Chapter 4

Bayesian ensemble learning of generative models

Harri Valpola, Antti Honkela, Juha Karhunen, Tapani Raiko, Xavier Giannakopoulos, Alexander Ilin, Erkki Oja

4.1 Bayesian modeling and ensemble learning

In unsupervised learning, the goal is to build a model which captures the statistical regularities in the observed data and provides a compact and meaningful representation for the data. From such a representation it is often much easier to understand the basic characteristics of the data than directly from the raw data.

Unsupervised learning methods are often based on a generative approach where the goal is to find a specific model which explains how the observations were generated. It is assumed that there exist certain source signals (also called factors, latent or hidden variables, or hidden causes) which have generated the observed data through an unknown mapping. The goal of generative learning is to identify both the source signals 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 good nonlinear model is a challenging task, because it is typically computationally much more demanding than for linear models, and the choice of an appropriate specific model is often difficult.

In Bayesian data analysis and estimation methods, all the uncertain quantities are modeled in terms of their joint probability density function (pdf). The key principle is to construct the joint posterior pdf for all the unknown quantities in a model, given the data sample. This posterior density 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].

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

$$p(\boldsymbol{\theta}|X,\mathcal{H}) = \frac{p(X|\boldsymbol{\theta},\mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{p(X|\mathcal{H})} \tag{4.1}$$

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 choice of a multilayer perceptron (MLP) network, 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 distribution $p(\boldsymbol{\theta}|X,\mathcal{H})$. 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].

Instead of searching for some point estimates, the correct Bayesian procedure is to perform estimation by averaging over the posterior distribution $p(\boldsymbol{\theta}|X,\mathcal{H})$. This means that the estimates will be sensitive to regions where the probability mass is large instead of being sensitive to high values of the pdf [4]. One can even make use of the complete set of models by computing the predicted values as averages over the predictions given by several models, weighted by their respective probabilities. Such a procedure appropriately 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 models 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 (4.1) 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. Ensemble learning, also called variational Bayes, is a recently developed [2, 3] approximative fully Bayesian method, which has become popular because of its good properties. It uses the Kullback-Leibler (KL) information between two probability distributions $q(v)$ and $p(v)$. The KL information is defined by the cost function

$$\mathcal{C}(q \parallel p) = \int q(v) \ln \frac{q(v)}{p(v)} dv \tag{4.2}$$

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

A key idea in ensemble learning is to minimize the misfit between the actual posterior pdf and its parametric approximation using the KL information. 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 ensemble learning is that it avoids overfitting which would be a difficult problem if ML or MAP estimates were used. Ensemble learning allows one to select a model having appropriate complexity, making often possible to infer the correct number of sources or latent variables. It has provided good estimation results in the very difficult unsupervised (blind) learning problems that we have considered.

Ensemble learning is closely related to information theoretic approaches which minimize the description length of the data, because the description length is defined to be the minus logarithm of the probability. Minimal description length thus means maximal probability. In the probabilistic framework, we try to find the sources or factors and the nonlinear mapping which most probably correspond to the observed data. In the information theoretic framework, this corresponds to finding the sources and the mapping that can generate the observed data and have the minimum total complexity. Ensemble learning has been discussed from information theoretic point of view in [2].

In the following subsections, we first consider the use of ensemble learning in nonlinear principal component (factor) analysis as well as in nonlinear independent component analysis. The static model used in nonlinear factor analysis is then extended to nonlinear dynamic state-space models. In these problems, we have employed multilayer perceptron networks as a flexible family of models. We then discuss construction of hierarchic models from a set of basic blocks in context with ensemble learning, and present applications of the developed Bayesian methods to inferring missing values from data, to detection of changes in process states, and to analysis of biomedical MEG data.

References

[1] C. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[2] G. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proc. of the 6th Annual ACM Conf. on Computational Learning Theory*, pages 5–13, Santa Cruz, California, USA, 1993.

[3] D. MacKay. Developments in Probabilistic Modelling with Neural Networks – Ensemble Learning. In *Neural Networks: Artificial Intelligence and Industrial Applications. Proc. of the 3rd Annual Symposium on Neural Networks*, pages 191–198, Nijmegen, Netherlands, 1995.

