Chapter 2

Bayesian learning of latent variable models

Juha Karhunen, Antti Honkela, Tapani Raiko, Markus Harva, Alexander Ilin, Matti Tornio, Harri Valpola

2.1 Bayesian modeling and variational learning: introduction

Unsupervised learning methods are often based on a generative approach where the goal is to find a latent variable model which explains how the observations were generated. It is assumed that there exist certain latent variables (also called in different contexts source signals, factors, or hidden variables) which have generated the observed data through an unknown mapping. The goal of generative learning is to identify both the latent variables 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 the observations have been generated by a nonlinear process. Unsupervised learning of a nonlinear model is a challenging task, because it is typically computationally much more demanding than for linear models, and flexible models require strong regularization for avoiding overfitting.

In Bayesian data analysis and estimation methods, all the uncertain quantities are modeled in terms of their joint probability distribution. The key principle is to construct the joint posterior distribution for all the unknown quantities in a model, given the data sample. This posterior distribution contains all the relevant information on the parameters to be estimated in parametric models, or the predictions in non-parametric prediction or classification tasks [1, 2].

Denote by $\mathcal{H}$ the particular model under consideration, and by $\boldsymbol{\theta}$ the set of model parameters that we wish to infer from a given data set X. The posterior probability density $p(\boldsymbol{\theta}|X, \mathcal{H})$ of the parameters given the data X and the model $\mathcal{H}$ can be computed from the Bayes' rule

$$p(\boldsymbol{\theta}|X, \mathcal{H}) = \frac{p(X|\boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{p(X|\mathcal{H})} \tag{2.1}$$

Here $p(X|\boldsymbol{\theta}, \mathcal{H})$ is the likelihood of the parameters $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}|\mathcal{H})$ is the prior pdf of the parameters, and $p(X|\mathcal{H})$ is a normalizing constant. The term $\mathcal{H}$ denotes all the assumptions made in defining the model, such as the choice of a particular model class and structure, specific noise model, etc.

The parameters $\boldsymbol{\theta}$ of a particular model $\mathcal{H}_i$ are often estimated by seeking the peak value of a probability distribution. The non-Bayesian maximum likelihood (ML) method uses to this end the distribution $p(X|\boldsymbol{\theta}, \mathcal{H})$ of the data, and the Bayesian maximum a posteriori (MAP) method finds the parameter values that maximize the posterior probability density $p(\boldsymbol{\theta}|X, \mathcal{H})$. However, using point estimates provided by the ML or MAP methods is often problematic, because the model order estimation and overfitting (choosing too complicated a model for the given data) are severe problems [1, 2].

Instead of searching for some point estimates, the correct Bayesian procedure is to use all possible models to evaluate predictions and weight them by the respective posterior probabilities of the models. This means that the predictions will be sensitive to regions where the probability mass is large instead of being sensitive to high values of the probability density [3, 2]. This procedure optimally solves the issues related to the model complexity and choice of a specific model $\mathcal{H}_i$ among several candidates. In practice, however, the differences between the probabilities of candidate model structures are often very large, and hence it is sufficient to select the most probable model and use the estimates or predictions given by it.

A problem with fully Bayesian estimation is that the posterior distribution (2.1) has a highly complicated form except for in the simplest problems. Therefore it is too difficult

to handle exactly, and some approximative method must be used. Variational methods form a class of approximations where the exact posterior is approximated with a simpler distribution [4, 2]. In a method commonly known as *Variational Bayes (VB)* [1, 3, 2] the misfit of the approximation is measured by the Kullback-Leibler (KL) divergence between two probability distributions $q(v)$ and $p(v)$. The KL divergence is defined by

$$D(q \parallel p) = \int q(v) \ln \frac{q(v)}{p(v)} dv \tag{2.2}$$

which measures the difference in the probability mass between the densities $q(v)$ and $p(v)$.

A key idea in the VB method is to minimize the misfit between the actual posterior pdf and its parametric approximation using the KL divergence. The approximating density is often taken a diagonal multivariate Gaussian density, because the computations become then tractable. Even this crude approximation is adequate for finding the region where the mass of the actual posterior density is concentrated. The mean values of the Gaussian approximation provide reasonably good point estimates of the unknown parameters, and the respective variances measure the reliability of these estimates.

A main motivation of using VB is that it avoids overfitting which would be a difficult problem if ML or MAP estimates were used. VB method allows one to select a model having appropriate complexity, making often possible to infer the correct number of latent variables or sources. It has provided good estimation results in the very difficult unsupervised (blind) learning problems that we have considered.

Variational Bayes is closely related to information theoretic approaches which minimize the description length of the data, because the description length is defined to be the negative logarithm of the probability. Minimal description length thus means maximal probability. In the probabilistic framework, we try to find the latent variables or sources and the nonlinear mapping which most probably correspond to the observed data. In the information theoretic framework, this corresponds to finding the latent variables or sources and the mapping that can generate the observed data and have the minimum total complexity. This information theoretic view also provides insights to many aspects of learning and helps to explain several common problems [5].

In the following subsections, we first discuss a natural conjugate gradient algorithm which speeds up learning remarkably compared with alternative variational Bayesian learning algorithms. We then briefly present a practical building block framework that can be used to easily construct new models. This work has been for the most part carried out already before the years 2006-2007 covered in this biennial report. After this we consider the difficult nonlinear blind source separation (BSS) problem using our Bayesian methods. This section has been placed into the Bayes chapter instead of the ICA/BSS because the methods used are all Bayesian. This section is followed by variational Bayesian learning of nonlinear state-space models, which are applied to time series prediction, improving inference of states, and stochastic nonlinear model predictive control. After this we consider an approach for non-negative blind source separation, and then principal component analysis in the case of missing values using both Bayesian and non-Bayesian approaches. We then discuss predictive uncertainly and probabilistic relational models. Finally we present applications of the developed Bayesian methods to astronomical data analysis problems. In most of these topics, variational Bayesian learning is used, but for relational models and estimation of time delays in astronomical applications other Bayesian methods are applied.

2.2 Natural conjugate gradient in variational inference

Variational methods for approximate inference in machine learning often adapt a parametric probability distribution to optimize a given objective function. This view is especially useful when applying variational Bayes (VB) to models outside the conjugate-exponential family. For them, variational Bayesian expectation maximization (VB EM) algorithms are not easily available, and gradient-based methods are often used as alternatives.

In previous machine learning algorithms based on natural gradients [6], the aim has been to use maximum likelihood to directly update the model parameters $\boldsymbol{\theta}$ taking into account the geometry imposed by the predictive distribution for data $p(\boldsymbol{X}|\boldsymbol{\theta})$. The resulting geometry is often very complicated as the effects of different parameters cannot be separated and the Fisher information matrix is relatively dense.

