DESIGN OF KPCA BASED MINIMUM SQUARED ERROR CLASSIFIER

Peng Peng

Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762 USA
email: peng@isip.mstate.edu

ABSTRACT

Kernel principal component analysis (KPCA) has recently been proposed as a nonlinear extension of PCA. [1] The basic idea is to first map the input space into a feature space via a nonlinear map and then compute the principal components in that feature space. Based on these principal components, this paper proposes a kernel version of minimum squared-error linear discriminant classifier (KL) [2], in which the MSE cost is minimized on extracted features in the feature space. Programs based on IFC were written to implement this KPCA based classifier. This paper will focus on analyzing the theory behind KPCA and KL classifier and improving the classifier's performance by adjusting the necessary parameters of KPCA.

1. INTRODUCTION

Principal Component Analysis is a technique used to linearly transform an original set of variables into a set of uncorrelated variables of smaller dimension that represents most of the information. Through kernel functions, it can also transform variables in a nonlinear fashion. Kernel Principal Component Analysis is a nonlinear extension of PCA where the principal components are computed in a high dimensional feature space, which is nonlinear related to the input space. [3]

The aim of this study is to illustrate the potential of KPCA for data classification. Accordingly, a kernel version of MSE linear classifier is proposed that uses kernel PCA for data feature extraction. By adopting a Gaussian kernel, the principal components are computed efficiently within the feature space of input data. Unsupervised classification is then performed using a MSE linear discriminant function. We thus investigate the mathematical properties of kernel functions and characterize the dependence of the performance of classifiers on the changes of the kernel function parameters.

The reminder of this paper is organized as follows. In the next section, we will describe the Kernel PCA algorithm. In section 3, we present a KPCA based version of minimum squared error linear discriminant classifier. Data classification experiments on two data sets are given in section 4, followed by a discussion of our methods (section 5).

2. KERNEL PCA THEORY OVERVIEW

Principal Component Analysis (PCA) is a powerful technique for extracting structure from possibly high-dimensional data sets. It is readily performed by solving an eigenvalues problem, or by using iterative algorithms which estimate principal components. PCA is an orthogonal transformation of the coordinate system. The new coordinate values by which we represent the data are called principal components. It is often the case that a small number of principal components are sufficient to account for most of the structure in the data.

In current work of pattern recognition, we are not interested in principal components in input space, but rather in principal components of variables, or features, which are nonlinearly related to the input variables. Kernel principal component analysis (KPCA) is proposed as a nonlinear extension of PCA. It computes the principal components in high-dimensional feature space F, which is nonlinearly related to the input space. To reduce the computation complexity, it is performed using kernel functions (the dot product of two data in F) without explicitly working in feature space F.

Given a set of centered observations x_k, $k = 1, \ldots, M$, $x_k \in R^N$, $\sum x_k = 0$, PCA diagonalizes the covariance matrix

$$C = \frac{1}{M} \sum_{j=1}^{M} x_j x_j^T \tag{1}$$

To do this, one has to solve the eigenvalue equation

$$\lambda v = Cv \tag{2}$$

for eigenvalues $\lambda \geq 0$ and $v \in R^N \backslash \{0\}$. As

$$Cv = \frac{1}{M} \sum_{j=1}^{M} (x_j \cdot v) x_j^T \tag{3}$$

all solutions v with $\lambda \neq 0$ must lie in the span of $x_1, \ldots, x_M$, hence equation (2) is equivalent to

$$\lambda (x_k \cdot v) = (x_k \cdot Cv) \tag{4}$$

for all $k = 1, \ldots, M$.

In kernel PCA, we have the same computation in another dot product space F, which is related to the input space by a nonlinear map

$$\Phi: \mathrm{R}^N \rightarrow F, \; x \rightarrow X \tag{5}$$

Note that F, which is referred to be the feature space, could have an arbitrarily large, possibly infinite, dimensionality.

