2012 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 23–26, 2012, SATANDER, SPAIN

# LOCAL LINEAR APPROXIMATION OF PRINCIPAL CURVE PROJECTIONS

*Peng Zhang, Esra Ataer-Cansizoglu, Deniz Erdogmus*

Cognitive Systems Laboratory, Northeastern University, Boston, MA

## ABSTRACT

In previous work we introduced principal surfaces as hyper-ridges of probability distributions in a differential geometrical sense. Specifically, given an $n$-dimensional probability distribution over real-valued random vectors, a point on the $d$-dimensional principal surface is a local maximizer of the distribution in the subspace orthogonal to the principal surface at that point. For twice continuously differentiable distributions, the surface is characterized by the gradient and the Hessian of the distribution. Furthermore, the nonlinear projections of data points to the principal surface for dimension reduction is ideally given by the solution trajectories of differential equations that are initialized at the data point and whose tangent vectors are determined by the Hessian eigenvectors. In practice, data dimension reduction using numerical integration based differential equation solvers are found to be computationally expensive for most machine learning applications. Consequently, in this paper, we propose a local linear approximation to achieve this dimension reduction without significant loss of accuracy while reducing computational complexity. The proposed method is demonstrated on synthetic datasets.

***Index Terms*—** Principal Curve Projection; Nonlinear Dimension Reduction, Manifold Learning

## 1. INTRODUCTION

Manifold learning has been a popular topic in machine learning and pattern recognition fields for many years. Researchers paid much attention on discovering the inner relationship of the given datasets. One important aspect of manifold learning is dimensionality reduction, by using which, the data structures can be visualized more clear, easier to store and processing, especially when we have large number of data samples or high dimension datasets.

Many approaches have been proposed to tackle this problem, either nonlinearly, like Isomap or linearly, like Principal Component Analysis (PCA). In terms of the relationship is using, existing techniques can be roughly divided into global and local algorithms. PCA[1] acts as a significant approach, uses an orthogonal transformation, has been widely used in exploring the main structure of datasets. PCA reveals the global information of the data, local structure will be omitted. Other global manifold learning algorithms, for example, Isomap, which is short for isometric feature mapping, using the geodesic distances among its $k$-nearest neighbors to discover their relationship and finding points in low-dimensional Euclidean space. Besides, Roweis and Saul proposed Local Linear Embedding (LLE)[2], which is a local approach for dimension reduction. It follows the idea that if the neighborhood is sufficiently small and the manifold is smooth, then the points can be linearly approximated by their neighbors. In addition, some other algorithms have been invented based on these, including manifold charting[3], hessian LLE [4] and local tangent space alignment (LTSA) [5]. Their advantages and weakness have been widely discussed. Since local information is more reliable, we intend to propose a new local algorithm, by following the idea of principal curves.

Hastie introduced self-consistent principal surfaces as those which satisfy the following property: a principal surface is self-consistent if and only if every point on this surface is the expected value of the data distribution in the subspace orthogonal to the surface at that point [6]. Since the expectation operation requires integration over the whole orthogonal subspace, in practice this led to some algorithmic challenges. We proposed to use local maximality instead of the expectation operator. Specifically, we have the following [7].

Given a random vector $\mathbf{x} \in \mathbf{R}^n$, let $p(\mathbf{x})$ be its log-pdf (so that if the distribution is Gaussian, $p$ is a quadratic function), $\mathbf{g}(\mathbf{x})$ be the transpose of the gradient, and $\mathbf{H}(\mathbf{x})$ be the Hessian of the probability density function. Also let $\{(\lambda_1(\mathbf{x}), \mathbf{q}_1(\mathbf{x})), \ldots, (\lambda_n(\mathbf{x}), \mathbf{q}_n(\mathbf{x}))\}$ be the eigenvalue-eigenvector pairs of $\mathbf{H}(\mathbf{x})$, where the eigenvalues are sorted such that $\lambda_1(\mathbf{x}) > \lambda_2(\mathbf{x}) > \ldots > \lambda_n(\mathbf{x})$ and $\lambda_i \neq 0$.

