ON-LINE VARIATIONAL BAYESIAN LEARNING

Antti Honkela and Harri Valpola

Helsinki University of Technology, Neural Networks Research Centre
P.O. Box 5400, FIN-02015 HUT, Espoo, Finland
{Antti.Honkela, Harri.Valpola}@hut.fi http://www.cis.hut.fi/projects/ica/bayes/

ABSTRACT

Variational Bayesian learning is an approximation to the exact Bayesian learning where the true posterior is approximated with a simpler distribution. In this paper we present an on-line variant of variational Bayesian learning. The method is based on collecting likelihood information as the training samples are processed one at a time and decaying the old likelihood information. The decay or forgetting is very important since otherwise the system would get stuck to the first reasonable solution it finds. The method is tested with a simple linear independent component analysis (ICA) problem but it can easily be applied to other more difficult problems.

1. INTRODUCTION

Variational Bayesian learning is an approximation to exact Bayesian learning. It is based on approximating the posterior distribution with another simpler distribution. The approximation makes the learning problem tractable while preserving many of the benefits of Bayesian learning such as resistance to over-fitting.

A variational method called ensemble learning has gained popularity during recent years. It has been successfully applied to various linear independent component analysis (ICA) problems [2, 10, 3, 14, 17] as well as nonlinear ICA [11] and nonlinear and switching state-space models [20, 6], to name a few examples. The benefits of ensemble learning include easy comparison of different models that can be used e.g. for determining the correct number of sources in an ICA problem, and easy incorporation of prior knowledge such as positivity constraints. In [22] a complete framework based on variational Bayesian learning for building many kinds of models is presented.

The previous algorithms using variational Bayesian learning operate in *batch* mode, i.e. they use the whole data set as a one large block. This can be a serious limitation as storing a large data set requires a lot of memory. It may also be desirable to be able to process the data in parallel with its collection which is difficult with batch algorithms. One way to deal with these problems is to use an *on-line* algorithm that processes the data one sample at a time. Unfortunately the straightforward Bayesian on-line learning method of taking the posterior after previous sample as a prior for the new sample suffers from the same problems as exact Bayesian learning in general and is computationally intractable. A simple approximation method of using the variational approximation to the posterior at each step would not work very well as the approximation is valid only locally and the learning would therefore get stuck to the first decent solution.

In this paper, we present a novel method for performing variational Bayesian learning in on-line mode.[1] Our method is based on maintaining a decaying history of the previous samples processed by the model. This way the updates are based on a longer history although the algorithm processes the samples one at a time. The decay ensures that the system has a chance to forget old solutions in favour of new better ones. The method is also efficient because the whole history can be compressed into a few aggregate statistics thus allowing the desired space savings. The method has been developed as an extension to the building block framework [22] and can thus easily be used for many different models, such as the ones presented in [19, 21]. Additionally the method could be applied to other previous models such as [2, 10, 3, 14, 17, 11, 20].

The paper is organised as follows. First we introduce the basic batch form of variational Bayesian learning. In Section 3, possible approaches to on-line Bayesian learning are discussed. Our algorithm is presented in Section 4. The results of a simple ICA experiment are presented in Section 5.

[1] While revising the paper we became aware that essentially the same method has been proposed earlier in [18].

2. VARIATIONAL BAYESIAN LEARNING

Denote by $\boldsymbol{\theta} = \{\theta_i|i\}$ the set of model parameters and by $\boldsymbol{S} = \{s_i(t)|t,i\}$ the set of source values that we wish to estimate from a given data set $\boldsymbol{X} = \{x_i(t)|t,i\}$. Here the values of $\boldsymbol{X}$ at a given time instant depend only on the corresponding values of $\boldsymbol{S}$ whereas they all depend on $\boldsymbol{\theta}$ in the similar manner. To make the derivations easier we assume that there are no temporal dependencies within $\boldsymbol{S}$. In Bayesian estimation methods, it is assumed that there is some prior information on $\boldsymbol{\theta}$ and $\boldsymbol{S}$ available. This is represented in the form of prior distribution $p(\boldsymbol{\theta}, \boldsymbol{S})$ of $\boldsymbol{\theta}$ and $\boldsymbol{S}$. After learning, all the information of the parameters is contained in the posterior probability density $p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X})$ of the parameters given the data $\boldsymbol{X}$. It can be computed from the Bayes' rule

$$p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{S})p(\boldsymbol{\theta}, \boldsymbol{S})}{p(\boldsymbol{X})}. \tag{1}$$

Here $p(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{S})$ is the likelihood of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{S}$, and $p(\boldsymbol{X})$ is a normalizing constant which can be evaluated if necessary by integrating the numerator $p(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{S})p(\boldsymbol{\theta}, \boldsymbol{S})$ over all the possible values of $\boldsymbol{\theta}$ and $\boldsymbol{S}$.

