NONLINEAR INDEPENDENT FACTOR ANALYSIS BY HIERARCHICAL MODELS

Harri Valpola, Tomas Östman and Juha Karhunen

Helsinki University of Technology, Neural Networks Research Centre
P.O. Box 5400, FIN-02015 HUT, Espoo, Finland
`firstname.lastname@hut.fi` `http://www.cis.hut.fi/projects/ica/bayes/`

ABSTRACT

The building blocks introduced earlier by us in [1] are used for constructing a hierarchical nonlinear model for nonlinear factor analysis. We call the resulting method hierarchical nonlinear factor analysis (HNFA). The variational Bayesian learning algorithm used in this method has a linear computational complexity, and it is able to infer the structure of the model in addition to estimating the unknown parameters. We show how nonlinear mixtures can be separated by first estimating a nonlinear subspace using HNFA and then rotating the subspace using linear independent component analysis. Experimental results show that the cost function minimised during learning predicts well the quality of the estimated subspace.

1. INTRODUCTION

Blind separation of sources from their nonlinear mixtures—known as nonlinear blind source separation (BSS)—is generally a very difficult problem, from both theoretical and practical point of view [2, 3]. The task is to extract the sources $\mathbf{s}(t)$ that have generated the observations $\mathbf{x}(t)$ through a nonlinear mapping $\mathbf{f}(\cdot)$:

$$\mathbf{x}(t) = \mathbf{f}[\mathbf{s}(t)] + \mathbf{n}(t)\,, \tag{1}$$

where $\mathbf{n}(t)$ is additive noise.

Theoretically, the task is difficult since both the nonlinear mapping and the underlying sources must be learned from the data in a blind manner, and the problem is highly ill-posed without a suitable regularisation [2, 3]. A related problem is that it is often quite difficult to infer the number of sources and the structure of the mapping $\mathbf{f}(\cdot)$. From practical point of view, efficiency and reliability of nonlinear BSS algorithms are critical issues. They have restricted the number of sources that can be separated in practice to be quite small in many instances.

Existing nonlinear BSS methods have been reviewed in Chapter 17 of [3] and in [4]. The method introduced in this paper stems from [5], where a variational Bayesian learning method called ensemble learning was used to estimate the generative nonlinear mixture model (1).

In this paper we study the approach outlined in [1]. We construct a hierarchical nonlinear generative model for the nonlinear mixtures and learn the model by Bayesian ensemble learning. The theoretical basis was introduced in [1] but the scope in that paper was much wider. Here we study in more detail the nonlinear BSS problem, introducing hierarchical nonlinear factor analysis (HNFA), an algorithm for extracting a nonlinear subspace. This provides a nonlinear PCA (principal component analysis) solution. The estimated subspace can subsequently be rotated by standard linear independent component analysis (ICA). This provides the desired solution to the nonlinear BSS problem.

Compared to nonlinear factor analysis (NFA) presented in [5], HNFA has several advantages: 1) Its computational complexity scales better for large models; 2) The learning method is always stable and converges better; and 3) Learning of the structure of the model has been improved. The disadvantage of HNFA is that the approximation of the posterior density is farther away from the true posterior density. This may occasionally lead to inferior performance [6]. However, experiments suggest that in many cases the advantages are more important.

2. VARIATIONAL BAYESIAN LEARNING

Variational Bayesian learning techniques are based on approximating the true posterior probability density of the unknown variables of the model by a function with a restricted form. Currently the most common tech-

This research has been funded by the European Commission project BLISS, and the Finnish Center of Excellence Programme (2000–2005) under the project New Information Processing Principles.

nique is ensemble learning [7, 8, 9] where the Kullback-Leibler divergence measures the misfit between the approximation and the true posterior. It has been applied to standard ICA and BSS [10, 11, 12, 13] and to their extensions [14, 15, 5, 9], as well as to a wide variety of other models.