**Definition 1.** A point $\mathbf{x}$ is an element of the $d$-dimensional principal set, denoted by $\mathcal{P}^d$ iff the inner product of $\mathbf{g}(\mathbf{x})$ with at least ($n$-$d$) eigenvectors of $\mathbf{H}(\mathbf{x})$ is zero and the eigenvalues corresponding to these orthogonal eigenvectors are all negative.

In [7], we have shown that the principal sets are invariant under monotonic transformations of the functions. Therefore, we prefer to focus on the logarithm of the probability density function (pdf) for the rest of the paper due to the fact that in the case of Gaussian distributions, this choice allows the

This work is supported by NSF.

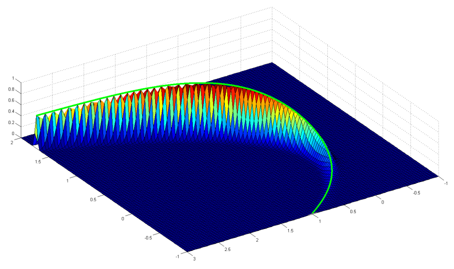


**Fig. 1**: Illustration of eigenvector based projections on negative Rosenbrock function. The principal curves are highlighted (green).

definition above to coincide with the usual linear PCA definition [7].

The principal surfaces provide a natural geometric solution for dimension reduction, manifold learning, and data denoising. Since the principal surface defined above is essentially the $d$-dimensional *ridge* of the pdf, they locally traverse the maximum likelihood regions of the data - thus exhibit the desired property of *passing through the data*. Under the particular definition given above, the ideal nonlinear dimension reduction procedure for projecting a point $\mathbf{x}$ to its $(n-1)$-dimensional counterpart would be as follows: starting at $\mathbf{x}$ solve $n$ differential equations following the eigenvectors of the Hessian along the trajectory where for each trajectory, the eigenvector direction is adjusted to be an ascent direction by selecting the appropriate sign. Then among the $n$ possible projection points, the one which lies on the smallest curvature principal surface is retained - this allows for local sorting of principal and minor directions as opposed to enforcing a global sorting, which may not be always suitable at every location. For the negative of Rosenbrock function, the eigenvector based projections are illustrated in Figure 1.

The definition principal curves and surfaces and the nonlinear dimensionality reduction procedure prescribed above could be very powerful in many applications in machine learning and signal processing including manifold learning, signal and data denoising, data compression, and dimension reduction for other purposes [8] [9]. The caveat is the computational complexity of the differential equation solver. In the illustration, we used Runge-Kutta-4 (RK4) iterations with a small step size to prevent significant error accumulation; the error accumulation for the RK4 integration method is $O(h^4)$, where $h$ is the fixed integration step length. At roughly four times the computational complexity, RK4 compares favorably to the $O(h^2)$ accumulation rate of Euler integration. In practice, the main problem arises when the distribution is estimated nonparametrically, for instance, using the popular kernel density estimation (KDE) method. Since most recent algorithms in machine learning utilize the kernel machine concept and KDE as the underlying framework, the computational complexity of numerical integration with RK4 on a surface characterized by a KDE constructed from $N$ samples and the popular Gaussian kernel (or any other infinite-support kernel) will quickly become prohibitive as $N$, the number of samples, increases.

Although ideally required, since the numerical integration procedure is not feasible in most practical applications dealing with large datasets, in this paper, we propose to use the fundamental concept behind local linear embedding (LLE) [2] in achieving approximate projections of date to principal surfaces. We will develop the concept such that it works for the general problem of dimension reduction from $n$ to $d$ dimensions, however, for illustration purposes, examples will be for projections from 2 to 1 dimension. The illustrations will utilize Gaussian mixture model (GMM) distributions for two reasons: (1) For Gaussian components that are well separated, the projections near the modes are approximately linear (since only a single Gaussian component is effective in this locality) so ideal projections can be visualized easily, (2) the Gaussian-KDE is a particular GMM that is commonly used in machine learning and signal processing applications.

