P1.5 KERNEL PCA ANALYSIS FOR REMOTE SENSING DATA

John Tan[*], Ruixin Yang, Menas Kafatos
Center for Earth Observing & Space Research (CEOSR)
George Mason University (GMU)
Fairfax, VA 22030

ABSTRACT

Kernel principal component analysis (KPCA) is an efficient generalization of traditional principal component analysis (PCA) that allows for the detection and characterization of low-dimensional nonlinear structure in multivariate datasets. As with the PCA algorithm KPCA maximizes the variance of the data points, but in a new coordinate system nonlinearly related to the original input space. Several well known algorithms for dimensionality reduction such as Isomap, graph Laplacian eigenmap, and locally linear embedding (LLE) can be recast as KPCA by using an appropriate kernel. The kernel itself can also be optimized or chosen to facilitate dimension reduction. This study presents the application of KPCA for dimension reduction on datasets with inherent non-linear structure, such as the Lorenz attractor. The potential of performing KPCA analysis on earth science data will also be discussed.

1. INTRODUCTION

Subspace methods such as Principal Component Analysis (PCA; *aka* empirical orthogonal function (EOF) analysis) and Kernel Principal Component Analysis (KPCA) are a family of algorithms that perform feature extraction and pattern recognition on potentially high dimensional data sets. Additional related applications involve dimension reduction and denoising of data sets. PCA and its nonlinear extension KPCA are subspace algorithms that define a new coordinate system to describe the data, in the expectation that it represents the features of interest. However, linear classifiers such as PCA do not deal well with data that is nonlinearly separable, or data that is noisy. One solution is to map the dataset into a richer feature space, one that may include nonlinear features; KPCA is an example of such a method.

KPCA is related to modern dimensionality reduction techniques that attempt to discover nonlinear sub-manifolds by utilizing a similarity measure between data points (Burges 2004). Isomap, graph Laplacian eigenmap, and locally linear embedding (LLE) are well known dimension reduction algorithms that can be described as KPCA by using or calculating the proper kernel (Ham et al. 2003). In addition to these relationships, specifying a proper kernel for particular data sets will allow KPCA to be used for dimensionality reduction (Weinberger, Sha, & Saul 2004).

2. KERNEL PRINCIPAL COMPONENT ANALYSIS

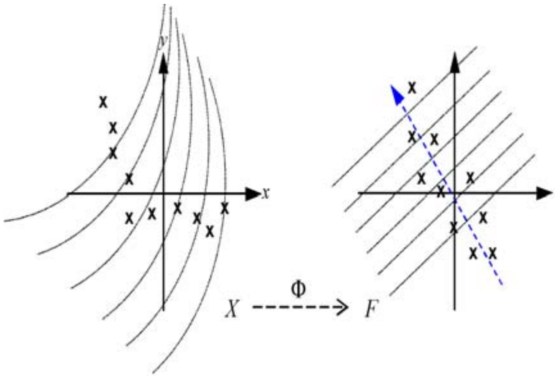


Figure 1. Demonstration of KPCA (reproduced based on Figure 1 in Schölkopf et al. 1999).

KPCA is simply the PCA algorithm applied to data points that are first mapped into another feature space *F* from the original input

[*] Corresponding author address: John Tan, MS 5C3, School of Computational Sciences, George Mason University, Fairfax, VA 22030; e-mail: jtan@gmu.edu.

space *X*, as demonstrated in Figure 1. This is performed through a mapping function Φ,

$$\Phi : X \rightarrow F \,.$$

The subspace of this feature space *F* is found using PCA and that is the feature space that we refer to when transforming from input space to feature space and back. This feature space is a finite subspace of the space *F* which can be infinite dimensional.

Schölkopf, Smola, and Muller (1998) introduced kernel PCA as a nonlinear generalization of PCA. The KPCA algorithm is the same as standard PCA but in the mapped space *F*. This mapping can be to a large even infinite dimensional space where the calculations can be computationally prohibitive or even impossible. To obtain the advantages of a rich high dimensional feature space without the cost, the substitution known as the kernel "trick" is applied. This is the replacement of all dot products in the algorithm with a kernel function:

$$K(\underline{x},\underline{y}) := (\Phi(\underline{x})\cdot\Phi(\underline{y})) = f\,(\underline{x},\underline{y}) \qquad (1)$$

The substitution of the kernel function removes the explicit use of the map Φ and along with it the need to perform calculations in the mapped space. The kernel can be any function that satisfy Mercer's theorem; it must be a (semi)positive definite, symmetric function.

