Proceedings of the 9th WSEAS International Conference on Applied Mathematics, Istanbul, Turkey, May 27-29, 2006 (pp611-616)

Sparsification of Probabilistic Canonical Correlation Analysis

DANIEL LIVINGSTONE and COLIN FYFE,
Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland.

Abstract We have recently developed several ways of performing Canonical Correlation Analysis [1, 5, 7, 4] with probabilistic methods rather than the standard statistical tools. However, the computational demands of training such methods scales with the square of the number of samples, making these methods uncompetitive with e.g. artificial neural network methods [3, 2]. In this paper, we examine a recent development which sparsifies a probabilistic method of performing principal component analysis and then use this method to sparsify a new probabilistic method of performing canonical correlation analysis.

Key-words: Probabilistic canonical correlation analysis, sparsification

1 Introduction

We have recently investigated several ways of creating probabilistic methods for performing Canonical Correlation Analysis (CCA) [1, 5, 7, 4]. In particular, we have shown that, while such methods can be used to reliably find the canonical correlations between data sets, the computational demands of the methods are such that they can only be used with relatively small data sets (< 1000 samples). In this paper, we investigate a method for sparsification of data sets by identifying samples which are most relevant in some way for application to the task in hand.

Canonical Correlation Analysis is used when we have two data sets which we believe have some underlying correlation. Consider two sets of input data, $\mathbf{x}_1 \in X_1$ and $\mathbf{x}_2 \in X_2$. Then in classical CCA, we attempt to find the linear combination of the variables which gives us maximum correlation between the combinations. Let $y_1 = \mathbf{w}_1^T \mathbf{x}_1$ and $y_2 = \mathbf{w}_2^T \mathbf{x}_2$. Then, for the first canonical correlaton, we find those values of $\mathbf{w}_1$ and $\mathbf{w}_2$ which maximises $E(y_1 y_2)$ under the constraint that $E(y_1^2) = E(y_2^2) = 1$.

[1]We assume centered data.

2 Probabilistic Principal Component Analysis

Perhaps the best known method of performing Probabilistic Principal Component Analysis is that of [11]. PCA is one of the oldest and most established of statistical techniques. Recently, [11] has shown how to give it a probabilistic underpinning. Consider a data set, $\mathbf{Y} \in R^{N \times D}$ of N samples of D dimensional data. Then latent variable modeling consists of relating a set of latent variables, $\mathbf{X} \in R^{N \times q}$ to the data through a particular model so that the data can be best explained by the latent variables. Generally the model is a probabilistic one and the parameters are found by adapting the model to make the data as likely as possible under the model. In the context of PCA[1], we have

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \eta_n, \forall n = 1, ..., N \tag{1}$$

where $\mathbf{W} \in R^{D \times q}$ is the matrix of parameters to be adjusted and η_n are drawn independently from zero mean spherical Gaussian noise in the data space:

$$p(\eta_n) = N(0, \beta^{-1} I) \tag{2}$$

with inverse noise variance β. Then

$$p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta) = N(\mathbf{W}\mathbf{x}_n, \beta^{-1}I) \quad (3)$$

[11] defines a prior distribution over the latent variables by

$$p(\mathbf{x}_n) = N(0, I) \quad (4)$$

which means that they may be integrated out to give

$$p(\mathbf{y}_n|\mathbf{W}, \beta) = N(0, \mathbf{W}\mathbf{W}^T + \beta^{-1}I) \quad (5)$$

We adjust the parameters, $\mathbf{W}$ to maximise the likelihood of the data, $\mathbf{Y}$ under this model.

Lawrence [6] takes an alternative approach: instead of integrating out the latent variables and optimising the weights, he integrates out the weights and optimises the positions of the latent variables in the q dimensional latent space. Thus he puts a prior on the weights so that

$$P(\mathbf{W}) = \prod_{i=1}^{D} P(\mathbf{w}_i) = \prod_{i=1}^{D} N(0, I) \quad (6)$$

Lawrence [6] derives the log likelihood as

$$L = -DN\ln(2\pi) - D\ln|\mathbf{K}| - trace(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \quad (7)$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^T + \beta^{-1}I$. Gradient ascent on this provides the method to maximise the likelihood of the data under this model:

$$\frac{\partial L}{\partial \mathbf{X}} = \mathbf{K}^{-1}\mathbf{S}\mathbf{K}^{-1}\mathbf{X} - D\mathbf{K}^{-1}\mathbf{X} \quad (8)$$

where $\mathbf{S} = \mathbf{Y}\mathbf{Y}^T$. A fixed point of this is given by the first q eigenvectors of $\mathbf{S}$, from which the principal component filters can be found (exactly as with Kernel PCA [9]) by treating these vectors as being defined in the basis described by the data points.

We have previously used a similar method to Lawrence for CCA [4].

