Using Kernel PCA for Initialisation of Variational Bayesian Nonlinear Blind Source Separation Method

Antti Honkela[1], Stefan Harmeling[2], Leo Lundqvist[1], and Harri Valpola[1]

[1] Helsinki University of Technology, Neural Networks Research Centre
P.O. Box 5400, FI-02015 HUT, Espoo, Finland
{antti.honkela,leo.lundqvist,harri.valpola}@hut.fi
[2] Fraunhofer FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
harmeli@first.fhg.de

Abstract. The variational Bayesian nonlinear blind source separation method introduced by Lappalainen and Honkela in 2000 is initialised with linear principal component analysis (PCA). Because of the multilayer perceptron (MLP) network used to model the nonlinearity, the method is susceptible to local minima and therefore sensitive to the initialisation used. As the method is used for nonlinear separation, the linear initialisation may in some cases lead it astray. In this paper we study the use of kernel PCA (KPCA) in the initialisation. KPCA is a rather straightforward generalisation of linear PCA and it is much faster to compute than the variational Bayesian method. The experiments show that it can produce significantly better initialisations than linear PCA. Additionally, the model comparison methods provided by the variational Bayesian framework can be easily applied to compare different kernels.

1 Introduction

Nonlinear blind source separation (BSS) and related nonlinear independent component analysis (ICA) are difficult problems. Several different methods have been proposed to solve them in a variety of different settings [1, 2]. In this work, we attempt to combine two different methodologies used for solving the general nonlinear BSS problem, the kernel based approach [3, 4] and the variational Bayesian (VB) approach [5, 6]. This is done by using sources recovered by kernel PCA as initialisation for the sources in the variational Bayesian nonlinear BSS method.

Kernel PCA (KPCA) [3] is a nonlinear generalisation of linear principal component analysis (PCA). It works by mapping the original data space nonlinearly to a high dimensional feature space and performing PCA in that space. With the kernel approach this can be done in a computationally efficient manner. One of the drawbacks of KPCA in general is the difficulty of mapping the extracted components back to the data space, but in the case of source initialisation, such mapping is not needed.

The variational Bayesian nonlinear BSS method presented in [5] is based on finding a generative model from a set of sources through a nonlinear mapping

C.G. Puntonet and A. Prieto (Eds.): ICA 2004, LNCS 3195, pp. 790–797, 2004.

to the data. The sources and the model are found by using an iterative EM-like algorithm. Because of the flexible multilayer perceptron (MLP) network used to model the nonlinearity and general ill-posed nature of the problem, the method requires a reasonable initialisation to provide good results. In the original implementation, the initialisation was handled by computing a desired number of first linear principal components of the data and fixing the sources to those values for some time while the MLP network was adapted. The linear initialisation is robust and seems to work well in general, but a nonlinear initialisation provided by KPCA should lead to better results and faster learning.

In the next section, kernel PCA and variational Bayesian nonlinear BSS methods will be presented in more detail. Experimental results of using KPCA initialisation for VB approach are presented in Section 3. The paper concludes with discussion and conclusions in Sections 4 and 5.

2 The Methods

In this section, kernel PCA and the variational Bayesian nonlinear BSS method will be introduced briefly. For more details, see the referenced papers.

2.1 Kernel PCA

Kernel principal component analysis (kernel PCA) was introduced in [3] as a nonlinear generalisation of principal component analysis. The idea is to map given data points from their input space $\mathbb{R}^n$ to some high-dimensional (possibly infinite-dimensional) feature space $\mathcal{F}$,

$$\Phi : \mathbb{R}^n \to \mathcal{F}, \tag{1}$$

and to perform PCA in $\mathcal{F}$. The space $\mathcal{F}$ and therewith also the mapping Φ might be very complicated. However, employing the so-called kernel trick, kernel PCA avoids to use Φ explicitly: PCA in $\mathcal{F}$ is formulated in such a way that only the inner product in $\mathcal{F}$ is needed (for details see [3]). This inner product can be seen as some nonlinear function, called *kernel function*,

$$\begin{aligned} \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R} \\ (\mathbf{x}, \mathbf{y}) &\mapsto \mathbf{k}(\mathbf{x}, \mathbf{y}), \end{aligned} \tag{2}$$

which calculates a real number for each pair of vectors from the input space. Deciding on the form of the kernel function, defines implicitly the feature space $\mathcal{F}$ (and the mapping Φ). The kernel functions used in this paper are shown in Table 1. These functions are not proper Mercer kernels and the "covariance matrix" evaluated in feature space is not positive semidefinite. Most eigenvalues are nevertheless positive and the corresponding components are meaningful, so the negative eigenvalues can be simply ignored.