Two examples of a Mercer's kernel are given in the following equations:

$$K(\underline{x},\underline{y}) = \underline{x}\cdot\underline{y} \qquad (2)$$

and

$$K(\underline{x},\underline{y}) = e^{\frac{-\left\|\underline{x}-\underline{y}\right\|^2}{2\sigma^2}} \qquad (3)$$

Equation 4 is the first order polynomial kernel, and when used the KPCA algorithm reduces to standard PCA. The other is the Gaussian Radial Basis Function (RBF) kernel. This RBF kernel is a mapping to infinite dimensions, which has been used successfully in denoising and classification applications. Both kernels are utilized in the examples below.

The transformation of points in the subspace of *F* back to the original input space is performed by calculating the preimage. For the linear case of PCA, this only requires a matrix transpose and multiplication. However this is a nontrivial problem for KPCA with nonlinear kernels. The solutions to this nonlinear problem are not guaranteed to be unique. Schölkopf et al. (1999) deals with the preimage problem for a general class of kernels by solving an optimization problem. This can be expressed as a point iteration formula for many kernels and is quite efficient.

3. DATA SETS

The first application of KPCA is to a three dimensional data set of 300 points. The structure of the data is generated by adding uniform random noise to a parabola.

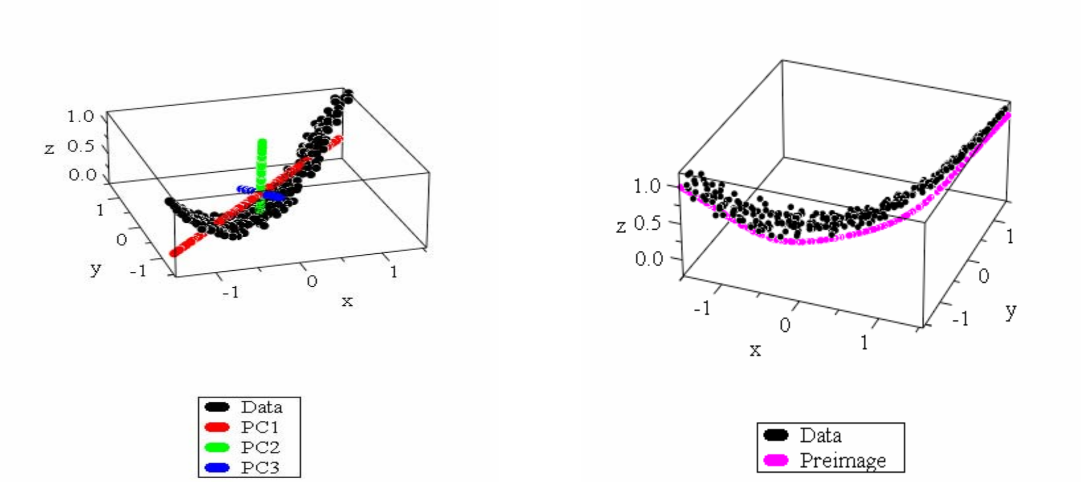


Figure 2. The preimage of principal components 1, 2, 3 of standard PCA (left) and the resulting preimage from projections onto principal components 1, 2, 3 of the RBF kernel (right).