## 2. METHOD

### 2.1. Ideal Projections to the Local Principal Surface

As discussed in Section 1, the ideal projections must follow differential equation trajectories emanating from the given point and following local Hessian eigenvectors in an ascent direction. The reason for this is that the eigenvector flow trajectories form a natural curvilinear coordinate system for each mode (local maximum) of the pdf. The curvilinear coordinates that one would obtain by following the eigenvectors of the Hessian of the pdf, log-pdf, or any other monotonic-increasing function of the pdf are acceptable and they yield the same principal surface, but the projection points and coordinates differ depending on the choice of the monotonic nonlinear function. As mentioned above, we prefer using log-pdf because in the case of Gaussian densities, this allows the proposed method to coincide exactly with linear PCA projections [7]. In general, at the initial point to be projected one may not know which eigenvectors to follow for the projection to end up at the local major surface, however, once all possible projections are found, the projection candidates can be compared and sorted from the most major to the least based on the eigenvalues of the Hessian that remain in the orthogonal subspace (eigenvectors that are orthogonal to the gradient). Specifically, locally the most major surface is the one whose orthogonal subspace eigenvalues are the smallest (i.e., these orthogonal subspace eigenvalues are the most negative and they cause the pdf to most sharply decrease in a quadratic manner as one gets away from the principal surface).

Assuming we have the probability density function (pdf) of data set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, denoted as $p(\mathbf{x})$. The new function, which is log-pdf, will be built as $f(\mathbf{x}) = lnp(\mathbf{x})$, where $ln$ denotes the natural logarithm. Therefore, we can derive the gradient and the Hessian of the function as:

$$\mathbf{g}(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \nabla p(\mathbf{x}) \tag{1}$$

$$\mathbf{H}(\mathbf{x}) = p(\mathbf{x})^{-1} \nabla^2 p(\mathbf{x}) - \mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T \tag{2}$$

Consider a GMM with $m$ components:

$$p(\mathbf{x}) = \sum_{i=1}^{m} w_i G_{\mathbf{\Sigma}_i}(\mathbf{x} - \mu_i)$$

where $w_i (i = 1, ..., m)$ are the weights of every component, $\mathbf{x}$ is an $n$-dimensional real vector, and $G_{\mathbf{\Sigma}_i}(\mathbf{x} - \mu_i)$ is a Gaussian distribution with mean $\mu_i$ and covariance matrix $\mathbf{\Sigma}_i$.

Defining $\mathbf{u}_i(\mathbf{x}) = \mathbf{\Sigma}_i^{-1}(\mathbf{x} - \mu_i)$ and $\mathbf{c}_i(\mathbf{x}) = G_{\mathbf{\Sigma}_i}(\mathbf{x} - \mu_i)$, the gradient and Hessian of $f(\mathbf{x})$ are respectively

$$\mathbf{g}(\mathbf{x}) = -\frac{1}{p(\mathbf{x})} \sum_{i=1}^{m} w_i \mathbf{c}_i(\mathbf{x}) \mathbf{u}_i(\mathbf{x}) \tag{3}$$

$$\mathbf{H}(\mathbf{x}) = \frac{1}{p(x)^2} \sum_{i=1}^{m} w_i \mathbf{c}_i(\mathbf{x}) \left[\mathbf{u}_i(\mathbf{x})\mathbf{u}_i^T(\mathbf{x}) - \mathbf{\Sigma}_i^{-1}\right] - \mathbf{g}(\mathbf{x})\mathbf{g}^T(\mathbf{x}) \tag{4}$$

The RK4 numerical differential equation solver requires the availability of a vector field. For a finite-state time-invariant differential equation of the form $\dot{\mathbf{y}} = \mathbf{v}(\mathbf{y})$, the $t^{th}$ RK4 iteration is $\mathbf{y}_{t+1} = (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)/6$, where $\mathbf{k}_1 = \mathbf{v}(\mathbf{y}_t)$, $\mathbf{k}_2 = f(\mathbf{y}_n + \mathbf{k}_1/2)$, $\mathbf{k}_3 = f(\mathbf{y}_n + \mathbf{k}_2/2)$, $\mathbf{k}_4 = f(\mathbf{y}_n + \mathbf{k}_3)$. When projecting a given point $\mathbf{x}$ to a principal surface of $d$-dimensions, we pick a subset of the eigenvectors of the Hessian as the orthogonal subspace to be eliminated during dimension reduction. Let $\mathbf{V}_\perp(\mathbf{x})$ denote the matrix whose $(n - d)$ columns contain the designated orthogonal subspace eigenvectors of $\mathbf{H}(\mathbf{x})$ at $\mathbf{x}$. Then at the initial RK4 iteration uses $\mathbf{x}_0 = \mathbf{x}$ and $\mathbf{v}(\mathbf{x}) = \mathbf{V}_\perp(\mathbf{x})\mathbf{V}_\perp^T(\mathbf{x})\mathbf{g}(\mathbf{x})$. At each subsequent and intermediate iteration, the orthogonal subspace is selected to be given by the eigenvectors that best align with the orthogonal subspace in the previous step. The iterations terminate naturally when the gradient is orthogonal to the orthogonal subspace eigenvectors. Once $\mathbf{x}$ is projected to all possible $d$ dimensional subspaces by using all possible $(n - d)$ eigenvector subsets as the orthogonal subspace, the major subspace projection can be identified by selecting the projection at which the Hessian eigenvalues are the largest $d$ eigenvalues of the local Hessian.

