Notes on Nonlinear PCA (for Quantitive Finance)

Prepared By: Romeil Sandhu

1. INTRODUCTION

These are a set of notes that I wrote for a quantitive finance group at Stony Brook University to provide an introduction of one form of nonlinear PCA (known as Kernel PCA). They cover several tacit areas that include the "pre-image" approximation and a general framework that has been used in previous for shape analysis. In particular, I was motivated to write these to provide a tutorial and excuse any misplace or abused notation. If you find a mistake in these notes in reviewing, please contact me and I will correct them accordingly.

2. PCA REVISITED

Let us consider a set of N observations x_k where $k = 1, ..., N$ and $x_k \in R^N$. We also assume the observations are centered, i.e., $\sum_{i=1}^{N} x_i = 0$. Then, PCA diagonalizes the covariance matrix $C = \frac{1}{N}\sum_{j=1}^{N} x_j x_j^T$ through the following eigenvalue problem:

$$\lambda \mathbf{v} = C\mathbf{v} \tag{1}$$

for eigvenvalues $\lambda_i \geq 0$ with eigenvectors $v = \{v_1, ..., v_N\}$. As $C\mathbf{v} = \frac{1}{N}\sum_{j=1}^{N}(x_j \cdot \mathbf{v})x_j^T$, all solutions v must lie in the span of $x_1, x_2, ..., x_N$. Then Equation (1) can be rewritten as

$$\lambda(x_k \cdot \mathbf{v}) = (x_k \cdot C\mathbf{v}) \text{ for all k = 1,..., N} \tag{2}$$

Nevertheless, PCA performs dimensional reduction by choosing an l-dim subspace (via l significant eigenvalues). Then reconstruction is performed as $\hat{x} = \sum_{i=1}^{l} \lambda_i v_i$. One could also perform prediction by assuming that a newly "unseen" out of sample observation x^* will lie in this l-dim subspace. Then, the underlying observation is that observation that is the closest, in the $L2$ sense, to this subspace:

$$E(w) = ||x^* - P^l_{\text{linear pca}}(x)||^2 \tag{3}$$

where $P^l_{\text{linear pac}}(x) = \sum_{i=1}^{l} w_i v_i$. Solving the optimization for the "weights" w_i (previously denoted as λ), this can be formulated as a filtering problem - i.e., state space is w_i. **Note**: One can deal with uncentered data trivially; however, special care must be taken care of this fact in nonlinear pca (e.g., how one centers in the feature space). In short, the predictive power of PCA in this formulation is driven by two items: 1) how accurate is our l-dim subspace 2) the underlying "dynamics" of our weights if our system evolves over time.

3. NONLINEAR PCA - INTRODUCTION

Let us again considered a set of N centered observations apart of an input space $\mathcal{M} \in \mathbb{R}^n$ for which there exists a possibly nonlinear map $\varphi : R^N \mapsto \mathcal{H}$. We denote $\mathcal{H}$ as the "feature space." Under this mapping, data is represented in a linear fashion and PCA can be performed. This said, the map φ is **explicitly hard to compute** and to this end, we let the covariance matrix in the "feature space" $\mathcal{H}$ be denoted as:

$$\bar{C} = \frac{1}{N}\sum_{j=1}^{N} \varphi(x_j)\varphi(x_j)^T \tag{4}$$

We have assumed the data is centered in the feature space (i.e., $\sum_{i=1}^{N} \varphi(x_i) = 0$) for now and we will return to this later. Again, the problem is formulated as the eigenvalue problem:

$$\lambda \mathbf{v} = \bar{C}\mathbf{v} \tag{5}$$

As we can not explicitly compute φ, we will solve this forward mapping problem implicitly. In particular, by the same argument in the previous section, $\mathbf{v}$ lies in the span of $\varphi(x_1), ..., \varphi(x_N)$, which provides two useful consequences. Firstly, we can consider the equivalent equation as

$$\lambda(\varphi(x_k) \cdot \mathbf{v}) = (\varphi(x_k) \cdot \bar{C}\mathbf{v}) \text{ for all k = 1,..., N} \tag{6}$$

and secondly, there exists a set of coefficients α_i (i=1,...,N) such that

$$\mathbf{v} = \sum_{i=1}^{N} \alpha_i \varphi(x_i). \tag{7}$$

From this, we can combine Equation (6) and Equation (7) to yield the following:

$$\lambda \sum_{i=1}^{N} \alpha_i(\varphi(x_k) \cdot \varphi(x_i)) = \frac{1}{N} \sum_{i=1}^{N} \alpha_i(\varphi(x_k) \cdot \sum_{j=1}^{N} \varphi(x_j))(\varphi(x_j) \cdot \varphi(x_i)) \text{ for all k = 1,...,N} \tag{8}$$

