KERNEL PRINCIPAL COMPONENT ANALYSIS FOR FEATURE REDUCTION IN HYPERSPECTRALE IMAGES ANALYSIS

Mathieu Fauvel [*◇], *Jocelyn Chanussot* [*]

[*]Laboratoire des Images et des Signaux
LIS-INPG
BP 46 - 38402 St Martin d'Heres - FRANCE

Jon Atli Benediktsson [◇]

[◇]University of Iceland
Dept. of Electrical and Computer Eng.
Hjardarhagi 2-6, 107 Reykjavik-ICELAND

ABSTRACT

Feature extraction of hyperspectral remote sensing data is investigated. Principal component analysis (PCA) has shown to be a good unsupervised feature extraction. On the other hand, this methods only focus on second orders statistics. By mapping the data onto another feature space and using nonlinear function, Kernel PCA (KPCA) can extract higher order statistics. Using kernel methods, all computation are done in the original space, thus saving computing time. In this paper, KPCA is used has a preprocessing step to extract relevant feature for classification and to prevent from the Hughes phenomenon. Then the classification was done with a backpropagation neural network on real hyperspectral ROSIS data from urban area. Results were positively compared to the linear version (PCA) and to a version of a algorithm specially designed to be use with neural network (DBFE).

1. INTRODUCTION

In the last decades, various *kernel methods* were applied successfully in pattern analysis, as well as in classification as in regression [1]. Ones of them are the well know Support Vector Machines [2]. The main idea is kernels allow to work in some feature space implicitly, while all computations are done in the input space. In practice, dot products in feature space is expressed in terms of kernel functions in input space. The major consequence from this is that *any algorithm which only uses scalar product can be turn to nonlinear version of it, using kernel methods* [3].

Principal Component Analysis (PCA) is classic linear techniques in statistical analysis. Given a set of multivariate measurements, PCA finds, using only second-order statistics, a smaller set where the feature are uncorrelated to each others. The nonlinear version of PCA, namely *Kernel Principal Component Analysis* (KPCA), is capable of capturing part of the high order statistics, thus provides more information from the original data set.

In the field of remote sensing, especially in hyperspectral imagery, reduction of the dimensionality is a key point for data analysis to prevent from Hughes phenomenon [4]. Typical method in hyperspectral processing are *Discriminant Analysis* (DAFE), which is a method that is intended to enhance class separability, and *Decision Boundary* (DBFE), which is a method that is extracted discriminately feature from the decision boundary between classes [5]. These method are linear, as PCA, but they are also focused on discriminating between classes. By definitions, both of them are supervised methods, i.e. some *a priori* informations are needed. However, these algorithms could be computationally intensive and their performance depend strongly on the training samples [5]. Unsupervised learning algorithms are an alternative way to reduce the dimensionality without any *a priori* information. Our interest in this paper lies in the application of KPCA in high dimensional space, such as hyperspectral images. Its influence on classification accuracy with neural network is thus investigated.

We start by recalling PCA and its nonlinear version KPCA. Then we describe the hyperspectral data and the experiments. The obtained results are compared to these obtained with PCA and DBFE. Finally conclusions are drawn.

2. KERNEL PRINCIPAL COMPONENT ANALYSIS

The starting point is a random vector $\mathbf{x} \in \mathbb{R}^n$ with N observations $\mathbf{x}_i,\ i \in [1,\dots,N]$. In PCA, data are first centered $\mathbf{x} \Leftarrow \mathbf{x} - E\{\mathbf{x}\}$. Then PCA diagonalizes the covariance matrix $C_{\mathbf{x}}$:

$$C_{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T. \tag{1}$$

This problem leads to solve the eigenvalue equation [6]:

$$\begin{array}{rcl} \lambda \mathbf{v} & = & C_{\mathbf{x}}\mathbf{v} \\ \|\mathbf{v}\| & = & 1 \end{array} \tag{2}$$

The authors would like to thank the IAPR - TC7 for provinding the data and Prof. Paolo Gamba and Prof. Fabio Dell'Acqua of the University of Pavia, Italy, for providing reference data. This research was supported in part by the Research Fund of the University of Iceland and the Jules Verne Program of the French and Icelandic governments (PAI EGIDE).

where $\lambda \geq 0$ are eigenvalues and $\mathbf{v} \in \mathbb{R}^n$ are eigenvectors. The projection on the eigenvector $\mathbf{v}^k$ is done by:

$$\mathbf{x}_{pc}^k = \mathbf{v}^k \cdot \mathbf{x}. \tag{3}$$

Now, suppose we first map the data onto another dot product space $\mathbb{H}$:

$$\begin{array}{rcl} \Phi : \mathbb{R}^n & \rightarrow & \mathbb{H} \\ \mathbf{x} & \rightarrow & \Phi(\mathbf{x}) \end{array} \tag{4}$$

