Group Project Summary Report on Paper: Non-Linear Component Analysis as a Kernel Eigen Value Problem   
 -- Huan Yan, Jifu Zhao, Pavan Kumar Nadiminti, and Vikram Idiga

This paper is a new approach to a traditional problem of identifying major modes/components of a given data. Traditionally Principal Component Analysis (PCA) is used to detect major structure within data. Essentially it is a eigen value problem in which we transform the coordinate axes in such a way that the variances of the data set we have, lie mainly along those axes. These directions are called as principal components. The PCA approach basically determines the components extracted from the data in the same space, i.e. the input space. However, in some cases it is useful to look at the components in a space of the original variables/features of the experiment, instead of the derived input space. Here the input space refers to the data we have in the current form, which might usually have been derived from a nonlinear transformation of variables’ data in feature space. The dimensionality of the feature space might also be different from that of the input space. This paper develops a frame work to handle this task of performing a form of nonlinear PCA by using kernel function approach. The kernel functions are essentially the dot products in the feature space transformed by the non-linearity relating the input space to the feature space. The key feature of the Kernel PCA method is that we do not need to carry out the mapping of the data points into the feature space. All calculations can be done using the kernel function in the input space.

Feature Space PCA

The entire analysis is done on centered data. Here we perform PCA analysis and explore ways to represent the PCA equations in terms of dot products. Basic PCA involves computing the covariance matrix:

and solving the eigen value problem λ**v** = C**v**

The same can be expressed in terms of a dot product as follows:

the non-linearity between input space and feature space is represented as Φ() .We the perform the PCA in the feature space with the use of these transformations.

,

the covariance matrix in the feature space and the eigen value problem we solve in the feature space is  **.**

We then obtain the normalized principal components in F space.

Kernel PCA Method Description:

Mercer’s Theorem states that if K(**x,y**) is a continuous kernel of a positive integral operator, there exists a mapping into a space where k acts as a dot product. Dot product in F space is represented as a Kernel so that the notation is simplified:

In kernel PCA, a non-trivial arbitrary is chosen, but is never calculated explicitly, allowing the possibility to use a very high dimensional for . Hence, we never actually compute the eigen vectors and eigen values of the covariance matrix in the high dimensional feature space. Elements in the K matrix represent the dot product of one point of the transformed data with respect to all the transformed points.

where is the eigen vector matrix, and λ1, λ2 ... λM, are the eigen values of K obtained by solving .

Properties of Kernel PCA:

The salient feature of the Kernel PCA is that when we are extracting the major modes/components from the data, we are not limited to extracting only up-to the dimensionality of our data set, but we can extract further components from the data. The usual properties of PCA continue to hold here, such as: The first q principal components (the projections of eigen vectors) carry more variance than any other q orthogonal directions. The mean squared error approximation is representing the observations by first q principal components is minimal. The principal components explain most of the variation in the data.

Computational Complexity of the Procedure of Kernel PCA:

Kernel PCA is not computationally intensive, because we are not transforming the data into feature space but we are computing the dot-products using the kernel function in the input space itself. Thus this procedure is equally computationally intensive as linear PCA. When there are large number of observations we estimate the Kernel Matrix using a subset of the input data to reduce the complexity of calculations.

**Experiments by the Authors:**

To explore the effectiveness of the Kernel PCA the authors explore two examples cases.  
*toy example:*

The authors generate an artificial 2-d data set using kernel of varying degrees. Linear PCA produces only two modes for only two non-zero eigen values as the data is in 2 dimensions. In contrast the non-linear PCA allows more components to be discerned. Also the first component from non-linear PCA fits the data better than the linear one. This method also better explained how different principle components extracted using Gaussian kernels separate the clusters in the input data.

*Character Recognition Example:*

A data set of hand written data from USPS is used to perform K-PCA and then the components’ utility is assessed to classify the images using character recognition classifier. The performance of non-linear PCA is found to be better than that of linear PCA. The main reason for this is that there are many higher order features in a image than there are pixels in it. Non-linear PCA addresses this issue better.

Advantages of Nonlinear (kernel) PCA:

* In the use of pattern recognition, the components from nonlinear PCA recognized the features better than the linear PCA.
* We do not perform any non-linear optimization in this procedure
* We do not need to specify the number of principal components in advance to perform this method.

Data Description:

Algorithm: