An Ensemble Learning Approach to Nonlinear Dynamic Blind Source Separation Using State-Space Models

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

Abstract - We propose a new method for learning a nonlinear dynamical state-space model in unsupervised manner. The proposed method can be viewed as a nonlinear dynamic generalization of standard linear blind source separation (BSS) or independent component analysis (ICA). Using ensemble learning, the method finds a nonlinear dynamical process which can explain the observations. The nonlinearities are modeled with multilayer perceptron networks. In ensemble learning, a simpler approximative distribution is fitted to the true posterior distribution by minimizing their Kullback-Leibler divergence. This also regularizes the studied highly ill-posed problem. In an experiment with a difficult chaotic data set, the proposed method found a much better model for the underlying dynamical process and source signals used for generating the data than the compared methods.

I. Introduction

The nonlinear state-space model (NSSM) is a very general and flexible model for time series data. The observation vectors $\mathbf{x}(t)$ are assumed to be generated from the hidden source vectors $\mathbf{s}(t)$ of a dynamical system through a nonlinear mapping $\mathbf{f}$ according to Eq. (1):

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

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

The sources follow the nonlinear dynamics $\mathbf{g}$ defined by Eq. (2). The terms $\mathbf{n}(t)$ and $\mathbf{m}(t)$ account for modeling errors and noise.

We propose an unsupervised method for learning nonlinear state-space models (1)-(2). Multi-layer perceptron (MLP) networks [1] are used to model the unknown nonlinear mappings $\mathbf{f}$ and $\mathbf{g}$, and the noise terms are assumed to be Gaussian. The MLP network provides an efficient parameterization for mappings in high-dimensional spaces, and it is a universal approximator for smooth functions. Similar models using a radial-basis function network [1] as nonlinearity have been proposed in [2], [3] and using an MLP network in [4].

In general, the nonlinear dynamical reconstruction problem addressed in this paper is severely ill-posed [5]. A wide variety of nonlinear transformations can be applied to the sources and then embedded in the functions $\mathbf{f}$ and $\mathbf{g}$, keeping the predictions unchanged. In this work, we apply ensemble learning to learn the parameters and hidden sources or states of the nonlinear state-space model. Ensemble learning is a recently developed practical method for fitting a parametric approximation to the exact posterior probability density function [6], [7]. We show how ensemble learning can be used to regularize the dynamical reconstruction problem by restricting the complexity of the posterior structure of the solution.

The proposed method is a nonlinear dynamical generalization of standard linear blind source separation (BSS) and independent component analysis (ICA) [8]. Several authors have recently applied ensemble learning or closely related Bayesian methods to the linear ICA problem [9], [10], [11], [12]. We have previously used ensemble learning also for nonlinear ICA [13], and shown how the approach can be extended for nonlinear dynamical models using nonlinear state-space models [14], [15]. A general discussion of nonlinear ICA and BSS with many references can be found in Chapter 17 of [8].

Even though the method presented in this paper can be regarded as a generalization of ICA, the recovered sources need not be independent. Our method tries to find the simplest possible explanation for the data, and hence avoids unnecessary dependencies between the recovered sources. If the process being studied cannot be described as a composition of one-dimensional independent processes, the method tries to split it to as small pieces as possible, as will be seen in the example in Section III.

II. Ensemble learning for the NSSM

This section briefly outlines the model and learning algorithm. A thorough presentation can be found in [16].