In ensemble learning, the posterior approximation $q(\boldsymbol{\theta})$ of the unknown variables $\boldsymbol{\theta}$ is required to have a suitably factorial form

$$q(\boldsymbol{\theta}) = \prod_i q_i(\boldsymbol{\theta}_i)\,, \tag{2}$$

where $\boldsymbol{\theta}_i$ denotes a subset of the unknown variables. The misfit between the true posterior $p(\boldsymbol{\theta} \mid \mathbf{X})$ and its approximation $q(\boldsymbol{\theta})$ is measured by the Kullback-Leibler divergence. An additional term $-\log p(\mathbf{X})$ is included to avoid calculation of the model evidence term $p(\mathbf{X}) = \int p(\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{\theta}$. The cost function then has the form [7, 8, 9]

$$\mathcal{C} = D(q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta}|\mathbf{X})) - \log p(\mathbf{X}) = \left\langle \log \frac{q(\boldsymbol{\theta})}{p(\mathbf{X}, \boldsymbol{\theta})} \right\rangle , \tag{3}$$

where $\langle\cdot\rangle$ denotes expectation over the distribution $q(\boldsymbol{\theta})$. Note that since $D(q \parallel p) \geq 0$, it follows that the cost function provides a lower bound $p(\mathbf{X}) \geq \exp(-\mathcal{C})$ for the model evidence $p(\mathbf{X})$.

During learning, the factors $q_i(\boldsymbol{\theta}_i)$ are typically updated one at a time while keeping the other factors fixed. In this paper, we apply the method introduced in [1]. The posterior has a maximally factorial form, which means that each unknown variable is approximated to be independent a posteriori of the rest of the variables. The computational complexity of each individual update is then proportional to the number of connections it has with other variables. Consequently, updating the posterior variance of all the variables in the model can be carried out in a time proportional to the total number of connections in the model.

For each update of the posterior approximation $q_i(\theta_i)$, the variable θ_i requires the prior distribution $p(\theta_i \mid \text{parents})$ given by its parents and the likelihood $p(\text{children} \mid \theta_i)$ obtained from its children[1]. The relevant part of the Kullback-Leibler divergence to be minimised is, up to a constant independent of $q_i(\theta_i)$

$$C(q_i(\theta_i)) = \left\langle \ln \frac{q_i(\theta_i)}{p(\theta_i \mid \text{parents}) p(\text{children} \mid \theta_i)} \right\rangle . \tag{4}$$

In ensemble learning, conjugate priors are commonly used because they make it very easy to solve the variational minimisation problem of finding the optimal $q_i(\theta_i)$ which minimises (4).

[1]In a graphical model representation, each variable is conditionally dependent on its parents.

As an example, consider linear mappings with Gaussian variables. First, note that in (4), the negative logarithm of the prior and likelihood is needed. We shall call this quantity the potential. Gaussian prior has a quadratic potential. The likelihood arising from a linear mapping to Gaussian variables also has a quadratic potential. The sum of the potential is quadratic and the optimal posterior approximation can be shown to be the Gaussian distribution whose potential has the same second and first order terms. The minimisation thus boils down to adding the coefficients of the second and first order terms of the prior and likelihood.

3. NONLINEAR FACTOR ANALYSIS

In this and the next section, we discuss how Bayesian ensemble learning can be applied to nonlinear models. Our approach stems from nonlinear factor analysis (NFA) [5], where the nonlinear generative model (1) was estimated by ensemble learning. We briefly introduce NFA, explaining how the computations can be made in linear time and the cost function analytical by splitting the nonlinear mapping into two parts. NFA is in many respects similar to the hierarchical NFA (HNFA) to be discussed in the next section. For instance posterior approximation is chosen to be maximally factorial in both these methods for achieving computational efficiency and both can in principle separate any nonlinear mixtures.

Unlike for linear models, no conjugate priors exist for the sources in the nonlinear model (1). It is therefore impossible in practice to solve the functional form of $q_i(\theta_i)$ by minimising (4). Instead of using such free form approximation [8], the terms $q_i(\theta_i)$ are restricted to be Gaussian in NFA. A justification for this choice is that it is the free-form solution if the mapping $\mathbf{f}(\cdot)$ is linear.

In [5], a multi-layer perceptron (MLP) network with one hidden layer was used for modelling the nonlinear mapping $\mathbf{f}(\cdot)$:

$$\mathbf{f}(\mathbf{s}; \mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}) = \mathbf{A} \tanh[\mathbf{B}\mathbf{s} + \mathbf{b}] + \mathbf{a}\,, \tag{5}$$

where $\mathbf{A}$ and $\mathbf{B}$ are weight matrices, $\mathbf{a}$ and $\mathbf{b}$ are bias vectors and the activation function tanh operates on each element separately.