Defining an N X N matrix K as $K_{ij} := (\varphi(x_i) \cdot \varphi(x_j))$, this can rewritten as follows

$$N\lambda K\alpha = K^2\alpha \tag{9}$$

where α denotes the column vector with entries $\alpha_1, ..., \alpha_N$. As K is symmetric, it has a set of eigenvectors which spans the whole space and solving the below

$$M\lambda\alpha = K\alpha \tag{10}$$

gives us all solutions α of Equation (9). The eigenvalues of K will exactly give the solutions of $M\lambda$ of Equation (9). We therefore only need to diagonalize K and circumvent computing $\bar{C}$. This said, we need to normalize the eigenvectors such that

$$(v^k \cdot v^k) = 1 \text{ for all k = 1,..., N} \tag{11}$$

Taking this into account, we find this normalization to be (by virtue of Equation (7) and Equation (10)):

$$1 = \sum_{i,j=1}^{N} \alpha_i^k \alpha_j^k (\varphi(x_i) \cdot \varphi(x_j)) \tag{12}$$

$$= \sum_{i,j=1}^{N} \alpha_i^k \alpha_j^k K_{ij} \tag{13}$$

$$= (\alpha^k \cdot K\alpha^k) \tag{14}$$

$$= \lambda(\alpha^k \cdot \alpha^k) \tag{15}$$

For the purpose of principal component extraction, we need to compute projections on the eigenvectors $\mathbf{v}^k$ in $\mathcal{H}$. As such, let $\mathbf{x}$ be a test point with an image $\varphi(\mathbf{x})$ in $\mathcal{H}$, then

$$(\mathbf{v}^k \cdot \varphi(\mathbf{x})) = \sum_{i=1}^{N} \alpha_i^k (\varphi(x_i) \cdot \varphi(\mathbf{x})) \tag{16}$$

may be called its nonlinear principal components corresponding to φ. Next, we present the case where data is not centered (as there is no guarantee if one centers data in the input space that it will be centered in the "feature" space).

4. NONLINEAR PCA - UNCENTERED CASE

Up until now, we assumed "centered" observations in $\mathcal{H}$ - this was done to introduce nonlinear pca. However, we are not guaranteed that even if data is centered in the input space, it will be centered in the "feature" space.

Again, we will consider the set of data points $\{x_1, x_2, ..., x_N\}$ that are members of the input space $\mathcal{M} \in \mathbb{R}^n$. This set can be mapped to a possibly higher dimensional feature space, denoted by $\mathcal{H}$, via the nonlinear map $\varphi : \mathbb{R}^n \mapsto \mathcal{H}$. One can introduce a Mercer kernel, which is defined to be a function $k(x_a, x_b) = \langle \varphi(x_a), \varphi(x_b) \rangle$ such that for all data points x_i, the kernel matrix

$$\mathbf{K} = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & \ddots & & \\ \vdots & & k(x_i, x_j) & \\ k(x_N, x_1) & & & k(x_N, x_N) \end{pmatrix}$$

is symmetric positive and was previously defined above. Further, according to Mercer's Theorem computing $k(x_a, x_b)$ as a function of $\mathcal{M} \times \mathcal{M}$, amounts to computing the inner scalar product in $\mathcal{H}$. This scalar product in $\mathcal{H}$ defines a distance $d_{\mathcal{H}}$. For example, the L_2 distance is $d^2_{\mathcal{H}}(\varphi(x_a), \varphi(x_b)) = \|\varphi(x_a) - \varphi(x_b)\|^2 = k(x_a, x_a) - 2k(x_a, x_b) + k(x_b, x_b)$.

Lifting the restriction of "centered" data, we introduce a centered kernel matrix $\tilde{\mathbf{K}}$ defined as

$$\begin{aligned} \tilde{\mathbf{K}}(i,j) &= \langle (\varphi(X_i) - \bar{\varphi}), (\varphi(X_j) - \bar{\varphi}) \rangle \\ &= \langle \tilde{\varphi}(X_i), \tilde{\varphi}(X_j) \rangle = \tilde{k}(X_i, X_j), \text{for}(i,j) \in [|1, N|] \end{aligned} \tag{17}$$

with

$$\bar{\varphi} = \frac{1}{N} \sum_{i=1}^{N} \varphi(X_i)$$

where

$$\tilde{\varphi}(X_i) = \varphi(X_i) - \bar{\varphi}.$$

In addition, since $\tilde{\mathbf{K}}$ is symmetric, it can be decomposed as

$$\tilde{\mathbf{K}} = USU^T, \tag{18}$$

where $\mathbf{S} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N)$ is a diagonal matrix containing the eigenvalues of $\tilde{\mathbf{K}}$. $\mathbf{U} = [u_1, u_2, ..., u_N]$ is an orthonormal matrix, where the columns $u_i = [u_{i1}, u_{i2}, ..., u_{iN}]^T$ are the eigenvectors corresponding to the eigenvalues λ_i's. Furthermore, it can be shown that

$$\tilde{\mathbf{K}} = \mathbf{HKH} \tag{19}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{l}\mathbf{l}^T$, $\mathbf{l} = [1, 1, ..., 1]^T$ is a $N \times 1$ vector, and $\mathbf{I}$ is a $N \times N$ identity matrix. Revisiting the covariance matrix in the "feature space" denoted as $\bar{C}$, we can now compute the n^{th} (orthonormal) eigenvector of $\bar{C}$ as follows

$$V_n = \sum_{i=1}^{N} \frac{u_{ni}}{\sqrt{\lambda_n}} \tilde{\varphi}(X_i) = \frac{1}{\sqrt{\lambda_n}} \tilde{\phi} u_n, \tag{20}$$

where $\tilde{\phi} = [\tilde{\varphi}(X_1), \tilde{\varphi}(X_2), ..., \tilde{\varphi}(X_N)]$. Now let X be *any* element of the input space $\mathcal{M}$. The projection of X on the kernel PCA space, spanned by the first l eigenvectors of $\mathbf{C}$, is then given by

$$P^l \varphi(X) = \sum_{n=1}^{l} \omega_n V_n + \bar{\varphi} \tag{21}$$

with $\omega = [\omega_1, \omega_2, ..., \omega_l]$ being the KPCA weights or coefficients. **This is our l-dim KPCA subspace**.