A. Model structure

The unknown nonlinear mappings $\mathbf{f}$ and $\mathbf{g}$ in (1) and (2) are modeled by multilayer perceptron (MLP) networks having one hidden layer of sigmoidal tanh nonlinearities. The function realized by the network can be written in vector notation as

$$\mathbf{f}(\mathbf{s}) = \mathbf{B}\tanh(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mathbf{b} \tag{3}$$

where the tanh nonlinearity is applied componentwise. $\mathbf{A}$ and $\mathbf{B}$ are the weight matrices and $\mathbf{a}$ and $\mathbf{b}$ the bias vectors of the network. The function $\mathbf{g}$ has a similar structure except that the MLP network is used to model only the change in the source values:

$$\mathbf{g}(\mathbf{s}) = \mathbf{s} + \mathbf{D}\tanh(\mathbf{C}\mathbf{s} + \mathbf{c}) + \mathbf{d} \tag{4}$$

The noise terms $\mathbf{n}(t)$ and $\mathbf{m}(t)$ are assumed to be Gaussian and white, so that the values at different time instants and different components at the same time instant are independent. Let us denote the observation set by $\boldsymbol{X} = (\mathbf{x}(1), \ldots, \mathbf{x}(T))$, source set by $\boldsymbol{S} = (\mathbf{s}(1), \ldots, \mathbf{s}(T))$ and all the model parameters by $\boldsymbol{\theta}$. The likelihood of the observations defined by the model can then be written as

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{\theta}) &= \prod_{i,t} p(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta}) \\ &= \prod_{i,t} N(x_i(t);\ f_i(\mathbf{s}(t)), \exp(2v_i)) \end{aligned} \tag{5}$$

where $N(x;\ \mu, \sigma^2)$ denotes a Gaussian distribution over x with mean μ and variance σ^2, $f_i(\mathbf{s}(t))$ denotes the ith component of the output of $\mathbf{f}$, and v_i is a hyperparameter specifying the noise variance. The probability $p(\boldsymbol{S}|\boldsymbol{\theta})$ of the sources $\boldsymbol{S}$ is specified similarly using the function $\mathbf{g}$. All the parameters of the model have hierarchical Gaussian priors. For example the noise parameters v_i of different components of the data share a common prior [13], [16].

The parameterization of the variances through $\exp(2v)$ where $v \sim N(\alpha, \beta)$ corresponds to log-normal distribution of the variance. The inverse gamma distribution would be the conjugate prior in this case, but log-normal distribution is close to it, and it is easier to build a hierarchical prior using log-normal distributions than inverse gamma distribution.

B. Posterior approximation and regularization

The goal of ensemble learning is to fit a parametric approximating distribution $q(\boldsymbol{\theta}, \boldsymbol{S})$ to the true posterior $p(\boldsymbol{\theta}, \boldsymbol{S}|\boldsymbol{X})$. The misfit is measured by the Kullback-Leibler divergence between the approximation and the true posterior:

$$D(q(\boldsymbol{S}, \boldsymbol{\theta})||p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})) = \mathrm{E}_{q(\boldsymbol{S},\boldsymbol{\theta})}\left[\log \frac{q(\boldsymbol{S}, \boldsymbol{\theta})}{p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})}\right] \tag{6}$$

where the expectation is calculated over the approximation $q(\boldsymbol{S}, \boldsymbol{\theta})$. The Kullback-Leibler divergence is always nonnegative. It attains its minimum of zero if and only if the two distributions are equal.

The posterior distribution can be written as $p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X}) = p(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X})/p(\boldsymbol{X})$. The normalizing term $p(\boldsymbol{X})$ cannot usually be evaluated, and the actual cost function used in ensemble learning is thus

$$\begin{aligned} C &= \mathrm{E}\left[\log \frac{q(\boldsymbol{S}, \boldsymbol{\theta})}{p(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X})}\right] \\ &= D(q(\boldsymbol{S}, \boldsymbol{\theta})||p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})) - \log p(\boldsymbol{X}) \geq -\log p(\boldsymbol{X}). \end{aligned} \tag{7}$$

Usually the joint probability $P(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X})$ is a product of simple terms due to the definition of the model. In this case $p(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X}) = p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{\theta})p(\boldsymbol{S}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ can be written as a product of univariate Gaussian distributions.

