An Expectation Maximization Approach to Nonlinear Component Analysis

Roman Rosipal and Mark Girolami
Computational Intelligence Research Unit
Department of Computing and Information Systems
University of Paisley
Paisley, PA1 2BE
{rosi-ci0, giro-ci0}@wpmail.paisley.ac.uk

Abstract

The proposal of considering nonlinear principal component analysis as a kernel eigenvalue problem has provided an extremely powerful method of extracting nonlinear features for a number of classification and regression applications. Whereas the utilization of Mercer kernels makes the problem of computing principal components in, possibly, infinite dimensional feature spaces tractable, there are still the attendant numerical problems of diagonalizing large matrices. In this contribution we propose an expectation maximization approach for performing kernel principal component analysis and show this to be a computationally efficient method especially when the number of data points is large.

1 Introduction

The notion of performing linear Principal Component Analysis (PCA) in a high (and possibly infinite) dimensional nonlinear feature space was first proposed in (Schölkopf et al., 1998). The key feature of the proposed method is the creation of the kernel matrix (Schölkopf et al., 1998) and its subsequent diagonalization. Each element of the kernel matrix is composed of the dot product of the nonlinearly mapped data points, and the elegant use of Mercer kernels allows these dot products in high dimensional feature space to be computed using simple kernel functions in data space (Schölkopf et al., 1999; Schölkopf et al., 1998). The dimension of the kernel matrix is equal to the number of data points and so its diagonalization allows *nonlinear principal components* to be extracted from the data, the number of which may be up to the number of observed data points. This implies that the number of principal components extracted can exceed the dimensionality of the data (Schölkopf et al., 1998).

In practice however if there is a substantial number of observations then the diagonalization of the associated kernel matrix can be somewhat problematic in terms of computational demands and numerical accuracy (Jolliffe, 1986; Rosipal et al., 2000). The problem of performing an eigenvalue decomposition on large covariance matrices can be alleviated by using the Expectation Maximization (EM) approach for PCA which emerges from considering PCA from a probabilistic perspective (Tipping & Bishop, 1999; Roweis & Ghahramani, 1999). Taking the EM approach to PCA in data space we now show that this can be modified to allow the efficient computation of the Kernel PCA (Schölkopf et al.,1998).

2 An EM Approach to Kernel PCA

The data space model for probabilistic PCA is given as $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{v}$ where $\mathbf{C} \in R^{p\times k}$ and the observation vector and latent variable vectors are given as $\mathbf{y} \in R^p$ and $\mathbf{x} \in R^k$, respectively. The latent variables are normally distributed with zero mean and identity covariance. The zero mean noise $\mathbf{v}$ is also normally distributed with a covariance matrix defined as $\mathbf{\Psi}$. It is shown in (Roweis & Ghahramani, 1999; Tipping & Bishop, 1999) that as the noise level in the model becomes infinitesimal the PCA model is recovered. The posterior density then becomes a delta function $P(\mathbf{x}|\mathbf{y}) = \delta(\mathbf{x} - (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y})$ and the EM algorithm is effectively a straightforward least squares projection (Roweis & Ghahramani, 1999) which is given below. We denote the matrix of data observations as $\mathbf{Y} \in R^{p\times n}$ and the matrix of latent variables as $\mathbf{X} \in R^{k\times n}$. Then

E-Step $\mathbf{X} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{Y}$

M-Step $\mathbf{C}^{new} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$

We now wish to transform this EM procedure to feature space $\mathcal{F}$. The implicit nonlinear map from input space to $\mathcal{F}$ is denoted as $\Phi(.)$ (Schölkopf et al., 1998). For clarity of exposition we now denote the matrix $\mathbf{Y}$ to be the matrix which has individual columns consisting of the following vectors $\Phi(\mathbf{y}_1), \ldots, \Phi(\mathbf{y}_n)$ of the mapped observed data.