As in the case of linear PCA, if we would like to perform the issue of simply dimensional reduction, then we should set the weights, ω_i, to be:

$$\omega_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N} u_{ij} \tilde{k}(\mathbf{x}, x_i) \tag{22}$$

where

$$\tilde{k}(x,y) = \langle \tilde{\varphi}(x), \tilde{\varphi}(y) \rangle = k(x,y) - \frac{1}{N}\mathbf{l}^T k_x - \frac{1}{N}\mathbf{l}^T k_y + \frac{1}{N^2}\mathbf{l}^T K \mathbf{l} \tag{23}$$

$$\text{with } k_x = [k(\mathbf{x}, x_1), ..., k(\mathbf{x}, x_N)]^T \tag{24}$$

We note that while the above formulation allows us to construct the projection from a linear combination involving the weights, **it is only valid in the feature space.** That is, given a test point $\mathbf{x} \in \mathcal{M}$, we would ideally like to compute its projection in the image space $\hat{\mathbf{x}} = \varphi^{-1}(P^l \varphi(\mathbf{x}))$. Unfortunately, because the map $\varphi : \mathbb{R}^n \mapsto \mathcal{H}$ is unknown, one can not directly obtain this projection. **This is referred to in the literature as the *pre-image problem*.** In what follows, we present a non-iterative pre-image result of to directly evolve the KPCA coefficients. However, before doing so, we first present a few popular kernels and their corresponding pre-image result.

5. POPULARIZED KERNELS

5.1 Linear PCA & (Nonlinear) Polynomial Kernels

We present the polynomial kernel which can be used to model both the bond yield curve and can also function as linear PCA. This polynomial kernel is given by

$$k_{\varphi_P}(\varphi_i, \varphi_j) = (c + \langle x_i, x_j \rangle)^d \tag{25}$$

where c is any constant, d is the degree (odd) of the polynomial, and x_i is the data points in our input space. Then by choosing $c = 0$ and $d = 1$, we arrive at the following kernel used to perform classical PCA.

$$k_{id}(x_i, x_j) = \langle x_i, x_j \rangle \tag{26}$$

The subscript *id* stands for the identity function: when performing linear PCA, the kernel used is the inner scalar product in the input space. Hence, the corresponding mapping function $\varphi = id$. One should note that this latter integral is infinite if there is no bound. However, since this is being used in a financial learning context, there is an assumption on the size of the data within the given training set that gives the necessary boundedness and finiteness result.

Another interesting option for the above, would be to choose $d = 2$. In particular, for the bond yield curve which by construction is nonlinear, this kernel may find a suitable proper result.

5.2 Exponential PCA

Choosing various types of nonlinear kernel functions $k(x_a, x_b)$ is the basis of nonlinear PCA. The exponential kernel has been a popular choice in the machine learning community and has proven to nicely extract nonlinear structures from data sets. This kernel is given by

$$k_{\varphi_\sigma}(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{27}$$

where σ^2 is a variance parameter estimated *a priori* and $\|x_i - x_j\|^2$ is the squared L_2-distance between two observations x_i and x_j. The subscript φ_σ stands for the nonlinear mapping corresponding to the exponential kernel. This mapping also depends on the choice of σ.

6. PRE-IMAGE PROBLEM AND APPROXIMATION

In this section, we revisit the closed-form pre-image approximation that was developed internally in the Tannenbaum group. It can be seen that although the φ map is not necessarily known, we ideally would like to reconstruct the pre-image $\hat{\mathbf{x}}$ of a corresponding test point $\mathbf{x} \in \mathcal{M}$ such that the distance between the feature point $\varphi(\hat{\mathbf{x}})$ and the projection in the PCA space $P^l \varphi(\mathbf{x})$ is minimized, i.e.,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathcal{M}} \|\varphi(\hat{\mathbf{x}}) - P^l \varphi(\mathbf{x})\|^2.$$

This can be achieved by minimizing the following error

$$\begin{aligned}\rho(\hat{x}) &= \|\varphi(\hat{\mathbf{x}}) - P^l\varphi(\mathbf{x})\|^2 \\ &= k(\hat{x},\hat{x}) - 2\langle\varphi(\hat{x}), P^l\varphi(x)\rangle + \|P^l\varphi(x)\|^2 \end{aligned} \tag{28}$$

where $P^l\varphi(x)$ is the projection of *any* observations in the kernel PCA space. Using the kernel notation, we can rewrite the middle term.