The cost function can be minimized efficiently if a suitably simple factorial form for the approximation is chosen. We use $q(\boldsymbol{\theta}, \boldsymbol{S}) = q(\boldsymbol{\theta})q(\boldsymbol{S})$, where $q(\boldsymbol{\theta}) = \prod_i q(\theta_i)$ is a product of univariate Gaussian distributions. Hence the distribution for each parameter θ_i is parameterized by its mean $\overline{\theta}_i$ and variance $\widetilde{\theta}_i$. These are the variational parameters of the distribution to be optimized.

The approximation $q(\boldsymbol{S})$ takes into account posterior dependences between the values of sources at consecutive time instants. The approximation can be written as a product $q(\boldsymbol{S}) = \prod_i [q(s_i(1)) \prod_t q(s_i(t)|s_i(t-1))]$. The value $s_i(t)$ depends only on $s_i(t-1)$ at previous time instant, not on the other $s_j(t-1)$ with $j \neq i$. The distribution $q(s_i(t)|s_i(t-1))$ is a Gaussian with mean that depends linearly on the previous value as in $\mu_i(t) = \overline{s}_i(t) + \breve{s}_i(t-1,t)(s_i(t-1) - \overline{s}_i(t-1))$, and variance $\mathring{s}_i(t)$. The variational parameters of the distribution are $\overline{s}_i(t)$, $\breve{s}_i(t-1,t)$ and $\mathring{s}_i(t)$.

A positive side-effect of the restrictions on the approximating distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$ is that the nonlinear dynamical reconstruction problem is regularized and becomes well-posed. With linear $\mathbf{f}$ and $\mathbf{g}$, the posterior distribution of the sources $\boldsymbol{S}$ would be Gaussian, while nonlinear $\mathbf{f}$ and $\mathbf{g}$ result in non-Gaussian posterior distribution. Restricting $q(\boldsymbol{S})$ to be Gaussian therefore favors smooth mappings and regularizes the problem. This still leaves a rotational ambiguity which is solved by discouraging the posterior dependences between $s_i(t)$ and $s_j(t-1)$ with $j \neq i$.

C. Evaluating the cost function and updating the parameters

The parameters of the approximating distribution are optimized with gradient based iterative algorithms. During one sweep of the algorithm all the parameters are updated once, using all the available data. One sweep consists of two different phases. The order of the computations in these two phases is the same as in standard supervised back-propagation [1] but otherwise the algorithm is different. In the forward phase, the distributions of the outputs of the MLP networks are computed from the current values of the inputs, and the value of the cost function is evaluated. In the backward phase, the partial derivatives of the cost function with respect to all the parameters are fed back through the MLPs and the parameters are updated using this information.

When the cost function (7) is written for the model defined above, it splits into a sum of simple terms. Most of the terms can be evaluated analytically. Only the terms involving the outputs of the MLP networks cannot be computed exactly. To evaluate those terms, the distributions of the outputs of the MLPs are calculated using a truncated Taylor series approximation for the MLPs. This procedure is explained in detail in [13], [16]. In the feedback phase, these computations are simply inverted to evaluate the gradients.

Let us denote the two parts of the cost function (7) arising from the denominator and numerator of the logarithm respectively by $C_p = \mathrm{E}_q[-\log p]$ and $C_q = \mathrm{E}_q[\log q]$. The term C_q is a sum of negative entropies of Gaussians, and has the form

$$C_q = \sum_i -\frac{1}{2}[1 + \log(2\pi\widetilde{\theta}_i)] + \sum_{t,i} -\frac{1}{2}[1 + \log(2\pi\mathring{s}_i(t))]. \tag{8}$$

The terms in the corresponding sum for C_p are somewhat more complicated but they are also relatively simple expectations over Gaussian distributions [13], [14], [16].