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

4.2 Nonlinear factor analysis and independent component analysis

The linear principal and independent component analysis (PCA and ICA) model the data as having been generated by independent sources through a linear mapping. The difference between the two is that PCA restricts the distribution of the sources to be Gaussian, whereas ICA does not, in general, restrict the distribution of the sources.

We have applied ensemble learning to nonlinear counterparts of PCA and ICA where the generative mapping form sources to data is not restricted to be linear [1, 2, 3]. The general form of the model is

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) + \mathbf{n}(t)\,. \tag{4.3}$$

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 parametrised mapping from source space to observation space. We have used multi-layer perceptron (MLP) network with tanh-nonlinearities to model the mapping $\mathbf{f}$:

$$\mathbf{f}(\mathbf{s}) = \mathbf{B}\tanh(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mathbf{b}\,. \tag{4.4}$$

The mapping $\mathbf{f}$ is thus parametrized by the matrices $\mathbf{A}$ and $\mathbf{B}$ and bias vectors $\mathbf{a}$ and $\mathbf{b}$. MLP networks are well suited for nonlinear PCA and ICA. 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, close to linear mappings with them. This makes it possible to learn high dimensional nonlinear representations in practice.

Traditionally MLP networks have been used for supervised learning where both the inputs and the desired outputs are known. Here sources correspond to inputs and observations correspond to desired outputs. The sources are unknown and therefore learning is unsupervised.

Usually the linear PCA and ICA models do not have an explicit noise term $\mathbf{n}(t)$ and the model is thus simply $\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) = \mathbf{A}\mathbf{s}(t) + \mathbf{a}$, where $\mathbf{A}$ is a mixing matrix and $\mathbf{a}$ is a bias vector. The corresponding PCA and ICA models which include the noise term are often called factor analysis and independent factor analysis (FA and IFA) models. The nonlinear models discussed here can therefore also be called nonlinear factor analysis and nonlinear independent factor analysis models.

Dimension reduction by nonlinear factor analysis

Just as their linear counterparts, the nonlinear versions of PCA and ICA can be used for instance in dimension reduction and feature extraction. The difference between linear and nonlinear PCA is depicted in Figure 4.1. In the linear PCA the data is described with a linear coordinate system whereas in the nonlinear PCA the coordinate system is nonlinear. The nonlinear PCA and ICA can be used for similar tasks as their linear counterparts, but they can be expected to capture the structure of the data better if the data points lie in a nonlinear manifold instead of a linear subspace.

Model order selection is an important issue is real applications: how many sources are there, how complex nonlinearities should be used, etc. One benefit of ensemble learning is that the cost function can be used for optimizing model structure by simply minimizing the cost. A sample case is shown in Figure 4.2. The data was created by nonlinearly mixing five Gaussian random sources. Ensemble learning is actually able to shut down unused parts of the model. The weights of the MLP network corresponding to the extra sources converge to zero which means that the extra sources are effectively pruned away from the model. This explains why the cost function saturates after five sources.

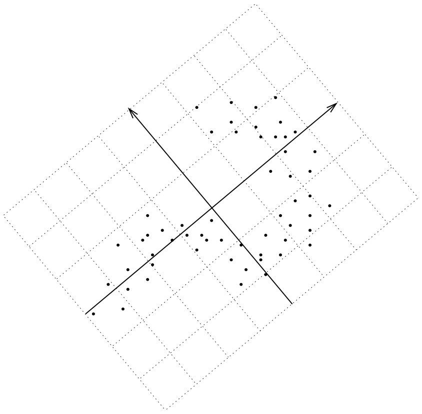

Figure 4.1: On the left hand side the data is described with a linear coordinate system. On the right hand side the coordinate system is nonlinear

