Project Report

Comparative Analysis of Kernel Methods for Decomposition

Gaurav Dhir

December 14, 2023

1 Overview

Principal Component Analysis attempts to diagonalize the covariance matrix of the data using a basis transformation and use the resulting transformation to analyze the dataset. More specifically, the projections on the eigenvectors corresponding to the largest eigenvalues of the covariance matrix are subsequently used for dimensionality reduction and denoising applications. Kernel PCA attempts to generalize the setting of PCA to a non-linear one by using a non-linear feature transformation to analyze the dataset. Kernel PCA has been used extensively for applications ranging from denoising [3, 5, 8, 6, 9], novelty detection [2] and pattern recognition [10]. Kernel PCA has also been used as a preprocessing step for other classification/prediction algorithms [1, 6]. The present project aims to understand and analyze kernel PCA for denoising applications via its application on different synthetic datasets. The project also aims to understand the feature basis transformations obtained via the application of different kernels in kernel PCA and the corresponding representations of non-linear eigenvectors within the input space for different test datasets.

Given a non-linear feature map $\Phi : R^M \rightarrow F$, kernel PCA attempts to find a diagonalization of the covariance matrix in the space F. Assuming centered data in the feature space $\{\widetilde{\Phi}(x_i)\}_{i=1}^N$, one can attempt to find a diagonalization of the Covariance matrix $C = \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i)\widetilde{\Phi}(x_i)^T$ by solving the eigenvalue problem $CV^k = \lambda V^k$. Here, $\{x_i\}_{i=1}^N$ represent the original data in the input space, V^k represents the eigenvectors in the space F, C represents the covariance matrix and N represents the number of samples in the input space.

The eigenvalue problem can also be obtained minimizing the lagrangian $L = \frac{1}{2N}\sum_{i=1}^N((V \cdot \Phi(x_i))^2 - \frac{\lambda}{2}((V \cdot V) - 1)$ with respect to V. Let M_F represent the dimensionality of F. Then, the dimensionality of the computed eigenspace will be $min(N, M_F)$.

It must be noted that no restriction is placed on the dimensionality of F. Hence, for high dimensional feature spaces, the eigenvalue problem in its original form cannot be explicitly solved. However, as in linear PCA, one can make the observation that the eigenvectors of the matrix C lie within the span of $\{\ \Phi(x_j)\}_{j=1}^{N}$ as described in Eq. 1.

$$V_k = \sum_{i=1}^{N} \alpha_i^k \Phi(x_i) \tag{1}$$

Hence, using $(\Phi(x_k) \cdot CV) = N\lambda(\Phi(x_k) \cdot V)$ and Eq. 1, an alternate eigenvalue problem of dimensionality $N \times N$ given as $K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$ can instead be solved. Here, K represents the kernel matrix. Each entry in K can be obtained by applying a kernel function $k(x, y)$ on the vectors in the input space. As a result, $K_{ij} = k(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$.

Furthermore, instead of explicitly specifying a suitable feature transformation Φ, one can try to directly specify a kernel function $k(x, y)$ (leading to a kernel matrix) and assume that a corresponding feature map Φ exists. It can be shown that for a positive semi-definite kernel matrix and a symmetric kernel function satisfying $k(u, v) = k(v, u)\ \forall u, v \in R^M$, there always exists a feature transformation Φ and an inner product space F. For instance, for the homogenous polynomial kernel of degree d, $k(x, y) = (x \cdot y)^d$, the feature vector contains all d products of the original vector components. In other words, $\Phi(x) = \left[\ldots, \sqrt{\binom{d}{j_1 j_2 j_3 \cdots j_d}}(x^{(1)})^{j_1}(x^{(2)})^{j_2} \ldots (x^{(d)})^{j_d}, \ldots\right]^T$ where $\binom{d}{j_1 j_2 j_3 \cdots j_d}$ represents a multinomial expression. For some choices of kernel functions (such as the Gaussian kernel), this feature map can be infinite dimensional.

To compute the projections of individual feature vectors on the principal components computed by kernel PCA, one can again use Eq. 1 to avoid explicit calculation of transformed feature vectors using the formulation presented in Eq. 2. The

$$< V_k, \Phi(x) >= \sum_{i=1}^{N} \alpha_i^k < \Phi(x_i), \Phi(x) > \tag{2}$$

1.1 Centering in Kernel PCA

As a preprocessing step, mean centering of the data is required. The mean centering process must be performed on both the initial data and the test vectors [4]. In the case of kernel PCA, this procedure must be performed in the transformed feature space F. However, the eigenvalue problem $KV = \lambda V$ could be solved in kernel PCA without access to the transformed feature vectors using the formulation shown in Eq. 3 [2].