An update rule for the posterior variances $\widetilde{\theta}_i$ is obtained by differentiating (7) with respect to $\widetilde{\theta}_i$, yielding [13], [16]

$$\frac{\partial C}{\partial \widetilde{\theta}_i} = \frac{\partial C_p}{\partial \widetilde{\theta}_i} + \frac{\partial C_q}{\partial \widetilde{\theta}_i} = \frac{\partial C_p}{\partial \widetilde{\theta}_i} - \frac{1}{2\widetilde{\theta}_i} \tag{9}$$

Equating this to zero yields a fixed-point iteration:

$$\widetilde{\theta}_i = \left[2\frac{\partial C_p}{\partial \widetilde{\theta}_i}\right]^{-1} \tag{10}$$

The posterior means $\overline{\theta}_i$ can be estimated from the approximate Newton iteration [13], [16]

$$\overline{\theta}_i \leftarrow \overline{\theta}_i - \frac{\partial C_p}{\partial \overline{\theta}_i}\left[\frac{\partial^2 C}{\partial \overline{\theta}_i^2}\right]^{-1} \approx \overline{\theta}_i - \frac{\partial C_p}{\partial \overline{\theta}_i}\widetilde{\theta}_i \tag{11}$$

The posterior means $\overline{s}_i(t)$ and variances $\mathring{s}_i(t)$ of the sources are updated similarly. The update rule for the posterior linear dependences $\breve{s}_i(t-1,t)$ is also derived by solving the zero of the gradient [14], [16].

D. Learning scheme

In general the learning proceeds in batches. After each sweep through the data the distributions $q(\boldsymbol{S})$ and $q(\boldsymbol{\theta})$ are updated. There are slight changes to the basic learning scheme in the beginning of training. The hyperparameters governing the distributions of other parameters are not updated to avoid pruning away parts of the model that do not seem useful at the moment. The data is also embedded to have multiple time-shifted copies to encourage the emergence of sources representing the dynamics. The embedded data is given by $\mathbf{z}^T(t) = [\mathbf{x}^T(t-d), \ldots, \mathbf{x}^T(t+d)]$ and it is used for the first 500 sweeps.

At the beginning, the posterior means of most of the parameters are initialized to random values. The posterior variances are initialized to small constant values. The posterior means of the sources $\mathbf{s}(t)$ are initialized using a suitable number of principal components of the embedded data $\mathbf{z}(t)$. They are frozen to these values for the first 50 sweeps, during which only the MLP networks $\mathbf{f}$ and $\mathbf{g}$ are updated. Updates of the hyperparameters begin after the first 100 sweeps.

III. Experimental results

The dynamical process used to test the NSSM method was a combination of three independent dynamical systems. The total dimension of the state space was eight; the eight original source processes are shown in Fig. 1a. Two of the dynamical systems were independent Lorenz systems, each having a three-dimensional nonlinear dynamics. The third dynamical system was a harmonic oscillator which has a linear two-dimensional dynamics. The three uppermost source signals in Figure 1a correspond to the first Lorenz process, the next three sources correspond to the second Lorenz process, and the last two ones to the harmonic oscillator.

The 10-dimensional data vectors $\mathbf{x}(t)$ used in learning are depicted in Fig. 1c. They were generated by nonlinearly mixing the five linear projections of the original sources shown in Fig. 1b, and then adding some Gaussian noise. The standard deviations of the signal and noise are 1 and 0.1, respectively. The nonlinear mixing was carried out using an MLP network having randomly chosen weights and using $\sinh^{-1}$ nonlinearity. The same mixing was used in one of the experiments in [13]. The dimension of the original state space was reduced to five in order to make the problem more challenging. Now the dynamics of the observations is needed to reconstruct the original sources as only five out of eight dimensions are visible instantaneously.

The posterior means of the sources of the estimated process after 1,000,000 sweeps are shown in Fig. 1d together with a predicted continuation. This can be compared with the continuation of the original process in Fig. 1a. Figure 2 shows a three-dimensional plot of the state trajectory of one of the original Lorenz processes (top) and the corresponding plot for the estimated sources of the same process (bottom). Note that the MLP networks modeling $\mathbf{f}$ and $\mathbf{g}$ could have represented any rotation of the source space. Hence separation of the independent processes results only from the form of the posterior approximation.