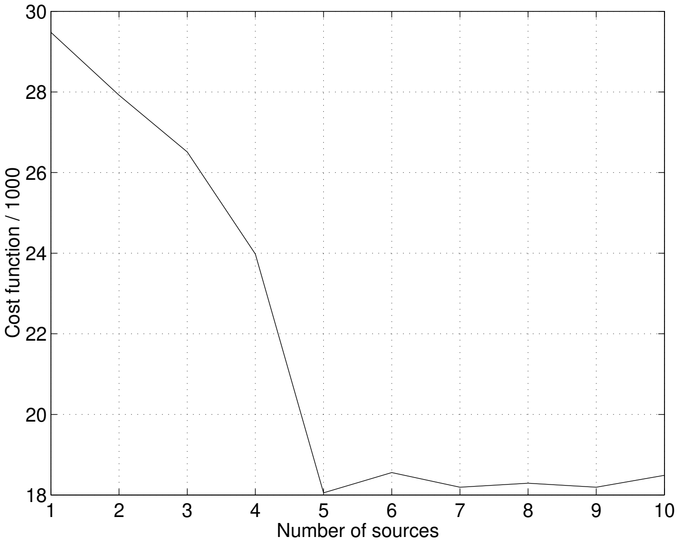

Figure 4.2: The value of the cost function is shown as a function of the number of sources. The MLP network had 30 hidden neurons. Ten different initialisations were tested to find the minimum value for each number of sources. The cost function saturates after five sources and the deviations are due to different random initialisation of the network

Nonlinear factor analysis is able to capture nonlinear structure in real data sets as can be seen in Figure 4.3. This data set consists of 2480 samples from 30 time series measured by different sensors from an industrial pulp process. An expert has preprocessed the signals by roughly compensating for time lags of the process which originate from the finite speed of pulp flow through the process. It appears that the data is quite nonlinear since the nonlinear factor analysis is able to explain as much data with 10 components as the linear factor analysis with 21 components.

Nonlinear independent factor analysis

Like in the linear case, Gaussian model for the sources results in rotational indeterminacy of the source space, whereas a non-Gaussian model for the sources fixes the rotation and

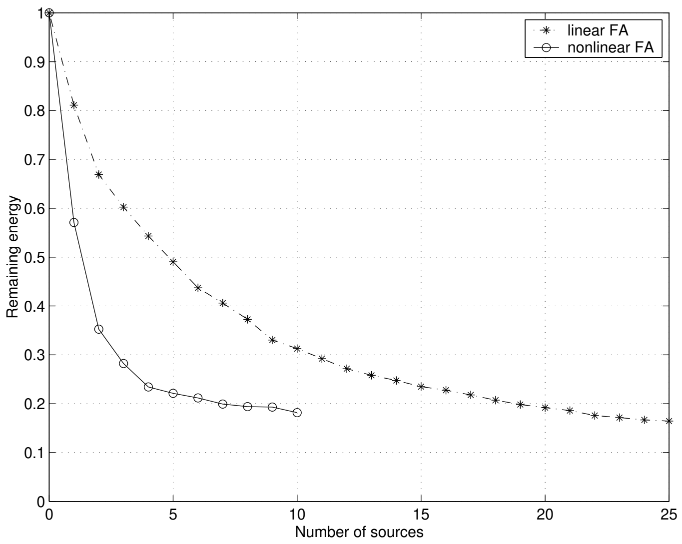

Figure 4.3: The graph shows the remaining energy in the process data as a function of the number of extracted components in linear and nonlinear factor analysis

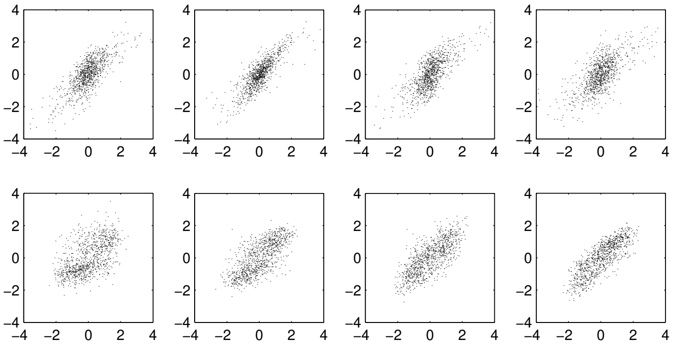

Figure 4.4: Original sources are on the x-axis of each scatter plot and the sources estimated by a linear ICA are on the y-axis. Signal to noise ratio is 0.7 dB

makes it possible to identify independent sources.