Recently, in [7], we propose using natural gradients for free energy minimisation in variational Bayesian learning using the simpler geometry of the approximating distributions $q(\boldsymbol{\theta}|\boldsymbol{\xi})$. Because the approximations are often chosen to minimize dependencies between different parameters $\boldsymbol{\theta}$, the resulting Fisher information matrix with respect to the variational parameters $\boldsymbol{\xi}$ will be mostly diagonal and hence easy to invert.

While taking into account the structure of the approximation, plain natural gradient in this case ignores the structure of the model and the global geometry of the parameters $\boldsymbol{\theta}$. This can be addressed by using conjugate gradients. Combining the natural gradient search direction with a conjugate gradient method yields our proposed *natural conjugate gradient (NCG)* method, which can also be seen as an approximation to the fully Riemannian conjugate gradient method.

The NCG algorithm was compared against conjugate gradient (CG) and natural gradient (NG) algorithms in learning a nonlinear state-space model [8]. The results for a number of datasets ranging from 200 to 500 samples of 21 dimensional speech spectrograms can be seen in Figure 2.1. The plain CG and NG methods were clearly slower than others and the maximum runtime of 24 hours was reached by most CG and some NG runs. NCG was clearly the fastest algorithm with the older heuristic method of [8] between these extremes. The results with a larger data set are very similar with NCG outperforming all alternatives by a factor of more than 10.

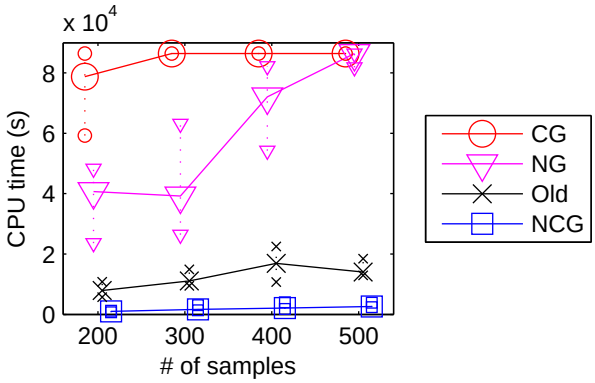

Figure 2.1: Convergence speed of the natural conjugate gradient (NCG), the natural gradient (NG) and the conjugate gradient (CG) methods as well as the heuristic algorithm (Old) with different data sizes. The lines show median times with 25 % and 75 % quantiles shown by the smaller marks. The times were limited to at most 24 hours, which was reached by a number of simulations.

The experiments in [7] show that the natural conjugate gradient method outperforms both conjugate gradient and natural gradient methods by a large margin. Considering univariate Gaussian distributions, the regular gradient is too strong for model variables with small posterior variance and too weak for variables with large posterior variance. The posterior variance of latent variables is often much larger than the posterior variance of model parameters and the natural gradient takes this into account in a very natural manner.

2.3 Building blocks for variational Bayesian learning

In graphical models, there are lots of possibilities to build the model structure that defines the dependencies between the parameters and the data. To be able to manage the variety, we have designed a modular software package using C++/Python called the Bayes Blocks [9]. The theoretical framework on which it is based on was published in [10] and a description of the software package was published in [11].

The design principles for Bayes Blocks have been the following. Firstly, we use standardized building blocks that can be connected rather freely and can be learned with local learning rules, i.e. each block only needs to communicate with its neighbors. Secondly, the system should work with very large scale models. We have made the computational complexity linear with respect to the number of data samples and connections in the model.

The building blocks include Gaussian variables, summation, multiplication, nonlinearity, mixture-of-Gaussians, and rectified Gaussians. Each of the blocks can be a scalar or a vector. Variational Bayesian learning provides a cost function which can be used for updating the variables as well as optimizing the model structure. The derivation of the cost function and learning rules is automatic which means that the user only needs to define the connections between the blocks. Examples of structures which can be build using the Bayes Blocks library can be found in Figure 2.2.

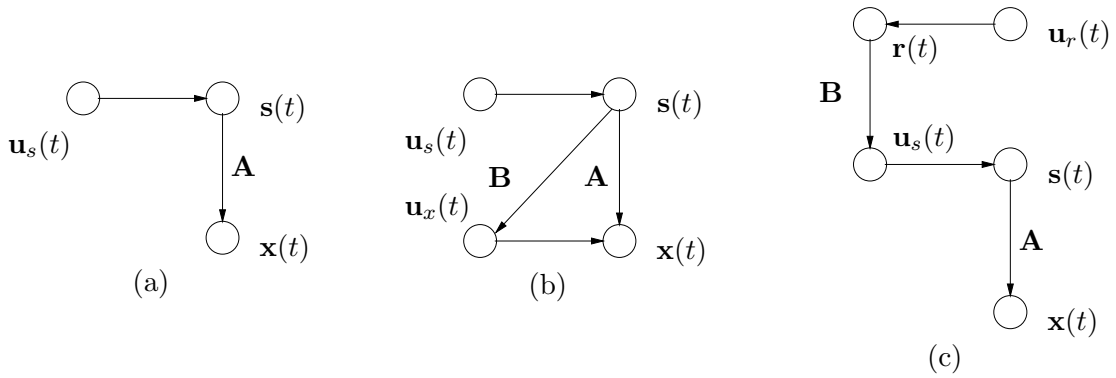

Figure 2.2: Various model structures utilizing variance nodes. Observations are denoted by $\mathbf{x}$, linear mappings by $\mathbf{A}$ and $\mathbf{B}$, sources by $\mathbf{s}$ and $\mathbf{r}$, and variance nodes by $\mathbf{u}$.

2.4 Nonlinear BSS and ICA

A fundamental difficulty in the nonlinear blind source separation (BSS) problem and even more so in the nonlinear independent component analysis (ICA) problem is that they provide non-unique solutions without extra constraints, which are often implemented by using a suitable regularization. Our approach to nonlinear BSS uses Bayesian inference methods for estimating the best statistical parameters, under almost unconstrained models in which priors can be easily added.

We have applied variational Bayesian learning to nonlinear factor analysis (FA) and BSS where the generative mapping from sources to data is not restricted to be linear. The general form of the model is

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_f) + \mathbf{n}(t) \tag{2.3}$$

This can be viewed as a model about how the observations were generated from the sources. The vectors $\mathbf{x}(t)$ are observations at time t, $\mathbf{s}(t)$ are the sources, and $\mathbf{n}(t)$ the noise. The function $\mathbf{f}(\cdot)$ is a mapping from source space to observation space parametrized by $\boldsymbol{\theta}_f$.