$$\begin{aligned}
&\widetilde{K}_{ij} =< \widetilde{\Phi}(x_i) \cdot \widetilde{\Phi}(x_j) > \\
&\widetilde{K}_{ij} = \left\langle \left(\Phi(x_i) - \frac{1}{N}\sum_{r=1}^{N} \Phi(x_r) \right) \cdot \left(\Phi(x_j) - \frac{1}{N}\sum_{s=1}^{N} \Phi(x_s) \right) \right\rangle \\
&\widetilde{K}_{ij} = K_{ij} - \frac{1}{N}\sum_{s=1}^{N} K_{is} - \frac{1}{N}\sum_{r=1}^{N} K_{rj} + \frac{1}{N^2}\sum_{r=1}^{N}\sum_{s=1}^{N} K_{rs} \\
&K = K - 1_N K - K 1_N + 1_N K 1_N \\
&where \quad 1_N = \frac{1}{N} * ones(N)
\end{aligned} \tag{3}$$

Furthermore, the projections of any mean centered test vector $\Phi(x)$ on the basis vectors V^k could also be obtained without resorting to the calculation of feature vectors.

$$\begin{aligned}
&\beta_k =< \widetilde{\Phi}(y) \cdot V^k >=< \widetilde{\Phi}(y) \cdot \sum_{j=1}^{N} \alpha_j^k \widetilde{\Phi}(x_k) >= \sum_{j=1}^{N} \alpha_j^k < \widetilde{\Phi}(y) \cdot \widetilde{\Phi}(x_k) > \\
&or \quad \beta_k = \sum_{j=1}^{N} \alpha_j^k \widetilde{k}(y, x_k) \\
&where \quad \widetilde{\Phi}(y) = \Phi(y) - \frac{1}{N}\sum_{i=1}^{N} \Phi(x_i) \quad and \quad \widetilde{\Phi}(x_k) = \Phi(x_k) - \frac{1}{N}\sum_{i=1}^{N} \Phi(x_i) \\
&\widetilde{k}(y, x_k) = k(y, x_k) - \frac{1}{N}\sum_{i=1}^{N} k(y, x_i) - \frac{1}{N}\sum_{i=1}^{N} k(x_i, x_k) + \frac{1}{N^2}\sum_{r=1}^{N}\sum_{s=1}^{N} k(x_r, x_s)
\end{aligned} \tag{4}$$

1.2 Denoising and Kernel PCA

From a dimensionality reduction perspective, linear PCA has been extensively used for approximate face reconstruction and compression applications. In the underlying context, reconstruction of original vectors from the transformed basis is considered a non-trivial task [7]. The non triviality arises from the observation that an inverse mapping from the transformed feature space to the original space might not exist. Bernhard [7] elucidate this observation by giving an example of a transformed basis represented using the basis induced by the Gaussian kernel. In this case, any vector in the transformed feature space can be represented as a linear superposition of Gaussian bumps on R^N.

A reconstruction can still be computed using the procedure provided in Bernhard [7] by finding a z which minimizes the error between the projection of z on the transformed basis and the actual transformed feature value. This has been stated explicitly in Eq. 5. Note that P_n represents the projection on the basis computed by the kernel PCA and $P_n\Phi(z) = \Phi(z)$ if the observation z can be exactly represented within this basis.

$$\begin{aligned}
&z = argmin_z \left\| P_n\Phi(z) - \Phi(z) \right\| \\
&where \quad P_n\Phi(x) = \sum_{k=1}^{N} \beta_k V_k
\end{aligned} \tag{5}$$

The objective function $\rho(z)$ in the stated optimization problem shown in Eq. 5 can be calculated without knowledge of a feature mapping $\Phi(z)$ using the formulation shown in Eq.

6. Here, Ω includes terms not dependent on z.

$$
\begin{aligned}
&\rho(z) = \|\Phi(z)\|^2 - 2 < \Phi(z), P_n\Phi(x) > +\Omega \\
&P_n\Phi(x) = \sum_{k=1}^{n} \beta_k V^k, \quad \beta_k =< \Phi(x), V^k > \\
&V^k = \sum_{i=1}^{l} \alpha_i^k \Phi(x_i) \\
&\beta_k = \sum_{i=1}^{l} \alpha_i^k k(x, x_i) \\
&P_n\Phi(x) = \sum_{k=1}^{n} \beta_k \sum_{i=1}^{l} \alpha_i \Phi(x_i) \\
&\rho(z) = k(z,z) - 2\sum_{k=1}^{n} \beta_k \sum_{i=1}^{l} \alpha_i < \Phi(x_i), \Phi(z) > +\Omega
\end{aligned}
\tag{6}
$$

For the Gaussian kernel, a fixed point iterative iteration could be devised by setting $\nabla_z \rho = 0$ as shown in Mika [5] and presented in more detail in Eq. 6.