We have modeled the distribution of the sources by a mixture of Gaussian distributions. This density model is well suited for generative models and moreover, given enough Gaussians in the mixture, any density can be modeled by arbitrary accuracy using it.

Figures 4.4 and 4.5 illustrate the difference between linear and nonlinear ICA in the case where the mixing is nonlinear. The data set consisted of 1000 20-dimensional vectors which were created by nonlinearly mixing eight non-Gaussian independent random sources. Nonlinear model is clearly required in order to capture to underlying nonlinear manifold.

References

[1] Harri Lappalainen and Antti 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.

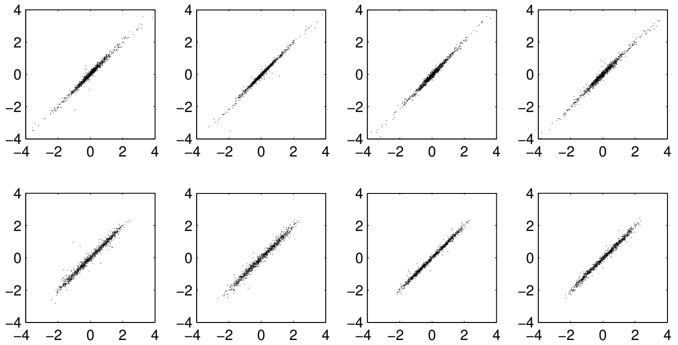

Figure 4.5: Nonlinear independent factor analysis is able to retrieve the original signals with small error. Signal to noise ratio is 17.3 dB

[2] Harri Valpola. Nonlinear independent component analysis using ensemble learning: Theory. In Petteri Pajunen and Juha Karhunen, editors, *Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2000*, pages 251–256, Helsinki, Finland, June 2000.

[3] Harri Valpola, Xavier Giannakopoulos, Antti Honkela, and Juha Karhunen. Nonlinear independent component analysis using ensemble learning: Experiments and discussion. In Petteri Pajunen and Juha Karhunen, editors, *Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2000*, pages 351–356, Helsinki, Finland, June 2000.

4.3 Nonlinear dynamic state-space models

In many cases, measurements originate from a dynamic system and form time series. In such cases, it is often useful to model the dynamics in addition to the instantaneous observations. We have extended the nonlinear factor analysis model by adding a nonlinear model for the dynamics of the sources $\mathbf{s}(t)$ [1, 2, 3]. This results in a state-space model where the sources can be interpreted as the internal state of the underlying generative process.

The model consists of the following set of equations:

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) + \mathbf{n}(t) \tag{4.5}$$

$$\mathbf{s}(t) = \mathbf{g}(\mathbf{s}(t-1)) + \mathbf{m}(t)\,, \tag{4.6}$$

where $\mathbf{x}(t)$ are the observations, $\mathbf{s}(t)$ are the sources (states), $\mathbf{n}(t)$ and $\mathbf{m}$ are Gaussian observation and process noise and $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are the nonlinear functions modeling the observations and dynamics, respectively.

As in nonlinear factor analysis, the nonlinear functions are modeled by MLP networks. The mapping $\mathbf{f}$ has the same functional form (4.4). 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:

$$\mathbf{g}(\mathbf{s}(\mathbf{t}-\mathbf{1})) = \mathbf{s}(t-1) + \mathbf{D}\tanh[\mathbf{C}\mathbf{s}(t-1) + \mathbf{c}] + \mathbf{d}\,. \tag{4.7}$$

An important advantage of the proposed new method 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. A process monitoring application is discussed in Section 4.5 in more detail.

References

[1] H. Valpola. Unsupervised learning of nonlinear dynamic state-space models. Technical Report A59, Lab of Computer and Information Science, Helsinki University of Technology, Finland, 2000.

[2] H. Valpola. *Bayesian Ensemble Learning for Nonlinear Factor Analysis*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 2000. Published in Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 108.

[3] H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 2002. Submitted, under revision.

