State Inference in Variational Bayesian Nonlinear State-Space Models

Tapani Raiko, Matti Tornio, Antti Honkela, and Juha Karhunen

Helsinki University of Technology, Neural Networks Research Centre,
P.O. Box 5400, FI-02015 HUT, Espoo, Finland
Email: {tapani.raiko, matti.tornio, antti.honkela, juha.karhunen}@hut.fi

Abstract. Nonlinear source separation can be performed by inferring the state of a nonlinear state-space model. We study and improve the inference algorithm in the variational Bayesian blind source separation model introduced by Valpola and Karhunen in 2002. As comparison methods we use extensions of the Kalman filter that are widely used inference methods in tracking and control theory. The results in stability, speed, and accuracy favour our method especially in difficult inference problems.

1 Introduction

Many applications of source separation methods involve data with some kind of relations between consecutive observations. Examples include relations between neighbouring pixels in images and time series data. Using information on these relations improves the quality of separation results, especially in difficult nonlinear separation problems. Nonlinear modelling of relations may also be useful in linear mixing problems as the dynamics of the time series, for instance, may well be nonlinear.

A method for blind source separation using a nonlinear state-space model is described in [1]. In this paper we study and improve ways of estimating the sources or states in this framework. Efficient solution of the state estimation problem requires taking into account the nonlinear relations between consecutive samples, making it significantly more difficult than source separation in static models. 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. We propose solving the problem by improving the variational Bayesian technique proposed in [1] by explicitly using the information on the relation between consecutive samples to speed up convergence.

To tackle just the state estimation (or source separation) part, we will simplify the blind problem by fixing the model weights and other parameters. In [2], linear and nonlinear state-space models are used for blind and semi-blind source separation. Also there the problem is simplified by fixing part of the model.

2 Nonlinear State-Space Models

In nonlinear state-space models, the observation vectors $\mathbf{x}(t)$, $t = 1, 2, \ldots, T$, are assumed to have been generated from unobserved state (or source) vectors $\mathbf{s}(t)$. The model equations are

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

$$\mathbf{s}(t) = \mathbf{g}(\mathbf{s}(t-1)) + \mathbf{m}(t), \tag{2}$$

Both the mixing mapping $\mathbf{f}$ and the process mapping $\mathbf{g}$ are nonlinear. The noise model for both mixing and dynamical process is often assumed to be Gaussian

$$p(\mathbf{n}(t)) = \mathcal{N}\left[\mathbf{n}(t); \mathbf{0}; \mathbf{\Sigma}_x\right] \tag{3}$$

$$p(\mathbf{m}(t)) = \mathcal{N}\left[\mathbf{m}(t); \mathbf{0}; \mathbf{\Sigma}_s\right], \tag{4}$$

where $\mathbf{\Sigma}_x$ and $\mathbf{\Sigma}_s$ are the noise covariance matrices. In blind source separation, the mappings $\mathbf{f}$ and $\mathbf{g}$ are assumed to be unknown [1] but in this paper we concentrate on the case where they are known.

2.1 Inference Methods

The task of estimating a sequence of sources $\mathbf{s}(1), \ldots, \mathbf{s}(T)$ given a sequence of observations $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ and the model is called inference. In case $\mathbf{f}$ and $\mathbf{g}$ in Eqs. (1) and (2) are linear, the state can be inferred analytically with an algorithm called the *Kalman filter* [3]. In a filter phase, evidence from the past is propagated forward, and in a smoothing phase, evidence from the future is propagated backwards. Only the most recent state can be inferred using the Kalman filter, otherwise the algorithm should be called the *Kalman smoother*. In [4], the Kalman filter is extended for blind source separation from time-varying noisy mixtures.

The idea behind *iterated extended Kalman smoother* [3] (IEKS) is to linearise the mappings $\mathbf{f}$ and $\mathbf{g}$ around the current state estimates using the first terms of the Taylor series expansion. The algorithm alternates between updating the state estimates by Kalman smoothing and renewing the linearisation. When the system is highly nonlinear or the initial estimate is poor, the IEKS may diverge.

The *iterative unscented Kalman smoother* [5, 6] (IUKS) replaces the local linearisation of IEKS by a deterministic sampling technique. The sampled points are propagated through the nonlinearities, and a Gaussian distribution is fitted to them. The use of nonlocal information improves convergence and accuracy at the cost of doubling the computational complexity[1]. Still there is no guarantee of convergence.