2.1 Sparse Probabilistic Principal Component Analysis

Tipping [10] proposes to sparsify Kernel PCA by specifying the covariance matrix of the data as

$$C = \sigma_n^2 I + \sum_{i=1}^{N} a_i \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T = \sigma_n^2 I + \Phi^T A \Phi \quad (9)$$

where the weights a_i are adjustable parameters which are positioned on the main diagonal of diagonal matrix , A, in the last equation and he has performed a nonlinear mapping of the data using the function, $\phi(.)$. Kernel PCA [9] utilises the "kernel trick": provided you can find the scalar product $K_{ij} = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$, you need never actually require the individual functions $\phi(.)$. By maximising the likelihood of the data under this model, he shows that we are performing a reduced PCA. Tipping derives an algorithm based on Kernel PCA [9] to find these weights by iterating

$$\Sigma = (A^{-1} + K)^{-1} \quad (10)$$

$$\mu_n = \sigma^{-2}\Sigma\mathbf{k}_n \quad (11)$$

$$a_i^{new} = \frac{\sum_{n=1}^{N}\mu_{ni}^2}{N(1 - \Sigma_{ii}/a_i)} \quad (12)$$

where K is the positive definite kernel matrix.

2.2 Experimental Results

We illustrate the effect of the sparsification on a two dimensional artificial data set shown in Figure 1. We also show in that figure the weights greater than 0.02 in the linear case (left) and non-zero weights when using the squared exponential covariance matrix (right). We see that in both cases an informative discrimination is taking place.

We can however examine the non-zero weights in greater depth. In Figure 2, left, we see positive weights in the outer circle, mainly zero in the centre but with two negative values at the edge of the central cluster. This feature

is valuable for identification of potential outliers in a data set. The same discrimination can be found with the squared exponential covariance function.

3 The Sphere-Concatenate Method for CCA

Now it may be shown [8] that a method of finding the canonical correlation directions is to solve the generalised eigenvalue problem

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \tag{13}$$

where Σ_{ij} is the covariance matrix between the i^{th} and j^{th} data streams. Note that this is equivalent to the standard eigenproblem

$$\begin{bmatrix} 0 & \Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}} \\ \Sigma_{22}^{-\frac{1}{2}}\Sigma_{21}\Sigma_{11}^{-\frac{1}{2}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$$

where

$$\begin{aligned} \mathbf{v}_1 &= \Sigma_{11}^{\frac{1}{2}}\mathbf{w}_1 \\ \mathbf{v}_2 &= \Sigma_{22}^{\frac{1}{2}}\mathbf{w}_2 \end{aligned}$$

This standard eigenproblem can now be seen to be equivalent to a decomposition of a cross-covariance matrix of sphered data.

This suggests the following algorithm:

1. Use Probabilistic PCA[2] on both data streams, independently. This gives us eigenvector matrices, $\mathbf{V}_1, \mathbf{V}_2$ and eigenvalues on the main diagonal of Λ_1, Λ_2.

2. Project each data stream onto their respective eigenvectors and divide by the square root of the eigenvalues. This gives us sphered data.

3. Concatenate these two sphered data streams.

4. Perform PPCA on this data, to get eigenvectors $\mathbf{V}_3$ and eigenvalues Λ_3.

5. To recover the CCA directions, $W_1 = \mathbf{V}_1\Lambda_1^{-\frac{1}{2}}\mathbf{V}_{3,1}$, $W_2 = \mathbf{V}_2\Lambda_2^{-\frac{1}{2}}\mathbf{V}_{3,2}$, where we have used the notation $\mathbf{V}_{3,i}$ to denote the appropriate part of $\mathbf{V}_3$.

This algorithm is easily shown to perform well on both artificial and real data.

3.1 Sparse Sphere-Concatenate CCA

The computational intensity of probabilistic methods are very dependent on the number of data samples we have and so one criticism of the method of this section might be that we are now performing PPCA in the Step 4 of the algorithm on a data set which is twice as long as previously.

We may address this problem with the Sparse Kernel Principal Component method [10]. We first calculate appropriate Kernel matrices of the two data streams separately giving K_1 and K_2. We may use Tipping's method in our algorithm replacing Step 1 with the iteration

$$\begin{aligned} \Sigma_1 &= (A^{-1} + K_1)^{-1} && (14) \\ \mu_{1,n} &= \sigma^{-2}\Sigma_1\mathbf{k}_{1,n} && (15) \\ \Sigma_2 &= (A^{-1} + K_2)^{-1} && (16) \\ \mu_{2,n} &= \sigma^{-2}\Sigma_2\mathbf{k}_{2,n} && (17) \\ a_i^{new} &= \frac{\sum_{n=1}^N \mu_{1,ni}^2}{N(1-\Sigma_{1,ii}/a_i)} + \frac{\sum_{n=1}^N \mu_{2,ni}^2}{N(1-\Sigma_{2,ii}/a_i)} \end{aligned}$$

Note that while we are sparsifying two data streams with different covariance matrices, we are utilising a single A matrix. We require this since we wish to identify pairs of important data points simultaneously.