In an earlier work [13] we have used multi-layer perceptron (MLP) network with tanh-nonlinearities to model the mapping $\mathbf{f}$:

$$\mathbf{f}(\mathbf{s}; \mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}) = \mathbf{B} \tanh(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mathbf{b}\,. \tag{2.4}$$

The mapping $\mathbf{f}$ is thus parameterized by the matrices $\mathbf{A}$ and $\mathbf{B}$ and bias vectors $\mathbf{a}$ and $\mathbf{b}$. MLP networks are well suited for nonlinear FA and BSS. First, they are universal function approximators which means that any type of nonlinearity can be modeled by them in principle. Second, it is easy to model smooth, nearly linear mappings with them. This makes it possible to learn high dimensional nonlinear representations in practice.

An important special case of general nonlinear mixtures in (2.3) is a post-nonlinear (PNL) mixing model. There linear mixing is followed by component-wise nonlinearities acting on each output independently of the others:

$$x_i(t) = f_i\left[\mathbf{a}_i^T \mathbf{s}(t)\right] + n_i(t) \qquad i = 1, \ldots, n \tag{2.5}$$

Such models are plausible in applications where linearly mixed signals are measured by sensors with nonlinear distortions f_i. The nonlinearities f_i can also be modelled by MLP networks.

Identification of models (2.3) or (2.5) assuming Gaussianity of sources $\mathbf{s}(t)$ helps to find a compact representation of the observed data $\mathbf{x}(t)$. Nonlinear BSS can be achieved by performing a linear rotation of the found sources using, for example, a linear ICA technique.

The paper [12] presents our recent developments on nonlinear FA and BSS. A more accurate linearization increases stability of the algorithm in cases with a large number of sources when the posterior variances of the last weak sources are typically large. A hierarchical nonlinear factor analysis (HNFA) model using the building blocks presented in Section 2.3 is applicable to larger problems than the MLP based method, as the computational complexity is linear with respect to the number of sources. Estimating the PNL factor analysis model in (2.5) using variational Bayesian learning helps achieve separation of signals in very challenging BSS problems.

2.5 Nonlinear state-space models

In many cases, measurements originate from a dynamical system and form a time series. In such instances, it is often useful to model the dynamics in addition to the instantaneous observations. We have used rather general nonlinear models for both the data (observations) and dynamics of the sources (latent variables) [8]. This results in a state-space model where the sources can be interpreted as the internal state of the underlying generative process.

The general form of our nonlinear model for the generative mapping from the source (latent variable) vector $\mathbf{s}(t)$ to the data (observation) vector $\mathbf{x}(t)$ at time t is the same as in Eq. (2.3):

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_f) + \mathbf{n}(t)\,. \tag{2.6}$$

The dynamics of the sources can be modelled by another nonlinear mapping, which leads to a source model [8]

$$\mathbf{s}(t) = \mathbf{g}(\mathbf{s}(t-1), \boldsymbol{\theta}_g) + \mathbf{m}(t)\,, \tag{2.7}$$

where $\mathbf{s}(t)$ are the sources (states) at time t, $\mathbf{m}$ is the Gaussian noise, and $\mathbf{g}(\cdot)$ is a vector containing as its elements the nonlinear functions modelling the dynamics.

As for the static models presented in Sec. 2.4, the nonlinear functions are modelled by MLP networks. The mapping $\mathbf{f}$ has the same functional form (2.4). Since the states in dynamical systems are often slowly changing, the MLP network for mapping $\mathbf{g}$ models the change in the value of the source:

$$\mathbf{g}(\mathbf{s}(\mathbf{t}-\mathbf{1})) = \mathbf{s}(t-1) + \mathbf{D}\tanh[\mathbf{C}\mathbf{s}(t-1) + \mathbf{c}] + \mathbf{d}\,. \tag{2.8}$$

The dynamic mapping $\mathbf{g}$ is thus parameterized by the matrices $\mathbf{C}$ and $\mathbf{D}$ and bias vectors $\mathbf{c}$ and $\mathbf{d}$.

Estimation of the arising state-space model is rather involved, and it is discussed in detail in our earlier paper [8]. An important advantage of the proposed nonlinear state-space method (NSSM) is its ability to learn a high-dimensional latent source space. We have also reasonably solved computational and over-fitting problems which have been major obstacles in developing this kind of unsupervised methods thus far. Potential applications for our method include prediction and process monitoring, control and identification. MATLAB software packages are available for both the static model (2.3)-(2.4) (under the name nonlinear factor analysis) and the dynamic model (2.7)-(2.8) (under the name nonlinear dynamical factor analysis) on the home page of our Bayes group [14].

Time series prediction

Traditionally, time series prediction is done using models based directly on the past observations of the time series. Perhaps the two most important classes of neural network based solutions used for nonlinear prediction are feedforward autoregressive neural networks and recurrent autoregressive moving average neural networks [15]. However, instead of modelling the system based on past observations, it is also possible to model the same information in a more compact form with a state-space model.

We have used the nonlinear state-space model and method [8] described in the beginning of this section to model a time series. The primary goal in the paper [16] was to apply our NSSM method and software [14] to the time series prediction task as a black box tool. The details of this application are given in [16].

We applied the NSSM method to the prediction of the nonlinear scalar time series provided by the organizers of the ESTSP'07 symposium. The original time series containing 875 samples is shown in Figure 2.3. It seems to be strongly periodic with a period of

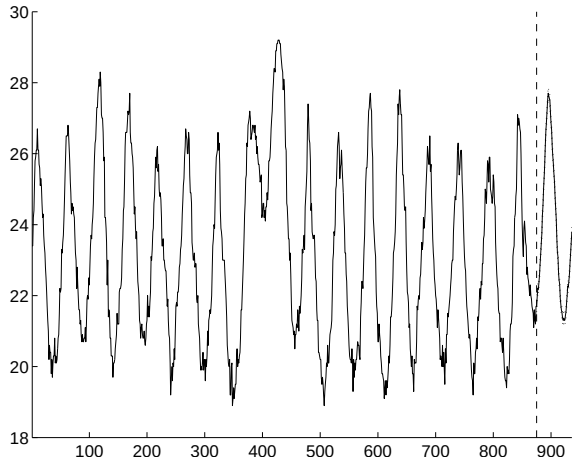

Figure 2.3: The original time series and the predicted 61 next time steps.