Because integration over parameter space that is needed to evaluate the normalising constant and to use the posterior otherwise is difficult, using the exact posterior is typically intractable, and some approximations are needed. The key idea of variational methods is to approximate the exact posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X})$ by another distribution $q(\boldsymbol{\theta}, \boldsymbol{S})$ that is computationally easier to handle [9]. The approximating distribution is usually chosen to be a product of several independent distributions, one for each parameter or a set of similar parameters. We use a particular variational method known as ensemble learning that has recently become very popular [8, 13, 12]. An example of a variational technique other than ensemble learning can be found in [7].

In ensemble learning the optimal approximating distribution is found by minimizing the Kullback-Leibler divergence between the approximate and true posterior. After some considerations [12, 13], this leads to the cost function

$$\begin{aligned}\mathcal{C} &= \left\langle \log \frac{q(\boldsymbol{\theta}, \boldsymbol{S})}{p(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{S})} \right\rangle_{q(\boldsymbol{\theta}, \boldsymbol{S})} \\ &= \langle \log q(\boldsymbol{\theta}, \boldsymbol{S}) \rangle_{q(\boldsymbol{\theta}, \boldsymbol{S})} - \langle \log p(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{S}) \rangle_{q(\boldsymbol{\theta}, \boldsymbol{S})}.\end{aligned} \tag{2}$$

Here $\langle \cdot \rangle_q$ denotes the expectation over distribution q. From now on all such expectations are taken over $q(\boldsymbol{\theta}, \boldsymbol{S})$ unless mentioned otherwise. With typical factorial approximation $q(\boldsymbol{\theta}, \boldsymbol{S}) = \prod_i q(\theta_i) \prod_{t,i} q(s_i(t))$ and i.i.d. noise, this simplifies into

$$\begin{aligned}\mathcal{C} = &\sum_i \langle \log q(\theta_i) \rangle - \langle \log p(\boldsymbol{\theta}) \rangle \\ &+ \sum_{t,i} \left(\langle \log q(s_i(t)) \rangle - \langle \log p(s_i(t)|\boldsymbol{\theta}) \rangle \right) \\ &- \sum_{t,i} \langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S}) \rangle .\end{aligned} \tag{3}$$

The cost function in Eq. (2) has the convenient property that

$$\mathcal{C} = D(q(\boldsymbol{\theta}, \boldsymbol{S})||p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X})) - \log p(\boldsymbol{X}) \geq - \log p(\boldsymbol{X}), \tag{4}$$

where $D(q||p)$ denotes the Kullback-Leibler divergence between distributions q and p. This way the cost $\mathcal{C}$ yields a lower bound to the model evidence, an important quantity in model comparison.

3. POSSIBLE APPROACHES TO BAYESIAN ON-LINE LEARNING

As noted at the beginning of Sec. 2, Bayesian learning is based on the idea of using Bayes rule to update the prior distribution $p(\boldsymbol{\theta}, \boldsymbol{S})$ to posterior $p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X})$. The same principle can be used iteratively to update the distribution taking into account the information given by each individual sample.

Let us denote by $\boldsymbol{X}_{1:T} = \{x_i(t)|t = 1, \ldots, T; i\}$ the set of observations on time interval from 1 to T similarly for $\boldsymbol{S}_{1:T}$. Let also $\boldsymbol{X}_t = \boldsymbol{X}_{t:t}$ and similarly $\boldsymbol{S}_t = \boldsymbol{S}_{t:t}$. For simplicity we assume that $p(\boldsymbol{X}_t|\boldsymbol{\theta}, \boldsymbol{S}) = p(\boldsymbol{X}_t|\boldsymbol{\theta}, \boldsymbol{S}_t)$, i.e. the observations $\boldsymbol{X}_t$ only depend on the source values $\boldsymbol{S}_t$ at the same time instant. The iterative Bayesian learning can now be represented by using the Bayes rule to derive the new posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{S}_{1:T+1}|\boldsymbol{X}_{1:T+1})$ from the old $p(\boldsymbol{\theta}, \boldsymbol{S}_{1:T}|\boldsymbol{X}_{1:T})$ as