However, sometimes we have no idea about the distribution of the data set. To solve this, we intend to use Kernel Density Estimation (KDE) to estimate the pdf. Based on this, we are able to propose a similar eigenvector-based projection algorithm with GMM.

Consider a $d$-dimensional data set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, which has $N$ samples. The kernel density estimator is given as:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$$

where $K_{\mathbf{H}}$ is the kernel function with $d$ by $d$ covariance matrix $\mathbf{H}$.

Kernels are defined as non-negative symmetric functions, which satisfy $\int_{-\infty}^{\infty} K(\mathbf{x}) = 1$. Many kernels have been proposed by researchers in the past decades years, like Triangular kernel, Epanechnikov kernel and Gaussian kernel etc. Due to its smoothness and symmetric property, we prefer to use Gaussian kernel.

However, we find that when we are doing with large size data. The covariance matrix can make the computation expensive. Instead, we use isotropic kernel, which is defined as $\mathbf{H} = \sigma^2 \mathbf{I}$ with kernel bandwidth $\sigma$. Therefore, the above equation turns out to be:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i)$$

where $G_\sigma$ is the Gaussian kernel.

The bandwidth can be determined by maximizing leave-one-out function, which is

$$L(\sigma) = \frac{1}{N} \sum_{j=1}^{N} log\left(\frac{1}{N-1} \sum_{i \neq j}^{N} G_\sigma(\mathbf{x}_j - \mathbf{x}_i)\right)$$

Fibonacci line search algorithm can be used to find the optimal bandwidth $\sigma_{opt}$.

Let $f(\mathbf{x}) = lnp(\mathbf{x})$, following the similar equations, we can have the gradient and Hessian for KDE

$$\mathbf{g}(\mathbf{x}) = -\frac{1}{Np(\mathbf{x})} \sum_{i=1}^{m} \mathbf{c}_i(\mathbf{x}) \mathbf{u}_i(\mathbf{x}) \tag{5}$$

$$\mathbf{H}(\mathbf{x}) = \frac{1}{Np(\mathbf{x})^2} \sum_{i=1}^{m} \mathbf{c}_i(\mathbf{x})[\mathbf{u}_i(\mathbf{x})\mathbf{u}_i^T(\mathbf{x}) - \mathbf{\Sigma}_i^{-1}] - \mathbf{g}(\mathbf{x})\mathbf{g}^T(\mathbf{x}) \tag{6}$$

where $\mathbf{u}_i(\mathbf{x}) = \mathbf{\Sigma}_i^{-1}(\mathbf{x} - \mathbf{x}_i)$ and $\mathbf{c}_i(\mathbf{x}) = G_{\mathbf{\Sigma}_i}(\mathbf{x} - \mathbf{x}_i)$

Then, by applying RK4, we can find the projection points. Clearly, this process is time-consuming; especially the RK4 iterations are computationally expensive and may prohibit the use of this nonlinear dimension reduction technique for large dimensional data.