Centering in the feature space can be carried out in a straightforward manner by "centering" the kernel matrix $\mathbf{K}$ (which is defined below) using the simple procedure which is outlined in (Schölkopf et al., 1998). Realizing that the columns of $\mathbf{C}$ are scaled and orthogonally rotated eigenvectors computed by diagonalization of the sample covariance matrix we can express the r^{th} column of $\mathbf{C}$ as $\mathbf{C}^r = \sum_{j=1}^{n} \gamma_j^r \Phi(\mathbf{y}_j)$ (Tipping & Bishop, 1999; Schölkopf et al., 1998). Using this fact we can write the (r,s) element of the matrix $\mathbf{C}^T\mathbf{C}$ as

$$\begin{aligned}(\mathbf{C}^T\mathbf{C})_{r,s} &= (\mathbf{C}^r)^T\mathbf{C}^s = \textstyle\sum_{i=1}^{n} \gamma_i^r \Phi^T(\mathbf{y}_i) \sum_{j=1}^{n} \gamma_j^s \Phi(\mathbf{y}_j) = \\ &= \textstyle\sum_{i,j=1}^{n} \gamma_i^r \gamma_j^s (\Phi(\mathbf{y}_i).\Phi(\mathbf{y}_j)) = \sum_{i,j=1}^{n} \gamma_i^r \gamma_j^s K(\mathbf{y}_i,\mathbf{y}_j).\end{aligned}$$

Which gives in matrix form $\mathbf{C}^T\mathbf{C} = \mathbf{\Gamma}^T\mathbf{K}\mathbf{\Gamma}$, where the columns of the matrix $\mathbf{\Gamma} \in R^{n\times k}$ consists of the $\{\boldsymbol{\gamma}^i\}_{i=1}^{k}$ vectors. Likewise the second term of the E-step expression can be written as follows,

$$(\mathbf{C}^T\mathbf{Y})_{r,s} = (\sum_{i=1}^{n} \gamma_i^r \Phi^T(\mathbf{y}_i))\Phi(\mathbf{y}_s) = \sum_{i=1}^{n} \gamma_i^r (\Phi(\mathbf{y}_i).\Phi(\mathbf{y}_s)) = \sum_{i=1}^{n} \gamma_i^r K(\mathbf{y}_i,\mathbf{y}_s),$$

then $(\mathbf{C}^T\mathbf{Y}) = \mathbf{\Gamma}^T\mathbf{K}$. So, finally we have the required E-step

$$\boxed{\mathbf{X} = (\mathbf{\Gamma}^T\mathbf{K}\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{K}}$$

Now let us consider the M-Step. Denote the following term as follows $\mathbf{A} \equiv \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. As a consequence of the fact that the columns of $\mathbf{C}$ lie in the span of $\Phi(\mathbf{y}_1),\ldots,\Phi(\mathbf{y}_n)$ we can consider the set of equations

$$\Phi^T(\mathbf{y}_j)\mathbf{C}^{new} = \Phi^T(\mathbf{y}_j)\mathbf{Y}\mathbf{A} \;\text{ for all }\; j = 1,\ldots,n$$

$$\left[\sum_{i=1}^{n} \gamma_i^1 K(\mathbf{y}_j,\mathbf{y}_i),\ldots,\sum_{i=1}^{n} \gamma_i^k K(\mathbf{y}_j,\mathbf{y}_i)\right] = [K(\mathbf{y}_j,\mathbf{y}_1),\ldots,K(\mathbf{y}_j,\mathbf{y}_n)]\,\mathbf{A}$$

for all $j = 1,\ldots,n$. We can write it in matrix form $\mathbf{K}\mathbf{\Gamma}^{new} = \mathbf{K}\mathbf{A}$[1]. The M–step then follows as below.

$$\boxed{\mathbf{\Gamma}^{new} = \mathbf{A} = \mathbf{X}^{\mathbf{T}}(\mathbf{X}\mathbf{X}^{\mathbf{T}})^{-\mathbf{1}}}$$