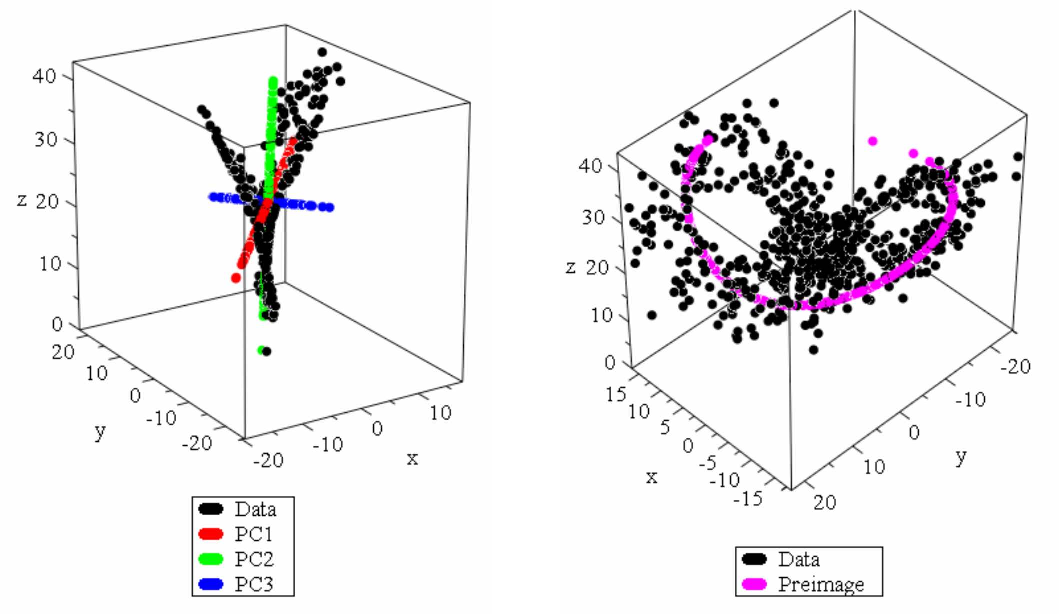
$$
\begin{aligned}
x(t) &= \cos(\frac{\pi}{4})(t-t_0)+rand_1 \\
y(t) &= \sin(\frac{\pi}{4})(t-t_0)+rand_2 \\
z(t) &= \frac{x(t)^2}{2} \\
rand_1&,\ rand_2 \le \pm 0.2
\end{aligned}
\tag{5}
$$

Standard PCA (via the first order polynomial kernel) applied to the data set generates three principal components, each capturing a fraction of the total variance. The preimages of the first principal component capture the most variance of the three as seen in Figure 2 (on left, red line). However the curve generated from the preimages using the Gaussian RBF kernel (Figure 2 on right) is nonlinear and so can follow the trend of the data more closely

The next data set analyzed is the Lorenz attractor. The 600 point data set is generated by sampling the solutions of the nonlinear ordinary differential equations describing the attractor:

$$
\begin{aligned}
\frac{dx}{dt} &= -sx+sy \\
\frac{dy}{dt} &= -xz+rx-y \\
\frac{dz}{dt} &= xy-bz
\end{aligned}
\tag{6}
$$

s=10, *r=28, and b=8/3.*



(a) (b)

Figure 3. The preimage of principal components 1, 2, 3 of standard PCA on the Lorenz data set (a). Three dimensional view of Figure 4 with principal components 1, 2 resulting from the RBF kernel (b).

This data set represents a fractal structure with a box-counting dimension near 2.04 (Monahan 2000). The butterfly-shaped attractor has a symmetry that indicates a U-shaped curve tracing the symmetry of the two lobes will explain much of its structure. Figure 3 (a) is the results of the standard PCA. The red line is the preimages for the first principal component, and this is the linear structure that captures the most variance of the attractor. Unfortunately it can only account for linear variation. Figure 3 (b) shows the nonlinear curve generated from the RBF kernel more closely follows the nonlinear structure of the attractor.

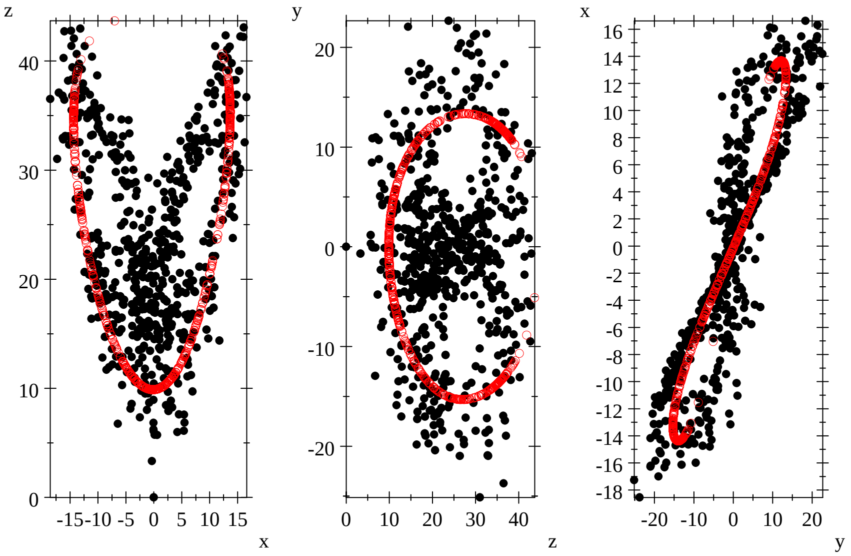


Figure 4. Preimage projections of the Lorenz data set using the first two nonlinear PC's that resulted from the RBF kernel. Black dots represent the data set. Red circles represent the preimage.