$$\begin{aligned}&p(\boldsymbol{\theta}, \boldsymbol{S}_{1:T+1}|\boldsymbol{X}_{1:T+1}) = \\ &\qquad \frac{p(\boldsymbol{X}_{T+1}|\boldsymbol{\theta}, \boldsymbol{S}_{T+1})p(\boldsymbol{\theta}, \boldsymbol{S}_{1:T}|\boldsymbol{X}_{1:T})}{p(\boldsymbol{X}_{T+1}|\boldsymbol{X}_{1:T})}.\end{aligned} \tag{5}$$

Unfortunately this approach suffers from the same problems as the exact Bayesian learning in general. Even representing the posterior depending explicitly on all the data can be very difficult if it is for instance a mixture density of exponentially growing number of components.

The approaches and approximations used to avoid the problems of exact Bayesian on-line learning are the same as in general Bayesian learning. Stochastic approximation approaches to on-line learning include sequential Monte Carlo methods and particle filtering [4].

In this paper we concentrate on variational approximation instead of the stochastic approximations.

The simplest variational approximation would be to simply use the standard factorial approximation at each step so that the posterior from the previous step becomes the new prior and so on. This approach would, however, lead to problems as the approximation is valid only locally and information of possible good solutions further away is lost. The factorial approximation is especially susceptible as all the correlations between variables are lost. In ICA this would effectively fix the rotation of the sources to the initial value or very close to it.

One way to partially deal with the problem is to approximate the posterior at each step with the full multivariate Gaussian distribution as in [16]. This will preserve some of the correlations but in large models the computational cost of estimating the full covariance is high. Moreover, even the Gaussian approximation with a full covariance matrix will be invalid outside the neighborhood of the current solution. The approach may also lose information of the prior that does not fit the Gaussian approximation.

4. ON-LINE VARIATIONAL LEARNING

Our approach is basically the same as the one presented by Ghahramani in [5] except that it incorporates forgetting of old samples. The prior is, however, not forgotten, only the likelihood information of the old data. The forgetting makes learning with even relatively small fixed data set practical as the same samples can be used again. With the basic approach, the same samples can only be used once.

4.1. The cost function

The on-line version of variational Bayesian learning is based on collecting the likelihood information gradually as the samples are processed one at a time. This approach works if the expectations $\langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S}_t)\rangle$ of the log-likelihood and $\langle \log p(\boldsymbol{S}_t|\boldsymbol{\theta})\rangle$ of the prior of the sources over the approximate posterior $q(\boldsymbol{\theta}, \boldsymbol{S}_t)$ can be expressed in form

$$\langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S}_t)\rangle = \sum_{k=1}^{n} \alpha_k(x_i(t), \boldsymbol{S}_t) f_k(\boldsymbol{\theta}) \qquad (6)$$

$$\langle \log p(\boldsymbol{S}_t|\boldsymbol{\theta})\rangle = \sum_{l=1}^{m} \beta_l(\boldsymbol{S}_t) g_l(\boldsymbol{\theta}), \qquad (7)$$

where $\alpha_k, k = 1, \ldots, n$ and $\beta_l, l = 1, \ldots, m$ are constants depending on the observation $x_i(t)$ and sources $\boldsymbol{S}_t$ while $f_k(\boldsymbol{\theta}), k = 1, \ldots, n$ and $g_l(\boldsymbol{\theta}), l = 1, \ldots, n$ are fixed functions of $\boldsymbol{\theta}$, i.e. they do not depend on the observations $x_i(t)$ and the source values $\boldsymbol{S}_t$.

As an example a Gaussian variable $x(t) \sim N(a \cdot s(t), \sigma^2)$ with a linear model for mean and a factorial posterior approximation $q(s(t), a, \sigma^2) = q(s(t))q(a, \sigma^2)$, has such a decomposition that can be written as

$$\left\langle \log p(x(t)|a, s(t), \sigma^2) \right\rangle = -\frac{1}{2} \left\langle \log(2\pi\sigma^2) \right\rangle \\ -\frac{1}{2} \left\langle s(t)^2 \right\rangle \left\langle \frac{a^2}{\sigma^2} \right\rangle - \frac{1}{2} x(t)^2 \left\langle \frac{1}{\sigma^2} \right\rangle + x(t) \left\langle s(t) \right\rangle \left\langle \frac{a}{\sigma^2} \right\rangle . \qquad (8)$$

The essentially same decomposition works even if σ^2 has a more complicated model or is time-dependent.