In NFA, the most time consuming part is the computation of the posterior variance $\mathrm{Var}(\mathbf{f})$ over the approximated posterior distribution $q(\boldsymbol{\theta})$. The estimate of variance is based on a linear approximation of the mapping $\mathbf{f}$ about the posterior mean of the sources and weights. This requires computation of the Jacobian matrix of $\mathbf{f}(\cdot)$ with respect to the sources, having a complexity which is essentially the same as multiplication of the matrices $\mathbf{A}$ and $\mathbf{B}$.

In NFA, neither the posterior mean nor the variance of $\mathbf{f}(\cdot)$ over $q(\boldsymbol{\theta})$ can be computed analytically. The approximation based on the Taylor series expansion can be inaccurate if the posterior variance for the input of the hidden nodes grows too large. This may cause the instability observed in some simulations.

4. HIERARCHICAL NONLINEAR FACTOR ANALYSIS

In [1], modular design principles and building blocks for latent variable models were introduced. The benefits of the approach are that the cost function and learning rules can be derived automatically once the structure of the model is given and learning is stable and computationally efficient. One of the building blocks was a Gaussian variable ξ followed by a nonlinearity ϕ:

$$\phi(\xi) = \exp(-\xi^2)\,. \tag{6}$$

The motivation for choosing this particular nonlinearity is that for Gaussian posterior approximation $q_\xi(\xi)$, the posterior mean and variance and consequently the cost function (3) can be evaluated analytically.

Using this construction—Gaussian variables followed by nonlinearity—it is possible to build nonlinear mappings for which the learning time is linear with respect to the size of the model. The key idea is to introduce latent variables $\mathbf{h}(t)$ before the nonlinearities and thus split the mapping (5) into two parts:

$$\begin{aligned}\mathbf{h}(t) &= \mathbf{B}\mathbf{s}(t) + \mathbf{b} + \mathbf{n}_h(t) && (7)\\ \mathbf{x}(t) &= \mathbf{A}\phi[\mathbf{h}(t)] + \mathbf{C}\mathbf{s}(t) + \mathbf{a} + \mathbf{n}_x(t)\,, && (8)\end{aligned}$$

where $\mathbf{n}_h(t)$ and $\mathbf{n}_x(t)$ are Gaussian noise terms. Note that we have included a short-cut mapping $\mathbf{C}$ from sources to observations. This means that hidden nodes only need to model the deviations from linearity.

In HNFA, the extra latent variables $\mathbf{h}(t)$ are not expected to represent the data independently but to operate on tight guidance of the upper layer[2]. They are included in the model merely to reduce the computational complexity. The algorithm still needs the posterior mean and variance of the mappings in (7) and (8), but now they all have analytic expressions which can be computed in linear time.

According to our posterior approximation $q(\boldsymbol{\theta})$, all variables are independent a posteriori. Introduction of the extra latent variables $\mathbf{h}(t)$ has the negative effect of increasing the misfit between the approximated and the true posterior density. Minimisation of the cost function (3) favours solutions where the misfit is as small as possible. In [6], it is shown how this can lead to suboptimal separation in linear ICA. It is difficult to analyse the situation in linear models mathematically, but it seems that models with fewer hidden nodes and thus more linear mappings are favoured. This should lead to conservative estimates of the nonlinearity of the model.

On the other hand, introducing $\mathbf{h}(t)$ has major benefits: computational complexity of learning is linear and convergence is stable because the cost function to be minimised is not based on series approximations.

5. LEARNING SCHEME

The learning scheme is designed to minimise the cost function (3). The basic operation during learning is an iteration where all the terms $q_i(\theta_i)$ of $q(\boldsymbol{\theta})$ are updated one at a time by minimising (4). In addition, several other operations are performed:

- addition of hidden nodes;
- addition of weights;
- pruning of weights; and
- line search.

Line search has been explained in [16]. The idea is to monitor the individual updates during one iteration and then perform a line search simultaneously for all $q_i(\theta_i)$. We applied the line search after every tenth iteration.

The addition and pruning operations aim at optimising the model structure. The cost function (3) relates to the model evidence $p(\mathbf{X} \mid \text{model})$ which can be used to find the most likely model structure.

In general, addition takes place randomly and pruning is based on estimating whether the cost function can be decreased by removing a weight. The motivation for this is that ensemble learning can effectively prune out parts of the model which are not needed. The weights in the matrix $\mathbf{B}$ corresponding to one hidden node can for instance approach zero. The cost function can usually be decreased by removing such weights. If all outgoing weights of a hidden node have been removed, the hidden node becomes useless and can be removed from the model. Ensemble learning cannot, however, actively make room for a part of the model which may be added in the future. It usually takes some time for the rest of the model to accommodate to additions.