Here, Φ could be a nonlinear function and $\mathbb{H}$ could have infinite dimensionality. PCA can be perform in $\mathbb{H}$ with the same procedure as previously: the data is centered and the covariance matrix is defined as:

$$C_{\Phi(\mathbf{x})} = \frac{1}{N} \sum_{i=1}^{N} \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \tag{5}$$

Similarly to PCA, one as to solve:

$$\begin{array}{rcl} \lambda \mathbf{v}_\Phi = C_{\Phi(\mathbf{x})} \mathbf{v}_\Phi & = & \left(\frac{1}{N} \sum_{i=1}^{N} \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T \right) \mathbf{v}_\Phi \\ & = & \frac{1}{N} \sum_{i=1}^{N} (\Phi(\mathbf{x}_i) \cdot \mathbf{v}_\Phi) \Phi(\mathbf{x}_i). \end{array} \tag{6}$$

From (6), it is clear that $\mathbf{v}_\Phi$ is lying in the span of $\Phi(\mathbf{x}_1) \ldots \Phi(\mathbf{x}_N)$, thus each eigenvector can be written as:

$$\mathbf{v}_\Phi = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{x}_i). \tag{7}$$

By multiplying (6) with $\Phi(\mathbf{x}_k)$ from the left and substituting (7) into it, we get:

$$\begin{array}{l} \lambda \sum_{i=1}^{N} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i) \right) = \\ \qquad \frac{1}{N} \sum_{i=1}^{N} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{N} \left(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) \right) \Phi(\mathbf{x}_j) \right) \end{array} \tag{8}$$

for $k \in [1, N]$.

Defining the $N \times N$ *Gram matrix* K by $K_{ij} := (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$, the above equation turns to:

$$\lambda K \boldsymbol{\alpha} = \frac{1}{N} K^2 \boldsymbol{\alpha} \tag{9}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^T$. The solution of (9) is found by solving the eigenvalue problem:

$$N \lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha} \tag{10}$$

for nonzero eigenvalues. Clearly, all solutions of (10) satisfy (9). However, it does not give all the solutions, eigenvector associate to zero eigenvalue is solution of (9) which is not a solution of (10). But, it can be shown that these solutions lead to null expansion of (7) and thus are irrelevant for the considered problem. Finally, to solve $C_{\Phi(\mathbf{x})}$'s eigenvalue equation is equivalent to solve K's eigenvalue equation.
The unitary norm condition from (2) is translated in $\mathbb{H}$ into $\lambda_k(\boldsymbol{\alpha}^k \cdot \boldsymbol{\alpha}^k) = 1$ (details in [7]). The projection in $\mathbb{H}$ is simply done by:

$$\Phi(\mathbf{x})_{kpc}^k = \mathbf{v}_\Phi^k \cdot \Phi(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})). \tag{11}$$

However, compute PCA in $\mathbb{H}$ has a high computational cost. Using *kernel trick*, it is possible to work implicitly in $\mathbb{H}$ while all computations is done in the input space. Using kernel function, the dot product in feature space is reduced to a (possibly nonlinear) function in input space:

$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{12}$$

The kernel function has to satisfy the Mercer's theorem to ensure that it is possible to construct a mapping into a space where κ acts as a dot product. The polynomial kernel and the Gaussian kernel are ones of the most used kernel:

$$\begin{array}{rcl} \kappa_{poly}(\mathbf{x}_i, \mathbf{x}_j) & = & (\mathbf{x}_i \cdot \mathbf{x}_j + r)^d \\ \kappa_{gauss}(\mathbf{x}_i, \mathbf{x}_j) & = & \exp\left(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \right). \end{array} \tag{13}$$

When builds with kernel functions, Gram matrix is also known as *Kernel Matrix*. Finally, the KPCA is done in the original space as follows:

1. Compute the Kernel Matrix: $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.
2. Center K (see [3, 7] for details):

   $$K_c = K - 1_N K - K 1_N + 1_N K 1_N$$

   where 1_N is a N square matrix for which $(1_N)_{ij} = \frac{1}{N}$, for all (i,j) in $[1, \ldots, N]$.
3. Diagonalize K_c and normalize eigenvectors:

   $$\lambda_k(\boldsymbol{\alpha}^k \cdot \boldsymbol{\alpha}^k) = 1.$$

4. Extract the k first principal components:

   $$\Phi(\mathbf{x})_{kpc}^k = \sum_{i=1}^{N} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})).$$

3. HYPERSPECTRAL DATA SET & FEATURE EXTRACTION

Hyperspectral images are characterized by a high number of bands, which are highly correlated side by side (Fig. 1, the

Table 1. Eigenvalues in percentage of variance for principal and kernel principal components.