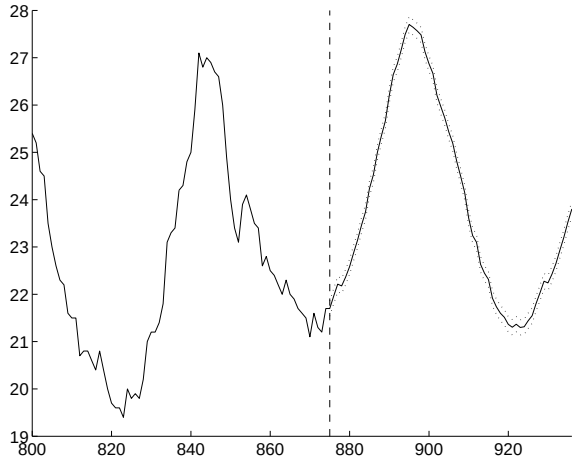

Figure 2.4: Bottom: The original time series starting from time instant 800 and the predicted 61 next time steps.

approximately 52 samples. Figure 2.3 shows also the predicted 61 next time steps, and Figure 2.4 in more detail the original time series starting from time instant 800 and the predicted 61 next time steps. The dotted lines in both figures represent pseudo 95 % confidence intervals. These intervals are, however, smaller than in reality as the variance caused by the innovation is ignored [16].

Improving state inference

The problem of state inference involves finding the source vectors $\mathbf{s}(t-1)$ given the data and the model. While this is an easier problem than finding both the model and the sources, it is more time critical, since it must often be computed in real-time. While the algorithm in [8] can be used for inference, it is very slow because of the slow flow of information through the time series. Standard algorithms based on extensions of the Kalman smoother work rather well in general, but may fail to converge when estimating the states over a long gap or when used together with learning the model.

When updates are done locally, information spreads around slowly because the states of different time slices affect each other only between updates. It is possible to predict this interaction by a suitable approximation. In [17], we derived a novel update algorithm

for the posterior mean of the states by replacing partial derivatives of the cost function with respect to state means $\overline{\mathbf{s}}(t)$ by (approximated) total derivatives

$$\frac{d\mathcal{C}_{\mathrm{KL}}}{d\overline{\mathbf{s}}(t)} = \sum_{\tau=1}^{T} \frac{\partial \mathcal{C}_{\mathrm{KL}}}{\partial \overline{\mathbf{s}}(\tau)} \frac{\partial \overline{\mathbf{s}}(\tau)}{\partial \overline{\mathbf{s}}(t)}. \tag{2.9}$$

They can be computed efficiently using the chain rule and dynamic programming, given that we can approximate the terms $\partial\overline{\mathbf{s}}(t)/\partial\overline{\mathbf{s}}(t-1)$ and $\partial\overline{\mathbf{s}}(t)/\partial\overline{\mathbf{s}}(t+1)$.

This is how we approximated the required partial derivatives. The posterior distribution of the state $\mathbf{s}(t)$ can be factored into three potentials, one from $\mathbf{s}(t-1)$ (the past), one from $\mathbf{s}(t+1)$ (the future), and one from $\mathbf{x}(t)$ (the observation). We linearized the nonlinear mappings so that the three potentials become Gaussian. Then also the posterior of $\mathbf{s}(t)$ becomes Gaussian with a mean that is the weighted average of the means of the three potentials, where the weights are the inverse (co)variances of the potentials. A change in the mean of a potential results in a change of the mean of the posterior inversely proportional to their (co)variances.

Experimental comparison in [17] showed that the proposed algorithm worked reliably and fast. The algorithms from the Kalman family (IEKS and IUKS) were fast, too, but they also suffered from stability problems when gaps of 30 consequtive missing observations were introduced into the data. Basic particle smoother performed very poorly compared to the iterative algorithms. It should be noted that many different schemes exists to improve the performance of particle filters.

Stochastic nonlinear model-predictive control

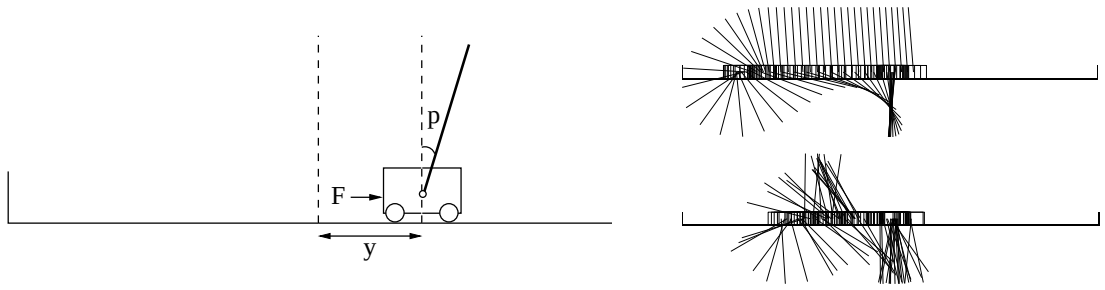

Figure 2.5: Left: The cart-pole system. The goal is to swing the pole to an upward position and stabilize it without hitting the walls. The cart can be controlled by applying a force to it. Top left: The pole is successfully swinged up by moving first to the left and then right. Bottom right: Our controller works quite reliably even in the presence of serious observation noise.

In [18], we studied such a system combining variational Bayesian learning of an unknown dynamical system with nonlinear model-predictive control. For being able to control the dynamical system, control inputs are added to the nonlinear state-space model. Then we can use stochastic nonlinear model-predictive control, which is based on optimising control signals based on maximising a utility function.

Figure 2.5 shows simulations with a cart-pole swing-up task. The results confirm that selecting actions based on a state-space model instead of the observation directly has many benefits: First, it is more resistant to noise because it implicitly involves filtering. Second, the observations (without history) do not always carry enough information about the system state. Third, when nonlinear dynamics are modelled by a function approximator such

as an multilayer perceptron network, a state-space model can find such a representation of the state that it is more suitable for the approximation and thus more predictable.

Continuous-time modeling

In [19], we have outlined an extension of the discrete-time variational Bayesian NSSM of [8] to continuous-time systems and presented preliminary experimental results with the method. Evaluation of the method with larger and more realistic examples is a very important item of further work. The main differences between continuous-time and discrete-time variational NSSMs are the different method needed to evaluate the predictions of the states and the different form of the dynamical noise or innovation.

2.6 Non-negative blind source separation

In linear factor analysis (FA) [20], the observations are modeled as noisy linear combinations of a set of underlying sources or factors. When the level of noise is low, FA reduces to principal component analysis (PCA). Both FA and PCA are insensitive to orthogonal rotations, and, as such, cannot be used for blind source separation except in special cases. There are several ways to solve the rotation indeterminacy. One approach is to assume the sources independent, which in low noise leads to independent component analysis. Another approach, the one discussed in this section, is to constrain the sources to be non-negative.