Again, we assume that we are dealing with centered data, i.e. $\sum \Phi(x_k) = 0$. Using the covariance matrix in F,

$$\bar{C} = \frac{1}{M} \sum_{j=1}^{M} \Phi(x_j) \Phi(x_j)^T \tag{6}$$

We now have to find eigenvalues $\lambda \geq 0$ and eigenvectors $V \in F \backslash \{0\}$ satisfying

$$\lambda V = \bar{C} V \tag{7}$$

Again, all solutions V with $\lambda \neq 0$ lie in the space of $\Phi(x_1), \ldots, \Phi(x_M)$. This has two useful consequences: first, we may instead consider the set of equations

$$\lambda (\Phi(x_k) \cdot V) = (\Phi(x_k) \cdot \bar{C} V) \tag{8}$$

for all $k = 1, \ldots, M$, and second, there exist coefficients $\alpha_i (i) (i = 1, \ldots, M)$ such that

$$V = \sum_{i=1}^{M} \alpha_i \Phi(x_i) \tag{9}$$

Combining (8) and (9), we get

$$\lambda \sum_{i=1}^{M} \alpha_i (\Phi(x_k) \cdot \Phi(x_i)) \tag{10}$$
$$= \frac{1}{M} \sum_{i=1}^{M} \alpha \left(\Phi(x_k) \cdot \sum_{j=1}^{M} \Phi(x_j) \right) (\Phi(x_j) \cdot \Phi(x_i))$$

Defining an $M \times M$ matrix K by

$$K_{ij} = (\Phi(x_i) \cdot \Phi(x_j)) \tag{11}$$

this reads

$$M \lambda K \alpha = K^2 \alpha \tag{12}$$

where α denotes the column vector with entries $\alpha_1, \ldots, \alpha_M$. To find solutions of (12), we solve the eigenvalue problem

$$M \lambda \alpha = K \alpha \tag{13}$$

for nonzero eigenvalues.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$ denote the eigenvalues of K (i.e. the solutions $M\lambda$ of (12)), and $\alpha^1, \ldots, \alpha^M$ the corresponding complete set of eigenvectors, with λ_p being the last nonzero eigenvalue (assuming $\Phi \neq 0$). We normalize $\alpha^1, \ldots, \alpha^p$ by requiring that the corresponding vectors in F be normalized, i.e.

$$(V^k \cdot V^k) = 1 \tag{14}$$

for all $k = 1, \ldots, p$. Combining (9) and (13), this

translates into a normalization condition for $\alpha^1,\ldots,\alpha^p$:

$$1 = \sum_{i,j=1}^{M} \alpha_i^k \alpha_j^k (\Phi(x_i) \cdot \Phi(x_j)) \quad (15)$$
$$= \sum_{i,j=1}^{M} \alpha_i^k \alpha_j^k K_{ij} = (\alpha^k \cdot K\alpha^k) = \lambda_k(\alpha^k \cdot \alpha^k)$$

For the purpose of principal component extraction, we need to compute projections onto the eigenvectors V^k in F ($k = 1, \ldots, p$). Let x be a test point, with an image $\Phi(x)$ in F, then

$$(V^k \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i^k (\Phi(x_i) \cdot \Phi(x)) \quad (16)$$

may be called its kernel principal components corresponding to Φ.

In summary, we use the following steps to compute the principal components: first, compute the matrix K; second, compute its eigenvectors and normalize them in F; third, compute projections of input vectors onto the eigenvectors as well as the test vectors. Therefore we can get extracted feature matrix K'

$$K'_{ij} = (V^j \cdot \Phi(x_i)) \quad (17)$$

where $i = 1, \ldots, M$, $j = 1, \ldots, p$, and extracted feature matrix K_{test}' for test vectors

$$K'_{ij \cdot test} = (V^j \cdot \Phi(x_{i \cdot test})) \quad (18)$$

where $i = 1, \ldots, l$, $j = 1, \ldots, p$, l is the number of test vectors.

In this paper, we focus on Gaussian kernels of the form

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{c}\right) \quad (19)$$

3. KERNEL MINIMUM SQUARED ERROR

A function $f:\mathrm{R}^l \to R$ is a discriminant (decision) rule for a class of objects $C \subset R^l$ if $f(x) > 0$ for $x \in C$ and $f(x) \le 0$ otherwise. Linear discriminant functions, defined by $f(x) = w \cdot x$ are frequently used because of their mathematical simplicity. A popular method to find a good weight vector w from input data is based on the solution of the following system of linear equations:

$$Xw = b \quad (20)$$

where X denotes the sample matrix and b is the vector of associated class labels: $b_i = +1$ if $x_i \in C$ and $b_i = -1$ otherwise.