4.4 Building blocks for ensemble learning

The methods for unsupervised learning of nonlinear latent variable models by ensemble learning introduced in Sections 4.2 and 4.3 represent a significant improvement in existing techniques in their ability to learn high-dimensional nonlinear representations of the data. However, the ability to scale these methods is limited by computational complexity which is roughly proportional to the number of data samples and the complexity of multiplying the matrices $\mathbf{A}$ and $\mathbf{B}$ in (4.4).

In order to be able to work with very large scale models, the computational complexity should be linear in the number of data samples and connections in the model. This can be achieved by utilizing the methods introduced in [1]. The idea is to construct large models from standardized building blocks which can be connected rather freely and can be learned by ensemble learning with local learning rules. Each block only needs to communicate with its neighbors which results in linear computational complexity.

The building blocks include continuous and discrete variables, summation, addition, nonlinearity and switching. Ensemble learning provides a cost function which can be used for updating the variables as well as optimising the model structure. The blocks are designed to fit together and to yield efficient update rules.

We have started to design a C++ library which can utilize these building blocks. The derivation of the cost function and learning rules is largely automatic which means that the user only needs to define the connections between the blocks. This research will be published in forthcoming papers.

References

[1] H. Valpola, T. Raiko, and J. Karhunen. Building blocks for hierarchical latent variable models. In *Proc. 3rd Int. Workshop on Independent Component Analysis and Signal Separation (ICA2001)*, pages 710–715, San Diego, California, December 2001.

4.5 Applications

In this section, applications of nonlinear factor analysis (Section 4.2) and nonlinear state-space models (Section 4.3) are discussed. Application of the nonlinear state-space model to biomedical magnetoencephalographic data is discussed in Section 6.3.

Missing values

Generative models can usually easilly deal with missing observations. For instance in self-organizing maps (SOM) the winning neuron can be found based on those observations that are available. The generative model can also be used to fill in the missing values. This way unsupervised learning can be used for a similar task as supervised learning as illustrated in Figure 4.6. Both the inputs and desired outputs of the learning data are treated equally. When a generative model for the combined data is learned, it can be used to reconstruct the missing outputs for the test data. The scheme used in unsupervised learning is more flexible because any part of the data can act as the cue which is used to complete the rest of the data. In supervised learning, the inputs always act as the cue.

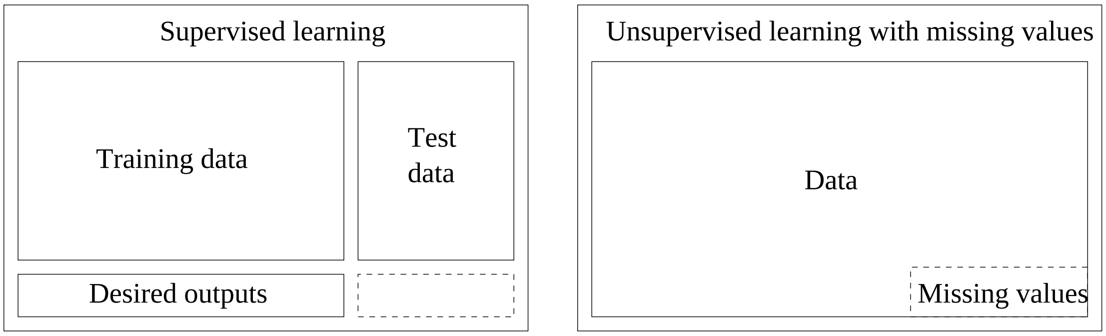

Figure 4.6: Unsupervised learning can be used for supervised learning by considering the outputs of the test data as missing values.

The quality of the reconstructions provides insight to the properties of different unsupervised models. The ability of self-organizing maps, linear principal component analysis and nonlinear factor analysis to reconstruct the missing values of various data sets have been studied in [2].

Figure 4.7 shows the results of reconstructing missing values in speech spectra. On the left, the reconstructed spectra are shown for a case where data were missing in patches and the data is not permutated. This corresponds to the upper row of the plots on the right. The other rows on the right show results for cases where the data was permutated and/or data was missing randomly instead of in patches. In the permutated case the learning data contained samples which were similar to the test data with missing values. This task does not require generalization but rather memorization of the learned data. SOM performs the best in this task because it has the largest amount of parameters in the model.