The quality of the estimate of the underlying process was tested by studying the prediction accuracy for new samples. It should be noted that since the Lorenz processes are chaotic, the best that any method can do is to capture its general long-term behavior - exact numerical prediction is impossible.

The proposed approach was compared to nonlinear autoregressive (NAR) model which makes the predictions directly in the observation space:

$$\mathbf{x}(t) = \mathbf{h}(\mathbf{x}(t-1), \ldots, \mathbf{x}(t-d)) + \mathbf{n}(t). \qquad (12)$$

The nonlinear mapping $\mathbf{h}(\cdot)$ was again modeled by an MLP network, but now standard back-propagation was applied in learning. The best performance was given by an MLP network with 20 inputs and one hidden layer of 30 neurons, and the number of delays was $d = 10$. The dimension of the inputs to the MLP network $\mathbf{h}(\cdot)$ was compressed from 100 to 20 using standard PCA. Figure 3 shows the results, averaged over 100 Monte Carlo simulations. The results of the NSSM are from several different simulations that used different initializations. At each stage, the one with the smallest cost function value was chosen. Low cost function values seem to correlate with good prediction performance even though this would not necessarily have to be so.

After the first 7500 sweeps the NSSM method was roughly comparable with the NAR-based method in predicting the process $\mathbf{x}(t)$. The performance improved considerably when learning was continued. The final predictions given by NSSM after 1,000,000 sweeps are excellent up to the time $t = 1013$ and good up to $t = 1022$, while the NAR method is quite inaccurate already after $t \geq 1003$. Here the prediction started at time $t = 1000$. We have also experimented with recurrent neural networks, which provide slightly better results than the NAR model but significantly worse than the proposed NSSM method.

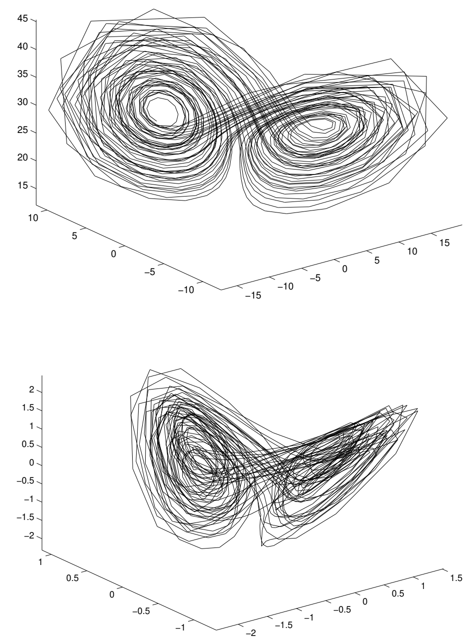

Fig. 2. The original sampled Lorenz process (top) and the corresponding three components of the estimated process (bottom).

IV. Discussion

The NSSM method is in practice able to learn up to about 15-dimensional latent spaces. This is clearly more than many other methods can handle. Currently learning requires a lot of computer time, taking easily days. Finding means to speed up it is therefore an important future research topic. The block approach presented in [17] has smaller computational complexity, and it could help to reduce the learning time, but we have not yet tried it with a NSSM.

The proposed method has several potential applications. In addition to the experiments reported here, essentially the same method has already been successfully