Using the analysis presented in Equation 7, for kernels of the form $k(x,y) = k(\|x-y\|^2)$, the minimization of $\rho(z)$ can be further simplified as shown in Equation 7.

$$
\begin{aligned}
&\rho(z) = -2\sum_{i=1}^{l} \gamma_i k(z, x_i) \quad where \quad \gamma_i = \sum_{k=1}^{n} \beta_k \alpha_i^k \\
&\nabla_z \rho(z) = \sum_{i=1}^{l} \gamma_i k'(\|z - x_i\|^2)(z - x_i) = 0 \\
&z = \frac{\sum_{i=1}^{l} \gamma_i k'(\|z - x_i\|^2) x_i}{\sum_{i=1}^{l} \gamma_i k'(\|z - x_i\|^2)} \\
&z = \sum_{i=1}^{l} \frac{\delta_i x_i}{\sum_{j=1}^{l} \delta_j} \quad where \quad \delta_i = \gamma_i k'(\|z - x_i\|^2)
\end{aligned}
\tag{7}
$$

A fixed point iteration shown in Eq. 8 can be used to solve the recurrence presented in Equation 7.

$$
z^{t+1} = \frac{\sum_{i=1}^{l} \delta_i x_i}{\sum_{j=1}^{l} \delta_j} \quad where \quad \delta_i = \gamma_i k'(\left\|z^t - x_i\right\|^2) \tag{8}
$$

For the gaussian kernel, it can be shown that $\|\Phi(x_i) - \Phi(z)\|^2 = -2\gamma_i$. Hence, the fixed point iteration gives more preference to the training samples closer to the estimate.

Despite the simplification accorded by the fixed point iteration, multiple starting points were needed to come to a good enough solution and the algorithm was found prone to stuck in a local minima [4]. As a result, the MATLAB's optimization toolbox with the **fmincon** solver was used to construct the denoised pre-images. The Sequential Quadratic Programming and the Interior Point algorithms were used for the optimization process. Global Search based solutions were also used for the optimization but were found extremely slow in arriving at a solution.

2 Experiments and Analysis

Three different experiments were performed to analyze kernel PCA and evaluate its efficacy within denoising and reconstruction applications. The experiments are described as follows.

Data was generated from 3 Gaussian distributions in two dimensions with point sources located at $(-0.5, -0.1), (0, 0.7), (0.5, 0.1)$ and $\sigma = 0.1$. 300 points were selected at random from each distribution and kPCA was performed on it. 30 Points were selected from each distribution as the test dataset. The initial data distribution has been shown in Fig. 1.

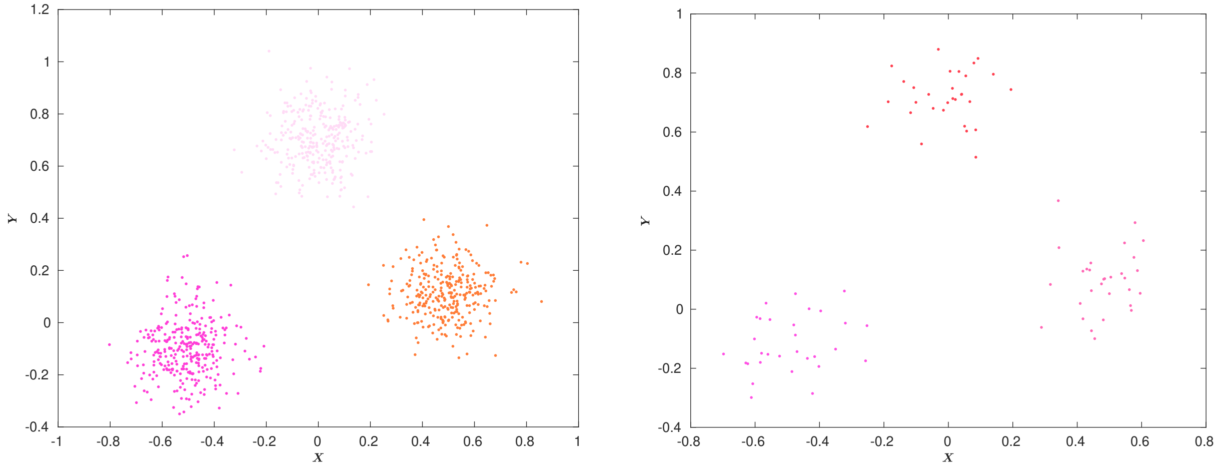
The resulting eigenvector contours of the first principal component obtained after performing PCA and kPCA can be observed in Fig. 2. It can be clearly observed that while on one hand, the level sets of the first principal component in the case of PCA analysis are limited to straight line directions, on the other hand, in the case of kPCA, the level sets seek to differentiate the three data distributions.