We illustrate the effects of this algorithm in Figure 3: we create 90 samples of two data sets, the first 30 of which are such that the

[2]Note that we retain all eigenvalues,vectors at this stage.

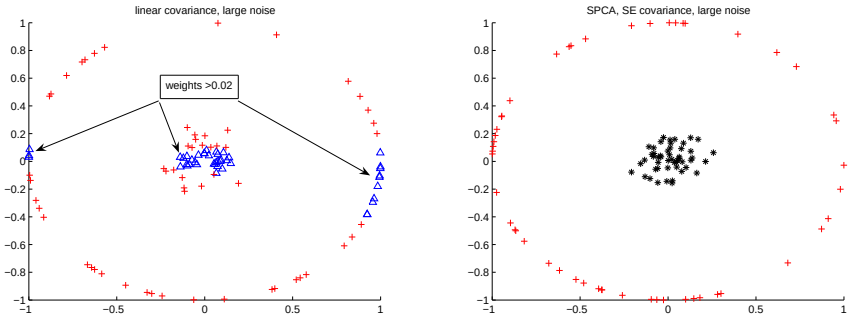

Figure 1: Left: the discrimination with the linear kernel. Right: discrimination with the squared exponential kernel; non-zero weights are shown with black '*'s.

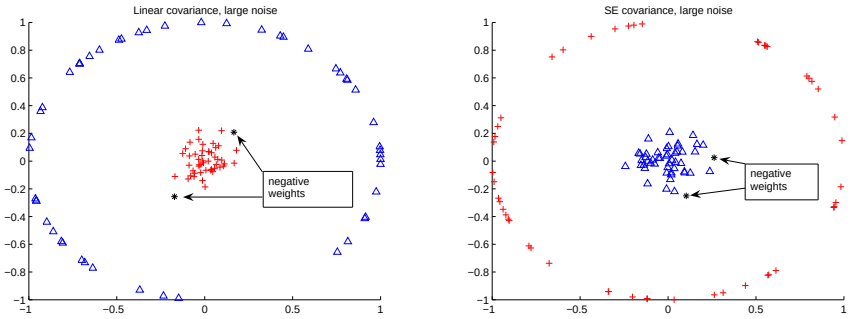

Figure 2: Left: the discrimination with the linear kernel. Right: discrimination with the squared exponential kernel.

corresponding elements come from related clusters (the black '*'s in Figure 3) while the last 60 samples contain no such relationship - $\frac{1}{2}$ of these samples in one data set come from one cluster while the other $\frac{1}{2}$ come from another cluster; in the other data set these samples are drawn from a widely dispersed cluster. We show the weights from both the linear and the squared exponential covariance matrices in that figure.

The degree of sparsification can be controlled by the σ parameter. Even if we set it low (and thereby do not get an appropriate degree of sparsification) we may identify samples which have been wrongly included by plotting the sphered data from the first data set against the sphered data in the second data set. This is illustrated in Figure 4 in which we show a simulation in which three samples from the non-matching clusters have been identified. We see that they are easily identified if we plot the first elements of the sphered data from each data stream.

4 Conclusion

We have illustrated an existing method of sparsifying probabilistic principal component analysis. We have developed a new method of performing canonical correlation analysis with probabilistic principal component analysis and sparsified the method using the sparse principal component analysis. Future work will investigate the method with real data sets.