A recent variant called *backward-smoothing extended Kalman filter* [8] searches for the maximum a posteriori solution to the filtering problem by a guarded

[1] An even better way of replacing the local linearisation when a multilayer perceptron network is used as a nonlinearity, is described in [7].

Gauss-Newton method. It increases the accuracy further and guarantees convergence at the cost of about hundredfold increase in computational burden.

Particle filter [9] uses a set of particles or random samples to represent the state distribution. It is a Monte Carlo method developed especially for sequences. The particles are propagated through nonlinearities and there is no need for linearisation nor iterating. Given enough particles, the state estimate approaches the true distribution. Combining the filtering and smoothing directions is not straightforward but there are alternative methods for that. In [10], particle filters are used for non-stationary ICA.

2.2 Variational Bayesian method

Nonlinear dynamical factor analysis (NDFA) [1] is a variational Bayesian method for learning nonlinear state-space models. The mappings $\mathbf{f}$ and $\mathbf{g}$ in Eqs. (1) and (2) are modelled with multilayer perceptron (MLP) networks whose parameters can be learned from the data. The parameter vector $\boldsymbol{\theta}$ include network weigths, noise levels, and hierarchical priors for them. The posterior distribution over the sources $\mathbf{S} = [\mathbf{s}(1), \ldots, \mathbf{s}(T)]$ and the parameters $\boldsymbol{\theta}$ is approximated by a Gaussian distribution $q(\mathbf{S}, \boldsymbol{\theta})$ with some further independency assumptions. Both learning and inference are based on minimising a cost function $\mathcal{C}_{\mathrm{KL}}$

$$\mathcal{C}_{\mathrm{KL}} = \int_{\boldsymbol{\theta}} \int_{\mathbf{S}} q(\mathbf{S}, \boldsymbol{\theta}) \ln \frac{q(\mathbf{S}, \boldsymbol{\theta})}{p(\mathbf{X}, \mathbf{S}, \boldsymbol{\theta})} d\mathbf{S} d\boldsymbol{\theta}, \tag{5}$$

where $p(\mathbf{X}, \mathbf{S}, \boldsymbol{\theta})$ is the joint probability density over the data $\mathbf{X} = [\mathbf{x}(1), \ldots, \mathbf{x}(T)]$, sources $\mathbf{S}$, and parameters $\boldsymbol{\theta}$. The cost function is based on Kullback-Leibler divergence between the approximation and the true posterior. It can be split into terms, which helps in studying only a part of the model at a time. The variational approach is less prone to overfitting compared to maximum a posteriori estimates and still fast compared to Monte Carlo methods. See [1] for details.

The variational Bayesian inference algorithm in [1] uses the gradient of the cost function w.r.t. state in a heuristic manner. We propose an algorithm that differs from it in three ways. Firstly, the heuristic updates are replaced by a standard conjugate gradient algorithm [11]. Secondly, the linearisation method from [7] is applied. Thirdly, the gradient is replaced by a vector of approximated total derivatives, as described in the following section.

2.3 Total Derivatives

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. We get a novel update algorithm for the posterior mean of the states by replacing partial derivatives of the cost function w.r.t. state means $\overline{\mathbf{s}}(t)$ by (approximated) total derivatives

$$\frac{d\mathcal{C}_{\mathrm{KL}}}{d\overline{\mathbf{s}}(t)} = \sum_{\tau=1}^{T} \frac{\partial \mathcal{C}_{\mathrm{KL}}}{\partial \overline{\mathbf{s}}(\tau)} \frac{\partial \overline{\mathbf{s}}(\tau)}{\partial \overline{\mathbf{s}}(t)}. \tag{6}$$

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

Before going into details, let us go through the idea. 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 will linearise 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.

The terms of the cost function (See Equation (5.6) in [1], although the notation is somewhat different) that relate to $\mathbf{s}(t)$ are

$$\begin{aligned}\mathcal{C}_{\mathrm{KL}}(\mathbf{s}(t)) = &\sum_{i=1}^{m}\left(-\frac{1}{2}\ln\widetilde{s}_{ii}(t) + \frac{1}{2}\Sigma_{sii}^{-1}\left\{\left[\overline{s}_i(t) - \overline{g}_i(\mathbf{s}(t-1))\right]^2 + \widetilde{s}_i(t)\right\}\right)\\ &+\sum_{j=1}^{m}\frac{1}{2}\Sigma_{sjj}^{-1}\left\{\left[\overline{g}_j(\mathbf{s}(t)) - \overline{s}_j(t+1)\right]^2 + \widetilde{g}_j(\mathbf{s}(t))\right\}\\ &+\sum_{k=1}^{n}\frac{1}{2}\Sigma_{xkk}^{-1}\left\{\left[\overline{f}_k(\mathbf{s}(t)) - \overline{x}_k(t)\right]^2 + \widetilde{f}_k(\mathbf{s}(t))\right\},\end{aligned} \tag{7}$$