It has been shown in (Tipping & Bishop, 1999), that in the case of infinitesimal noise in our model, i.e. $\mathbf{\Psi} = \lim_{\sigma^2\to 0}\sigma^2\mathbf{I}$, the matrix $\mathbf{C}$ at convergence

[1] In general, $\mathbf{K}$ is a positive semidefinite matrix, however we can guarantee positive definiteness by adding arbitrary small positive values on the diagonal.

will be equal to $\mathbf{C} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{R}$, where the columns of the $\mathbf{U}$ matrix are the eigenvectors of the sample covariance matrix with corresponding eigenvalues $\lambda_1, ..., \lambda_k$ creating the diagonal matrix $\mathbf{\Lambda}$, and $\mathbf{R}$ is an arbitrary orthogonal rotation matrix. In (Tipping & Bishop, 1999) the authors also pointed out that taking the columns of $\mathbf{R}^T$ to be equal to the eigenvectors of the $\mathbf{C}^T\mathbf{C}$ matrix we can recover the true principal axes. Thus, in our case the projection of the test point $\mathbf{y}$ onto the k nonlinear principal components is now given by

$$\boldsymbol{\beta}(\mathbf{y}) := \mathbf{R}^T(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\Phi(\mathbf{y}) = \mathbf{R}^T(\mathbf{\Gamma}^T\mathbf{K}\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{K}_y \ ,$$

where $\mathbf{K}_y$ is the "centered" vector $[K(\mathbf{y}_1, \mathbf{y}), ..., K(\mathbf{y}_n, \mathbf{y})]^T$ (Schölkopf et al., 1998).

We should note a number of points regarding this method for performing Kernel PCA. Firstly, due to the use of the Mercer kernels the method is independent of the dimensionality of the input space. Secondly, the computational complexity, per iteration, of the proposed EM method for Kernel PCA is $\mathcal{O}(kn^2)$ where n is the number of data points and k is the number of extracted components. Where a small number of eigenvectors require to be extracted and there are a large number of data points available this method is comparable in complexity to the iterative power method which has complexity $\mathcal{O}(n^2)$. Direct diagonalization of a symmetric $\mathbf{K}$ matrix to solve the eigenvalue problem for Kernel PCA (Schölkopf et al., 1998) has complexity of the order $\mathcal{O}(n^3)$.

3 Experiments

We tested the proposed EM algorithm for Kernel PCA on two artificial two dimensional data sets. Thus, by projecting the test input data to the extracted principal components we can visually compare the results even although we may have the situation where the non-linear mapping to feature space is not explicitly known. In the first example we generated two parabolic shapes vertically and horizontally mirrored. The data were generated by the function $y = x^2 + 0.6$ where the x values have a uniform distribution in $[-1, 1]$. We used 500 data points uniformly divided for each parabolic shape. The polynomial kernels of degree 2 and 3 were used.

In the second example three two-dimensional Gaussian clusters with means [-0.5, -0.2; 0.0, 0.6; 0.5, 0.0] and common variance 0.1 were generated. Each cluster consisted of five hundred data points. The Gaussian kernel $K(\mathbf{x},\mathbf{y}) = exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{0.1})$ was used. In each case the initial values of the $\mathbf{\Gamma}$ matrix components were randomly generated with uniform distribution in $[-1,1]$.

From Fig. 1a) and 1b) we can see the equivalence of the first two eigenvectors found by our EM approach (bottom)and the MATLAB `eig` procedure (top) in the case of using the polynomial kernel of degree 2. We ran the EM algorithm until the dot product between the eigenvectors of the 2-dimensional subspace found by EM and Kernel PCA was less than 0.999 (appr. 2.5^o). In this setting we found that on average less than two EM steps were sufficient to extract the two required eigenvectors. We also investigated the convergence of the proposed algorithm by running 200 simulations with different initializations of the $\mathbf{\Gamma}$ matrix. The average value of the dot product between the eigenvectors of the 2-dimensional subspace, which were found, was 0.995 (appr. 5.7^o). In six cases we observed that at convergence the dot product was less than 0.98 (appr. 11.5^o), but greater than 0.96 (appr. 16.2^o). Slightly better convergence results were achieved using the third-order polynomial kernel. In that case the average value of the dot product was 0.998 (appr. 3.6^o) and only in four cases the value of the dot product were in the $0.96 - 0.98$ range. The use of the MATLAB `eig` function requires 646 times the number of flops required by the proposed EM approach. The MATLAB `eigs` function[2] which is based on the iterative power method still requires 5.8 times as many flops as two iterations of our EM method.