Figure 3 shows the negative of the objective function plot for the 3 different test vectors. It can be observed that only one peak is observed for each test vector showing that the optimization problem is well formed.

Figure 4 shows the reconstructions obtained after projecting the data on the first principal component in case of PCA and 2 principal components in the case of kPCA with different optimization options. It can be observed that in the case of PCA, the entire dataset is projected on a straight line while on the other hand, the denoised solutions in kPCA are reconstructed on the center of the Gaussian corresponding to the initial data distribution.

Figures 5d and 6d show the reconstructions obtained using PCA and kPCA (with varying kernel width) when the initial data distribution was randomly generated around a half circle and a square respectively. PCA again seeks to reconstruct data along a straight line direction corresponding to the first principal direction. On the other hand, kPCA shows different reconstructions based on the value of the kernel width. For both the experiments, it was observed that due to extremely small kernel widths, some data was reconstructed around the origin. This occurs due to the fact that in the limit of inifinitesimally small kernel widths, the points away from the centers of the Gaussians corresponding to the principal component show almost 0 projections as explained in Twining [9]. For very large values of kernel widths, kPCA with the Gaussian kernel either reconsructs data in a similar fashion as PCA or reconstructs data along a Gaussian centered at the origin as can be observed in Fig. 8b.

As a result, an optimal kernel width exists when the reconstruction error is minimal. This width can also be observed explicitly from Fig. 7 which shows the denoising error variation



(a) Initial Training Data for Analysis

(b) Test Data for Evaluation

Figure 1: Initial Data Distribution for Denoising data involving 3 Gaussians in R^2

with the kernel width.

Finally, the same experiments were also attempted for polynomial kernels. However, as can be seen in Fig. 8 the objective function had a saddle point at the origin for most test vectors. As a result, on using the MATLAB's SQP and Interior Point solvers, all data was found reconstructed at the origin. Hence, further analysis for polynomial kernels was not attempted but can be the subject of future work.

References

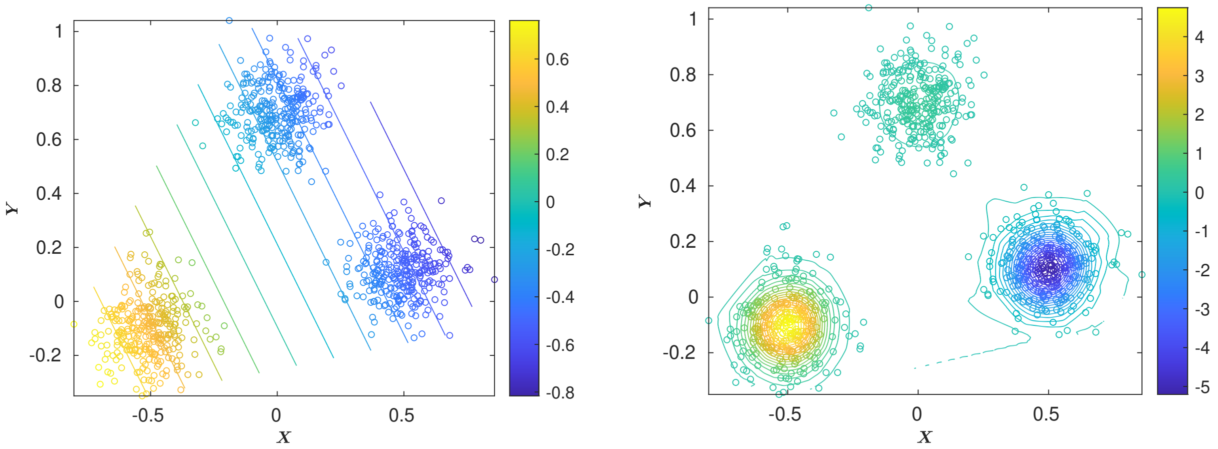
[1] Hala M. Ebied. Feature extraction using pca and kernel-pca for face recognition. In *2012 8th International Conference on Informatics and Systems (INFOS)*, pages MM–72–MM–77, 2012.

[2] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[3] A.M. Jade, B. Srikanth, V.K. Jayaraman, B.D. Kulkarni, J.P. Jog, and L. Priya. Feature extraction and denoising using kernel pca. *Chemical Engineering Science*, 58(19):4441–4448, 2003.

[4] Xiang Ma and Nicholas Zabaras. Kernel principal component analysis for stochastic input model generation. *Journal of Computational Physics*, 230(19):7311–7331, 2011.

[5] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998.



