NONLINEAR INDEPENDENT COMPONENT ANALYSIS USING ENSEMBLE LEARNING: EXPERIMENTS AND DISCUSSION

Harri Valpola[1], *Xavier Giannakopoulos*[2], *Antti Honkela*[1], *and Juha Karhunen*[1]

[1]Helsinki University of Technology, Neural Networks Research Centre
P.O.Box 5400, FIN-02015 HUT, Espoo, Finland
[2]IDSIA, Galleria 2, CH-6928 Manno, Switzerland
E-mail: Harri.Valpola@hut.fi, Xavier@idsia.ch, Antti.Honkela@hut.fi, Juha.Karhunen@hut.fi URL: http://www.cis.hut.fi/

ABSTRACT

In this paper, we present experimental results on a nonlinear independent component analysis approach based on Bayesian ensemble learning. The theory of the method is discussed in a companion paper. Simulations with artificial and natural data demonstrate the feasibility and good performance of the proposed approach. We also discuss the relationships of the method to other existing methods.

1. INTRODUCTION

The nonlinear independent component analysis (ICA) algorithm discussed here is based on generative learning. This means that we try to find a model which allows a compact description of the observations in the hope of discovering some of the underlying causes of the observations. Whether the explanation is successful depends on the class of models we use. If the observations are generated by a process which is very difficult to describe with the chosen generative model, there is not much hope of recovering the original causes.

Linear ICA is suitable when it is reasonable to assume that the observations have been generated by a linear mixing from some independent source signals. In many realistic cases the process which generates the observations is nonlinear, however, and then a nonlinear generative model is needed in order to recover the original independent causes or independent components.

We shall demonstrate that the nonlinear ICA algorithm described in [8] can be used for estimating the independent components which have generated the observations through a nonlinear mapping. The algorithm uses multi-layer perceptron (MLP) network to model the nonlinear mapping from sources to observations and ensemble learning to estimate the posterior distributions of the unknown variables of the model, consisting of the parameters of the MLP network, source signals, noise levels, etc.

2. LEARNING SCHEME

The learning algorithm is a gradient based second order method [8, 2]. It is able to efficiently prune away superfluous parts of the network as will be shown later. This ability is linked to the robustness of the learning algorithm against overfitting. It is necessary when fitting a flexible nonlinear model such as an MLP network to observations. The pruning capability can also be harmful in the beginning of learning when the network has not yet found a good representation of the observations because the network can prematurely prune away parts which are not useful in explaining the observations. These parts could be useful later on when the network refines its representation.

This problem can be avoided by making sure there is something reasonable to learn. In the beginning the sources are initialised to the values given by principal components of the observations. The sources are then kept fixed for 50 sweeps through the data and only the parameters of the MLP are updated. After the MLP has learned a mapping from PCA sources to observations also the sources will be adapted. After another 50 sweeps both the sources and the parameters of the MLP have reasonable values, and after that also the noise level of each observation channel and the distribution of the sources are updated.

The distribution of the sources is modelled by a mixture of Gaussians. In the beginning when the MLP has not yet found the correct nonlinear subspace where the observations lie, a complex model for the distribution of the sources is unnecessary, however, and therefore only one Gaussian is used in the mixture for the first 2000 sweeps. After that the sources are rotated

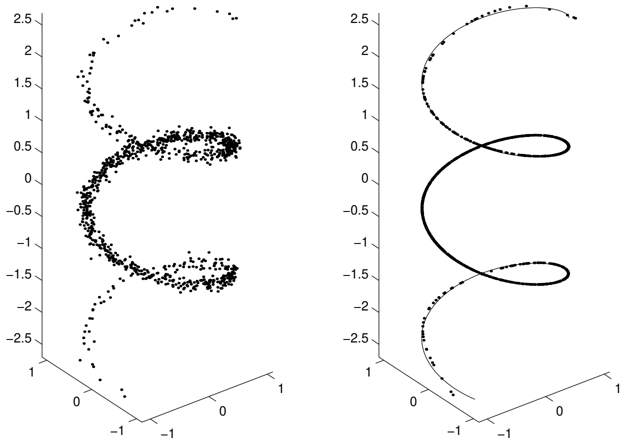

Figure 1: The noisy data points are shown on the left and the outputs of the MLP network (points) together with the underlying helical subspace (solid line) are shown on the right.

