NONLINEAR SOURCE SEPARATION USING ENSEMBLE LEARNING AND MLP NETWORKS

Harri Lappalainen[1], *Antti Honkela*[1], *Xavier Giannakopoulos*[2], *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
Email: `Harri.Lappalainen@hut.fi`, `Antti.Honkela@hut.fi`, `xavier@idsia.ch`,
`Juha.Karhunen@hut.fi` URL: `http://www.cis.hut.fi`

ABSTRACT

We consider extraction of independent sources from their nonlinear mixtures. Generally, this problem is very difficult, because both the nonlinear mapping and the underlying sources are unknown and should be learned from the data. In this paper, we use multilayer perceptrons as nonlinear generative models for the data. The model indeterminacy problem is resolved by applying ensemble learning. This Bayesian method selects the most probable generative data model. In simulations with artificial data, the network is able to find the underlying sources from the observations only, even though the data generating mapping is strongly nonlinear. We have applied the developed method also to real-world process data.

1. INTRODUCTION

Unsupervised learning methods are often based on a generative approach, where the goal is to find a specific model which explains how the observations were generated. It is assumed that there exist certain source signals (also called factors, latent or hidden variables, or hidden causes) which have generated the observed data through an unknown mapping. The goal of generative learning is to identify both the source signals and the unknown generative mapping.

The success of a specific model depends on how well it captures the structure of the phenomena underlying the observations. Various linear models have been popular, because their mathematical treatment is fairly easy. However, in many realistic cases it is reasonable to assume that the observations have been generated by a nonlinear process. Take for example the effect of pressure and temperature on the properties of the end product of a chemical process. Although the effects can be locally linear, the overall effects in real world are almost always nonlinear. Also, there are usually several sources whose nature and effect on the observations are completely unknown and which cannot be measured directly.

Our goal is to develop methods for inferring the original sources from the observations alone. The nonlinear mapping from the unknown sources to the observations is modeled with the familiar multi-layer perceptron (MLP) network [6, 3], but the learning procedure is quite different from standard backpropagation.

2. ENSEMBLE LEARNING

A flexible model family, such as MLP networks, provides infinitely many possible explanations of different complexity for the observed data. Choosing too complex a model results in overfitting, where the model tries to make up meaningless explanations for the noise in addition to the true sources or factors. Choosing too simple a model results in underfitting, leaving hidden some of the true sources that have generated the data.

The solution to the problem is that no single model should actually be chosen. Instead, all the possible explanations should be taken into account and weighted according to their posterior probabilities. This approach, known as Bayesian learning, optimally solves the tradeoff between under- and overfitting. All the relevant information needed in choosing an appropriate model is contained in the posterior probability density functions (pdfs) of different model structures. The posterior pdfs of too simple models are low, because they leave a lot of the data unexplained. Also too complex models occupy little probability mass, though they often show a high but very narrow peak in their posterior pdfs corresponding to the overfitted parameters.

In practice, exact treatment of the posterior pdfs of the models is impossible. Therefore, some suitable approximation method must be used. Ensemble learning [7, 2, 13], also known as variational learning, is a recently developed method for parametric approximation of posterior pdfs where the search takes into account the probability *mass* of the models. Therefore, it does not suffer from overfitting. The basic idea in ensemble learning is to minimize the misfit between the posterior pdf and its parametric approximation.

Let $P(\boldsymbol{\theta}|X)$ denote the exact posterior pdf and $Q(\boldsymbol{\theta})$ its parametric approximation. The misfit is measured with the Kullback-Leibler (KL) divergence C_{KL} between P and Q, defined by the cost function

$$C_{\mathrm{KL}} = E_Q \left\{ \log \frac{Q}{P} \right\} = \int d\boldsymbol{\theta} Q(\boldsymbol{\theta}) \log \frac{Q(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|X)}. \quad (1)$$

Because the KL divergence involves an expectation over a distribution, it is sensitive to probability mass rather than to probability density.

3. MODEL STRUCTURE

We use MLP networks which have the universal approximation property for smooth continuous mappings. They are well suited for modeling both strongly and mildly nonlinear mappings.