2.2 Variational Bayesian Nonlinear BSS

Denoting the observed data by $\boldsymbol{X} = \{\mathbf{x}(t)|t\}$ and the sources by $\boldsymbol{S} = \{\mathbf{s}(t)|t\}$, the generative model for the VB nonlinear BSS method can be written as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_\mathbf{f}) + \mathbf{n}(t), \tag{3}$$

Table 1. Summary of the kernels used in the experiments

Function	Values of parameter κ used
$\tanh(\kappa(\mathbf{x} \cdot \mathbf{y}))$	$10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{1.5}, 10^{2}$
$\operatorname{arsinh}(\kappa(\mathbf{x} \cdot \mathbf{y}))$	$10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{1.5}, 10^{2}$

where $\mathbf{f}$ is the unknown nonlinear (mixing) mapping modelled by a multilayer perceptron (MLP) network with weights and parameters $\boldsymbol{\theta}_{\mathbf{f}}$, and $\mathbf{n}(t)$ is Gaussian noise. The sources $\boldsymbol{S}$ are usually assumed to have a Gaussian prior, which leads to a PCA like nonlinear factor analysis (NFA) model. This can be extended to a full nonlinear BSS method by either using a mixture-of-Gaussians source prior or using standard linear ICA as post-processing for the sources recovered by NFA. As the latter method is significantly easier and produces almost as good results, it is more commonly used [5, 6].

The NFA model is learned by a variational Bayesian learning method called ensemble learning. As a variational Bayesian method, ensemble learning is based on finding a simpler approximation to the true posterior distribution $p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})$ of the sources and model parameters $\boldsymbol{\theta}$. The approximation $q(\boldsymbol{S}, \boldsymbol{\theta})$ is fitted by minimising the cost function

$$\mathcal{C} = E_q \left[\log \frac{q(\boldsymbol{S}, \boldsymbol{\theta})}{p(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X})} \right] = D_{KL}(q(\boldsymbol{S}, \boldsymbol{\theta}) || p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})) - \log p(\boldsymbol{X}), \tag{4}$$

where $D_{KL}(q||p)$ denotes the Kullback-Leibler divergence between the distributions q and p. The remaining evidence term is a constant with respect to the parameters of the model so the cost is minimised when the Kullback-Leibler divergence is minimised. Because the Kullback-Leibler divergence is always non-negative, the cost function yields an upper bound for $-\log p(\boldsymbol{X})$ and consequently a lower bound for model evidence $p(\boldsymbol{X})$. The values of the cost function can be thus used for model comparison with smaller values indicating larger lower bounds on model evidence [7, 8]. In our case, the approximating distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$ is restricted to be a multivariate Gaussian with a diagonal covariance.

2.3 Learning and Initialisation of the VB Method

The variational Bayesian learning algorithm of the NFA model is based on iterative updates of the parameters of the approximating distribution. The means and diagonal elements of the covariance correspond to estimated values and variances of the different sources and weights. The sources and MLP network weights are updated by minimising the cost in Eq. (4) with a gradient based algorithm. The optimal values of other model parameters such as noise variances and parameters of the hierarchical priors can be solved exactly if the other parameters are assumed to be fixed.

Because of the iterative nature of the update algorithms and especially because the MLP network is very prone to local optima, the method needs a good

initialisation to produce good results. Earlier, a given number of first linear PCA components has been used as initialisation of the posterior means of the sources while the means of the weights have been initialised randomly. The variances of all parameters are initialised to small constant values. The means of the sources are then kept fixed for the first 50 iterations while the network adapts to model the mapping from the PCA sources to the observations [5].

In this work, the principal components extracted with the linear algorithm are replaced with components extracted with the nonlinear kernel PCA algorithm. Otherwise the learning proceeds in the same way as before. The flow of information in the method is illustrated in Fig. 1.

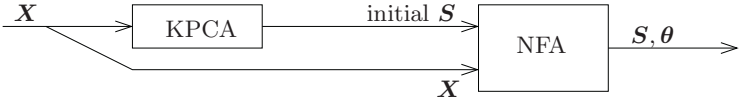

Fig. 1. A block diagram of the learning method

3 Experiments

The experiments were conducted using the same artificial data set that was used in [9]. The data was generated by mapping 4 super-Gaussian and 4 sub-Gaussian sources with a random MLP to a 20 dimensional space and adding some noise. The number of samples used was 1000. The NFA model used an MLP network with 10 inputs (sources), 30 hidden neurons and 20 outputs. The model can prune unneeded sources so using too many causes no problems[1].