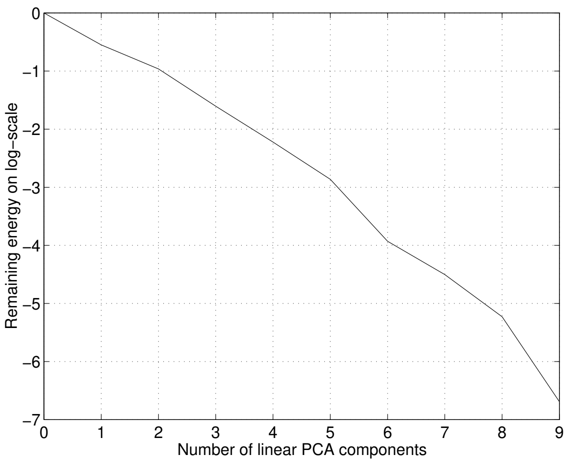

Figure 2: Remaining energy in the data as a function of extracted linear PCA components.

by a linear ICA algorithm in order to find independent sources. The source distributions are thereafter modelled by mixtures of Gaussians. A total of 7500 sweeps through the data was used in all simulations.

This procedure can be seen as first using nonlinear PCA to estimate a nonlinear subspace and then using nonlinear ICA to refine the model. This is analogous to the linear case where linear PCA is often used for estimating a linear subspace for the linear ICA. Since the algorithm estimates the noise level on each channel separately, it is more appropriately called nonlinear independent factor analysis (IFA) or, when using only one Gaussian, nonlinear factor analysis (FA).

3. SIMULATIONS

3.1. Helix

We start with a toy problem which is easy to visualise. A set of 1000 data points, shown on the left of Fig. 1, was generated from a normally distributed source s into a helical subspace. The mapping onto x-, y- and z-axes were $x = \sin(\pi s)$, $y = \cos(\pi s)$ and $z = s$. Gaussian noise with standard deviation 0.05 was added to all three data components.

Several different initialisations of the MLP networks with different number of hidden neurons were tested and the network which produced the lowest value of the cost function was chosen. The best network had 16 hidden neurons and it had estimated the noise level of different data components to be 0.052, 0.055 and 0.050. The outputs of the network for each estimated value of the source signal[1] together with the original helical subspace are shown on the right of Fig.1. It is evident that the network has learned to represent the underlying one-dimensional subspace and has been able to separate the signal from noise.

3.2. Nonlinear Artificial Data

The cost function which is based on ensemble learning and which the algorithm tries to minimise can be interpreted as the description length of the data [2]. The following experiments show that the cost function can be used for optimising the structure of the MLP network in addition to learning the unknown variables of the model.

This data set of 1000 vectors was generated by a randomly initialised MLP network with five inputs, 20 hidden neurons and ten outputs. The inputs were all normally distributed. Gaussian noise with standard deviation 0.1 was added to the data. The nonlinearity for the hidden neurons was chosen to be the inverse hyperbolic sine, while the MLP network which was estimated by the algorithm had hyperbolic tangent as its nonlinearity.

Figure 2 shows how much of the energy remains in the data when a number of linear PCA components are extracted. This measure is often used to infer the linear dimension of the data. As the figure shows, there is no obvious turn in the curve and it is difficult to tell what the linear dimension is. At least it is not five which is the underlying nonlinear dimension of the data.

[1] Our algorithm estimates the posterior distribution for all unknown parameters [8]. Posterior means are shown in all figures.

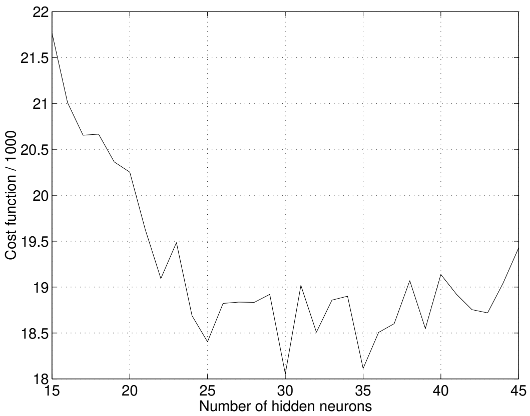

Figure 3: Several different initialisations of the MLP network were tested and the smallest attained value of the cost function is shown for each number of hidden neurons.