The data model used in this work is as follows. Let $\mathbf{x}(t)$ denote the observed data vector at time t, and $\mathbf{s}(t)$ the vector of source signals (latent variables) at time t. The matrices $\mathbf{B}$ and $\mathbf{A}$ contain the weights of the output and the hidden layer of the network, respectively, and $\mathbf{a}$ is the bias vector of the hidden layer. The vector of nonlinear activation functions is denoted by $\mathbf{f}(\cdot)$, and $\mathbf{n}(t)$ is the Gaussian noise vector corrupting the observations. Using these notations, the data model is

$$\mathbf{x}(t) = \mathbf{B}\left[\mathbf{f}\left(\mathbf{A}\mathbf{s}(t) + \mathbf{a}\right)\right] + \mathbf{n}(t). \quad (2)$$

We have used as the activation function the sigmoidal tanh nonlinearity, which is a typical choice in MLP networks. Other continuous activation functions are possible, too. The sources (latent variables) are assumed to be independent and mixtures of Gaussians. The independence assumption is natural, because the goal of the model is to find the underlying independent causes of the observations. By using mixtures of Gaussians, one can model sufficiently well any non-Gaussian distribution of sources [3]. We have earlier applied successfully this type of representation to the standard linear ICA model in [11].

The parameters of the network are: (1) the weight matrices $\mathbf{A}$ and $\mathbf{B}$ and the vector of biases $\mathbf{a}$; (2) the parameters of the distributions of the noise, source signals and column vectors of the weight matrices; (3) hyperparameters used for defining the distributions of the biases and the parameters in the group (2). All the parameterized distributions are assumed to be Gaussian except for the sources, which are modeled as mixtures of Gaussians. This does not limit severely the generality of the approach, but makes computational implementation simpler and much more efficient. The hierarchical description of the distributions of the parameters of the model used here is a standard procedure in probabilistic Bayesian modeling. Its strength lies in that knowledge about equivalent status of different parameters can be easily incorporated. For example all the variances of the noise components have a similar status in the model. This is reflected by the fact that their distributions are assumed to be governed by common hyperparameters.

4. LEARNING PROCEDURE

Usually MLP networks learn the nonlinear input-output mapping in a supervised manner using known input-output pairs, for which the mean-square mapping error is minimized using the back-propagation algorithm or some alternative method [6, 3]. In our case, the inputs are the *unknown* source signals $\mathbf{s}(t)$, and only the outputs of the MLP network, namely the observed data vectors $\mathbf{x}(t)$, are known. Hence, unsupervised learning must be applied. Due to space limitations, we give an overall description of the proposed unsupervised learning procedure in this paper. A more detailed account of the method and discussion of potentially appearing problems can be found in [12].

The practical learning procedure used in all the experiments was the same. First, linear PCA (principal component analysis) is applied to find sensible initial values for the posterior means of the sources. Even though PCA is a linear method, it yields clearly better initial values than a random choice. The posterior variances of the sources are initialized to small values. Good initial values are important for the method because the network can effectively prune away unused parts. Initially the weights of the network have random values, and the network has quite a bad representation for the data. If the sources were adapted from random values, too, the network would consider many of the sources useless for the representation and prune them away. This would lead to a local minimum from which the network might not recover.

Therefore the sources were fixed at the values given by linear PCA for the first 50 sweeps through the entire data set. This allows the network to find a meaningful mapping from sources to the observations, thereby justifying using the sources for the representation. For the same reason, the parameters controlling the distributions of the sources, weights, noise and the hyperparameters are not adapted during the first 100 sweeps. They are adapted only after the network has found sensible values for the variables whose distributions these parameters control.

Furthermore, we first used a simpler nonlinear model where the sources had Gaussian distributions instead of mixtures of Gaussians. This is called nonlinear factor analysis model in the following. After this phase, the sources were rotated using an efficient linear ICA algorithm called FastICA [9, 5]. The rotation of the sources was compensated by an inverse rotation of the weight matrix $\mathbf{A}$ of the hidden layer. The final representation of the data was then found by continuing learning using now the mixture-of-Gaussians model for the sources. In [12], this representation is called nonlinear independent factor analysis.

5. COMPUTATION OF THE COST FUNCTION

In this section, we consider in more detail the Kullback-Leibler cost function C_{KL} defined earlier in Eq. (1). For approximating and then minimizing it, we need two things: the exact formulation of the posterior density $P(\boldsymbol{\theta}|X)$ and its parametric approximation $Q(\boldsymbol{\theta})$.