	1	2	3	4	5	6	7
PCA (%)	65.10	94.19	98.33	98.88	99.17	99.37	99.53
KPCA (%)	25.55	43.59	59 .60	68.42	73.98	78.30	81.02

whiteness indicate the correlation). Due to the high correlation for neighboring bands, it is possible to reduce the dimensionality without losing significant information and separability. Our test image are from the ROSIS 03 sensor, the number of band is 103 with spectral coverage is from 0.43 through $0.86\mu m$. The image area is 610 by 340 pixels. PCA and KPCA were applied on that data. The kernel function used was the Gaussian kernel, where the parameter γ were set to 0.01. The Kernel Matrix were computed with 50% of the total number of pixel in the image, pixels were selected randomly. The results for the eigenvalues are shown in Table 1 and the first principal components are shown in Fig. 2.

The correlation between the first principal component extracted with PCA and KPCA is -0.54, which is significantly different. The variance of principal components provided by KPCA is smaller than those provides by the PCA. In a sense, it proves that the information of hyperspectral data could not be reduced to a very few number of bands without discarding information. By requiring higher order statistics, the number of principal components is increased, and more information are extracted.

Note that in PCA 95% of the total eigenvalue sum is achieved with the first three components while with KPCA 28 components are needed. However, the total number of components with PCA is equal to the number of channel, while with KPCA it is equal to the size of the Kernel Matrix, i.e. the number of training samples used, which is significantly higher.

The PCA and KPCA were computed using C++ and GSL library. KPCA was more time consuming, since the matrix to diagonalize was in general bigger. If a too large kernel matrix is defined, some memory problem could appear with KPCA. Anyway, this problem could be solve by selecting less pixels to build the kernel matrix.

4. EXPERIMENTS

In this experiment, the features extracted previously were used as an input of a back-propagation neural network classifier with one hidden layer. The number of neurons in the hidden layer is twice the number of outputs, i.e. the number of classes. A neural classifier was used to compare KPCA to other feature extraction methods designed for a neural network (DBFE [5]). The training set was composed of 3921 pixels with labels. 9 classes were used in the classification: asphalt, meadow, gravel, tree, metal sheet, bare soil, bitumen, brick and shadow. A quarter of the labeled samples were uses for training, the others samples were used for testing. The result were compared with classification of the full spectrum and DBFE's transformed formats.

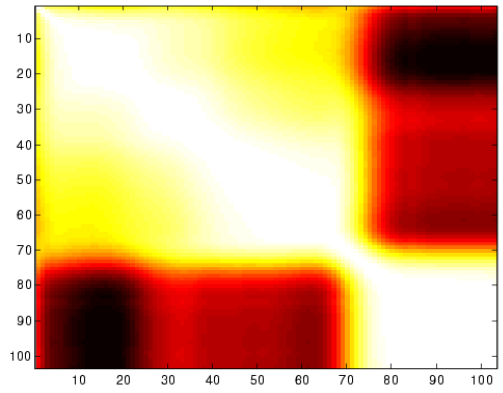
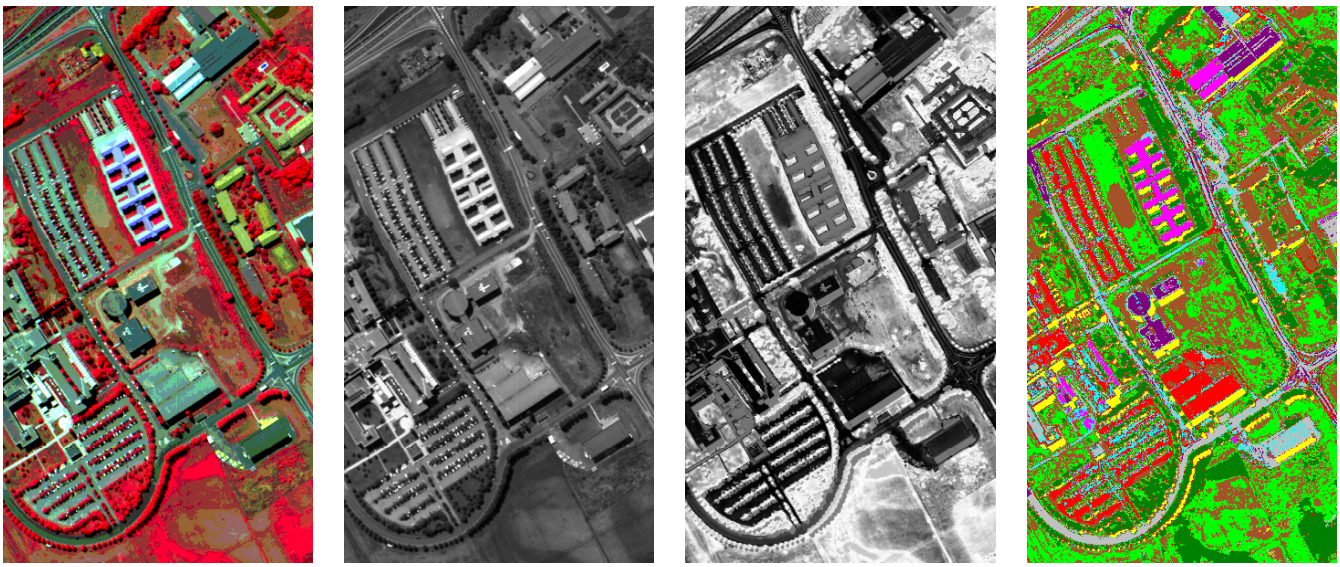