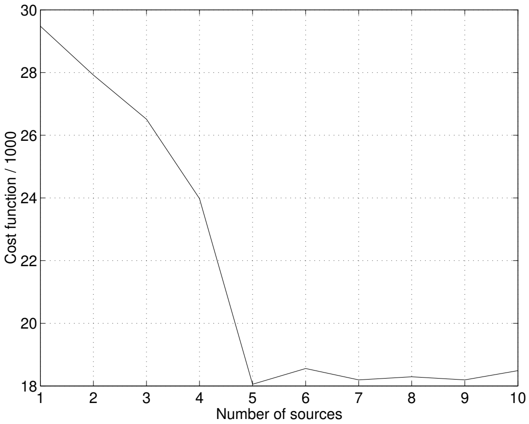

Figure 4: Several different initialisations of the MLP network were tested and the smallest attained value of the cost function is shown for each number of sources.

With the nonlinear IFA by MLP networks, not only the number of sources but also the number of hidden neurons needs to be estimated. With the cost function based on ensemble learning this is not a problem as is seen in Figs. 3 and 4. The cost function exhibits a broad minimum as a function of the number of hidden neurons and saturates after five sources when plotted as a function of sources.

The value of the cost function can be interpreted as the description length of the whole data. It is also possible to have a closer look at the terms of the cost function and interpret them as the description lengths of individual parameters [2]. The amount of bits which the network has used for describing a parameter can then be used to judge whether the parameter can be pruned away.

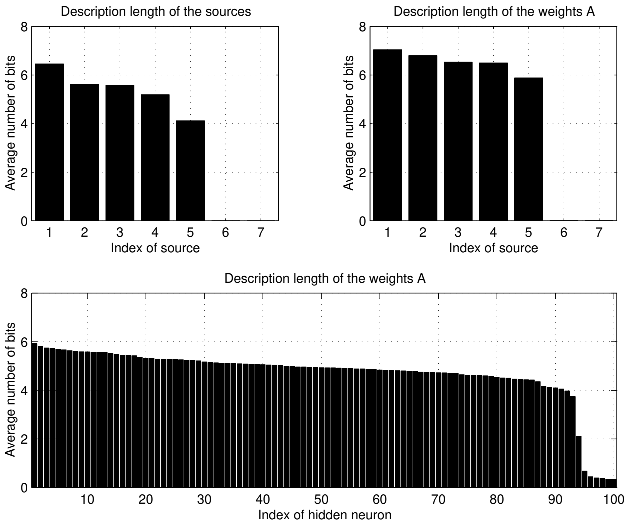

Figure 5: Average number of bits used by the network to describe various variables.

Figure 5 shows average description lengths for different variables when the data was the same as in previous simulation and an MLP network with seven inputs and 100 hidden neurons was used for estimating the sources. Clearly only five out of seven sources were used by the network. However, only a few hidden neurons were effectively pruned which shows that there is not much pressure for the network to prune away extra hidden neurons. The overall value of the cost function was higher than for models with equal number of sources but fewer hidden neurons.

3.3. Non-Gaussian Sources

The following simulations demonstrate the capability of the algorithm to discover the underlying causes of the observations. The observations were generated by a randomly initialised MLP network as before. The generating MLP network had eight inputs, 30 hidden neurons and 20 outputs. Four of the sources were super-Gaussian and four were sub-Gaussian. Several MLP networks with different structures and initialisations were used for estimating the sources and the results obtained by the network which reached the lowest value of the cost function are presented here. This network had 50 hidden neurons.

FastICA, a well-known linear ICA algorithm, gives the sources shown in Fig. 6. On each of the eight scatter plots one of the original sources is plotted against the estimated source which best correlates with the original source. An optimal result would be a straight line

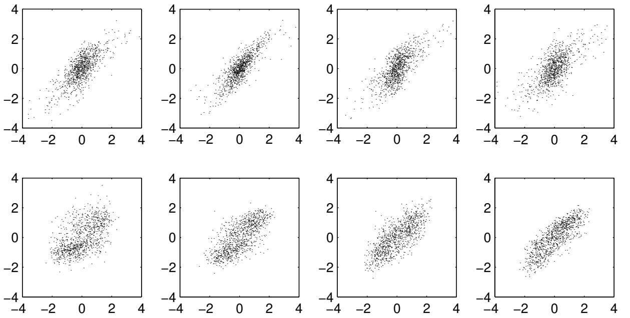

Figure 6: Sources estimated by linear ICA.