Let us denote $X = \{\mathbf{x}(t)|t\}$ and $S = \{\mathbf{s}(t)|t\}$, and let $\boldsymbol{\theta}$ denote all the unknown parameters of the model. According to the Bayes' rule, the posterior pdf of the unknown variables S and $\boldsymbol{\theta}$ is

$$P(S, \boldsymbol{\theta}|X) = \frac{P(X|S, \boldsymbol{\theta})P(S|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(X)} \tag{3}$$

The term $P(X|S, \boldsymbol{\theta})$ is obtained from the equation (2). Let us denote the mean vector of $\mathbf{n}(t)$ by $\boldsymbol{\mu}$ and the variance vector of the noise term $\mathbf{n}(t)$ by $\boldsymbol{\sigma^2}$. The distribution $P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})$ is thus Gaussian with mean $\mathbf{b}_i^T\mathbf{f}(\mathbf{As}+\mathbf{a}) + \mu_i$ and variance σ_i^2. Here $\mathbf{b}_i^T$ denotes the ith row vector of $\mathbf{B}$. As usually, the noise components $n_i(t)$ are assumed to be independent, and therefore $P(X|S, \boldsymbol{\theta}) = \prod_{t,i} P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})$.

The terms $P(S|\boldsymbol{\theta})$ and $P(\boldsymbol{\theta})$ in (3) are also products of simple Gaussian distributions, and they are obtained directly from the definition of the model structure [12]. The term $P(X)$ does not depend on the model parameters and can be neglected.

The approximation $Q(S, \boldsymbol{\theta})$ must be simple for mathematical tractability and computational efficiency. We assume that it is a Gaussian density with a diagonal covariance matrix. This implies that the approximation is a product of independent distributions: $Q(\boldsymbol{\theta}) = \prod_i Q_i(\theta_i)$. The parameters of each Gaussian component density $Q_i(\theta_i)$ are its mean $\bar{\theta}_i$ and variance $\tilde{\theta}_i$.

Both the posterior density $P(\boldsymbol{\theta}|X)$ and its approximation $Q(\boldsymbol{\theta})$ are products of simple Gaussian terms, which simplifies the cost function (1) considerably: it splits into expectations of many simple terms. The terms of the form $E_Q\{\log Q_i(\theta_i)\}$ are negative entropies of Gaussians, having the exact values $-(1 + \log 2\pi\tilde{\theta}_i)/2$. The most difficult terms are of the form $-E_Q\{\log P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})\}$. They are approximated by applying second order Taylor's series expansions of the nonlinear activation functions as explained in [12]. The remaining terms are expectations of simple Gaussian terms which can be computed as in [11].

The cost function C_{KL} is a function of the posterior means $\bar{\theta}_i$ and variances $\tilde{\theta}_i$ of the source signals and the parameters of the network. This is because instead of finding a point estimate, the joint posterior pdf of the sources and parameters is estimated in ensemble learning. The variances give information about the reliability of the estimates.

Let us denote the two parts of the cost function (1) by $C_p = -E_Q\{\log P\}$ and $C_q = E_Q\{\log Q\}$. The variances $\tilde{\theta}_i$ are obtained by differentiating (1) with respect to $\tilde{\theta}_i$ [12]:

$$\frac{\partial C}{\partial \tilde{\theta}} = \frac{\partial C_p}{\partial \tilde{\theta}} + \frac{\partial C_q}{\partial \tilde{\theta}} = \frac{\partial C_p}{\partial \tilde{\theta}} - \frac{1}{2\tilde{\theta}} \tag{4}$$

Equating this to zero yields a fixed-point iteration for updating the variances:

$$\tilde{\theta} = \left[2\frac{\partial C_p}{\partial \tilde{\theta}}\right]^{-1} \tag{5}$$

The means $\bar{\theta}_i$ can be estimated from the approximate Newton iteration [12]

$$\bar{\theta} \leftarrow \bar{\theta} - \frac{\frac{\partial C_p}{\partial \bar{\theta}}}{\frac{\partial^2 C}{\partial \bar{\theta}^2}} \approx \bar{\theta} - \frac{\partial C_p}{\partial \bar{\theta}}\tilde{\theta} \tag{6}$$

6. SIMULATIONS

In all our simulations, the total number of sweeps was 7500, where one sweep means going through all the observations once. As explained before, a nonlinear factor analysis (subspace) representation using plain Gaussians as model distributions for the sources was estimated first. We used 2000 first sweeps for finding this intermediate representation. After a linear ICA rotation, the final mixture-of-Gaussians representation of the sources was then estimated during the remaining 5500 sweeps. In the following, we first present results on an experiment with artificially generated nonlinear data and then on real-world process data.