Assuming the expectation of the likelihood term has a decomposition of the form shown in Eq. (6), the term of the cost function arising from the likelihood (the last term in Eq. (3)) can be written as

$$-\sum_{t,i} \langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S}_t)\rangle \\ = -\sum_{k=1}^{n} f_k(\boldsymbol{\theta}) \sum_{t,i} \alpha_k(x_i(t), \boldsymbol{S}_t). \qquad (9)$$

The latter part of the second term involving $p(\boldsymbol{S}_t|\boldsymbol{\theta})$ can be similarly written as

$$-\sum_{t} \langle \log p(\boldsymbol{S}_t|\boldsymbol{\theta})\rangle = -\sum_{l=1}^{n} g_l(\boldsymbol{\theta}) \sum_{t} \beta_l(\boldsymbol{S}_t). \qquad (10)$$

Adding a new observation is now very easy as it only affects the innermost sums on Eqs. (9) and (10) to which new terms corresponding to the new observation and the new sources must be added.

4.2. Learning procedure

The basic outline of the learning procedure in an online ICA algorithm is as follows. The data is processed one sample at a time so that only the source values corresponding to the current sample are updated. After moving on to the next sample, the old source values are no longer changed. The mixing matrix and possible other time-independent parameters are of course updated throughout the whole learning process.

The standard ensemble learning proceeds by minimising the cost function with respect to different variables or sets of variables so that the others are kept fixed while one is updated. All the variables are updated cyclically. The algorithm bears close resemblance to the EM algorithm which can actually be seen as a minimisation of very similar cost function as the one in Eq. (2) [15]. Because of this many authors call the variational procedure described here variational EM algorithm.

In order to take into account the on-line nature of our approach, we must divert slightly from the standard alternating updates of the EM algorithm. Contrary to standard EM, a typical model used in variational Bayesian learning has more than two independent groups of variables that are updated separately. This allows many new update strategies as there are many possible orders in which the variables can be updated.

In our method we update the time-independent parameters, i.e. the mixing matrix, noise distribution and similar parameters only once for each processed sample whereas the sources and other time-dependent parameters are updated several times. This is done because there are several groups of time-dependent parameters and it is not possible to find the optimal values for all of them in a single step. It is clear that the bad initial values of the sources do not give much relevant information that could be utilised in updating the mixing matrix so the mixing matrix is only updated after the current source values have more or less converged.

4.3. Forgetting

A typical posterior approximation used in variational Bayesian learning ignores dependencies between different variables. This restricts its validity to the immediate neighborhood of the current operating point. In a typical ICA example, the posterior approximation does not show that rotating the mixing matrix and the sources in a corresponding way mostly preserves the solution. From the approximation, it typically seems that changing the mixing matrix by a relatively large amount is always a bad idea and changing the sources correspondingly only makes things worse.

In on-line learning for ICA, only the sources corresponding to the current time index are updated at a time while the past sources always remain the same. These past values effectively fix the rotation to a given value with no room for change. Introducing forgetting to the method helps avoid this problem as the sources corresponding to the old rotation are gradually forgotten and the learning can proceed to find a better one.

The forgetting is implemented by decaying the likelihood terms that correspond to observations from the past. Thus $p(\boldsymbol{X}_{t_0}|\boldsymbol{\theta}, \boldsymbol{S}_{t_0})$ is replaced at time step $t > t_0$ by $p(\boldsymbol{X}_{t_0}|\boldsymbol{\theta}, \boldsymbol{S}_{t_0})^{d(t,t_0)}$, where $0 < d(t, t_0) \leq 1$. The priors for old observations $p(\boldsymbol{S}_{t_0}|\boldsymbol{\theta})$ are handled similarly. The value $d(t, t_0) = 1$ corresponds to unchanged likelihood whereas $d(t, t_0) = 0$ corresponds to a totally flat distribution with no information content at all. This approach is somewhat similar to the deterministic annealing used in [6] except that in our case it works backwards, starting from the annealed state (low temperature) and then gradually increasing the temperature to forget the exact value.