$$\begin{aligned}\langle\varphi(\hat{x}), P^l\varphi(x)\rangle &= \varphi(\hat{x})^T\Big[\sum_{n=1}^{l}\omega_n V_n + \bar{\varphi}\Big] \\ &= \varphi(\hat{x})^T\Big[\sum_{n=1}^{l}\omega_n\Big[\Big(\sum_{i=1}^{N}\frac{u_{ni}}{\sqrt{\lambda_n}}\,\varphi(x_i)\Big) \\ &\qquad - \bar{\varphi}\Big(\sum_{i=1}^{N}\frac{u_{ni}}{\sqrt{\lambda_n}}\;\Big)\Big] + \bar{\varphi}\Big] \\ &= \sum_{i=1}^{N}\tilde{\gamma}_i k(\hat{x}, x_i)\end{aligned} \tag{29}$$

where

$$\gamma_i = \sum_{n=1}^{l}\frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} \qquad \text{and} \qquad \tilde{\gamma}_i = \gamma_i + \frac{1}{N}\Big(1 - \sum_{j=1}^{N}\gamma_j\Big).$$

Plugging the result of (29) into (28), the extremum can be obtained by setting $\nabla_{\hat{x}}\rho = 0$. **Also, we prefer that the pre-image have a closed-form solution so that it can be used to reconstruct out-of-sample observations with the eventual ability to filter over this possible state-space**.

6.1 Pre-Image for Linear PCA

We begin with a pre-image approximation for the polynomial kernel presented in Section 5.2. From this, we can then simplify the resulting pre-image approximation to that of linear PCA. By setting $\nabla_{\hat{x}}\rho = 0$, the following pre-image of the projection in the KPCA space is

$$\begin{aligned}\hat{x} &= \sum_{i=1}^{N}\tilde{\gamma}_i\Big(\frac{k(\hat{x},x_i)}{k(\hat{x},\hat{x})}\Big)\cdot\Big(\frac{k(\hat{x},x_i)}{k(\hat{x},\hat{x})}\Big)^{-\frac{1}{d}}x_i \\ &= \sum_{i=1}^{N}\tilde{\gamma}_i\Big(\frac{k(\hat{x},x_i)}{k(\hat{x},\hat{x})}\Big)^{\frac{d-1}{d}}x_i \\ &= \sum_{i=1}^{N}\tilde{\gamma}_i\Big(\frac{c+\langle\hat{x},x_i\rangle}{c+\langle\hat{x},x\rangle}\Big)^{d-1}x_i.\end{aligned} \tag{30}$$

However, if we use the approximation $\varphi(\hat{x}) \approx P^l\varphi(x)$, which amounts to assuming that $d^2_{\mathcal{H}}(P^l\varphi(x),\varphi(x_i)) \propto d^2_{\mathcal{H}}(\varphi(\hat{x}),\varphi(x_i))$, one has

$$k(\hat{x},x_i) = \frac{1}{2}\big(\|P^l\varphi(x)\|^2 + k(x_i,x_i) - d^2_{\mathcal{H}}(P^l\varphi(x),\varphi(x_i))\big)$$

Then, the following expression to reconstruct the pre-image is obtained:

$$\hat{x} = \sum_{i=1}^{N}\tilde{\gamma}_i\Big(\frac{\|P^l\varphi(x)\|^2 + k(x_i,x_i) - d^2_{\mathcal{H}}(P^l\varphi(x),\varphi(x_i))}{2\|P^l\varphi(x)\|^2}\Big)^{\frac{d-1}{d}}x_i.$$

Interestingly, if we now take $d = 1$ in Equation (30), one gets the expression for performing linear PCA, i.e.,

$$\hat{x} = \sum_{i=1}^{N} \tilde{\gamma}_i x_i. \tag{31}$$

More importantly, we can now see that the pre-image approximation $\hat{x}$ presented in Equation (31) depends on the weight ω_i. **This is the same result that we have with classic linear PCA!**

6.1.1 Pre-Image for NonLinear PCA

For the exponential kernel in Equation (27), setting $\nabla_{\hat{x}}\rho = 0$ yields

$$\begin{aligned}\hat{x} &= \frac{\sum_{i=1}^{N} \tilde{\gamma}_i k(\hat{x}, x_i) x_i}{\sum_{i=1}^{N} \tilde{\gamma}_i k(\hat{x}, x_i)} \\ &= \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(-\|\hat{x} - x_i\|^2/(2\sigma^2)\big) x_i}{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(-\|\hat{x} - x_i\|^2/(2\sigma^2)\big)}\end{aligned} \tag{32}$$

Again, we assume that $\varphi(\hat{x}) \approx P^l \varphi(x)$. Moreover, they also assume that the Mercer kernel which is chosen can be inverted. Thus, for an exponential kernel of Equation (27), one has

$$d^2_{\mathcal{M}}(\hat{x}, x_i) = -2\sigma^2 \log\{\frac{1}{2}(2 - d^2_{\mathcal{H}}(\varphi(\hat{x}), \varphi(x_i)))\}.$$

Plugging this result into Equation (32) with the above approximation yields the following

$$\begin{aligned}\hat{x} &= \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(-d^2_{\mathcal{M}}(\hat{x}, x_i)/(2\sigma^2)\big) x_i}{\sum_{i=1}^{N} \tilde{\gamma}_i \exp\big(-d^2_{\mathcal{M}}(\hat{x}, x_i)/(2\sigma^2)\big)} \\ &\approx \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \Big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\Big) x_i}{\sum_{i=1}^{N} \tilde{\gamma}_i \Big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\Big)}.\end{aligned} \tag{33}$$

Of course, the above approximations were made so that one can reconstruct an out-of-sample observation for which we will detail next.