Non-negativity constraints in linear factor models have received a great deal of interest in a number of problem domains. In the variational Bayesian framework, positivity of the factors can be achieved by putting a non-negatively supported prior on them. The rectified Gaussian distribution is particularly convenient, as it is conjugate to the Gaussian likelihood arising in the FA model. Unfortunately, this solution has a technical limitation: the location parameter of the prior has to be fixed to zero; otherwise the potentials of both the location and the scale parameter become awkward.

To evade the above mentioned problems, the model is reformulated using rectification nonlinearities. This can be expressed in the form of Eq. (2.4) using the following nonlinearity

$$\mathbf{f}(\mathbf{s};\, \mathbf{A}) = \mathbf{A}\, \mathbf{cut}(\mathbf{s}) \tag{2.10}$$

where **cut** is the componentwise rectification (or cut) function such that $[\mathbf{cut}(\mathbf{s})]_i = \max(s_i, 0)$. In [21], a variational learning procedure was derived for the proposed model and it was shown that it indeed overcomes the problems that exist with the related approaches (see Figure 2.6 for a controlled experiment). In Section 2.10 an application of the method to the analysis of galaxy spectra is presented. There the underlying sources were such that the zero-location rectified Gaussian prior was highly inappropriate, which motivated the development of the proposed approach.

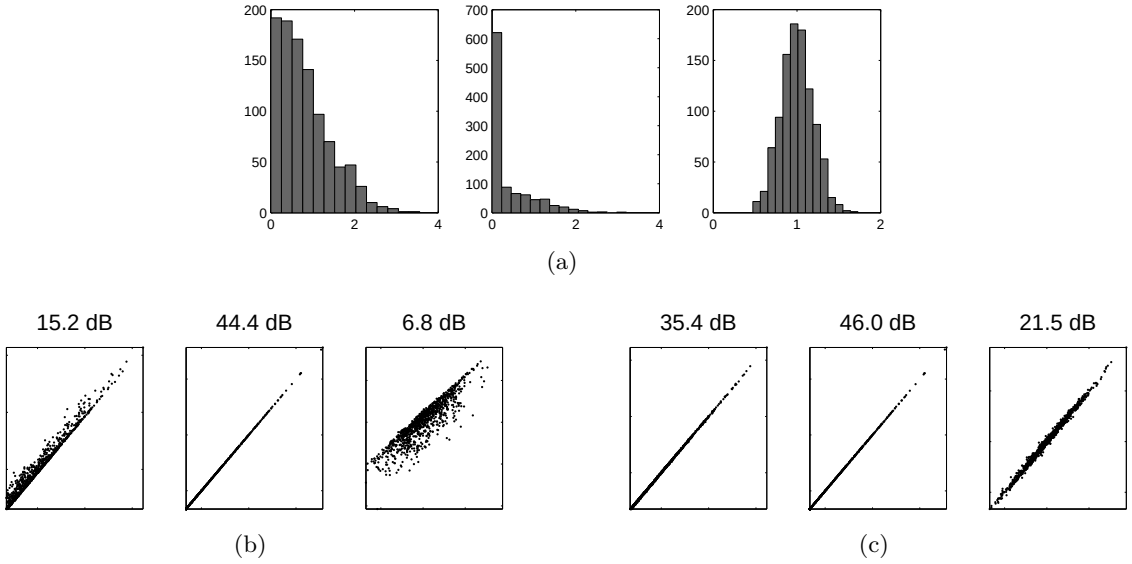

Figure 2.6: (a) The histograms of the true sources to be recovered. (b) and (c) The estimated sources plotted against the true sources with the signal-to-noise ratios printed above each plot. In (b), rectified Gaussian priors have been used for the sources. In (c), the proposed approach employing rectification nonlinearities has been used.

2.7 PCA in the presence of missing values

Principal component analysis (PCA) is a classical data analysis technique. Some algorithms for PCA scale better than others to problems with high dimensionality. They also differ in the ability to handle missing values in the data. In our recent papers [22, 23], a case is studied where the data are high-dimensional and a majority of the values are missing.

In the case of very sparse data, overfitting becomes a severe problem even in simple linear models such as PCA. Regularization can be provided using the Bayesian approach by introducing prior for the model parameters. The PCA model can then be identified using, for example, maximum a posteriori estimates (regularized PCA) or variational Bayesian (VB) learning. We study both approaches in the papers [22, 23].

The proposed learning algorithm is based on speeding up a simple principal subspace rule in which the model parameters are updated as

$$\theta_i \leftarrow \theta_i - \gamma \left(\frac{\partial^2 C}{\partial \theta_i^2} \right)^{-\alpha} \frac{\partial C}{\partial \theta_i}, \tag{2.11}$$

where α is a control parameter that allows the learning algorithm to vary from the standard gradient descent ($\alpha = 0$) to the diagonal Newton's method ($\alpha = 1$). These learning rules can be used for standard PCA learning and extended to regularized PCA and variational Bayesian (VB) PCA.

The algorithms were tested on the Netflix problem (`http://www.netflixprize.com/`), which is a task of predicting preferences (or producing personal recommendations) by using other people's preferences. The Netflix problem consists of movie ratings given by 480189 customers to 17770 movies. There are 100480507 ratings from 1 to 5 given, and the task is to predict 2817131 other ratings among the same group of customers and movies. 1408395 of the ratings are reserved for validation. Thus, 98.8% of the values are missing. We tried to find 15 principal components from the data using a number of methods. The results confirm that the proposed speed-up procedure is much faster than any of the compared methods, and that VB-PCA method provides more accurate predictions for new data than traditional PCA or simple regularized PCA (see Fig. 2.7).