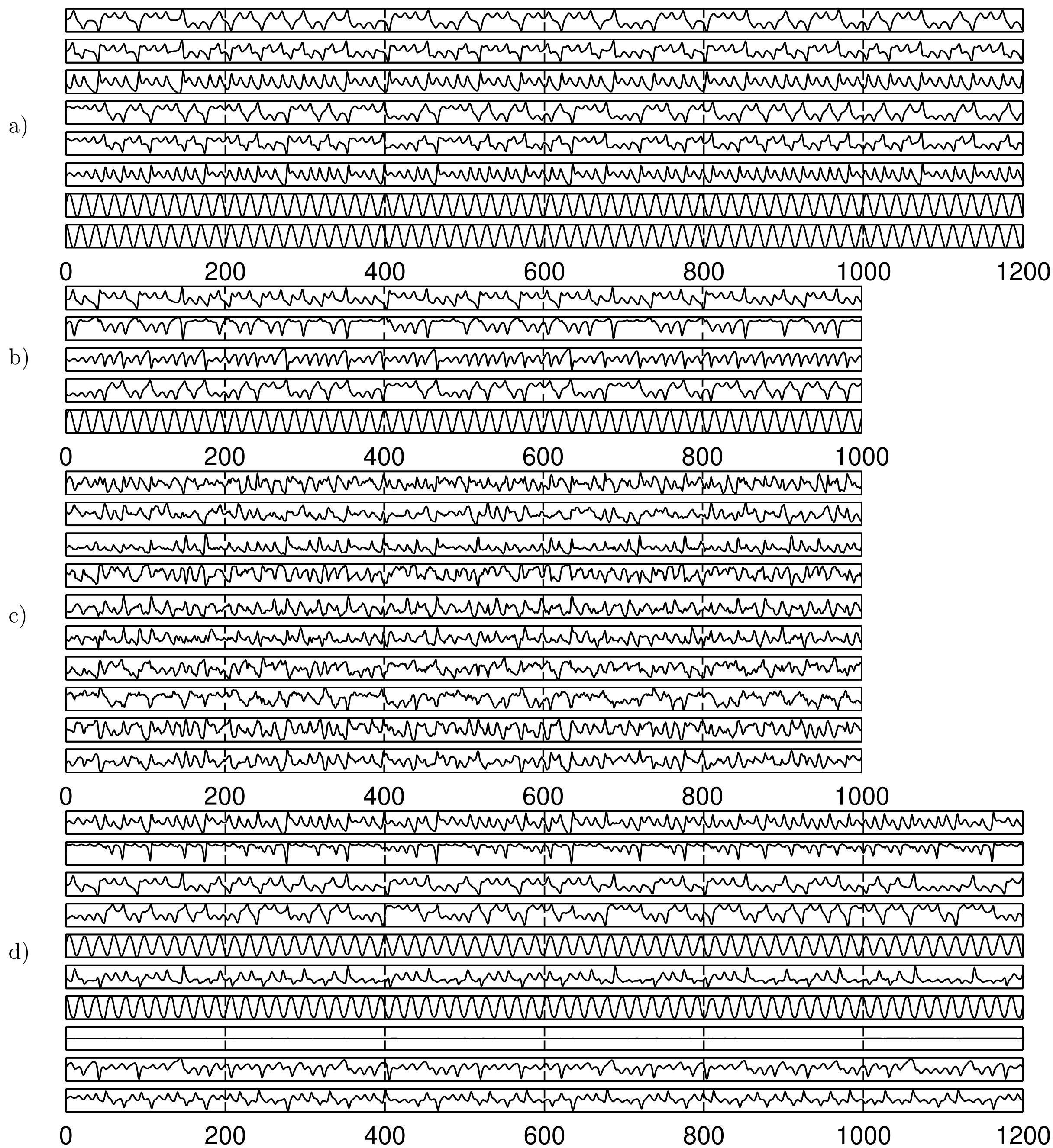

Fig. 1. a) The eight source signals $\mathbf{s}(t)$ of the three original dynamical processes. b) The five linear projections of the sources. c) The 1000 ten dimensional data vectors $\mathbf{x}(t)$ generated by mixing the projection b) nonlinearly and adding noise. d) The states of the estimated process ($t \leq 1000$) and predicted continuation ($t > 1000$). They can be compared with the original process and its continuation in a).