Fig. 1. Correlation image for the 103 bands in the ROSIS imagery for image Fig. 2. The correlation image was computed using (1) with N equal to number of pixels in the image.

The results are listed in Table 2. The classification of the full spectrum enlighten the Hughes phenomenon: the number of training samples (980) was to small with ratio to the dimensionality of the data (103), this leads to a bad classification in terms of accuracy: 27.3%. The classification with 1 principal component gave slightly better results for PCA and KPCA. For PCA, with 3 principals components (corresponding to 95% of the total variance) the overall classification accuracy on the testing set is 73.1%, adding more band does not improve significantly the classification accuracy and adding too many bands deteriorated the classification, as expected with the Hughes phenomenon. For the KPCA, the classification accuracy reached to 74.5% with 7 features corresponding to 81.02% of the variance. For 6 to 10 bands, the results remained nearly equal ($\approx 74.5\%$) and then decreased. For DBFE, with 8 features, corresponding to 62.2% of the total variance, the classification is slightly worse than with two principal components, while with 26 feature, corresponding to 95% of the variance, the classification is even worse. Nevertheless, with more training samples, the DBFE should give better results.

Using KPCAs' first principal components improved a few the classification accuracy, but contrary to PCA, 95% does not seem to be the best value of variance which led to the best classification accuracy. In these experiments, 80% of the variance gave best results. However, this value is kernel function dependant, i.e. if polynomial kernels were used, the percentage of the variance that lead to the best classification accuracy should change.



(a) (b) (c) (d)

Fig. 2. Rosis University area: (a) is the original image, (b) is the first pc, (c) is the first kpc and (d) is the classified image with the 7 first kpc with the neural network. Classes description: asphalt, meadow, gravel, tree, metal sheet, bare soil, bitumen, brick, shadow.

Table 2. Overall classification accuracy for Image 2.(a)

	Testing Set (%)	Training Set (%)
1 pca	37.7	39.4
2 pca	69.9	71.2
3 pca	73.1	74.1
1 kpca	37.3	39.4
2 kpca	66.0	66.6
3 kpca	72.4	75.3
4 kpca	73.1	76.8
5 kpca	74.4	77.9
6 kpca	74.9	79.1
7 kpca	74.5	78.1
28 kpca	55.1	55.9
8 dbfe	64.1	68.2
26 dbfe	43.4	43.6
ALL	27.3	30.4

5. CONCLUSION

A unsupervised nonlinear feature extraction method was investigated. Based on kernel methods, linear PCA was turn to nonlinear KPCA. This method was used to extract features that are uncorrelated in some feature space. In the experiment, KPCA used as feature extraction on hyperspectral data, performed well in terms of accuracy. However, more developments are needed to define the amount of variance which is optimal for classification.

In this article, we use the Gaussian kernel which has the property that the projection is done on infinite dimensional space; but another kernel functions could be use. For example, kernels defined for hyperspectral data [8] could be specially used for remotely sensed images.

6. REFERENCES

[1] K.R Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–202, Mar. 2001.

[2] C.J.C Burges, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, pp. 121–167, 1998.

[3] B. Schölkopf, A.J. Smola, and K.-R Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[4] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, 1968.

[5] D.A. Landgrebe, *Signal theory methods in multispectral remote sensing*, NJ: Wiley, 2003.

[6] A. Hyvärinen, J. Karhunen, and E. Oja, *Independant component analysis*, John Wiley and Sons, New York, 2001.

[7] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, 2004.

[8] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Evaluation of kernels for multiclass classification of hyperspectral remote sensing data," in *Proc. ICASSP*, May 2006.