[2] A hierarchical model where the middle layer had a more independent role in representing the observations was presented in [1].

5.1. Evidence node

During learning, it is necessary to initialise some variables and keep them fixed for a while until other parts of the model have accommodated appropriately. We use evidence nodes, as we call them. They are attached to a variable θ_i, whose value we want to set, and provide a term for the likelihood $p(\text{children} \mid \theta_i)$. When $q_i(\theta_i)$ is updated, θ_i will be close to the value set by evidence node if the likelihood term has a narrow peak but θ_i can accommodate to other parts of the model if the likelihood term is wide. After each iteration, the extra term for the likelihood is decayed a little on the logarithmic scale, and the evidence node is removed when the extra term vanishes. The persistance of the initialisation can be controlled by the life-span of the evidence node.

5.2. Phases of learning

The model is built in stages. First, only the linear mapping $\mathbf{Cs}(t)$ is used, so that there are no hidden nodes. The sources $\mathbf{s}(t)$ are initialised by principal component analysis (PCA) using evidence nodes with a life-span of 40 iterations in order to estimate a reasonable $\mathbf{C}$. The linear model is learned for 100 iterations.

After that, 50 randomly initialised hidden nodes are added to the model and estimation of the model structure begins. That is, weights are added and pruned and hidden nodes are added every now and then. Every time new hidden nodes are added, five of them are selected from a pool of 1,000 random candidates. After each addition of hidden nodes, there is a period of 30 iterations during which no pruning is applied. This gives the new hidden nodes enough time to fit themselves into the model.

Hidden nodes are added a limited number of times. After that, learning continues with pruning and random additions of weights. The number of weights to be added decreases with time. Finally, only line searches are applied for the last 1,000 iterations. The total number of iteratons in the simulations is 10,000 unless otherwise stated.

5.3. Addition of hidden nodes

The hidden nodes are latent variables which can independently represent some aspects of the observations. Due to our model structure, this usually corresponds to a local minimum of the cost function. It is better that the sources $\mathbf{s}(t)$ represent the data since they can share their information with all hidden nodes. The local minimum can be avoided by evidence nodes which keep the variance of the Gaussian noise $n_{hi}(t)$ of each newly added hidden node $h_i(t)$ low. Once the sources take the responsibility for the representation, the variances of hidden nodes no longer grow significantly. The life-span of these evidence nodes was 500 iterations in our experiments.

When hidden nodes are added, their incoming weights $\mathbf{B}$ are initialised to random values. However, after the first addition, the added hidden nodes are selected from a large pool of candidate initialisations. The hidden nodes which correlate best with the remaining modelling error of the observations are selected. The first hidden nodes are able to model many of the large scale nonlinearities in the data. It is much more difficult to find useful hidden nodes by random initialisations later on since the new neurons tend to be quickly pruned away.

6. EXPERIMENTS

In this section, we report experiments with an artificial data set which demonstrate the ability of the HNFA method to extract a nonlinear subspace. We show that it is possible to estimate the correct number of sources. The value of the cost function correlates well with the quality of the subspace. Linear ICA is used for rotating the extracted subspace to separate the independent source signals.

Our data set closely resembles the one used in [5, 3]. The data set consists of 1,000 samples from nonlinear mixtures of eight sources. Four of the sources are super-Gaussian and the remaining four are sub-Gaussian. We used the same nonlinear mixing as in [5, 3] where the nonlinear mapping was a randomly initialised MLP network with $\sinh^{-1}$ as the activation function for hidden nodes. Now we also included additive Gaussian noise whose standard deviation was one tenth of that of the signal. This corresponds to a signal-to-noise ratio (SNR) of 20 dB.

6.1. Model selection

We tested both linear models, which were otherwise like HNFA models but lacked the nonlinear hidden nodes, and HNFA models with varying number of sources. The values of the cost function attained after learning were compared. The best linear model had 14 sources while the best HNFA model had eight sources. Hence the HNFA method is able to infer the correct subspace dimension, while the linear model tries to model nonlinear effects using extra dimensions.