In Fig. 1c) we demonstrate results found in the second example using three EM steps. We can see that first two principal components nicely separate the three clusters in coincidence with results reported on Kernel PCA (Schölkopf et al., 1998).

4 Conclusion

We have proposed an EM based approach for performing a principal components decomposition in the kernel space $\mathcal{F}$. This provides another method for performing kernel based principal component analysis which, in many cases, is extremely efficient in terms of the computing resources required. Further work in this area includes the study of the convergence behaviour of the EM approach to nonlinear kernel based PCA.

[2]The default MATLAB setting with convergence tolerance equal to 1×10^{-10} was used here.

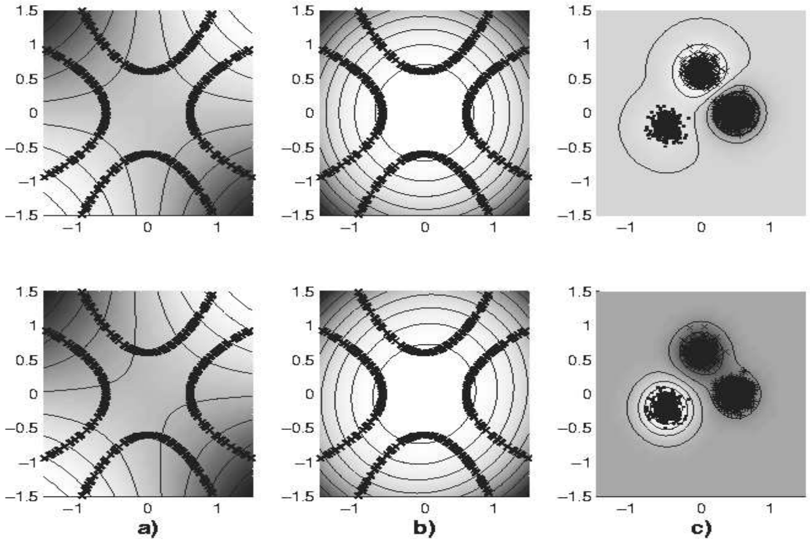

Figure 1: a)–b) First two principal components extracted by diagonalization of the $\mathbf{K}$ matrix (top) and by the proposed EM algorithm (bottom) on the first example using the second-order polynomial kernel. c) First two principal components extracted by the proposed EM algorithm on the second example. The greyscales in the figures represent the principal component values and the contours represent lines of constant principal component value.

Acknowledgments

The first author is funded by a research grant for the project "Objective Measures of Depth of Anaesthesia"; University of Paisley and Glasgow Western Infirmary NHS trust, and is partially supported by Slovak Grant Agency for Science (grants No. 2/5088/00 and No. 00/5305/468).

References

Jolliffe, I.T. (1986). *Principal Component Analysis.* New York: Springer-Verlag.

Rosipal, R., Girolami, M., & Trejo, L. (2000). *Kernel PCA for Feature Extraction and De-Noising in Non-linear Regression* (Technical Report No. 4). Paisley, Scotland: CIS Department, University of Paisley.

Roweis, S., & Ghahramani, Z. (1999). A unifying Review of Linear Gaussian Models. *Neural Computation, 11*, 305–345.

Schölkopf, B., Bruges, C., & Smola, A. (1999). *Advances in Kernel Methods - Support Vector Learning.* Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K.R. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation, 10*, 1299–1219.

Tipping, M. E. & Bishop, C.M. (1999). Probabilistic principal component analysis. *J. R. Statist. Soc. B, 61*, 611–622.