The task where the data was missing randomly and not in patches of several neighboring frequencies does not require a very nonlinear model but rather an accurate representation of a high-dimensional latent space. Linear and nonlinear factor analysis perform better than SOM whose parametrization is not well suited for very high-dimensional latent spaces. The conclusion of these experiments was that in many respects the properties of

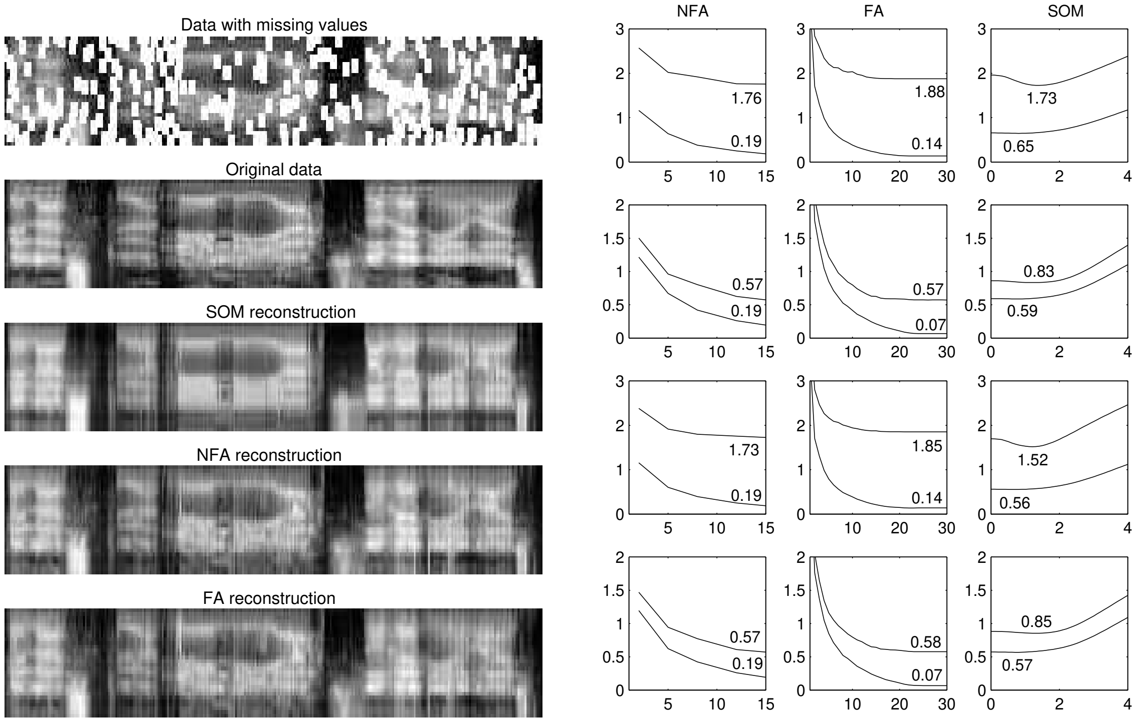

Figure 4.7: Left: Speech data reconstruction example with best parameters of each algorithm. Right: Mean square reconstruction errors of speech data as a function of number of factors or softening width. Reading from up to down, on the first and third row the values are missing in patches, and on the second and fourth, randomly. In the upper rows the test data is new words and in the lower rows the data is permuted randomly. Upper curve in each plot is the test result for missing values and lower is the reconstruction of observed values.

nonlinear factor analysis are closer to linear factor analysis than highly nonlinear mappings such as SOM. Nonlinear factor analysis is nevertheless able to capture nonlinear structure in the data and performed as well or better than linear factor analysis in all the reconstruction tasks.

Detection of process state changes

One potential application for the nonlinear dynamic state-space model discussed in Section 4.3 is process monitoring. In [1], ensemble learning was shown to be able to learn a model which is capable of detecting an abrupt change in the underlying dynamics of a fairly complex nonlinear process.