## 2.2. Local Linear Approximate Projections

Suppose that with a sufficiently dense training sample the data space can be partitioned into a simplex-mesh such that for points within a given simplex the ideal projection can be approximated relatively accurately using a linear combination of the projections of the simplex vertices. This simplex mesh could be, for instance, the Delaunay triangulation based partitioning of the data space. As the dual of Voronio regions, Delaunay simplexes have the nice property that they connect edges connecting neighboring Voronoi cells, which represent a compact vector quantization of the space using Euclidean distances. In the $n$-dimensional data space, each Delaunay simplex will have $(n+1)$ vertices. Suppose that for a given simplex, these vertices are the training data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_{n+1}\}$ and their corresponding $d$-dimensional projections for a particular orthogonal subspace selection are $\{\mathbf{p}_1, \ldots, \mathbf{p}_{n+1}\}$. For a point $\mathbf{x}$ that is inside this simplex, we propose to approximate its ideal projection $\mathbf{p}$ using the following linear form: $\hat{\mathbf{p}} = [\mathbf{p}_1, \ldots, \mathbf{p}_{n+1}]\,\alpha$, where the linear combination coefficient vector $\alpha$ is uniquely determined by solving for the following system of linear equations simultaneously: $[\mathbf{x}_1, \ldots, \mathbf{x}_{n+1}]\,\alpha = \mathbf{x}$ and $\mathbf{1}^T\alpha = 1$. This is similar with the fundamental principal behind the popular LLE algorithm which attempts to find a projection that preserves the linear combination coefficients as much as possible when representing a data point with a linear combination of its neighbors. If $\mathbf{x}$ is inside the simplex defined by the vertices above, then all entries of $\alpha$ should be nonnegative; otherwise one must determine the correct simplex which contains the point and recalculate its approximate projection using these correct vertices. The simplex in which a given point falls can be determined using the nearest neighbors of $\mathbf{x}$ and limiting the evaluations of simplexes to those which contain these vertices.

## 3. EXPERIMENTAL RESULTS

We illustrate the concept and quantify accuracy in low-dimensional GMM distributed data and synthetic data scenarios (Thank Kegl for the data). The selection of GMM is motivated for the two reasons mentioned earlier. For test data generated form a given GMM, the RK4 projection and local linear approximation methods are employed. The true underlying GMM is assumed to be known and available for both algorithms where needed. For the data whose pdf is unknown, KDE is applied to estimate its distribution. With the parameters, both projection and approximation methods are applied. Algorithms are implemented using Matlab and run on a Dell T7500 workstation with an Intel Xeon 3GHz CPU and 16GB RAM (in 8 cores mode; parallel computing toolbox is used).

Figure 2 shows the principal curve projections using RK4 for a 2-dimensional dataset, where the points are randomly generated from a curve shaped GMM. The projections of the points to the principal curve are plotted in red. Although the regions between mixtures seem ambiguous, the algorithm precisely finds the projections for the points in these regions.

Instead of GMM, we apply our eigenvector-based projection algorithm on some real life data set, which comes from Kegl's website[10]. As figure 3 and figure 4 show, the projection results on 4 datasets, the red points are the original data and the blue curves are the principal curves of the data, which is their main strictures.

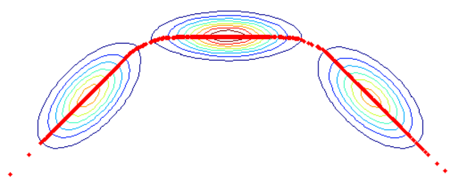


**Fig. 2**: Principal Curve Projections (plotted in red) on Gaussian Mixture Models

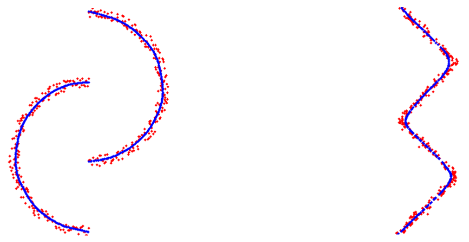

(a) Crescents (b) Zigzag

**Fig. 3**: Principal Curve Projections (plotted in blue) with KDE on crescents data (left) and zigzag data (right)

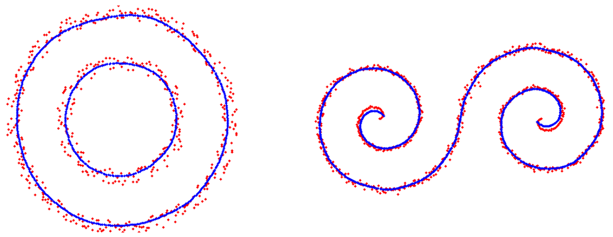
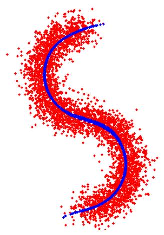

