the kernel weight is decreased fast with the radius, the contribution of the points resides far from our interested area is ignorable. As a result, our model works well with a relatively larger h, which can be seen in the rightmost partition of Figure (4).

In Figure (4), we randomly samples 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 underlining $\mathcal{M}$ 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 is the result obtained from the projection onto the nonlinear $\mathcal{M}_{\mathcal{A}}$ and the middle figure stands for the result obtained from the projection onto $\mathcal{M}_{\ell}$.

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 could similarly get $d(\hat{\mathcal{M}}, \mathcal{M})$.

The rightmost figure shows that with the increasing of the bandwidth h, the nonlinear projection result has a better performance (under the measurement of $c(\hat{\mathcal{M}})$) in data recovery aspect compared with the linear projection.

5.2. Real Data

The images of the same person or object taken under the different angles or illumination can be thought as data drawn from a low-dimensional manifold. The images can be thought as vector residing in the relatively high-dimensional space, because the dimension is $m * n$ for an image of size $m \times n$. In our algorithm, we need to estimate the local tangent space from the eigen-decomposition. The computational complexity is $O(m^3 * n^3)$ for the eigen-decomposition of the covariance matrix of shape $(m * n) \times (m * n)$, which is very expensive. Therefore, if we can apply some methods to reduce the dimension of images by keeping the majority useful information, the computation requirement will be largely reduced.

In order to speed up the algorithm, eigen-face is good idea to apply for images. Denote Img as the reshape-operation by transforming a vector of size $m * n$ into a matrix of shape $m \times n$. Therefore, we 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 drawn from a d dimensional manifold. The length of the vector $\tilde{x}$ is p, which indicates 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, as a result, we can recover U_p from the image set through the eigen-decomposion operation. Then, from the assumption, we know the p dimensional ambient space which the coordinates

$$\{v_i = U_p^T \mathrm{vec}(I_i), i = 1, ..., k\},$$

of the images under the basis U_p approximately lies on the d dimensional manifold. Therefore, we just need to recover a d dimensional manifold in the ambient space of dimension p.

COIL-20 (S. A. Nene & Murase, 1996) is a classical manifold image dataset where the photos are taken by rotating an object with a constant speed. As a consequence, the photos of the same object can be thought drawn from a one-dimensional manifold and disturbed by some noise where the intrinsic variable is the rotated angle.

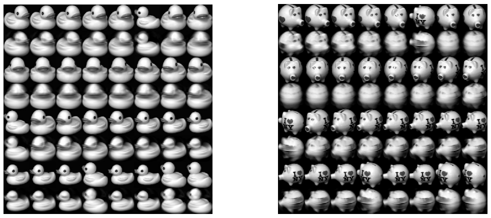

Figure 5. The performance of nonlinear manifold with Coil20 dataset

With the inputing of the original photo, we could recover a new photo which can be thought as a sample from the underlying smooth manifold. In Figure 5, the odd rows are the inputed original photos and the even rows are the output photos of our algorithm. We can see that comparing from the original ones, the sudden change areas of the input photos are smoothed and the rotating trend becomes 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.

Fefferman, C., Ivanov, S., Kurylev, Y., Lassas, M., and Narayanan, H. Fitting a putative manifold to noisy data. In *Conference On Learning Theory*, pp. 688–720, 2018.

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.

Sober, B. and Levin, D. Manifold approximation by moving least-squares projection (mmls). *Constructive Approximation*, pp. 1–46, 2019.

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

Yao, Z. and Xia, Y. Manifold fitting under unbounded noise. *arXiv preprint arXiv:1909.10228*, 2019.

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.