The KPCA results for the Lorenz attractor are very similar in structure with solutions from Nonlinear Principal Component Analysis (NLPCA) using feed forward neural networks (Monahan 2000). Additionally, the results from NLPCA have been shown empirically to agree with Principal Curves and Surfaces (PCS) for continuous projection functions (Monahan 2000). PCS is the true generalization of PCA to nonlinear components with a solid theoretical grounding.

4. DISCUSSION

The advantage of using KPCA over other nonlinear feature extraction algorithms can be significant computationally. KPCA does not require solving a nonlinear optimization problem which is expensive computationally and the validity of the solution as optimal is typically a concern. KPCA only requires the solution of an eigenvalue problem. This reduces to using linear algebra to perform PCA in an arbitrarily large, possibly infinite dimensional, feature space. The kernel "trick" greatly simplifies calculations in this case. The preimage calculation, if required for the problem being solved, can require an optimization routine. However, the preimage formula from Schölkopf et al. (1999) provides an avenue for developing fast and accurate point iteration techniques for various kernels.

An additional advantage of KPCA is that the number of components does not have to be specified in advance. Solving the eigenvalue problem using algorithms for a symmetric matrix such as the Jacobi iteration method will return all the components at once. KPCA reduces to PCA in the case of the first order polynomial kernel. Furthermore the capability to use various kernels for different data sets and/or different goals is compelling. This same advantage does lead to the problem of selecting the right parameters and the right kernel for the task at hand.

KPCA compared to NLPCA and other neural approaches can have a disadvantage if the data set used for training is very large. The eigenvalue problem is for a symmetric *M*x*M* matrix, where *M* is the size of the training set. Typical eigenvalue solvers require O(M^3) operations and O(M^2) storage (Press et al. 1992). So it is easy to see how the problem can become intractable if *M* is sufficiently large. However recently, sparse greedy methods have become available for performing approximate KPCA (Schölkopf & Smola 2002).

Another limitation of the KPCA algorithm is that it can be hard to interpret results in input space. The eigenvectors are not guaranteed to exist in input space for every kernel (though a good approximation can always be calculated). Additionally the variance measurements in feature space do not translate back in the input space. So compared to PCS, KPCA is harder to interpret in input space. Though some kernels are well understood, for example, the polynomial kernels have a clear interpretation in terms of higher-order features.

Standard PCA is well established and used widely in many diverse fields. KPCA is a useful generalization that can be applied to these domains where nonlinear features require a nonlinear feature extraction tool.

We plan to use the KPCA algorithm on real earth science data such as the sea surface temperature (SST) or normalized difference vegetation index (NDVI). The resulting information from KPCA can be correlated with signals such as the Southern Oscillation Index (SOI) for determining relationships with the El Nino phenomenon. KPCA can be used to discover nonlinear correlations in data that may otherwise not be found using standard PCA. The information generated about a data set using KPCA captures nonlinear features of the data. These features correlated with known spatial-temporal signals can discover nonlinear relationships. KPCA offers improved analysis of datasets that have nonlinear structure.

REFERENCES

Burges J. 2004: Geometric Methods for Feature Extraction and Dimensional Reduction: A Guided Tour. Technical Report MSR-TR-2004-55, Microsoft Research

Ham J., D. Lee, S. Mika, B. Schölkopf. 2003: A kernel view of the dimensionality reduction of manifolds. Technical Report 110, Max-Planck-Institut fur biologische Kybernetik.

Monahan, A. H. 2000. Nonlinear principal component analysis by neural networks: theory and application to the Lorenz system. *J. Climate*, 13, 821-835.

Schölkopf B., A. J. Smola, 2002: Learning with Kernels. The MIT Press.

Schölkopf B., A. Smola, K. Muller. 1998: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319.

Schölkopf B., S. Mika, C. J. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. J. Smola, 1999: Input Space Versus Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, Vol 10, No. 5

Weinberger K., F. Sha, L. Saul. 2004: Learning a Kernel Matrix for Nonlinear Dimensionality Reduction. Proceedings of the 21st International Conference on Machine Learning, Banff, Canada.

Press W., B. Flannery, S. Teukolsky, W. Vetterling. 1992: Numerical Recipes in C (2nd ed.), Cambridge University Press, New York.