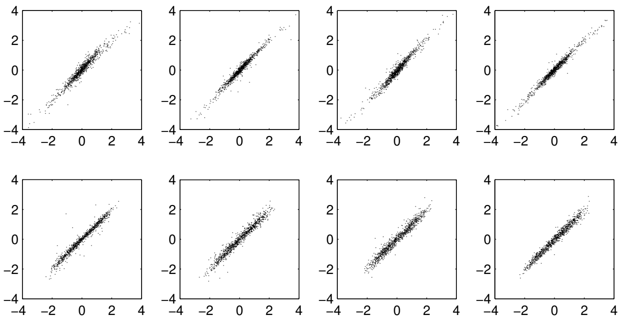

Figure 7: The nonlinear subspace has first been estimated by nonlinear FA (2000 sweeps). The the sources have been rotated by linear ICA.

on each plot. Judging from the plots in Fig. 6, linear ICA is not able to retrieve the original sources. This is also evident from the signal to noise ratio which is 0.7 dB. The inability of the linear ICA to find the original sources is caused by the mismatch between the actual generating model, which is nonlinear, and the assumed linear model.

After 2000 sweeps with the nonlinear FA, that is, using only one Gaussian for modelling the distribution of each source, and a rotation with the FastICA, the sources have greatly improved as can be seen in Fig. 7. The nonlinear FA has been able to detect the nonlinear subspace in which the data points lie. The rotation ambiguity inherent in FA has been solved by the linear ICA. At this stage the signal to noise ratio is 13.2 dB.

Now the sources have non-Gaussian distributions and it is reasonable to use mixtures of Gaussians to model the distribution of each source. Three Gaussians were used for each mixture, but it would have been possible to optimise also the number of Gaussians. The results after another 5500 sweeps through the data are depicted in Fig. 8. The signal to noise ratio has further improved to 17.3 dB. Part of the improvement is due to fine-tuning of the nonlinear subspace which would

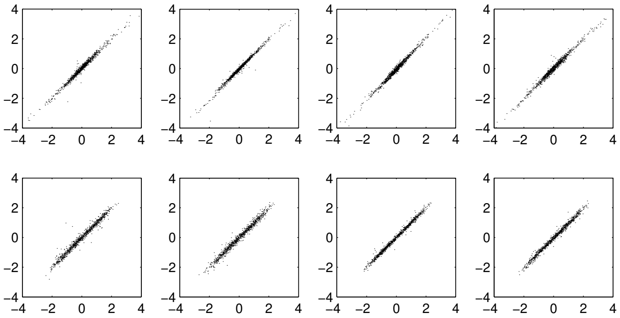

Figure 8: The sources have further been refined by nonlinear IFA for another 5500 sweeps.

have taken place even if only nonlinear FA were applied. However, the signal to noise ratio achieved by pure nonlinear FA applied for 7500 iterations is only 14.9 dB which shows that the network has also taken into account the non-Gaussian models of the sources.

3.4. Process Data

This data set consists of 2480 measurements from 30 sensors of an industrial pulp process. An expert has preprocessed the signal by roughly compensating for time lags of the process which originate from the finite speed of pulp flow through the process.

In order to get an idea of the dimensionality of the data, linear FA was applied to the data and compared with the nonlinear FA. It turned out that linear FA needs over twice as many sources for representing as much data as the nonlinear FA [2], which is a clear evidence for the nonlinearity of the data manifold.

Again several different structures and initialisations for the MLP network were tested and the cost function was found to be minimised by a model having 10 sources and 30 hidden neurons. The estimated sources are shown in Fig. 9 and the nonlinear reconstruction from sources to observations together with the original time series are shown in Fig. 10. Many of the reconstructions are strikingly accurate and in some cases it seems that the reconstructions have even less noise than the original signals. This is somewhat surprising since the time dependencies in the signal were not included in the model. The observation vectors could be arbitrarily shuffled and the model would still produce the same results.

3.5. Inversion by Auxiliary MLP network

During learning, the sources have been co-adapted with the network. The mapping has first been fairly smooth and gradually evolved into a more nonlinear one. Since

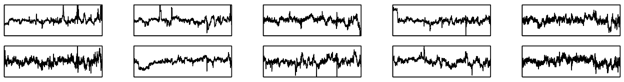

Figure 9: Ten source signals estimated from the industrial pulp process. Time increases form left to right.