(a) PCA Component

(b) kPCA Component

Figure 2: Level Sets of the First Principal Direction for PCA and kPCA using the Gaussian Kernel

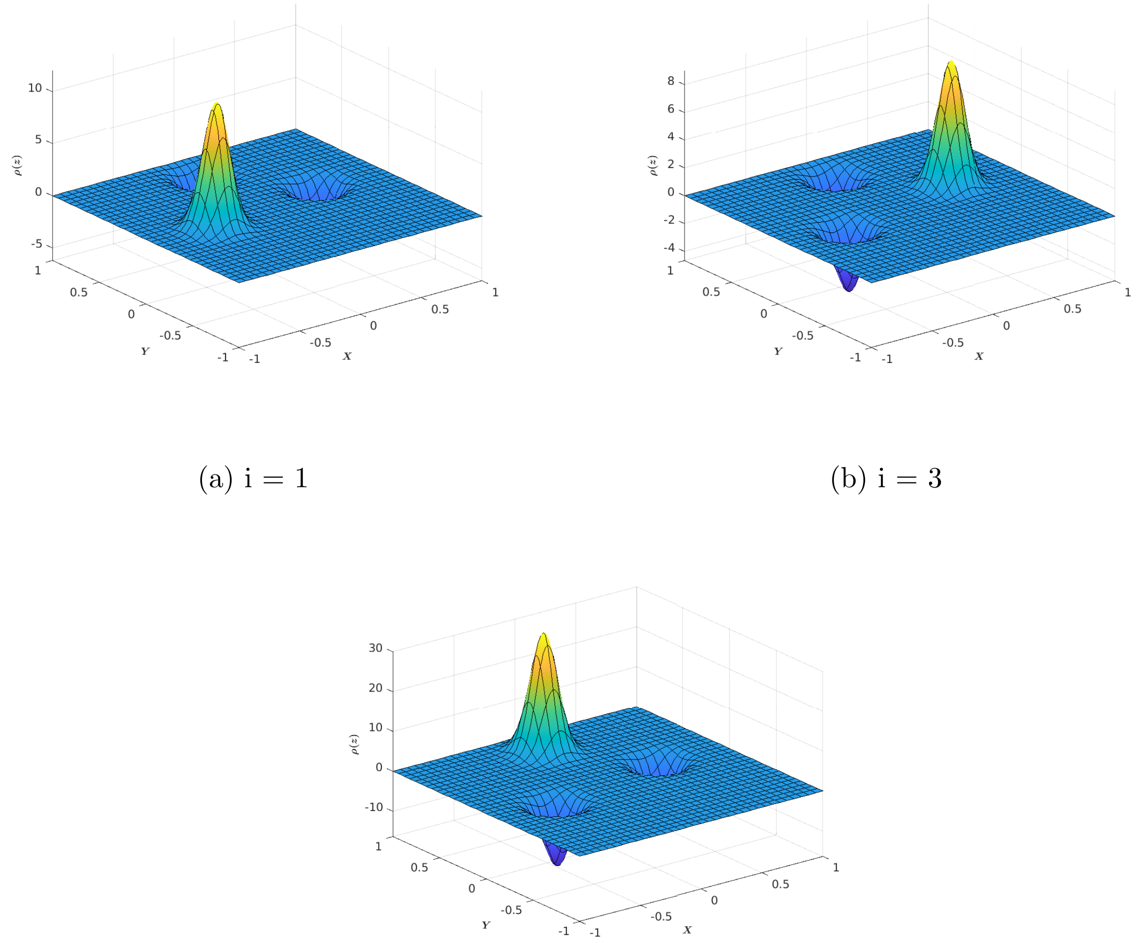
[6] Roman Rosipal, Mark Girolami, Leonard J. Trejo, and Andrzej Cichocki. Kernel pca for feature extraction and de-noising in nonlinear regression. *Neural Computing & Applications*, 10(3):231–243, Dec 2001.

[7] Bernhard Schölkopf, Sebastian Mika, Alex Smola, Gunnar Rätsch, and Klaus-Robert Müller. Kernel pca pattern reconstruction via approximate pre-images. In Lars Niklasson, Mikael Bodén, and Tom Ziemke, editors, *ICANN 98*, pages 147–152, London, 1998. Springer London.

[8] Takashi Takahashi and Takio Kurita. Robust de-noising by kernel pca. In José R. Dorronsoro, editor, *Artificial Neural Networks — ICANN 2002*, pages 739–744, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[9] C.J. Twining and C.J. Taylor. The use of kernel principal component analysis to model data distributions. *Pattern Recognition*, 36(1):217–227, 2003.

[10] Quan Wang. Kernel principal component analysis and its applications in face recognition and active shape models. *ArXiv*, abs/1207.3538, 2012.