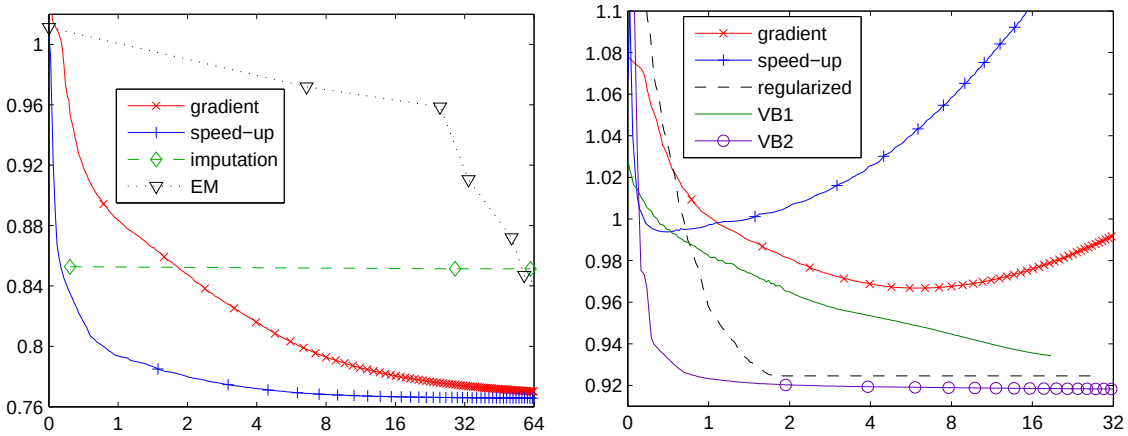

Figure 2.7: *Left:* Training error against computation time in hours in the Netflix problem for unregularized PCA algorithm based on gradient descent and the proposed speed-up. Two alternative methods are shown for comparison. *Right:* The error on test data for the two versions of unregularized PCA, regularized PCA and two variants of variational Bayesian PCA. The time scale is linear below 1 and logarithmic above 1.

2.8 Predictive uncertainty

In standard regression, we seek to predict the value of a response variable based on a set of explanatory variables. Here, the term *predictive uncertainty* is used to refer to a task similar to regression with the exception that we predict not only the mean outcome of the response variable, but also the uncertainty related to its value. For example, consider predicting the concentration of an air pollutant in a city, based on meteorological conditions measured some time in advance. In this task it is the extreme events, namely those occasions when the concentration of the air pollutant rises over a certain threshold, that are interesting. If the conditional distribution of the response variable is not tightly concentrated around its mean value, the mean value by itself will be a poor indicator of the extreme events occurring, and hence predictions based on those alone might lead to policies with ill consequences.

In [26], a method for predictive uncertainty is presented. The method is based on conditioning the scale parameter of the noise process on the explanatory variables and then using MLP networks to model both the location and the scale of the output distribution. The model can be summarised as

$$\begin{aligned} y_t &\sim N(f(\mathbf{x}_t, \boldsymbol{\theta}_y), e^{-u_t}) \\ u_t &\sim N(f(\mathbf{x}_t, \boldsymbol{\theta}_u), \tau^{-1}) \end{aligned} \tag{2.12}$$

Above, y_t is the response variable and $\mathbf{x}_t$ is the vector of explanatory variables. The function f, representing the MLP network, has essentially the same form as in Eq. (2.4). When the latent variable u_t is marginalised out of the model the predictive distribution for y_t becomes super-Gaussian. The extent to which this happens depends on the uncertainty in u_t as measured by the precision parameter τ which is adapted in the learning process. This adaptive nongaussianity of the predictive distribution is highly desirable as then the uncertainty in the scale parameter can be accommodated by making the predictive distribution more robust.

The problem with heteroscedastic models is that learning them using simple methods can be difficult as overfitting becomes a serious concern. Variational Bayesian (VB) methods can, however, largely avoid these problems. Unfortunately, VB methods for nonlinear models, such as that in Eq. (2.12), become involved both in analytic as well as in computational terms. Therefore the learning algorithm in [26] is based on the slightly weaker approximation technique, the variational EM algorithm, and only the "important" parameters have distributional estimates. These parameters include the latent variables u_t, the precision parameter, and the second layer weights of the MLPs. The rest of the parameters, that is, the first layer weights of the MLPs, have point estimates only.

The method summarized in this section was applied to all four datasets in the 'Predictive uncertainty in environmental modelling' competition held at World Congress on Computational Intelligence 2006. The datasets varied in dimensionality from one input variable to 120 variables. The proposed method performed well with all the datasets where heteroscedasticity was an important component being the overall winner of the competition.

2.9 Relational models

In the previous sections, we have studied models belonging to two categories: static and dynamic. In static modeling, each observation or data sample is independent of the others. In dynamic models, the dependencies between consecutive observations are modeled. A generalization of both types of models is that the relations are described in the data itself, that is, each observation might have a different structure.

Logical hidden Markov models

Many real-world sequences such as protein secondary structures or shell logs exhibit rich internal structures. In [24], we have proposed logical hidden Markov models as one solution. They deal with logical sequences, that is, sequences over an alphabet of logical atoms. This comes at the expense of a more complex model selection problem. Indeed, different abstraction levels have to be explored. Logical hidden Markov models (LOHMMs) upgrade traditional hidden Markov models to deal with sequences of structured symbols in the form of logical atoms, rather than characters. Our recent paper [24] formally introduces LOHMMs and presents solutions to the three central inference problems for LOHMMs: evaluation, most likely hidden state sequence, and parameter estimation. The resulting representation and algorithms are experimentally evaluated on problems from the domain of bioinformatics (see Figure 2.8).

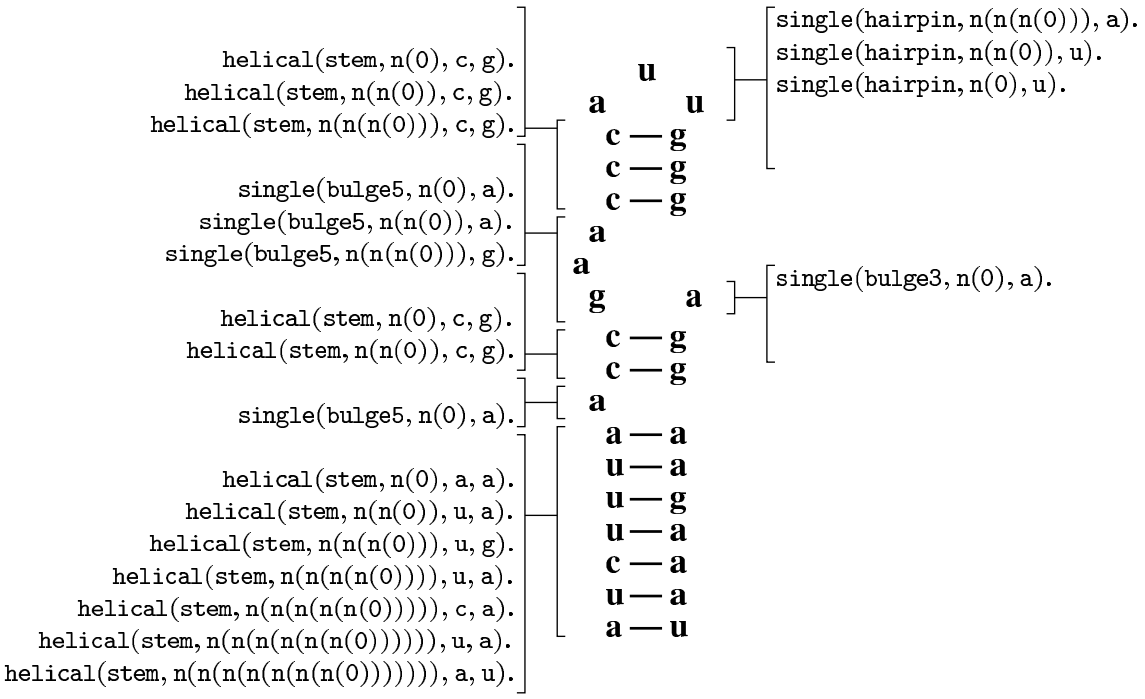

Figure 2.8: Representation of mRNA signal structures as a sequence of logical atoms to be analyzed with a logical hidden Markov model.