where $\overline{\alpha}$ and $\widetilde{\alpha}$ denote the mean and (co)variance of α over the posterior approximation q respectively and n and m are the dimensionalities of $\mathbf{x}$ and $\mathbf{s}$ respectively. Note that we assume diagonal noise covariances $\mathbf{\Sigma}$. Nonlinearities $\mathbf{f}$ and $\mathbf{g}$ are replaced by the linearisations

$$\widehat{\mathbf{f}}(\mathbf{s}(t)) = \overline{\mathbf{f}}(\mathbf{s}_{\mathrm{cur}}(t)) + \mathbf{J}_f(t)\left[\mathbf{s}(t) - \overline{\mathbf{s}}_{\mathrm{cur}}(t)\right] \tag{8}$$

$$\widehat{\mathbf{g}}(\mathbf{s}(t)) = \overline{\mathbf{g}}(\mathbf{s}_{\mathrm{cur}}(t)) + \mathbf{J}_g(t)\left[\mathbf{s}(t) - \overline{\mathbf{s}}_{\mathrm{cur}}(t)\right], \tag{9}$$

where the subscript cur denotes a current estimate that is constant w.r.t. further changes in $\mathbf{s}(t)$. The minimum of (7) with linearisations can be found at the zero of the gradient:

$$\widetilde{\mathbf{s}}_{\mathrm{opt}}(t) = \left[\mathbf{\Sigma}_s^{-1} + \mathbf{J}_g(t)^{\mathrm{T}}\mathbf{\Sigma}_s^{-1}\mathbf{J}_g(t) + \mathbf{J}_f(t)^{\mathrm{T}}\mathbf{\Sigma}_x^{-1}\mathbf{J}_f(t)\right]^{-1} \tag{10}$$

$$\begin{aligned}\overline{\mathbf{s}}_{\mathrm{opt}}(t) = \widetilde{\mathbf{s}}_{\mathrm{opt}}(t)\big\{&\mathbf{\Sigma}_s^{-1}\left[\overline{\mathbf{g}}(\mathbf{s}_{\mathrm{cur}}(t-1)) + \mathbf{J}_g(t-1)(\overline{\mathbf{s}}(t-1) - \overline{\mathbf{s}}_{\mathrm{cur}}(t-1))\right]\\ &+\mathbf{J}_g(t)^{\mathrm{T}}\mathbf{\Sigma}_s^{-1}\left[\overline{\mathbf{s}}(t+1) - \overline{\mathbf{g}}(\mathbf{s}_{\mathrm{cur}}(t))\right]\\ &+\mathbf{J}_f(t)^{\mathrm{T}}\mathbf{\Sigma}_x^{-1}\left[\overline{\mathbf{x}}(t) - \overline{\mathbf{f}}(\mathbf{s}_{\mathrm{cur}}(t))\right]\big\}.\end{aligned} \tag{11}$$

The optimum mean reacts to changes in the past and in the future by

$$\frac{\partial \overline{\mathbf{s}}_{\mathrm{opt}}(t)}{\partial \overline{\mathbf{s}}(t-1)} = \widetilde{\mathbf{s}}_{\mathrm{opt}}(t)\mathbf{\Sigma}_s^{-1}\mathbf{J}_g(t-1) \tag{12}$$

$$\frac{\partial \overline{\mathbf{s}}_{\mathrm{opt}}(t)}{\partial \overline{\mathbf{s}}(t+1)} = \widetilde{\mathbf{s}}_{\mathrm{opt}}(t)\mathbf{J}_g(t)^{\mathrm{T}}\mathbf{\Sigma}_s^{-1}. \tag{13}$$

Finally, we assume that the Equations (12) and (13) apply approximately even in the nonlinear case when the subscripts opt are dropped out. The linearisation matrices $\mathbf{J}$ need to be computed anyway [7] so the computational overhead is rather small.