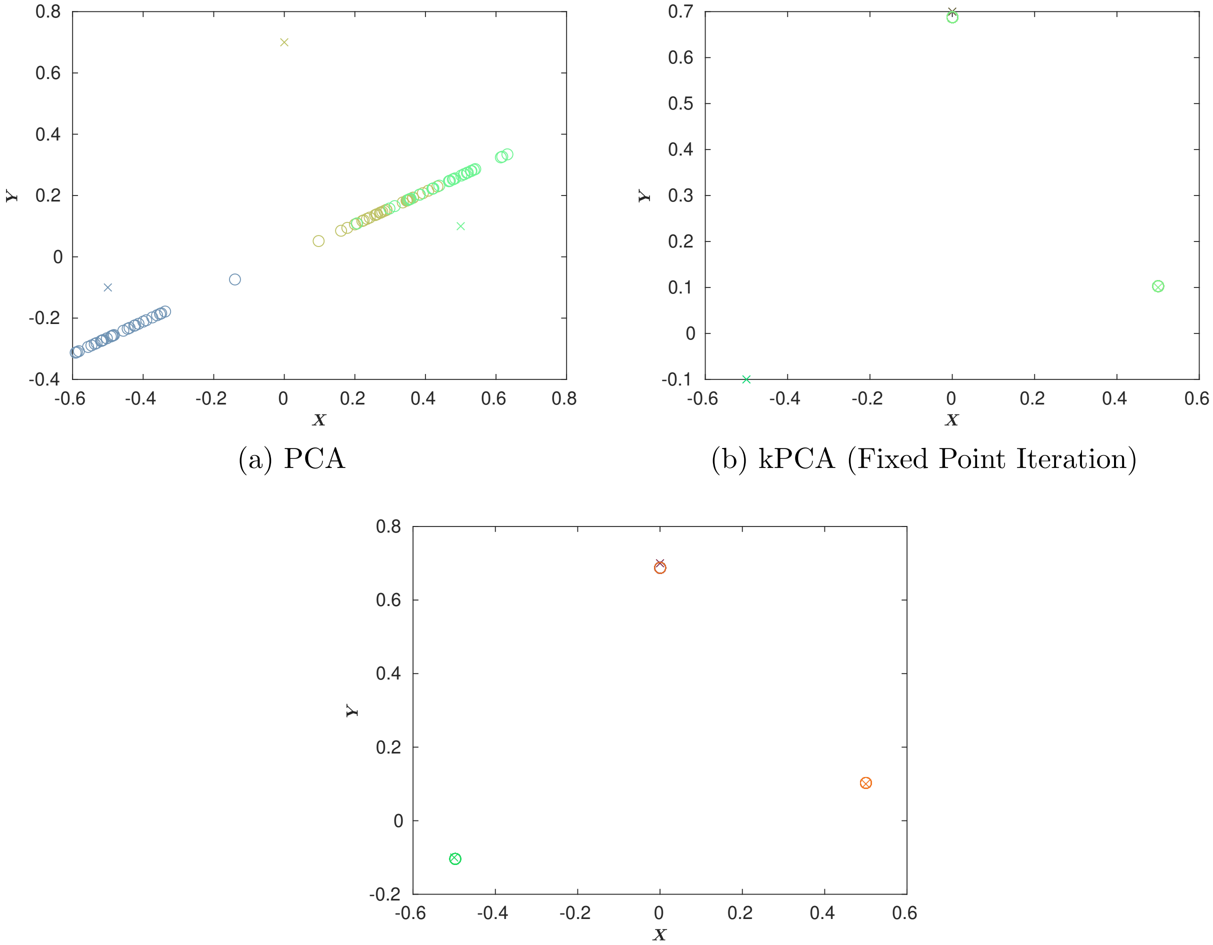


(a) i = 1

(b) i = 3

(c) i = 2

Figure 3: Objective Function Shape ($-\rho(z)$)