In the first experiment, there were eight sources, of which four were sub-Gaussian and four super-Gaussian

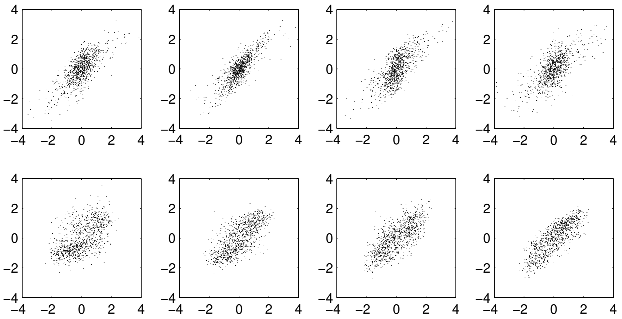

Figure 1: Original sources are on the x-axis of each scatter plot and the sources estimated by a linear ICA are on the y-axis. Signal to noise ratio is 0.7 dB.

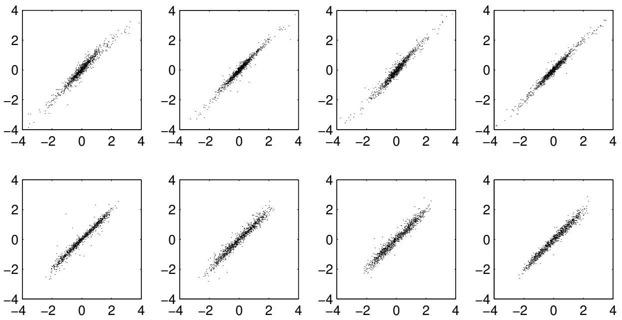

Figure 2: Scatter plots of the sources after 2000 sweeps of nonlinear factor analysis followed by a rotation with a linear ICA. Signal to noise ratio is 13.2 dB.

ones. The data were generated from these sources through a nonlinear mapping, which was obtained by using a randomly initialized MLP network having 30 hidden neurons and 20 output neurons. Gaussian noise having a standard deviation of 0.1 was added to the data. The nonlinearity used in the hidden neurons was chosen to be the inverse hyperbolic sine $\sinh^{-1}(x)$, which means that the nonlinear source separation algorithm using the MLP network with tanh nonlinearities cannot use exactly the same weights.

Several different numbers of hidden neurons were tested in order to optimize the structure of the MLP network, but the number of sources was assumed to be known. This assumption is reasonable because it is possible to optimize the number of sources simply by minimizing the cost function as experiments with purely Gaussian sources have shown [12]. The network which minimized the cost function turned out to have 50 hidden neurons. The number of Gaussians in each of the mixtures modelling the distribution of each source was chosen to be three, and no attempt was made to optimize this number.

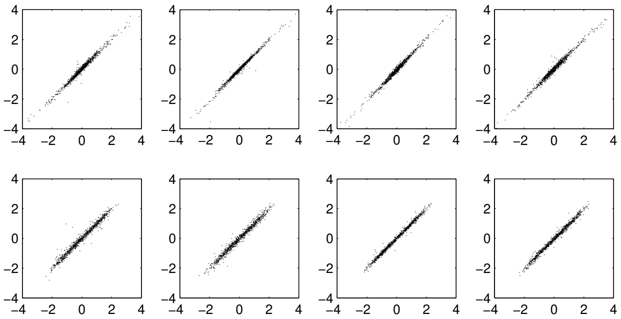

Figure 3: The final separation results after using a mixture-of-Gaussians model for the sources for the last 5500 sweeps. Signal to noise ratio is 17.3 dB.

The results are depicted in Figs. 1, 2 and 3. Each figure shows eight scatter plots, each of them corresponding to one of the eight sources. The original source which was used for generating the data is on the x-axis and the estimated source in on the y-axis of each plot. Each point corresponds to one data vector. The upper plots of each figure correspond to the super-Gaussian and the lower plots to the sub-Gaussian sources. Optimal result is a straight line implying that the estimated values of the sources coincide with the true values.