Higher order statistics in play-out analysis

A second relational study involves game playing. There is a class of board games called connection games for which traditional artificial intelligence approach does not provide a good computer player. For such games, it is an interesting option to play out the game

from the current state to the end many times randomly. Play-outs provide statistics that can be used for selecting the best move. In [25], we introduce a method that selects relevant patterns of moves to collect higher order statistics. Play-out analysis avoids the horizon effect of regular game-tree search. The proposed method is especially effective when the game can be decomposed into a number of subgames. Preliminary experiments on the board games of Hex and Y are reported in [25].

2.10 Applications to astronomy

Two astronomical applications are discussed in this section: analysis of galaxy spectra and estimation of time delays in gravitational lensing.

Analysis of galaxy spectra

We have applied rectified factor analysis [21] described in Section 2.6 to the analysis of real stellar population spectra of elliptical galaxies. Ellipticals are the oldest galactic systems in the local universe and are well studied in physics. The hypothesis that some of these old galactic systems may actually contain young components is relatively new. Hence, we have investigated whether a set of stellar population spectra can be decomposed and explained in terms of a small set of unobserved spectral prototypes in a data driven but yet physically meaningful manner. The positivity constraint is important in this modelling application, as negative values of flux would not be physically interpretable.

Using a set of 21 real stellar population spectra, we found that they can indeed be decomposed to prototypical spectra, especially to a young and old component [27]. Figure 2.9 shows one spectrum and its decomposition to these two components. The right subfigure shows the ages of the galaxies, known from a detailed astrophysical analysis, plotted against the first weight of the mixing matrix. The plot clearly shows that the first component corresponds to a galaxy containing a significant young stellar population.

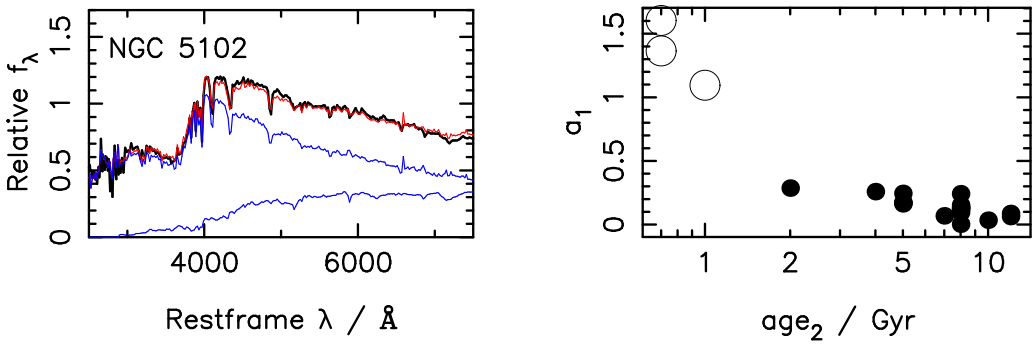

Figure 2.9: Left: the spectrum of a galaxy with its decomposition to a young and old component. Right: the age of the dominating stellar population against the mixing coefficient of the young component.

Estimation of time delays in gravitational lensing

Gravitational lensing occurs when the light coming from a distant bright source is bent by the gravitational potential of an intermediate galaxy such that several images of the source are observed (see the left panel of Figure 2.10 for an example system). Relativistic effects and the different lengths of the paths affect the time it takes for the photons originating from the source to travel to the observer. This is perceived as a delay in the intensity variations between the images (see the right panel of Figure 2.10). The significance of estimating the delays in such systems stems from the early observation that they can be used in determining important cosmological quantities [28].

The delay estimation problem is difficult for various reasons. The main challenge is the uneven sampling rate, as the sampling times are determined by factors one cannot control such as observing conditions and scheduling. The signal-to-noise ratio in the

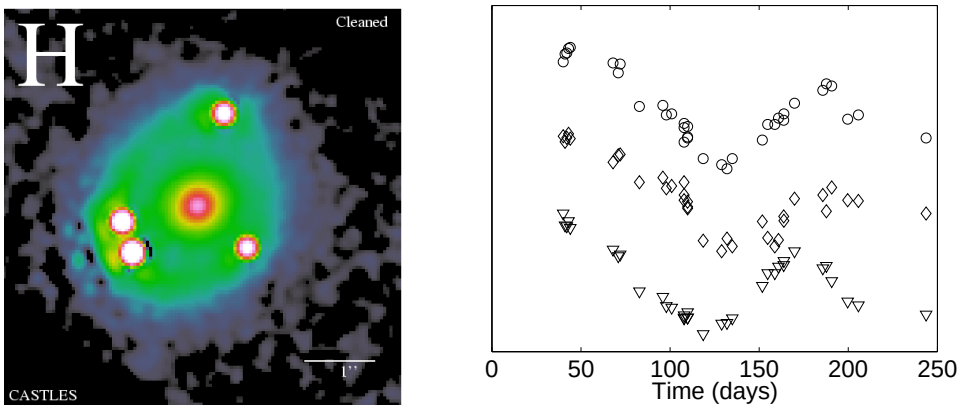

Figure 2.10: Left: The four images of PG1115+080. Right: The corresponding intensity measurements (the two images closest to each other are merged).

observations is often poor too, although this varies somewhat between datasets. Classical delay estimation methods usually rely on the cross-correlation function which is easy to evaluate between regularly sampled signals. The obvious way to attack the problem with unevenly sampled signals would then be to interpolate them appropriately to obtain evenly sampled signals and then apply the cross correlation method. But with all the gaps and the noise in the data, the interpolation can introduce spurious features to the data which make the cross-correlation analysis go awry [29].