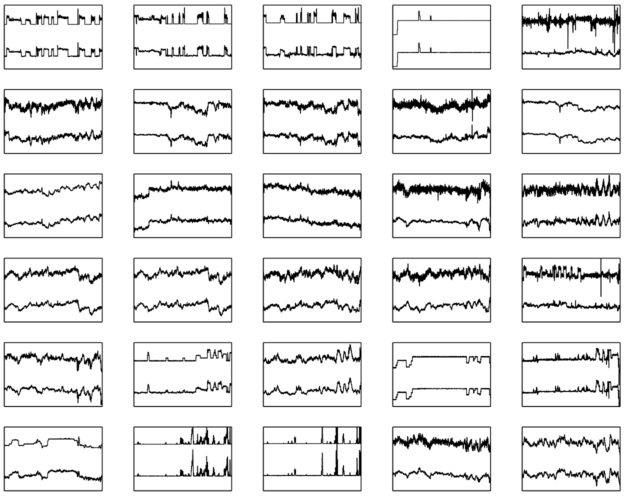

Figure 10: Each plot shows one of the thirty original time series on top of the nonlinear reconstruction made from the sources shown in Fig. 9.

the model defines the mapping from sources to observations, it is not trivial to find the sources given the observations, that is, to invert the model. In many applications it is necessary to estimate the sources for new observations which have not been seen by the network during learning, however, and it can be asked whether the gradient based method is able to invert the network or will it get stuck in local minima.

To test this, a new set of 1000 observation vectors were generated with the same generating MLP network as in the experiment with non-Gaussian artificial data in Sect. 3.3. Then several different techniques were tested for initialising the sources for the gradient descent based inversion of the same network whose results are shown in Fig. 8.

The best method turned out to be an auxiliary MLP network which was taught to approximate the inverse of the nonlinear mapping using Matlab Neural Network Toolbox. It had the same number of hidden neurons as the model MLP network and the numbers of inputs and output neurons had been switched to account for the inverse mapping. To teach the auxiliary MLP network we used the original data which was used in Sect. 3.3 and the sources estimated for that data. It is then possible to use the auxiliary MLP network to initialise

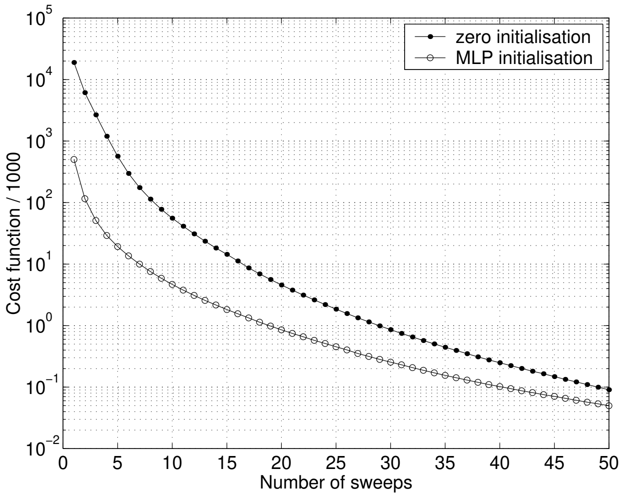

Figure 11: Cost function as the number of sweeps.

the sources for new observations. A local minimum was detected only with four observations out of 1000.

The naive initialisation with zeros is compared with the initialisation obtained by the auxiliary MLP network in Fig. 11. The case where all the sources have been set to the best values found is used as the baseline. On one hand, the figure shows that auxiliary MLP network gives a good initialisation, but on the other hand, it shows also that the auxiliary MLP network alone does not reach the quality obtained by gradient descent.

4. DISCUSSION

Learning nonlinear ICA can be based on several different criteria, but they all aim at finding models which could describe as large part of the observations as possible with as compact description of the sources as possible. The nonlinear ICA algorithms presented in the literature can be roughly divided in two classes: generative approaches which estimate the generative model and signal transformation (ST) approaches which estimate the the recognition model, that is, the inverse of the generative model.

Since the generative model is not directly estimated in the ST approach, it is difficult to measure how large part of the observations can be described with the sources except in the case when there are as many sources as there are observations. Then the observations can be perfectly reconstructed from the sources as long as the recognition mapping is invertible. To the best of our knowledge, all existing ST approaches are restricted to this case. The problem then reduces to transforming the observations into sources which are statistically as independent as possible. For an account on ST approaches for nonlinear ICA, see for instance [9, 4, 7] and references therein.