3 Experiments

To experimentally measure the performance of our proposed new method, we used two different data sets. The first data set was generated using a simulated double inverted pendulum system with known dynamics. As the second data set we used real-world speech data with unknown dynamics.

In all the experiments, IEKS and IUKS were run for 50 iterations and NDFA algorithm for 500 iterations. In most cases this was long enough for the algorithms to converge to a local minimum. For comparison purposes, the NDFA experiments were also repeated without using the total derivatives.

Even with a relatively high number of particles, particle smoother performed poorly compared to the iterative algorithms. The results for particle smoother are therefore omitted from the figures. They are however discussed where appropriate. Even though the particle smoother performed relatively poorly, it should be noted that many different schemes exists to improve the performance of particle filters [9], and therefore direct comparison between the iterative algorithms and the plain particle filter algorithm used in these experiments may be somewhat unjustified. The experiments were also repeated with the original NDFA algorithm presented in [1]. The results were quite poor, as was to be excepted, as the heuristic update rules are optimized for learning.

3.1 Double Inverted Pendulum

The double inverted pendulum system [6] (see Figure 1) is a standard benchmark in the field of control. The system consists of a cart and a two-part pole attached to the cart. The system has six states which are cart position on a track, cart velocity, and the angles and the angular velocities of the two attached pendulums. The single control signal is the lateral force applied to the cart. The dynamical equations for the double inverted pendulum system can be found e.g. in [6], in this experiment a discrete system with a time step of $\Delta t = 0.05$ s was simulated using the MATLAB ordinary differential equation solver `ode23`.

To make sure that the learning scheme did not favour the proposed algorithm, standard backpropagation algorithm was used to learn an MLP network to model the system dynamics using a relatively small sample of 2000 input-output pairs. To make this problem more challenging, only the velocity and position of the cart and the angle of the upper pendulum were available as observations, and the rest of the state had to be inferred from these. Experiments were run on ten different data sets with 50 samples each using 5 different initialisations. The final results can be seen in Figure 1.

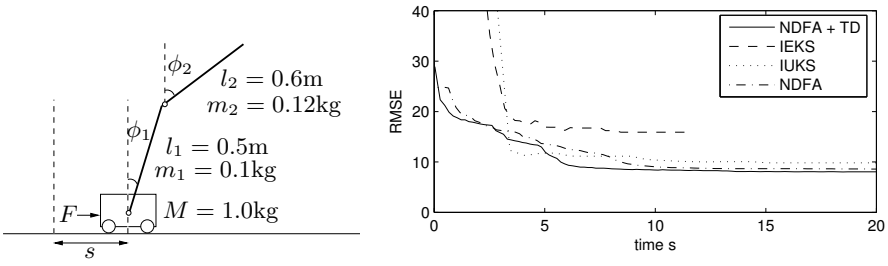

Fig. 1. Inference with the double inverted pendulum system. On the left the schematic of the system, on the right root mean square error plotted against computation time.

IEKS suffered from quite serious convergence problems with this data set. These problems were especially bad during the early iterations, but several runs failed to converge to a meaningful result even after the iteration limit was reached. IUKS performed somewhat better, but suffered from some stability problems too. The proposed method was much more robust and did not suffer from stability issues and also performed better on average than the two Kalman filter based algorithms. It should be noted, however, that in some experiments both IEKS and IUKS converged in only a few iterations, resulting in a superior performance compared to the proposed method. Therefore the problem with IEKS and IUKS may at least partially be related to poor choice of initialisations.

3.2 Speech Spectra

As a real world data set we used speech spectra. The data set consisted of 11200 21 dimensional samples which corresponds to 90 seconds of continuous human speech. The first 10000 samples were used to train a seven dimensional state-space model with the method from [1] and the rest of the data was used in the experiments. This data set poses a somewhat different problem from the double inverted pendulum system. The nonlinearities are not as strong as in the first experiment but the dimensionality of the observation and state spaces are higher, which emphasises the scalability of the methods.

The test data set was divided into three parts each consisting of 300 samples and all the algorithms were run for each data set with four random initialisations. The final results represent an average over both the different data sets and initialisations.