Let us assume that for all t, $d(t, t) = 1$. The decay of the likelihood affects the cost function in the last term of Eq. (3). Considering Eq. (9), the term becomes

$$\begin{aligned} -\sum_{t=1}^{T}\sum_{i} &\left\langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S})^{d(T,t)} \right\rangle \\ &= -\sum_{t=1}^{T} d(T,t) \sum_{i} \langle \log p(x_i(t)|\boldsymbol{\theta}, \boldsymbol{S})\rangle \\ &= -\sum_{k=1}^{n} f_k(\boldsymbol{\theta}) \sum_{t=1}^{T} d(T,t) \sum_{i} \alpha_k(x_i(t), \boldsymbol{S}_t). \quad (11) \end{aligned}$$

The corresponding modification to Eq. (10) yields

$$\begin{aligned} -\sum_{t=1}^{T} &\left\langle \log p(\boldsymbol{S}_t|\boldsymbol{\theta})^{d(T,t)} \right\rangle \\ &= -\sum_{l=1}^{n} g_l(\boldsymbol{\theta}) \sum_{t=1}^{T} d(T,t)\beta_l(\boldsymbol{S}_t). \quad (12) \end{aligned}$$

In order to keep the computations simple, we require that there exists a sequence of numbers c_i, $i = 1, 2, \ldots$ such that

$$d(T,t) = \prod_{i=t+1}^{T} c_i. \quad (13)$$

Denoting the innermost sum from Eq. (12) by $k(T, l) = \sum_{t=1}^{T} d(T, t)\beta_l(\boldsymbol{S}_t)$, there is now a simple update rule

$$k(T+1, l) = c_{T+1}k(T, l) + \beta_l(\boldsymbol{S}_{T+1}), \quad (14)$$

i.e. the old value of the sum is multiplied by the decay constant c_{T+1} and a new term corresponding to the current observation is added. The sum in Eq. (11) is handled similarly.

4.4. Practical issues of forgetting

Choosing the details of the forgetting procedure is very important for good results and performance. With too fast forgetting the results will be bad as very little data is used. Too little forgetting will hinder performance and may even stop learning to a suboptimal solution.

As noted in Eq. (13), the forgetting procedure is completely specified by the decay constant sequence c_i, $i = 1, 2, \ldots$. In order to study the properties of the forgetting, it is more convenient to use instead a related quantity of *effective memory length*

$$L(t) = \sum_{i=1}^{t-1} d(t,i). \tag{15}$$

The effective memory length measures the effective sample size that is seen by the learning algorithm. Specifying the effective memory lengths also uniquely determines the decay constants as $c_t = L(t)/\left(L(t-1)+1\right)$. In our experiments we have used the memory length $L_{\mathrm{ramp}}(t) = \min(t, L_{\mathrm{limit}})$.

5. EXPERIMENTS

In this section we present the results a simple preliminary experiment of on-line learning with a linear ICA model. The model had sources with time-varying variance (variance neurons) that defined a super-Gaussian source distribution [22, 19]. The same approach could relatively easily be used for more complicated models such as the ones presented in [19, 21].

In the experiment we compared on-line learning with and without forgetting. The data set used in the experiment was artificially created by mixing 4 super-Gaussian sources to 10 mixtures with some additive noise. The sources were generated from Gaussian random samples be applying the hyperbolic sine function to the values. The artificial data set was the only reasonable choice for the experiment because of the huge amount of data needed. The model was initialised to a solution of batch ICA algorithm with a small fraction of the data set.

The results of the experiment are presented in Fig. 1. The results are reported according to the rejection index suggested by Amari et al. [1]. They show that without forgetting the learning gets stuck rather quickly and there is practically no improvement in the separation. The result of the version with forgetting improves steadily until convergence. The final result is comparable to the result of a batch algorithm on a data set of same size as the effective memory length of 500 samples of the on-line algorithm.