The vector w^* that minimizes the MSE cost

$$J(w) = \|Xw - b\|^2 \quad (21)$$

can be computed as

$$w^* = X^\dagger b \quad (22)$$

where $X^\dagger$ is called the pseudoinverse of matrix X

$$X^\dagger = (X^T X)^{-1} X^T \quad (23)$$

Thus a kernel version of the MSE linear discriminant on extracted test feature can be obtained

$$f(t_i) = K'^\dagger b \cdot t_i \quad (24)$$

where t_i is the i row vector of the extracted feature matrix K_{test}', $i = 1, \ldots, l$.

A natural generalization of the MSE procedures to multi-class case is to consider the problem as a set of c two-class problems. The ith problem is to obtain a weight vector w_i that is minimum-squared-error solution to the equations

$$Xw = 1 \text{ for all } x \in C_i \quad (25)$$
$$Xw = -1 \text{ for all } x \notin C_i \quad (26)$$

where C_i is the ith class.

4. EXPERIMENTS

To test the feasibility of the proposed algorithm, we run two classification experiments on two different data sets. [4] Set 1 is a static classification problem with 11 classes. Each class vector has 10 elements. Totally there are 528 training vectors, 379 test vectors and 83 evaluation vectors. Set 2 is a temporal modeling problem with 5 classes. Each class vector has 39 elements but 5 class is a continuous sequence (same classes sequentially in time). Totally there are 925 training vectors (sets of 5 vectors for each class), 350 test vectors and 225 evaluation vectors.

SET 1		**SET 2**	
c	**ERR(%)**	c	**ERR(%)**
0.005	70.18	0.05	54.29
0.010	41.42	0.10	25.71
0.015	***34.30***	0.15	20.00
0.020	36.68	0.20	20.00
0.025	37.73	0.25	***18.57***
0.030	37.73	0.30	***18.57***
0.035	39.58	0.35	20.00
0.040	39.58	0.40	21.43
0.045	39.58	0.45	21.43
0.050	41.69	0.50	21.43

Table 1: Test error rates on the two data sets for MSE linear classifier trained on kernel principal components extracted by PCA with kernel (19), corresponding various values of c.

These data sets were performed using Gaussian kernels of the form (19) and MSE linear classifier in equation (24). The factor c was adjusted to obtain different test error rates from these data sets. The test error rates corresponding different c values are shown in Table 1. From the table, we can see $c = 0.015$ yields the best test error rate at 34.30% for test data set 1; $c = 0.25$ or $c = 0.30$ yields the best test error rate at 18.57 for test data set 2.

5. CONCLUSION

Kernel PCA is a nonlinear generalization of PCA in the sense that (1) it is performing PCA in feature spaces of arbitrarily large (possibly infinite) dimensionality, and (2) if we use the kernel $k(x, y) = (x \cdot y)$, we recover standard PCA. The main advantage of using KPCA is that no nonlinear optimization is involved. It is essentially linear algebra, as simple as standard PCA. It requires only the solution of an eigenvalue problem. Another advantage is that the performance for kernel principal components analysis can be further improved by using more components.

Based on KPCA, a particular simple kernel version of MSE linear classifier (KL) is proposed, which includes a number of generalized linear models. Satisfactory results in experiments are obtained. For large databases, KL based on KPCA can be both accurate and efficient.

Further work can be done on trying to implement a linear soft margin classifier based on KPCA, which is a special case of Support Vector Machines.

REFERENCES

[1] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K. R. Miller, G. Ratsch and A. Smola, “Input Space versus Feature Space in Kernel-based Methods,” *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000-1017, September 1999.

[2] A. Ruiz and P.E. Lopez-de-Teruel, “Nonlinear Kernel-Based Statistical Pattern Analysis,” *IEEE Transactions on Neural Networks*, vol. 12, no. 1, January 2001.

[3] K.I. Kim, K. Jung, S.H. Park, H.J. Kim, “Kernel Principal Component Analysis for Texture Classification,” *IEEE Signal Processing Letters*, vol. 8, no. 2, February 2001.

[4] J. Picone, “Common Evaluation,” *http://www.isip.msstate.edu/publications/courses/ece_8990_pr/exams/2001/*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA, May 2001.