7. NONLINEAR PCA: RECONSTRUCTING OUT-OF-SAMPLE OBSERVATIONS

In this section, we detail the derivation in the probable event that we would like to reconstruct the out-of-sample observation with respect to our l-dim KPCA subspace. While linear PCA, this can be done trivially, we must take proper care for nonlinear PCA. Let us begin with the idea that we observe an out-of-sample observation $\mathbf{x}$, we would like to find the set of weights ω_i that approximates and reconstructs this observation with respect to our *learnt* space. That is, our out-of-sample observation x^{**} is assumed to lie within our subspace (as in this case in PCA) when used to estimate one's predictive power.

$$E(\omega) = ||x^{**} - \hat{x}(\omega)||^2 \tag{34}$$

From this, one can see that the gradient w.r.t. to ω is

$$\frac{\partial E}{\partial \omega_i} = 2 \cdot ||x^{**} - \hat{x}(\omega)|| \cdot \frac{\partial \hat{x}(\omega)}{\partial \omega} \tag{35}$$

If we choose opt to utilize linear PCA, then $\frac{\partial \hat{x}(\omega)}{\partial \omega} = v_i$. However, if we opt to utilize the exponential kernel k_{φ_σ}, $\hat{x}(\omega)$ which is given as

$$\hat{x}(\omega) = \frac{\sum_{i=1}^{N} \tilde{\gamma}_i \Big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\Big) x_i}{\sum_{i=1}^{N} \tilde{\gamma}_i \Big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\Big)}. \tag{36}$$

then $\frac{\partial \hat{x}(\omega)}{\partial \omega_i}$ can be shown to be:

$$\frac{\partial \hat{x}}{\partial \omega_i} = \frac{\sum_i^N \eta_i \cdot x_i}{\sum_i^N \mathcal{J}_i} - \frac{\big(\sum_i^N \eta_i\big)\big(\sum_i^N \mathcal{J}_i \cdot x_i\big)}{(\sum_i^N \mathcal{J}_i)^2}, \tag{37}$$

where

$$\mathcal{J}_i = \tilde{\gamma_i}\big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\big)$$

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\big) + ...$$
$$\frac{\tilde{\gamma_i}}{\sqrt{\lambda_n}}\big(\frac{1}{N^2}(\mathbf{l}^T\mathbf{K}\mathbf{l})\mathbf{l} - \frac{1}{N}\mathbf{K}\mathbf{l} + \mathbf{k}_{\varphi_\mathbf{i}} - \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\big)^T u_n - \omega_n.$$

and where $\mathbf{k}_{\varphi_\mathbf{i}} = [k_{\varphi_\sigma}(x_i, x_1), \ldots, k_{\varphi_\sigma}(x_i, x_N)]^T$.

7.1 Derivation of Equation (37)

If we let

$$\mathcal{J}_i = \tilde{\gamma_i}\big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\big) \quad \text{and} \quad \eta_i = \frac{\partial \mathcal{J}_i}{\partial \omega_n}$$

then the general gradient form is

$$\frac{\partial \hat{x}}{\partial \omega_n} = \frac{\sum_i^N \eta_i \cdot x_i}{\sum_i^N \mathcal{J}_i} - \frac{\big(\sum_i^N \eta_i\big)\big(\sum_i^N \mathcal{J}_i \cdot x_i\big)}{(\sum_i^N \mathcal{J}_i)^2}$$

Thus, taking the derivative of $\mathcal{J}_i$ w.r.t to ω_i yields

$$\eta_i = \underbrace{\nabla_{\omega_n}\tilde{\gamma_i}}_{(a)} \cdot \big(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\big) -$$
$$\frac{1}{2}\tilde{\gamma_i} \cdot \underbrace{\nabla_{\omega_n} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))}_{(b)}$$

Now taking the derivative of (a) above by using Equation (30), we get

$$\tilde{\gamma_i} = \sum_n^l \frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} + \frac{1}{N}\left(1 - \sum_j^N \sum_n^l \frac{\omega_n u_{nj}}{\sqrt{\lambda_n}}\right)$$
$$\nabla_{\omega_n}\tilde{\gamma_i} = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}}$$

Next, we compute the gradient of part (b). Recall that

$$d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i)) = \|P^l\varphi(x)\|^2 + \|\varphi(x_i)\|^2 - 2\langle P^l\varphi(x), \varphi(x_i)\rangle$$

From this, we can begin to express each term's dependency on the shape weight ω_n. The first term can be expressed as

$$\begin{aligned}
\|P^l\varphi(x)\|^2 &= \big(\sum_n^l \omega_n V_n + \bar{\varphi}\big)^T \big(\sum_n^l \omega_n V_n + \bar{\varphi}\big) \\
&= \big(\sum_n^l \sum_n^l \omega_n V_n^T V_n \omega_n\big) + \bar{\varphi}^T\bar{\varphi} + \big(2\bar{\varphi}^T \sum_n^l \omega_n V_n\big) \\
&= \big(\sum_n^l \omega_n^2\big) + \bar{\varphi}^T\bar{\varphi} + \big(2\bar{\varphi}^T \sum_n^l \omega_n V_n\big) \\
&= \big(\sum_n^l \omega_n^2 + 2\frac{\omega_n}{\sqrt{\lambda_n}}\bar{\varphi}^T\tilde{\phi}u_n\big) + \bar{\varphi}^T\bar{\varphi}
\end{aligned}$$