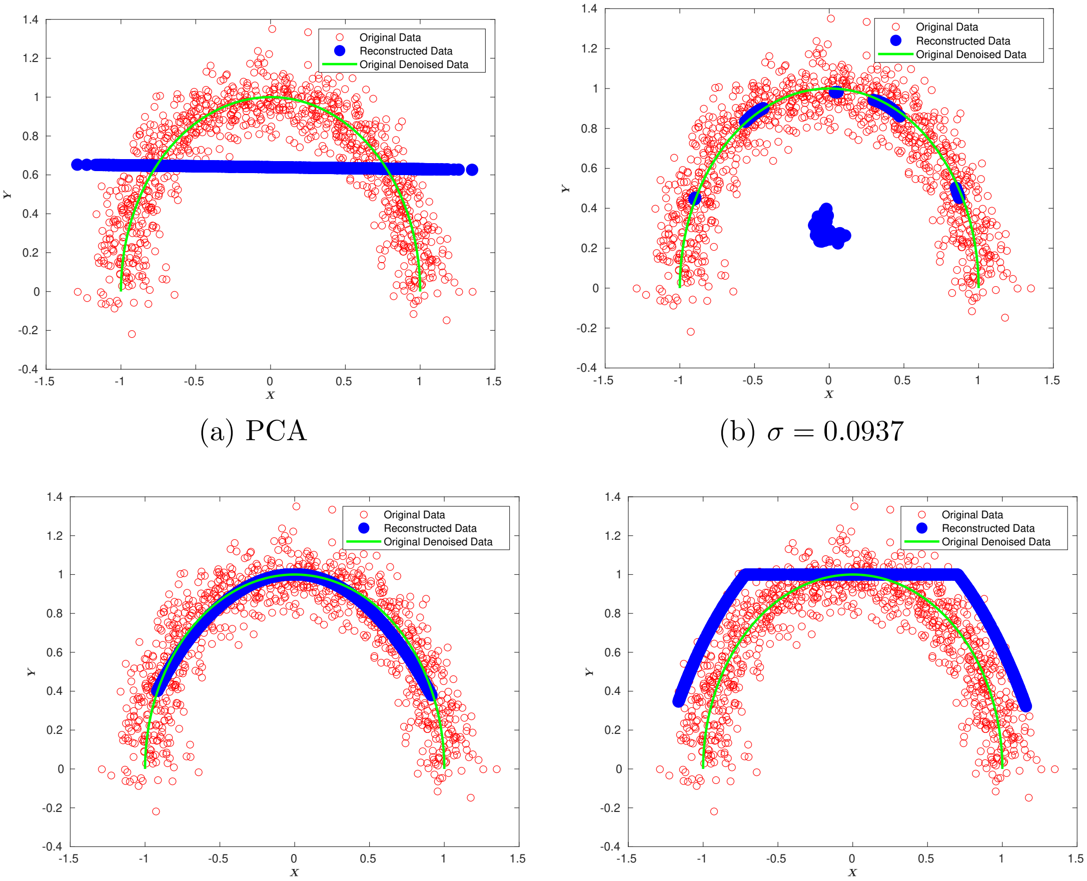


(a) PCA

(b) kPCA (Fixed Point Iteration)

(c) kPCA (MATLAB SQP)