In the ST approaches, the problem of model indeterminacy inherent in nonlinear ICA has usually been solved by restricting the model structure. The number of hidden neurons is the same as the number of observations in [9]. In [4], the number of hidden neurons controls the complexity of the reconstruction model and [7] is restricted to post-nonlinear mixtures. Principled way of making the trade-off between the complexity of the mapping and the sources has not been presented in the general case for the ST approach.

In the generative approaches it is easy to measure how large part of the observations is explained by the sources, and consequently, easy to assess the quality of the model. It might seem that the estimation of the sources would be a problem, but this is not the case as shown here. During learning, small changes in the generative model result in small changes in the optimal values of the sources and it is therefore easy to track the source values by gradient descent.

Although it is possible to measure the complexity of the mapping and the sources in generative approaches, no algorithms which would do this for nonlinear ICA have been proposed apart from our algorithm. Most often the maximum a posteriori (MAP) or the maximum likelihood (ML) estimate is used at least for some of the unknown variables. In coding terms, a point estimate means that it is impossible to measure the description length because the accuracy of description of the variable is neglected. In nonlinear ICA it is necessary to use better estimates for the posterior density of the unknown variables or otherwise there will be problems with overfitting which can be overcome only by restricting the model structure.

Self-organising maps (SOM) and generative topographic mapping (GTM) have been used for nonlinear ICA. In [6], GTM was used for modelling the nonlinear mapping from sources to observations. The number of parameters grows exponentially as a function of sources both in SOM and GTM, which makes these mappings unsuitable for larger problems. ML estimate was used for the parameters of the mapping.

MLP networks have been used as generative models in [3, 5, 1]. In [3], the model for the sources is Gaussian and computationally expensive stochastic approximation is used for estimating the distribution of the unknown parameters. Only a very simple network with the structure 2-16-2 was tested. ML estimate for the sources and the parameters of the MLP was used in [5], while in [1], the posterior distribution of the parameters of an auto-associative MLP network was approximated. The distribution of the sources was not modelled in neither paper.

Although MLP networks are universal models, which means that any nonlinear mapping can be approximated with arbitrary accuracy given enough hidden neurons, it is difficult to approximate some mappings with MLP networks. This problem cannot be completely escaped by any model since for each model there are mappings which are difficult to represent. MLP networks are in wide use because they have been found to be good models for many naturally occurring processes. There are also modifications to the basic MLP network structure which further increase its representational power. For simplicity we have used the standard MLP structure but it would be possible to use many of these extensions in our algorithm also. It is evident that for instance the signals of the pulp process in Fig. 10 have strong time dependencies, and taking them into account will be an important extension.

5. REFERENCES

[1] S. Hochreiter and J. Schmidhuber. Feature extraction through LOCOCODE. *Neural Computation*, 11(3):679–714, 1999.

[2] H. Lappalainen and A. Honkela. Bayesian nonlinear independent component analysis by multi-layer perceptrons. In M. Girolami, ed., *Advances in Independent Component Analysis*. Springer, Berlin, 2000. In Press.

[3] D. J. C. MacKay and M. N. Gibbs. Density networks. In J. Kay, ed., *Proceedings of Society for General Microbiology Edinburgh meeting*, 1997.

[4] G. C. Marques and L. B. Almeida. Separation of nonlinear mixtures using pattern repulsion. In *Proc. ICA'99*, pp. 277–282, Aussois, France, 1999.

[5] J.-H. Oh and H. S. Seung. Learning generative models with the up-propagation algorithm. In M. I. Jordan, M. J. Kearns, and S. A. Solla, eds., *Advances in Neural Information Processing Systems 10*, pp. 605–611. MIT Press, 1998.

[6] P. Pajunen and J. Karhunen. A maximum likelihood approach to nonlinear blind source separation. In *Proceedings of the 1997 Int. Conf. on Artificial Neural Networks (ICANN'97)*, pp. 541–546, Lausanne, Switzerland, 1997.

[7] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 47(10):2807–2820, October 1999.

[8] H. Valpola. Nonlinear independent component analysis using ensemble learning: Theory. In *Proc. ICA 2000*. In press.

[9] H. H. Yang, S. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in nonlinear mixture. *Signal Processing*, 64:291–300, 1998.