6.2. Subspace estimation

The HNFA method as such cannot find the original sources for the same reason as PCA cannot find inde-

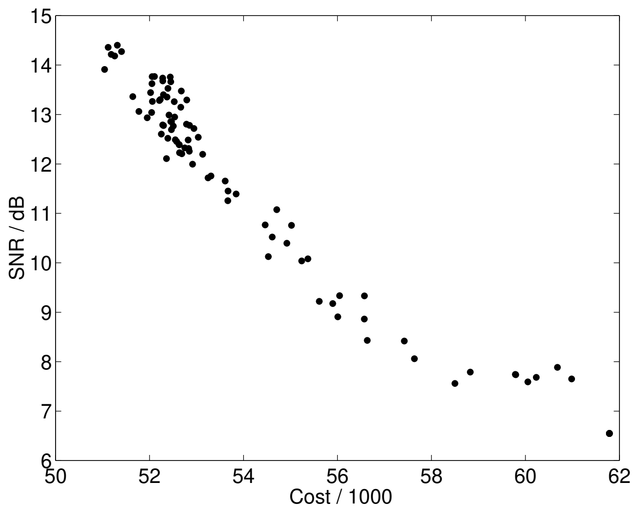

Fig. 1. The signal-to-noise ratio of the estimated subspace as a function of the cost function value attained in different simulations.

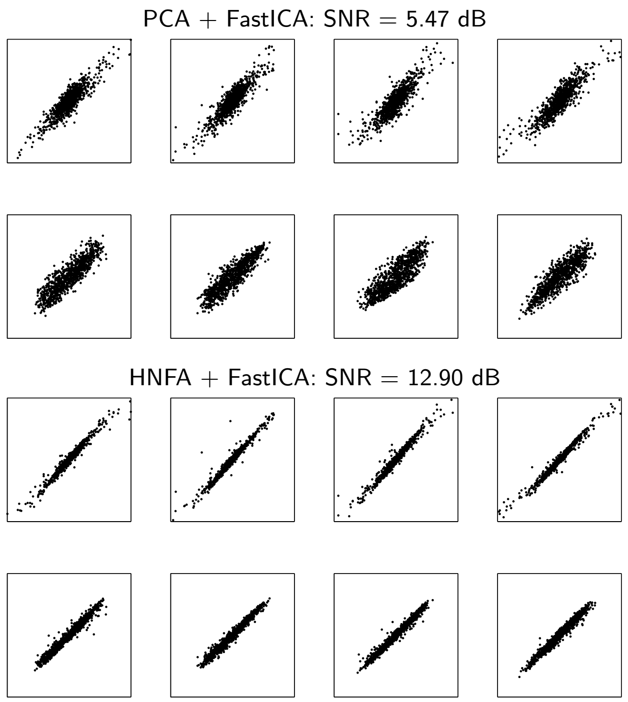

Fig. 2. Each scatter plot shows the values of one original source signal plotted against the best corresponding estimated source signal after a rotation with FastICA.

pendent components: the Gaussian source model has a rotational indeterminacy. We can, however, measure the quality of the estimated nonlinear subspace by measuring how accurately the original sources can be reconstructed as linear combinations of the estimated sources. We refer to this as the SNR of the optimal reconstruction.

The eight-dimensional linear subspace extracted by linear PCA yielded an SNR of 7.63 dB for optimal reconstruction while the HNFA simulation which reached the lowest value for the cost function attained an SNR of 13.91 for optimal reconstruction.

An interesting question is whether the value of the cost function (3) correlates with the quality of the estimated nonlinear subspace. Figure 1 shows how the SNR of the optimal reconstruction depends on the cost function reached by the simulations with eight sources. We deliberately generated some models which had fairly few hidden nodes to reach costs between 53,000 and 61,000. The linear model is shown, too, and its cost is slightly below 62,000. The figure clearly shows that the cost function is a reliable indicator of the quality of the subspace.

6.3. Rotation

After a nonlinear subspace has been estimated by HNFA, we can use standard linear ICA algorithms [3] for rotating the subspace to obtain independent source signals. As in [5], we used the FastICA method [17, 3], but this time its symmetric version instead of the deflation one. We found that symmetric FastICA method always converged to the same solution regardless of the initialisation when the quality of the subspace was adequate.

As expected, we found that the SNR of the sources provided by FastICA is closely correlated with the SNR of the optimal reconstruction, being typically about 1 dB lower. Figure 2 shows the scatter plots of the original sources and the sources obtained after a rotation by FastICA. On the first two rows, standard linear PCA has been used to estimate the subspace. The lower two rows show the results when the subspace was estimated using the HNFA method.