(a) Ring (b) Spiral

**Fig. 4**: Principal Curve Projections (plotted in blue) with KDE on ring data (left) and spiral data (right)

The result of our local linear approximation method is visualized in figure 5 for a 2D dataset. The red points are the real data, from which we intend to find the principal curve. The left figure shows the real projection result with our eigenvector-based algorithm, while the right figure is the approximation result. To illustrate, we just put the approximation results with 4000 training data and 4000 test data. As can be seen from the figure, true and approximate projections coincide with each other except some erroneous points. Note that, although we give the KDE results, it can generalize to other datasets as well. Mean Squared Error (MSE) metric is


(a) Real Projection

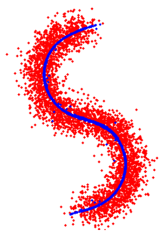

(b) Approximation

**Fig. 5**: Real principal curve projections and its approximation

employed in order to evaluate the performance of the approximation technique. MSE is computed as the expected value of the squared error $E[\|\hat{\mathbf{p}} - \mathbf{p}\|^2]$, where error is the norm of the difference between true projection $\mathbf{p}$ and approximate projection $\hat{\mathbf{p}}$. For a randomly generated test dataset consisting of $M = 4000$ samples, we analyze the relation between number of training samples $N$ and MSE by increasing the size of training samples each time. Keeping the test dataset fixed and varying the number of training samples ($N$), we compute the mean and variance of MSE values for each $M, N$ pair. Figure 6 shows the graph of MSE versus number of training samples ($N$). Errorbars indicate standard deviation of MSE for given $M, N$ values. MSE decreases as we increase the number of training samples.

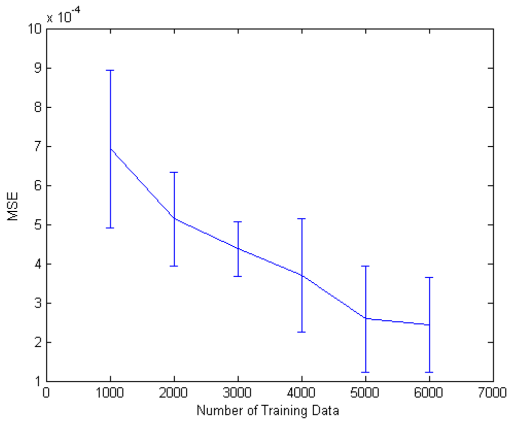


**Fig. 6**: Errorbar for MSE for different sizes of training data. Bars show the mean and standard deviation of the MSE for given $N$ value.

Since the contribution of the study is to compute accurate projections in a relatively shorter time compared to the original projection algorithm, we also report the running time analysis in table 1. The approximation algorithm runs at least 20 times faster than doing projection for all points. Moreover, the approximations yield small MSE values as seen in figure 6.

**Table 1**: Run-time comparison

| | | | |
|---|---|---|---|
| #test samples: | 4000 | 4000 | 4000 |
| #training samples: | 6000 | 7000 | 8000 |
| Approx.time(second): | 0.5083 | 0.5060 | 0.5053 |
| Proj. time(second): | 10.5482 | 10.5211 | 10.5207 |

As explained before, principal curve projections are also useful for dimension reduction. A spiral dataset is presented to explain our idea. We illustrate a scenario where original 2D points are reduced to 1-dimensional space. Simply, for an original test point, we can find the distance of its projection to the mode by tracing the gradient trajectory with a small stepsize. Figure 7 displays two points with positive and negative 1D coordinates in the reduced space. The gradient trajectories are displayed with green. Accumulated distances between each iteration construct the 1D coordinate of the point in the reduced space.The signs of the coordinates can be arranged based on the direction in which mode is reached(the coordinates of two points are -1.561 and 0.417). In the reduced space, the origin is selected by the user. And the negative and positive directions will be formed automatically. In the new space, the unit is set to be correlated with the step size in the RK4 process. Hence, it can be Note that, local linear approximation algorithm can also be used to approximate the coordinate of the points in the reduced space.