The process was artificially generated by nonlinearly mixing some of the states of three independent dynamical systems: two independent Lorenz processes and one harmonic oscillator. Figure 4.8 shows the observations on the left and the estimated underlying states on the right.

The first 1000 samples of the observations were used to learn the nonlinear dynamical system. The model was then fixed and applied to new observations. In the middle of the new data set, at time 1500, the underlying dynamics of one of the Lorenz processes abruptly changes. It is very difficult to detect this from the observed nonlinear mixtures whereas the estimated sources clearly exhibit the change. The cost function used in ensemble learning can readily be used for monitoring the change as can be seen in Figure 4.9. It

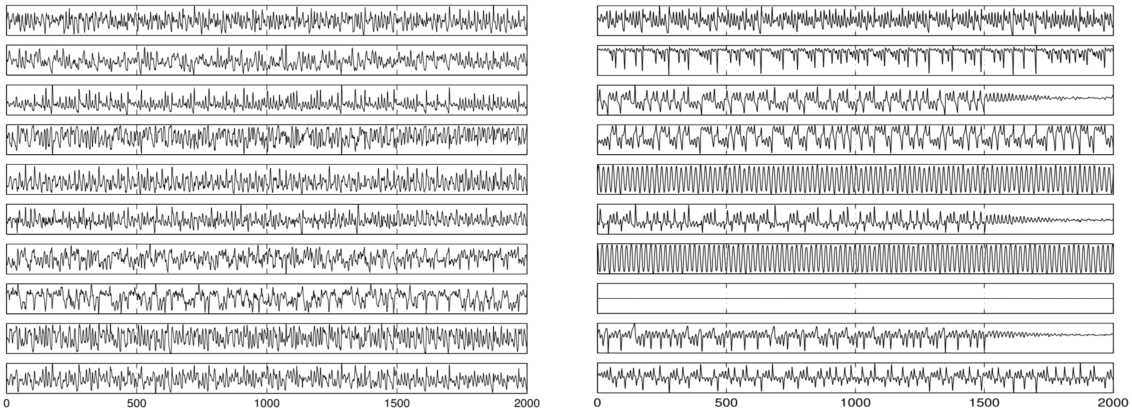

Figure 4.8: Observations on the left and reconstructed sources on the right. The process changes abruptly at time 1500 but this is visible only in the sources.

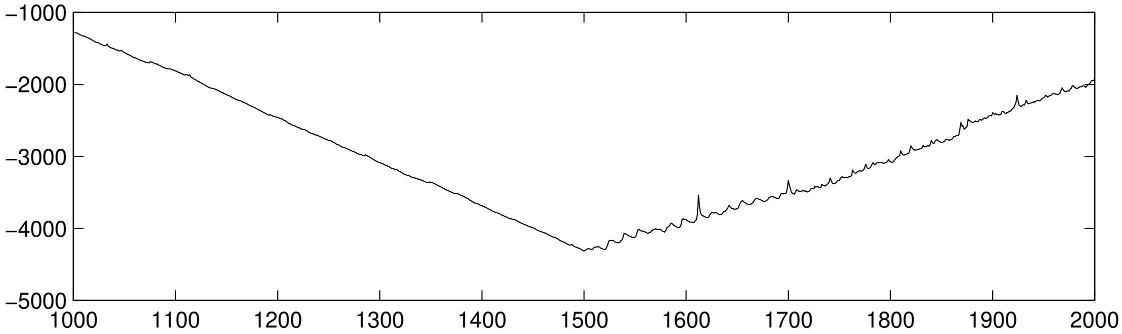

Figure 4.9: The change in the process can be detected by monitoring the cost function.

was shown experimentally that the method outperforms several standard change detection methods in this task [1].

References

[1] A. Iline, H. Valpola, and E. Oja. Detecting process state changes by nonlinear blind source separation. In *Proc. of the 3rd Int. Workshop on Independent Component Analysis and Signal Separation (ICA2001)*, pages 704–709, San Diego, California, December 2001.

[2] T. Raiko and H. Valpola. Missing values in nonlinear factor analysis. In *Proc. ICONIP 2001*, pages 822–827, Sanghai,China, November 2001.