Figure 1 shows the result of a linear FastICA algorithm [9]. The linear ICA is able to retrieve the original sources with only 0.7 dB SNR (signal-to-noise ratio). In practice a linear method could not deduce the number of sources, and the result would be even worse. The poor signal-to-noise ratio shows that the data really lies in a nonlinear subspace. Figure 2 depicts the results after 2000 sweeps with Gaussian sources (nonlinear factor analysis) followed by a rotation with linear FastICA. Now the SNR is considerably better, 13.2 dB, and the sources have clearly been retrieved. Figure 3 shows the final results after another 5500 sweeps where the mixture-of-Gaussians model have been used for the sources. The SNR has further improved to 17.3 dB.

Another data set consisted of 30 time series of length 2480 measured using different sensors from an industrial pulp process. A human expert preprocessed the measured signals by roughly compensating for the time lags of the process originating from the finite speed of pulp flow through the process.

For studying the dimensionality of the data, linear factor analysis was applied to the data. The result is shown in Fig. 4. The figure shows also the results with nonlinear factor analysis. It is obvious that the data are quite nonlinear, because nonlinear factor analysis is able to explain the data with 10 components equally

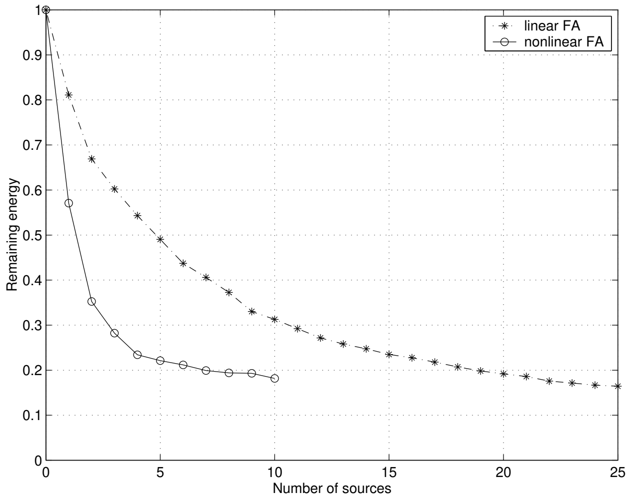

Figure 4: The remaining energy of the process data as a function of the number of extracted components using linear and nonlinear factor analysis

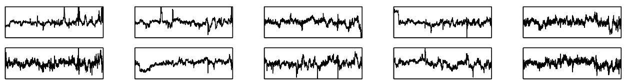

Figure 5: The ten estimated sources of the industrial pulp process. Time increases from left to right.

well as linear factor analysis with 21 components.

Different numbers of hidden neurons and sources were tested with random initializations using Gaussian model for sources (nonlinear factor analysis). It turned out that the cost function was minimized for a network of 10 sources and 30 hidden neurons. The same network size was chosen for nonlinear blind source separation using mixture-of-Gaussians model. After 2000 sweeps using the nonlinear factor analysis model the sources were rotated with FastICA, and each source was modeled with a mixture of three Gaussian distributions. The resulting sources are shown in Fig. 5.

Figure 6 shows 30 subimages, each corresponding to a specific measurement made from the process. The original measurement appears as the upper time series in each subimage, and the lower time series is the reconstruction of this measurement given by the network. These reconstructions are the posterior means of the outputs of the network when the inputs were the estimated sources shown in Fig. 5. The original measurements show a great variability, but the reconstructions are strikingly accurate. In some cases it even seems that the reconstruction is less noisy than the original signal.

The experiments are suggesting that the estimated

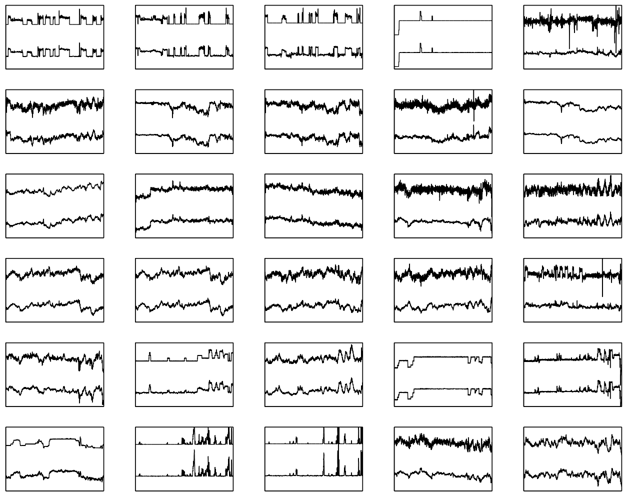

Figure 6: The 30 original time series are shown on each plot on top of the reconstruction made from the sources shown in Fig. 5

source signals can have meaningful physical interpretations. The results are encouraging, but further studies are needed to verify the interpretations of the signals.