In [30, 31], a method for estimating the delay between irregularly sampled signals is presented. Since interpolation on the gappy and noisy data can be venturesome, that is avoided. Instead the two observed signals, $x_1(t)$ and $x_2(t)$, are postulated to have been emitted from the same latent source signal $s(t)$, the observation times being determined by the actual sampling times and the delay. The source is then assumed to follow the Wiener process: $s(t_{i+1}) - s(t_i) \sim N(0, [(t_{i+1} - t_i)\,\sigma]^2)$. This prior encodes the notion of "slow variability" into the model which is an assumption implicitly present in many of the other methods as well. The model is estimated using exact marginalization, which leads to a specific type of Kalman-filter, combined with the Metropolis-Hastings algorithm.

We have used the proposed method to determine the delays in several gravitational lensing systems. Controlled comparisons againts other methods cannot, however, be done with real data as the true delays are unknown to us. Instead, artificial data, where the ground truth is known, must be used. Figure 2.11 shows the performance of several methods in an artificial setting.

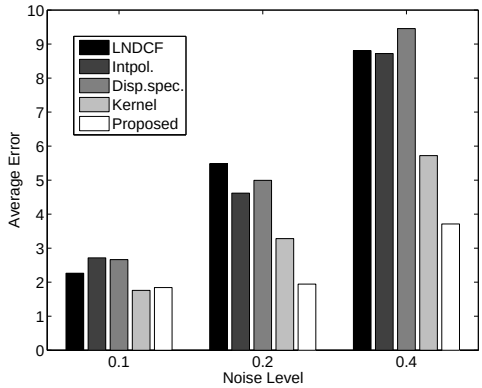

Figure 2.11: Average errors of the methods for three groups of datasets.

References

[1] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[2] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

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

[4] 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*, MIT Press, 1999, pages 105–161.

[5] A. Honkela and H. Valpola. Variational learning and bits-back coding: an information-theoretic view to Bayesian learning. *IEEE Transactions on Neural Networks*, 15(4):267–282, 2004.

[6] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[7] A. Honkela, M. Tornio, T. Raiko, and J. Karhunen. Natural conjugate gradient in variational inference. In *Proc. of the 14th Int. Conf. on Neural Information Processing (ICONIP 2007)*, Kitakyushu, Japan, November 2007.

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

[9] H. Valpola, A. Honkela, M. Harva, A. Ilin, T. Raiko, and T. Östman. Bayes Blocks software library. `http://www.cis.hut.fi/projects/bayes/software/`, 2003.

[10] T. Raiko, H. Valpola, M. Harva, and J. Karhunen. Building blocks for variational Bayesian learning of latent variable models. *Journal of Machine Learning Research*, Vol. 8, pp. 155-201, January 2007.

[11] A. Honkela, M. Harva, T. Raiko, H. Valpola, and J. Karhunen. Bayes Blocks: A Python toolbox for variational Bayesian learning. *NIPS2006 Workshop on Machine Learning Open Source Software*, Whistler, B.C., Canada, 2006.

[12] A. Honkela, H. Valpola, A. Ilin and J. Karhunen. Blind separation of nonlinear mixtures by variational Bayesian learning. *Digital Signal Processing*, Vol. 17, No 2, pp. 914–934, 2007.

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

[14] Home page of our Bayes group: `http://www.cis.hut.fi/projects/bayes/`.

[15] A. Trapletti, *On Neural Networks as Statistical Time Series Models.* PhD Thesis, Technische Universität Wien, 2000.

[16] M. Tornio, A. Honkela, and J. Karhunen. Time series prediction with variational Bayesian nonlinear state-space models. In *Proc. European Symp. on Time Series Prediction (ESTSP'07)*, pages 11–19, Espoo, Finland, February 2007.

[17] T. Raiko, M. Tornio, A. Honkela, and J. Karhunen. State inference in variational Bayesian nonlinear state-space models. In *Proc. of the 6th Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA 2006)*, pages 222–229, Charleston, South Carolina, USA, March 2006.

[18] M. Tornio and T. Raiko. Variational Bayesian approach for nonlinear identification and control. In *Proc. of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, NMPC FS06*, pp. 41-46, Grenoble, France, October 9-11, 2006.

[19] A. Honkela, M. Tornio, and T. Raiko. Variational Bayes for continuous-time nonlinear state-space models. In *NIPS2006 Workshop on Dynamical Systems, Stochastic Processes and Bayesian Inference*, Whistler, B.C., Canada, 2006.

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

[21] M. Harva and A. Kabán. Variational learning for rectified factor analysis. *Signal Processing*, 87(3):509–527, 2007.

[22] T. Raiko, A. Ilin and J. Karhunen. Principal component analysis for large scale problems with lots of missing values. In *Proc. of the 18th European Conf. on Machine Learning (ECML 2007)*, pages 691–698, Warsaw, Poland, September 2007.

[23] T. Raiko, A. Ilin, and J. Karhunen. Principal component analysis for sparse high-dimensional data. In *Proc. of the 14th Int. Conf. on Neural Information Processing (ICONIP 2007)*, Kitakyushu, Japan, November 2007.

[24] K. Kersting, L. De Raedt, and T. Raiko. Logical hidden Markov models. *Journal of Artificial Intelligence Research (JAIR)*, Vol. 25, pp. 425–456, April 2006.

[25] T. Raiko. Higher order statistics in play-out analysis. *Proc. of the Scandinavian Conf. on Artificial Intelligence, SCAI2006*, pp. 189–195, Espoo, Finland, October 25-27, 2006.

[26] M. Harva. A variational EM approach to predictive uncertainty. *Neural Networks*, 20(4):550–558, 2007.

[27] L. Nolan, M. Harva, A. Kabán, and S. Raychaudhury. A data-driven Bayesian approach for finding young stellar populations in early-type galaxies from their UV-optical spectra. *Monthly Notices of the Royal Astronomical Society*, 366(1):321–338, 2006.

[28] S. Refsdal. On the possibility of determining Hubble's parameter and the masses of galaxies from the gravitational lens effect. *Monthly Notices of the Royal Astronomical Society*, 128:307–310, 1964.

[29] J. C. Cuevas-Tello, P. Tino, and S. Raychaudhury. How accurate are the time delay estimates in gravitational lensing? *Astronomy & Astrophysics*, 454:695–706, 2006.

[30] M. Harva and S. Raychaudhury. Bayesian estimation of time delays between unevenly sampled signals. In *Proc. Int. Workshop on Machine Learning for Signal Processing (MLSP'06)*, pages 111–116. Maynooth, Ireland, 2006.

[31] M. Harva and S. Raychaudhury. Bayesian estimation of time delays between unevenly sampled signals. *Neurocomputing*, 2008. To appear.