In order to get the initialisations for the sources, kernel PCA was applied to the data. A number of different types of kernels and parameters were used as listed in Table 1. These were then all used for brief simulations with the NFA algorithm to see which provided the best results. The results in terms of cost function value attained after 1000 iterations are illustrated in Fig. 2. The figure shows that larger parameter values tend to produce better results although variations between neighbouring values can be large.

The results of the experiments were evaluated based on both the attained values of the cost function in Eq. (4) and the signal-to-noise ratios (SNRs) of the optimal linear reconstruction from the estimated source subspace to the true sources. The two statistics are strongly correlated, as illustrated in Fig. 3. This shows that the ensemble learning cost function is a very good measure of the quality of the found solution. This is in agreement with the results reported in [9] for a hierarchical nonlinear model.

Based on the results shown in Fig. 2, the parameter value $\kappa = 10^{1.5} \approx 31.6$ was chosen as best candidate for the tanh kernel and the value $\kappa = 100$ for the arsinh kernel. The simulations for these kernels and linear initialisation were

[1] Matlab code for KPCA and NFA methods used in the experiments is available at http://www.lis.inpg.fr/pages_perso/bliss/deliverables/d20.html.

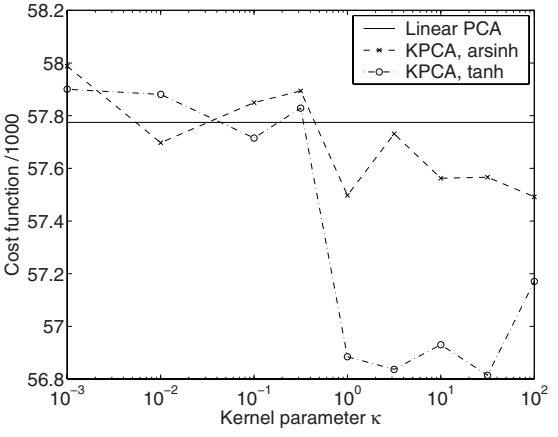

Fig. 2. Comparison of cost function values attained with different kernels and their parameter values after 1000 iterations of the NFA algorithm. The lines show the mean result of 10 simulations with different random MLP initialisations for kernel PCA with tanh and arsinh kernels and linear PCA

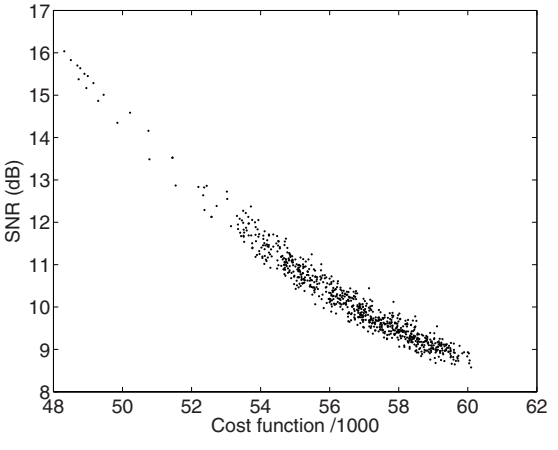

Fig. 3. Signal-to-noise ratio of the optimal linear reconstruction of the true sources from the estimated source subspace as a function of the cost function value attained in different stages of different simulations, some of which were run for up to 50000 iterations

then continued for 4000 more iterations. The SNRs attained at different stages of learning on average in 10 simulations with these initialisations are illustrated in Fig. 4. The results show that kernel PCA is able to provide a consistent improvement of about 1 dB in signal-to-noise ratio to the results attained in equal time with linear PCA initialisation.

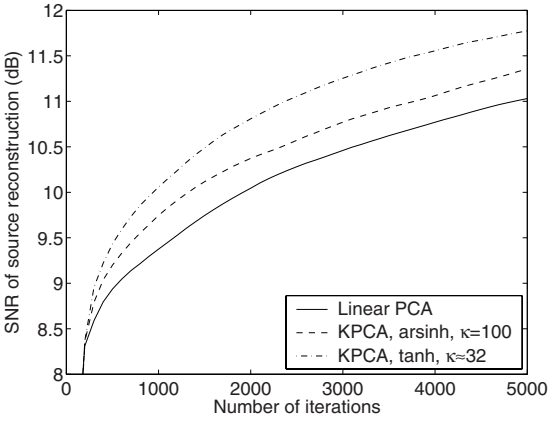

Fig. 4. Comparison of signal-to-noise ratios attained with linear PCA and kernel PCA initialisations. The results shown here are the mean of 10 simulations with different random MLP initialisations