Figure 4: Reconstructed (Denoised) Solutions



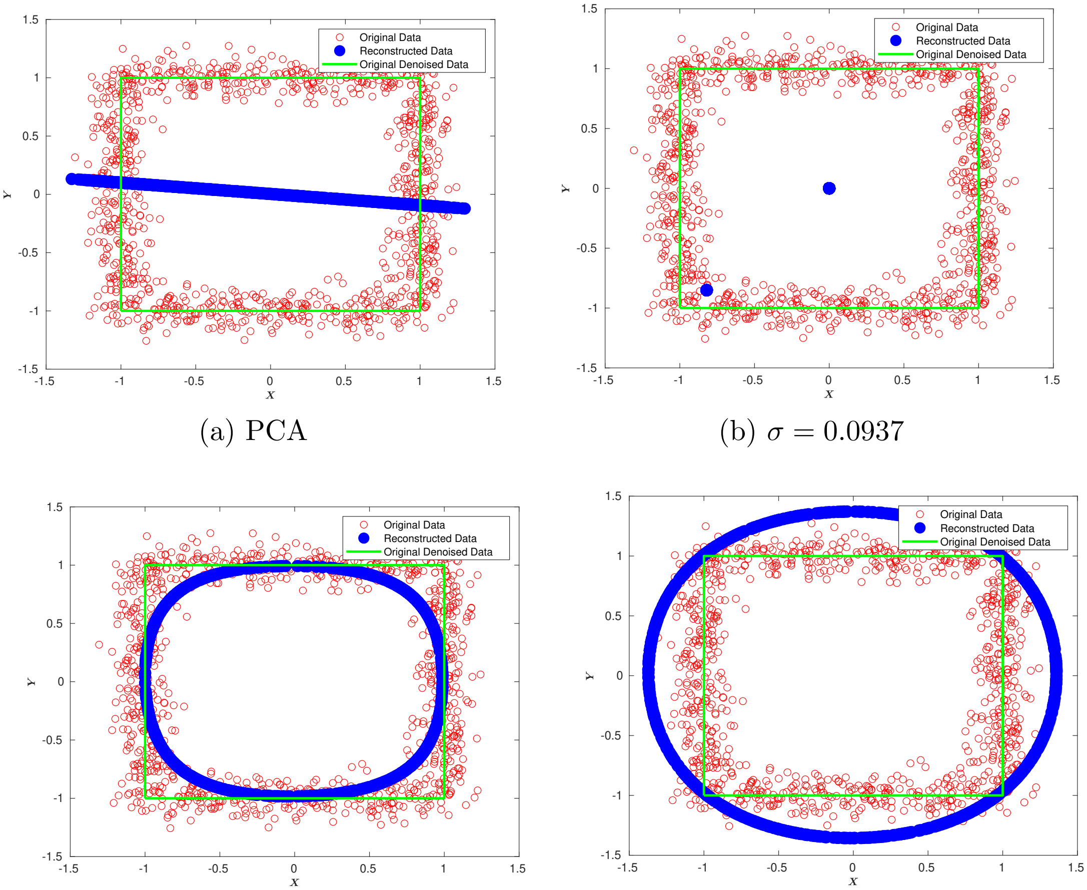
(a) PCA

(b) $\sigma = 0.0937$

(c) $\sigma = 0.468$

(d) $\sigma = 0.937$

Figure 5: Initial Noisy Half Circle Data Distribution and Reconstructed (Denoised) Solutions using 1 PCA and 2 kPCA Components with varying values of Gaussian Kernel Parameter σ



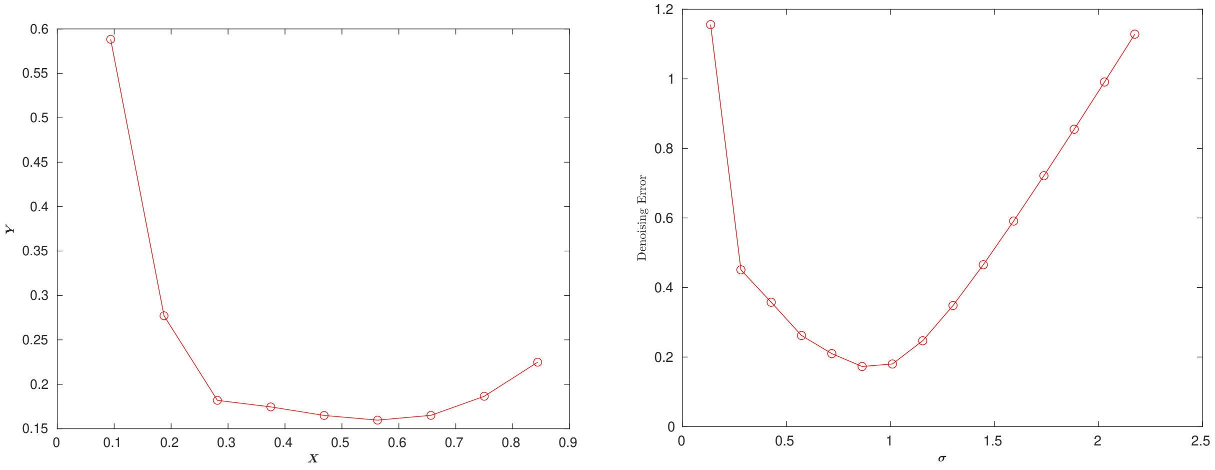
(a) PCA

(b) $\sigma = 0.0937$

(c) $\sigma = 0.679$

(d) $\sigma = 1.22$

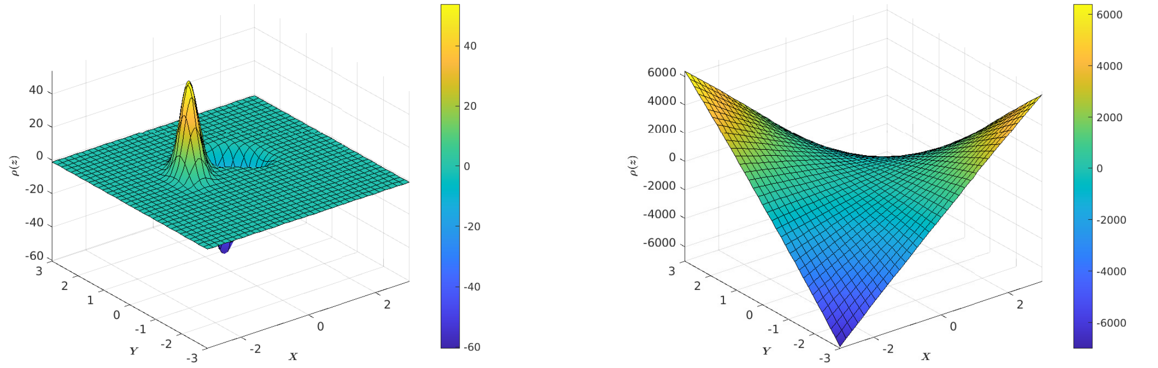
Figure 6: Initial Noisy Data Distribution on a Square and Reconstructed (Denoised) Solutions using 1 PCA and 2 kPCA Components with varying values of Gaussian Kernel Parameter σ



(a) Half Circle Data Distribution

(b) Square Data Distribution

Figure 7: Denoising Error Variation with different values of kPCA Parameter σ



(a) Gaussian Kernel with Half Circle Data Distribution

(b) Polynomial Kernel with Square Data Distribution

Figure 8: Objective Function Shape for different data distributions and different kernels