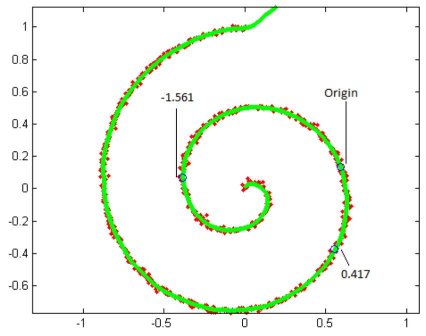
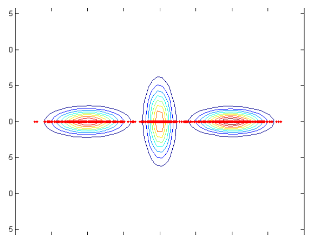


**Fig. 7**: Tracing (plotted in green) with spiral data (plotted in red) and locate new coordinates for two points (plotted with circles)

Principal component analysis (PCA) is a famous algorithm for dimensionality reduction, which was proposed in 1901 by Karl Pearson. It has been called one of the most valuable results from applied linear algebra, which is used abundantly in all forms of analysis from neuroscience to computer graphics. It is a simple, non-parametric method of extracting relevant information from confusing datasets [11]. In contrasting with our proposed algorithm, PCA is a linear projection method. It's based on the eigenvector decomposition of the covariance of the data. Given a dataset $\mathbf{x}$, we can compute its mean $\mu$ and its variance $\mathbf{\Sigma}$. Eigenvalue decomposition will be the next step. To do this, one can either find the eigenvalues directly or apply Singluar Value Decomposition (SVD) to have the projected points in new space. Generally, it is a very powerful and effective way to reduce dimension, but in some cases, it may fail. However, our nonlinear projection algorithm works in these cases.


(a) PCA Result

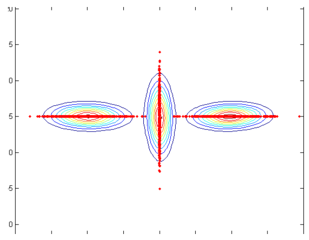

(b) Projection

**Fig. 8**: Comparison of PCA (left) and principal curve projection (Right)

As figure 8 shows, the left figure is the projection result from PCA on GMM, while the right figure is the projection result from our eigenvector-based algorithm. It is obvious to see, with the help of PCA we can only find the main structure of the whole data set, but our method can detect local structure of the data, which is more reasonable and accurate. The reason of the difference is we use local information for principal curve projection.

## 4. CONCLUSION

In this paper, a local linear approximation method is proposed for dimension reduction. Each data point is represented with a linear combination of its neighbors in the original space where the same relation is used for corresponding points in the projected space. The experiments are carried out on 2D simulated data where Gaussian Mixture Models and Gaussian-KDE are utilized. The results show that the technique provides precise approximations indicated with small mean squared error, while reducing the time complexity.

## 5. REFERENCES

[1] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[2] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[3] M. E. Brand, "Charting a manifold," *Advances in Neural Information Processing Systems (NIPS)*, pp. 961–968, 2003.

[4] D. Donoho and C. Grimes, "Hessian eigenmaps: new tools for nonlinear dimensionality reduction," *Proceedings of National Academy of Science*, pp. 5591–5596, 2003.

[5] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Scientific Computing*, vol. 26, pp. 313–338, 2004.

[6] T. Hastie and W. Stuetzle, "Principal curves," *Journal of American Statistical Association*, 1989.

[7] U. Ozertem and D. Erdogmus, "Locally Defined Principal Curves and Surfaces," *Journal of Machine Learning Research*, pp. 1249–1286, 2011.

[8] E. Bas and D. Erdogmus, "Sampling on locally defined principal manifolds," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2276–2279.

[9] U. Ozertem and D. Erdogmus, "Signal denoising using principal curves: Application to timewarping," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 3709–3712.

[10] Kegl, "Synthetic data," `http://www.iro.umontreal.ca/~kegl/research/pcurves/`.

[11] J. Shlens, "A tutorial on principal component analysis," 2005.