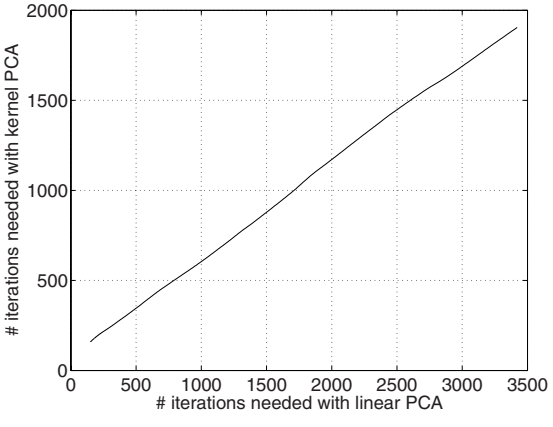

Fig. 5. The number of iterations needed on average to attain the same level of cost function value with linear PCA initialisation as a function of number of iterations needed with kernel PCA initialisation

Looking at the same result from time perspective, the kernel PCA initialisation can speed up learning significantly. This can be seen from Fig. 5, which shows a comparison of numbers of iterations needed with different initialisations on average in 10 simulations with tanh kernel to reach a given level of cost function value. The figure shows that as good results can be attained with kernel PCA initialisation using only slightly more than half of the time needed with linear PCA initialisation.

4 Discussion

The signal-to-noise ratios reported in the experiments were evaluated for optimal linear reconstruction from the estimated source subspace to the true sources. As noted in [9], these optimal results are presumably about 1 dB higher than completely blind application of linear ICA would produce. The optimal reconstruction was selected for comparison because it needed to be evaluated often and was more efficient to evaluate than running linear ICA every time and avoided a possible source of error.

In order to find out which kernels were the best ones, the signal-to-noise ratios were also evaluated for the components extracted with linear PCA and kernel PCA with different kernels. Surprisingly these SNRs had little correlation with how well NFA worked with different initialisations. The best SNR among the initialisations was attained by linear PCA followed by the kernels that were closest to linear. These were however not the ones that produced the best overall results. Fortunately the best kernels could be identified rather quickly from the cost function values attained during learning.

5 Conclusions

The experiments show that kernel PCA can provide significantly better initialisation for nonlinear factor analysis than linear PCA. The lower bound of model evidence provided by the cost function correlates strongly with the quality of the results as measured by the signal-to-noise ratio of optimal linear reconstruction of true sources from the estimated sources, thus allowing easy evaluation of results. The cost function can also be evaluated in more realistic situations, whereas the SNR cannot.

From variational Bayesian perspective, the kernel PCA initialisations are good complement to the nonlinear BSS method. Considering the significant computational demands of the basic method, the computation time required for kernel PCA and even kernel selection is more or less negligible. From kernel point of view, the variational Bayesian NFA is an interesting complement to KPCA as it allows relatively easy comparison of different kernels and parameter values.

Acknowledgements

This work was supported in part by the IST Programme of the European Community, under the project BLISS, IST-1999-14190, and under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. J. Wiley, 2001.

2. C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pp. 245–256, 2003. Invited paper in the special session on nonlinear ICA and BSS.
3. B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
4. S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller, "Kernel-based nonlinear blind source separation," *Neural Computation*, vol. 15, no. 5, pp. 1089–1124, 2003.
5. H. Lappalainen and A. Honkela, "Bayesian nonlinear independent component analysis by multi-layer perceptrons," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 93–121, Berlin: Springer-Verlag, 2000.
6. H. Valpola, E. Oja, A. Ilin, A. Honkela, and J. Karhunen, "Nonlinear blind source separation by variational Bayesian learning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 3, pp. 532–541, 2003.
7. G. E. Hinton and D. van Camp, "Keeping neural networks simple by minimizing the description length of the weights," in *Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, (Santa Cruz, CA, USA), pp. 5–13, 1993.
8. D. J. C. MacKay, "Developments in probabilistic modelling with neural networks – ensemble learning," in *Neural Networks: Artificial Intelligence and Industrial Applications. Proc. of the 3rd Annual Symposium on Neural Networks*, pp. 191–198, 1995.
9. H. Valpola, T. Östman, and J. Karhunen, "Nonlinear independent factor analysis by hierarchical models," in *Proc. 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, (Nara, Japan), pp. 257–262, 2003.