Similarly, the third term can be expressed as

$$\begin{aligned}
\langle P^l\varphi(x), \varphi(x_i)\rangle &= \Big(\sum_n^l \omega_n V_n + \bar{\varphi}\Big)^T \varphi(x_i) \\
&= \Big(\sum_n^l \big(\omega_n V_n\big)^T \varphi(x_i)\Big) + \bar{\varphi}^T\varphi(x_i) \\
&= \Big(\sum_n^l \omega_n (\frac{1}{\sqrt{\lambda_n}}\tilde{\phi}u_n)^T\varphi(x_i)\Big) + \bar{\varphi}^T\varphi(x_i) \\
&= \Big(\sum_n^l \frac{\omega_n}{\sqrt{\lambda_n}}(u_n^T\tilde{\phi}^T\varphi(x_i))\Big) + \bar{\varphi}^T\varphi(x_i)
\end{aligned}$$

Combining these terms then gives us

$$\begin{aligned}
d_{\mathcal{H}}^2 = &\Big(\sum_n^l \omega_n^2 + 2\frac{\omega_n}{\sqrt{\lambda_n}}(\tilde{\phi}^T\bar{\varphi})^T u_n - 2\frac{\omega_n}{\sqrt{\lambda_n}}(\tilde{\phi}^T\varphi(x_i))^T u_n\Big) \\
&-2\bar{\varphi}^T\varphi(x_i) + \bar{\varphi}^T\bar{\varphi} + \|\varphi(x_i)\|^2
\end{aligned}$$

Now, taking the derivative w.r.t ω_n, we arrive at the following:

$$\begin{aligned}
\frac{d(d_{\mathcal{H}}^2)}{d\omega_n} &= 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}}\cdot\big(\tilde{\phi}^T\bar{\varphi} - \tilde{\phi}^T\varphi(x_i)\big)^T u_n\Big) \\
&= 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}}\cdot\big((\phi-\bar{\phi})^T\bar{\varphi} - (\phi-\bar{\phi})^T\varphi(x_i)\big)^T u_n\Big) \\
&= 2\Big(\omega_n + \frac{1}{\sqrt{\lambda_n}}\cdot\big(\phi^T\bar{\varphi} - \bar{\phi}^T\bar{\varphi} - \phi^T\varphi(x_i) + \bar{\phi}^T\varphi(x_i)\big)^T u_n\Big) \\
&= 2\Big(\frac{1}{\sqrt{\lambda_n}}(\frac{1}{N}\mathbf{K}\mathbf{l} - \frac{1}{N^2}(\mathbf{l}^T\mathbf{K}\mathbf{l})\mathbf{l} - \mathbf{k}_{\varphi_\mathbf{i}} + \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\big)^T u_n + \omega_n\Big)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{k}_{\varphi_\mathbf{i}} &= [k_{\varphi_\sigma}(x_i, x_1), \ldots, k_{\varphi_\sigma}(x_i, x_N)]^T \\
\phi &= [\varphi(x_1), \varphi(x_2), ..., \varphi(x_N)] \qquad \text{and} \qquad \bar{\phi} = \phi - \tilde{\phi}.
\end{aligned}$$

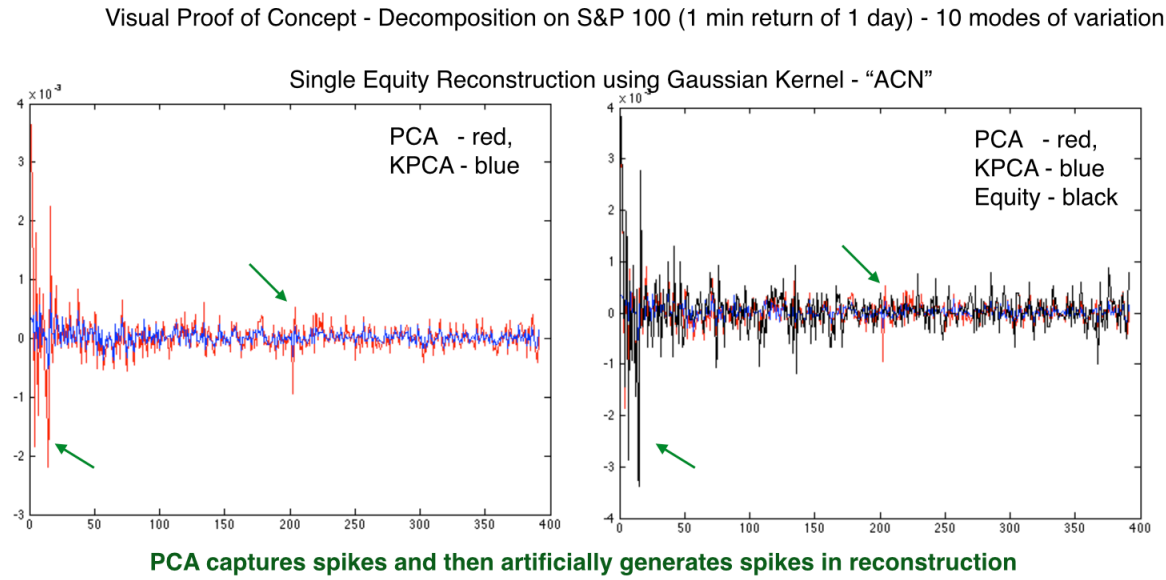

Figure 1. This is a reconstruction result of the equity "ACN" for the Gaussian/Exponential kernel. One can see that we are not as sensitive to outliers like that of linear PCA. We "push" these outliers as noise.