Linear PCA yielded an average SNR of 4.32 dB with FastICA, which is less than expected from the SNR of the optimal reconstruction. The reason is that although most runs of FastICA yielded an SNR close to 5.47 dB, there seems to be another local minimum around 1.8 dB.

6.4. Comparison to NFA

With our data set, the NFA method achieves significantly smaller values of the cost function and better SNRs. The number of iterations needed by NFA was much larges because we did not use line searches, but in 100,000 iterations we reached a cost of 36,368 and an SNR of 22.58 for optimal reconstruction.

However, we have a reason to believe that this is because the data is unusually well suited for NFA. Although both methods can in principle model any nonlinear mixing, the $\sinh^{-1}$ activation function matches better the tanh activation function in NFA than the nonlinearity (6). Our experiments with speech data suggest that with real-world data sets, the HNFA method can outperform NFA. This is probably due to enhanced structural learning in HNFA. Otherwise, one can expect the NFA method to reach lower values of the cost function because its posterior approximation $q(\boldsymbol{\theta})$ should fit the true posterior better.

7. DISCUSSION

The present version of HNFA is already a ready-to-use tool for nonlinear subspace estimation, but it is still possible to improve and automatise the structural estimation procedure. Now we estimated the number of sources by manually going through different models. It would be possible to estimate the number of sources in a similar manner as the number of hidden nodes was estimated. Moreover, pruning of hidden nodes should be improved. Now only weights were pruned, and sometimes a hidden node is left with some outgoing weights although the cost would decrease by removing the whole hidden node.

An important line of research will be the modelling of dynamics. In [9], the NFA method was extended to include a model for the dynamics of the sources. A similar extension for HNFA would lead to hierarchical nonlinear dynamical factor analysis.

To conclude, HNFA is a powerful method for estimating nonlinear subspaces. The cost function provides a lower bound for evidence and can be reliably used to estimate the dimension of the subspace. The HNFA method provides a nonlinear extension of standard linear PCA, which is often used as the whitening stage in linear ICA algorithms [3]. We have used FastICA for the final rotation of the sources. We showed that the cost function used in the HNFA method correlates very well with the quality of the estimated subspace.

8. REFERENCES

[1] H. Valpola, T. Raiko, and J. Karhunen, "Building blocks for hierarchical latent variable models," in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 710–715, 2001.

[2] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, 1999.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. J. Wiley, 2001.

[4] C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in *Proc. ICA2003*, 2003. Invited paper in the special session on nonlinear ICA and BSS.

[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] A. Ilin and H. Valpola, "On the effect of the form of the posterior approximation in variational learning of ICA models," in *Proc. ICA2003*, 2003. Submitted.

[7] D. Barber and C. Bishop, "Ensemble learning in Bayesian neural networks," in *Neural Networks and Machine Learning* (M. Jordan, M. Kearns, and S. Solla, eds.), pp. 215–237, Berlin: Springer, 1998.

[8] H. Lappalainen and J. Miskin, "Ensemble learning," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 75–92, Berlin: Springer-Verlag, 2000.

[9] H. Valpola and J. Karhunen, "An unsupervised ensemble learning method for nonlinear dynamic state-space models," *Neural Computation*, vol. 14, no. 11, pp. 2647–2692, 2002.

[10] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.

[11] H. Lappalainen, "Ensemble learning for independent component analysis," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, (Aussois, France), pp. 7–12, 1999.

[12] J. Miskin and D. MacKay, "Ensemble learning for blind image separation and deconvolution," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 123–141, Springer-Verlag, 2000.

[13] R. Choudrey, W. Penny, and S. Roberts, "An ensemble learning approach to independent component analysis," in *Proc. of the IEEE Workshop on Neural Networks for Signal Processing, Sydney, Australia, December 2000*, IEEE Press, 2000.

[14] H. Attias, "ICA, graphical models and variational methods," in *Independent Component Analysis: Principles and Practice* (S. Roberts and R. Everson, eds.), pp. 95–112, Cambridge University Press, 2001.

[15] K. Chan, T.-W. Lee, and T. Sejnowski, "Variational learning of clusters of undercomplete nonsymmetric independent components," in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 492–497, 2001.

[16] A. Honkela, "Speeding up cyclic update schemes by pattern searches," in *Proc. of the 9th Int. Conf. on Neural Information Processing (ICONIP'02)*, (Singapore), pp. 512–516, 2002.

[17] "The FastICA MATLAB package." Available at `http://www.cis.hut.fi/projects/ica/fastica/`, 1998.