The proposed method can be extended in several directions. An obvious extension is inclusion of time delays into the data model, making use of the temporal information often present in the sources. This would probably help to describe the process data even better. Using the Bayesian framework, it is also easy to treat missing observations or only partly known inputs.

7. RELATED WORK

Autoassociative MLP networks have been used for learning similar mappings as we have done. Both the generative model and its inversion are learned simultaneously, but separately without utilizing the fact that the models are connected. Autoassociative MLPs have shown some success in nonlinear data representation (see for example [4]), but generally they suffer from slow learning prone to local minima. Most works on autoassociative MLPs use point estimates for weights and sources obtained from minimization of the mean-square representation error for the data. It is then impossible to reliably choose the structure of the model, and problems with over- or underfitting can be severe. Hochreiter and Schmidhuber [8] have used a minimum description length based method which does estimate the distribution of the weights, but has no model for the sources. It is then impossible to measure the description length of the sources. Anyway, their method shows interesting connections with ICA [8].

MacKay and Gibbs [15] briefly report on using stochastic approximation to learn a generative MLP network which they called a density network because the model defines a density of the observations. The results are encouraging, but the model is very simple; noise level is not estimated from the observations and the latent space has only two dimensions. The computational complexity of the method is significantly greater than in the parametric approximation of the posterior presented here.

Several authors have tried to generalize linear ICA and blind source separation for nonlinear models; see [10, 14] for further information and references. In particular, a MLP model is used for separating sources from their nonlinear mixtures in [16]. This method is an extension of the well-known natural gradient method for linear blind source separation [14], and its performance for practical data is still somewhat unclear. A particular problem with nonlinear ICA is that the problem is highly non-unique without extra constraints [10]. In our method, ensemble learning provides the necessary regularization for nonlinear ICA by choosing the model and sources that have most probably generated the observed data. A similar generative model as here is applied to the linear ICA/BSS problem in [1]. Our work uses a nonlinear data model, and applies a fully Bayesian treatment to the hyperparameters of the network or graphical model, too.

8. CONCLUSIONS

We have proposed a fully Bayesian approach based on ensemble learning for solving the difficult nonlinear blind source separation problem. The MLP network used suits well for modeling both mildly and strongly nonlinear mappings. The proposed unsupervised ensemble learning method tries to find the sources and the mapping that have most probably generated the observed data. We believe that this provides an appropriate regularization for the nonlinear source separation problem. The results are encouraging for both artificial and real-world data. The proposed approach allows nonlinear source separation for larger-scale problems than some previously proposed computationally demanding approaches, and it can be easily extended in various directions.

9. REFERENCES

[1] H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

[2] D. Barber and C. M. Bishop. Ensemble learning in Bayesian neural networks. In M. Jordan, M. Kearns, and S. Solla, editors, *Neural Networks and Machine Learning*, pages 215–237. Springer, Berlin, 1998.

[3] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[4] G. Cottrell, P. Munro, and D. Zipser. Learning internal representations from grey-scale images: An example of extensional programming. In *Proc. of the 9th Annual Conf. of the Cognitive Science Society*, pages 461–473, 1987.

[5] A. Hyvärinen et al. *The FastICA MATLAB toolbox.* Available at http://www.cis.hut.fi/projects/ica/fastica/, Helsinki Univ. of Technology, Finland, 1998.

[6] S. Haykin. *Neural Networks – A Comprehensive Foundation, 2nd ed.* Prentice-Hall, 1998.

[7] G. 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*, pages 5–13, Santa Cruz, Calif., 1993.

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

[9] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[10] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(2):209–219, February 1999.

[11] H. Lappalainen. Ensemble learning for independent component analysis. In *Proc. of the First Int. Workshop on Independent Component Analysis and Signal Separation*, pages 7–12, Aussois, France, January 1999.

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

[13] H. Lappalainen and J. Miskin. Ensemble learning. In M. Girolami, editor, *Advances in Independent Component Analysis*. Springer, Berlin, 2000. In press.

[14] T.-W. Lee. *Independent Component Analysis – Theory and Applications*. Kluwer, 1998.

[15] D. MacKay and M. Gibbs. Density networks. In *Proc. of Society for General Microbiology Edinburgh Meeting*, 1997.

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