Putting all of this together, we arrive at the desired gradient.

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N}\sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \left(1 - \frac{1}{2} d^2_{\mathcal{H}}(P^l\varphi(x), \varphi(x_i))\right) + \ldots$$
$$\frac{\tilde{\gamma_i}}{\sqrt{\lambda_n}}\left(\frac{1}{N^2}(\mathbf{l}^T\mathbf{K}\mathbf{l})\mathbf{l} - \frac{1}{N}\mathbf{K}\mathbf{l} + \mathbf{k}_{\varphi_\mathbf{i}} - \frac{1}{N}\mathbf{l}\mathbf{l}^T\mathbf{k}_{\varphi_\mathbf{i}}\right)^T u_n - \omega_n.$$

We now have all of the components necessary to compute the overall gradient used above.

8. RESULTS &SUMMARY

We present reconstruction (in-sample) results below to show difference between PCA and Nonlinear PCA (using the exponential kernel). The framework itself can (at the least) serve as a lower bound (linear PCA) and we do not loose any capabilities by generalizing to KPCA.

9. REFERENCES

[1] S. Mika, B. Scholkopf, A. J. Smola, K.-R. Muller, M. Scholz, and G. Ratsch. Kernel PCA and de?noising in feature spaces. In Advancesin Neural Information Processing Systems, volume 11. MIT Press, 1999.
[2]. S. Dambreville, Y. Rathi, and A. Tannenbaum. A framework for image segmentation using shape models and kernel space shape priors. TPAMI, 30(8):1385?1399, 2008.
[3] J. Kwok and I. Tsang. The pre-image problem in kernel methods. In IEEE transactions on neural networks, volume 15, pages 15171525, 2004.
[4] B. Scholkopf, A. Smola, and K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, tech. rep., Max-Planck-Institute fur biologische Kybernetik, 1996.
[5] Sandhu, R., Dambreville, S., Yezzi, A., Tannenbaum, A.: A Nonrigid Kernel-Based Framework for 2D-3D Pose Estimation and 2D Image Segmentation. T-PAMI 33 (2011) 10981115.

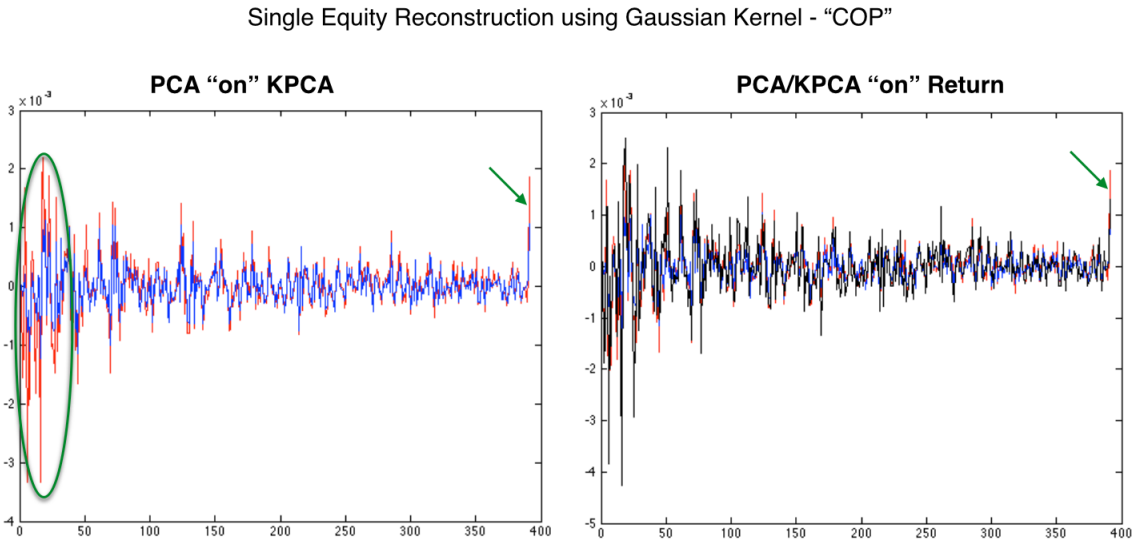

Figure 2. This is a reconstruction result of the equity “COP” for the Gaussian/Exponential kernel. One can see that we are not as sensitive to outliers like that of linear PCA. We ”push” these outliers as noise.