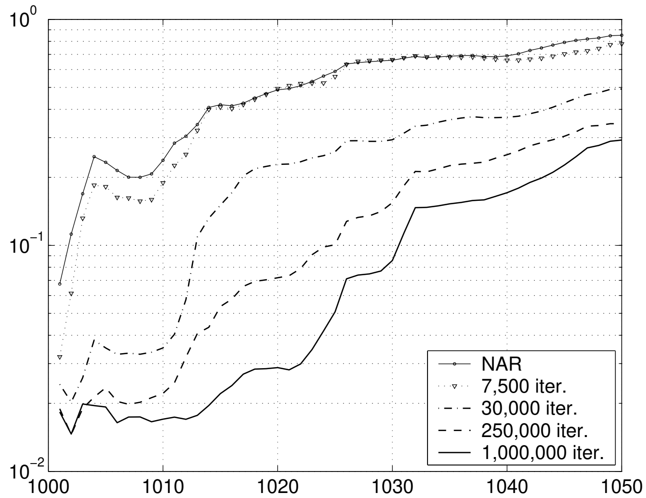

Fig. 3. The average cumulative squared prediction error for the nonlinear autoregressive (NAR) model (solid line with dots) and for our dynamic algorithm with different numbers of sweeps.

applied to detecting changes in the modeled process in [18] and to analyzing magnetoencephalographic (MEG) signals measured from the human brain in [19].

Acknowledgments

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.

References

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

[2] Z. Ghahramani and S. Roweis, “Learning nonlinear dynamical systems using an EM algorithm,” in *Advances in Neural Information Processing Systems 11* (M. Kearns, S. Solla, and D. Cohn, eds.), (Cambridge, MA, USA), pp. 599–605, The MIT Press, 1999.

[3] S. Roweis and Z. Ghahramani, “An EM algorithm for identification of nonlinear dynamical systems,” in *Kalman Filtering and Neural Networks* (S. Haykin, ed.). To appear.

[4] T. Briegel and V. Tresp, “Fisher scoring and a mixture of modes approach for approximate inference and learning in nonlinear state space models,” in *Advances in Neural Information Processing Systems 11* (M. Kearns, S. Solla, and D. Cohn, eds.), (Cambridge, MA, USA), pp. 403–409, The MIT Press, 1999.

[5] S. Haykin and J. Principe, “Making sense of a complex world,” *IEEE Signal Processing Magazine*, vol. 15, no. 3, pp. 66–81, May 1998.

[6] G. Hinton and D. van Camp, “Keeping neural networks simple by minimizing the description length of the weights,” in *Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, (Santa Cruz, CA, USA), pp. 5–13, 1993.

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

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

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

[10] J. Miskin and D. MacKay, “Ensemble Learning for blind source separation,” in *Independent Component Analysis: Principles and Practice* (S. Roberts and R. Everson, eds.), pp. 209–233, Cambridge University Press, 2001.

[11] R. Choudrey and S. Roberts, “Flexible Bayesian independent component analysis for blind source separation,” in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 90–95, 2001.

[12] P. Højen-Sørensen, L. K. Hansen, and O. Winther, “Mean field implementation of Bayesian ICA,” in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 439–444, 2001.

[13] H. Lappalainen and A. Honkela, “Bayesian nonlinear independent component analysis by multi-layer perceptrons,” in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 93–121, Springer-Verlag, 2000.

[14] H. Valpola, “Unsupervised learning of nonlinear dynamic state-space models,” Tech. Rep. A59, Lab of Computer and Information Science, Helsinki University of Technology, Finland, 2000.

[15] H. Valpola, A. Honkela, and J. Karhunen, “Nonlinear static and dynamic blind source separation using ensemble learning,” in *Proc. Int. Joint Conf. on Neural Networks (IJCNN'01)*, (Washington D.C., USA), pp. 2750–2755, 2001.

[16] H. Valpola and J. Karhunen, “An unsupervised ensemble learning method for nonlinear dynamic state-space models,” 2001. Manuscript submitted to Neural Computation.

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

[18] A. Iline, H. Valpola, and E. Oja, “Detecting process state changes by nonlinear blind source separation,” in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 704–709, 2001.

[19] J. Särelä, H. Valpola, R. Vigário, and E. Oja, “Dynamical factor analysis of rhythmic magnetoencephalographic activity,” in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, (San Diego, USA), pp. 451–456, 2001.