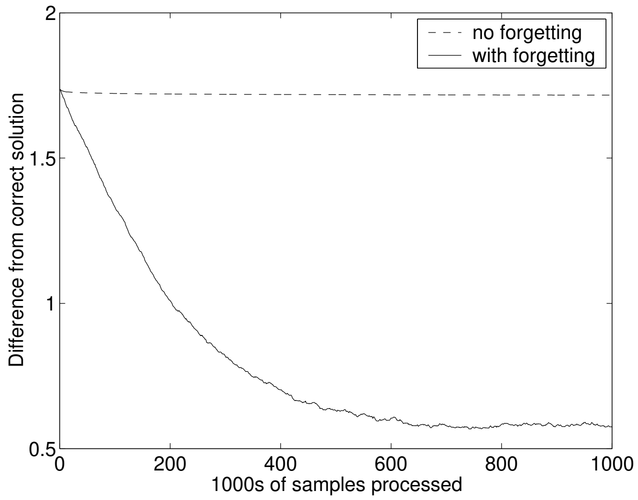

Fig. 1. Performance of on-line learning with and without forgetting with respect to number of samples.

6. DISCUSSION

Many practical unsupervised learning problems require a large amount of memory to store the full data set and parameters of the model when batch algorithms are used. On-line learning helps by requiring only one data sample and corresponding sources to be available at a time. The lower memory requirement often comes with a price of more iterations and computation time needed for the same result a batch algorithm would attain, if such an algorithm is applicable to the problem. In situations where the processing must be done in parallel with data collection the on-line approach is the only viable choice. It also provides a natural way to handle non-stationary problems.

The one-sample-at-a-time approach presented here is one extreme on a scale whose other end would be traditional batch learning. It gives the largest memory savings but also largest difficulties. For most practical learning problems, a suitable compromise of using many smaller batches would probably yield better overall result than either extreme.

As demonstrated by the experiment, some form of forgetting is necessary for the on-line approach to work. The optimal approach would be to use exact Bayesian inference to update the posterior of the parameters as new samples are observed but unfortunately this is computationally intractable. Our method provides one computationally tractable method of handling the problem, but further study is needed on the practical details.

It should be stressed that the experiment reported in this paper is only a very preliminary one. The same method can easily be applied to various variational linear ICA methods [2, 10, 3, 14, 17] as well as nonlinear and other extensions [22, 19, 21, 11, 20].

Acknowledgements

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.

7. REFERENCES

[1] S. Amari, A. Cichocki, and H. Yang. A new learning algorithm for blind source separation. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing 8 (Proc. NIPS'95)*, pages 757–763. MIT Press, Cambridge, MA, 1996.

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

[3] H. Attias. ICA, graphical models and variational methods. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 95–112. Cambridge University Press, 2001.

[4] R. Everson and S. Roberts. Particle filters for non-stationary ICA. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 280–298. Cambridge University Press, 2001.

[5] Z. Ghahramani. Online variational Bayesian learning. Slides from talk presented at NIPS 2000 workshop on Online Learning. Available from `http://www.gatsby.ucl.ac.uk/~zoubin/papers.html`, 2000.

[6] Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):963–996, 2000.

[7] M. Girolami. Variational method for learning sparse and overcomplete representations. *Neural Computation*, 13(11):2517–2532, 2001.

[8] 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*, pages 5–13, Santa Cruz, CA, USA, 1993.

[9] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In M. Jordan, editor, *Learning in Graphical Models*, pages 105–161. The MIT Press, Cambridge, MA, USA, 1999.

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

[11] H. Lappalainen and A. Honkela. Bayesian nonlinear independent component analysis by multi-layer perceptrons. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 93–121. Springer-Verlag, Berlin, 2000.

[12] H. Lappalainen and J. Miskin. Ensemble learning. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 75–92. Springer-Verlag, Berlin, 2000.

[13] 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*, pages 191–198, 1995.

[14] J. Miskin and D. J. C. MacKay. Ensemble Learning for blind source separation. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 209–233. Cambridge University Press, 2001.

[15] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. The MIT Press, Cambridge, MA, USA, 1999.

[16] M. Opper. A Bayesian approach to online learning. In D. Saad, editor, *On-line Learning in Neural Networks*, pages 363–378. Cambridge University Press, 1998.

[17] W. Penny, R. Everson, and S. Roberts. ICA: model order selection and dynamic source models. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 299–314. Cambridge University Press, 2001.

[18] M. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, 2001.

[19] H. Valpola and M. Harva. Hierarchical models of variance sources. In *Proc. ICA2003 (these proceedings)*, 2003.

[20] H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.

[21] H. Valpola, T. Östman, and J. Karhunen. Nonlinear independent factor analysis by hierarchical models. In *Proc. ICA2003 (these proceedings)*, 2003.

[22] 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)*, pages 710–715, San Diego, USA, 2001.