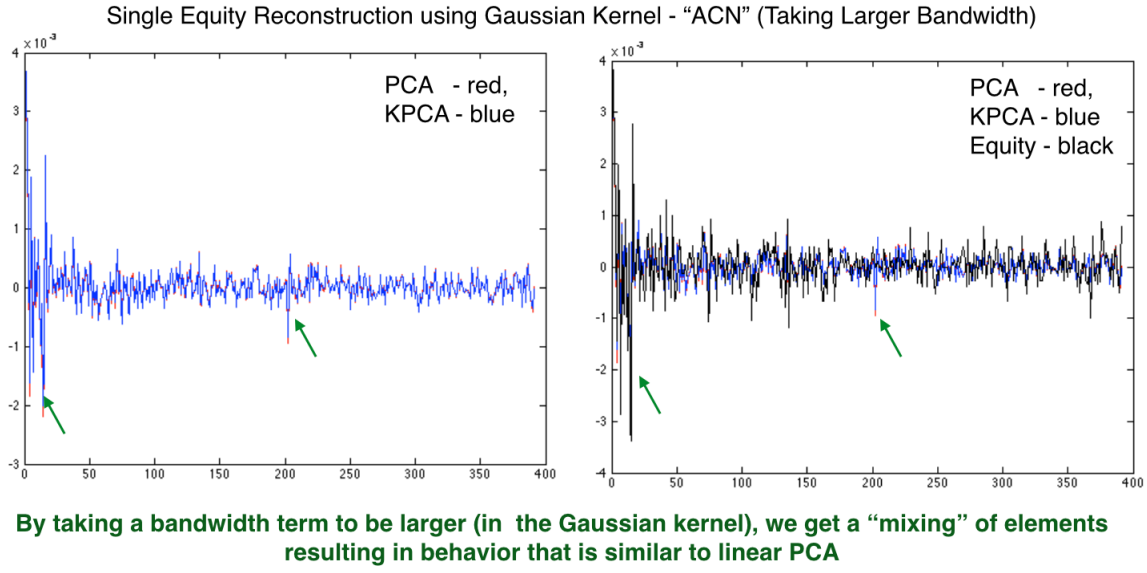

Figure 3. This is a reconstruction result when we take a larger bandwidth term. In short, this term can be interpreted as a mixing element that when taking quite large, will mimic PCA results. Another manner to mimic PCA results is to also simply choose a linear pac kernel.

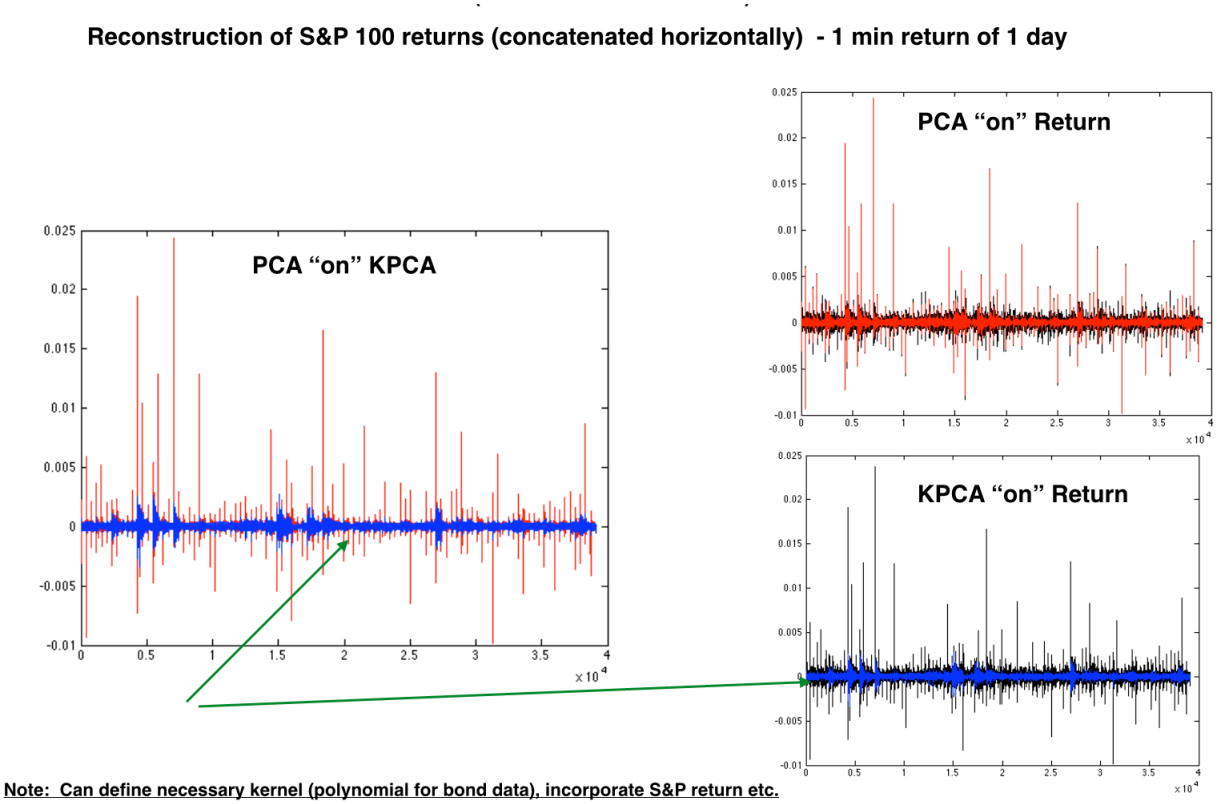

Figure 4. Results are concentrated to see the overall behavior of all equities for 1 min returns for 1 day. We see the behavior and ability to not capture outliers.