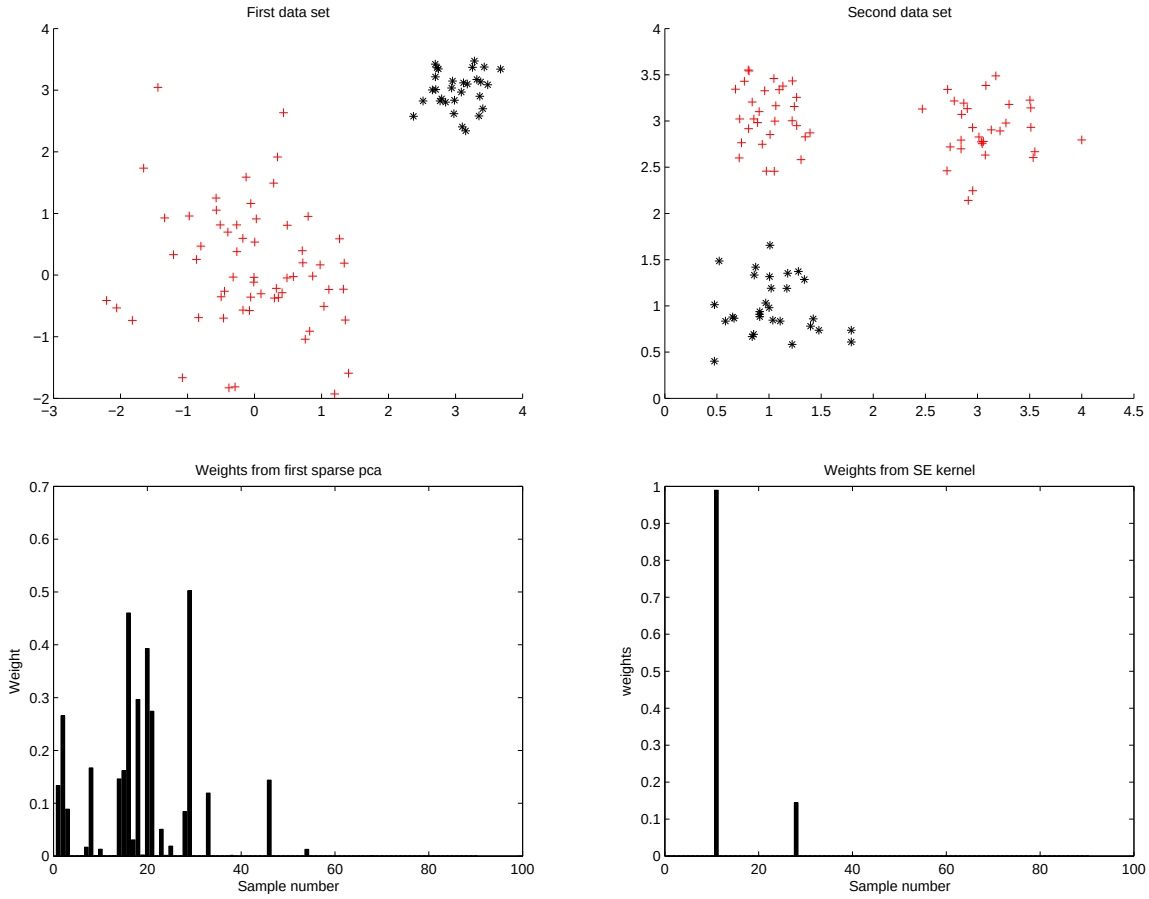

Figure 3: The top row shows the two data sets. The bottom row the weights found with the linear kernel (left) and the squared exponential kernel (right).

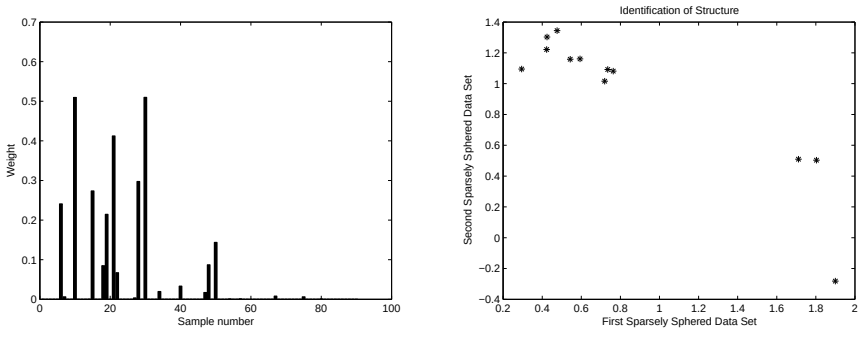

Figure 4: Left: weights found by linear covariance method. Plots of first sphered coordinate in both data sets. The wrongly included elements are clearly identified.

References

[1] C. Fyfe and G. Leen. Stochastic processes for canonical correlation analysis. In *14th European Symposium on Artificial Neural Networks*, 2006.

[2] Z. K. Gou and C. Fyfe. A canonical correlation neural network for multicollinearity and functional data. *Neural Networks*, 2003.

[3] P. L. Lai and C. Fyfe. A neural network implementation of canonical correlation analysis. *Neural Networks*, 12(10):1391–1397, Dec. 1999.

[4] P. L. Lai and C. Fyfe. A latent variable implementation of canonical correlation analysis for data visualisation. In *International Joint Conference on Neural Networks*, 2006.

[5] P. L. Lai, G. Leen, and C. Fyfe. A comparison of stochastic processes and artificial neural networks for canonical correlation analysis. In *International Joint Conference on Neural Networks*, 2006.

[6] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[7] G. Leen and C. Fyfe. A gaussian process latent variable model formulation of canonical correlation analysis. In *14th European Symposium on Artificial Neural Networks*, 2006.

[8] K. V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[9] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[10] M. Tipping. Sparse kernel principal component analysis. In *NIPS*, 2003.

[11] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999.