Since the true state is unknown in this experiment, the mean square error of the reconstruction of missing data was used to compare the different algorithms. Experiments were done with sets of both 3 and 30 consecutive missing samples. The ability to cope with missing values is very important when only partial

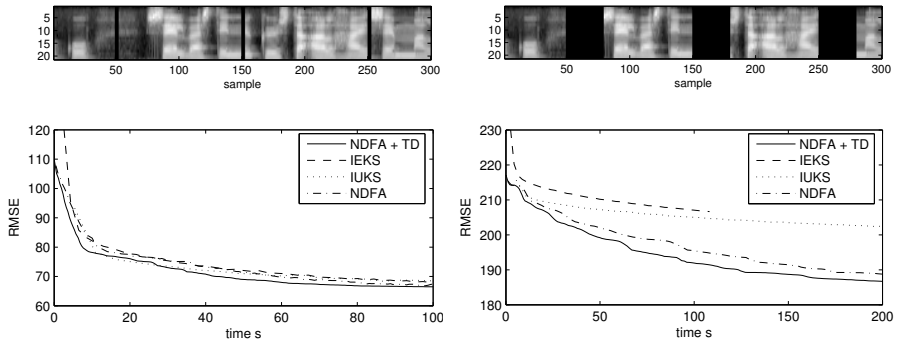

Fig. 2. Inference with the speech data and missing values. On the top one of the data sets used in the experiments (missing values marked in black), on the bottom root mean square error plotted against computation time. Left side figures use a small gap size, right side figures a large gap size.

observations are available or in the case of failures in the observation process. It also has interesting applications in the field of control as reported in [12].

Results can be seen in Figure 2. When missing values are present, especially in the case of the large gap size, the proposed algorithm performs clearly better than the rest of the compared algorithms. Compared to the double inverted pendulum data set, the stability issues with IEKS and IUKS were not as severe, but neither method could cope very well with long gaps of missing values.

4 Discussion and Conclusions

We proposed an algorithm for inference in nonlinear state-space models and compared it to some of the existing methods. The algorithm is based on minimising a variational Bayesian cost function and the novelty is in propagating the gradient through the state sequence. The results were slightly better than any of the comparison methods (IEKS and IUKS). The difference became large in a high-dimensional problem with long gaps in observations.

Our current implementation requires that the nonlinear mappings are modelled as multilayer perceptron networks. Part of the success of our method is due to a linearisation that is specialised to that case [7]. The idea presented in this paper applies in general.

When an algorithm is based on minimising a cost function, it is fairly easy to guarantee convergence. While the Kalman filter is clearly the best choice for inference in linear Gaussian models, the problem with many of the nonlinear generalisation (e.g. IEKS and IUKS) is that they cannot guarantee convergence. Even when the algorithms converge, convergence can be slow. A recent fix for convergence comes with a large computational cost [8] but this paper shows that stable inference can be fast, too.

While this paper concentrates on the case where nonlinear mappings and other model parameters are known, we aim at the case where they should be learned from the data [1]. Blind source separation involves a lot more iterations than the basic source separation. The requirements of a good inference algorithm change, too: There is always the previous estimate of the sources available and most of the time it is already quite accurate.

Acknowledgements

This work was supported in part by the Finnish Centre of Excellence Programme (2000-2005) under the project New Information Processing Principles and by the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. 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.
2. A. Cichocki, L. Zhang, S. Choi, and S.-I. Amari, "Nonlinear dynamic independent component analysis using state-space and neural network models," in *Proc. of the 1st Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, (Aussois, France, January 11-15), pp. 99–104, 1999.
3. B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
4. V. Koivunen, M. Enescu, and E. Oja, "Adaptive algorithm for blind separation from noisy time-varying mixtures," *Neural Computation*, vol. 13, pp. 2339–2357, 2001.
5. S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.
6. E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), pp. 221–280, New York: Wiley, 2001.
7. A. Honkela and H. Valpola, "Unsupervised variational Bayesian learning of nonlinear models," in *Advances in Neural Information Processing Systems 17* (L. Saul, Y. Weiss, and L. Bottou, eds.), pp. 593–600, Cambridge, MA, USA: MIT Press, 2005.
8. M. Psiaki, "Backward-smoothing extended Kalman filter," *Journal of Guidance, Control, and Dynamics*, vol. 28, Sep–Oct 2005.
9. A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
10. R. Everson and S. Roberts, "Particle filters for non-stationary ICA," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 23–41, Springer-Verlag, 2000.
11. R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.
12. T. Raiko and M. Tornio, "Learning nonlinear state-space models for control," in *Proc. Int. Joint Conf. on Neural Networks (IJCNN'05)*, (Montreal, Canada), pp. 815–820, 2005.