VARIATIONAL INFERENCE AND LEARNING FOR NON-LINEAR STATE-SPACE MODELS WITH STATE-DEPENDENT OBSERVATION NOISE

Veli Peltola and Antti Honkela

Department of Information and Computer Science,
Aalto University School of Science and Technology, Helsinki, Finland
{veli.peltola,antti.honkela}@tkk.fi

ABSTRACT

In many real world dynamical systems, the inherent noise levels are not constant but depend on the state. Such aspects are often ignored in modelling because they make inference significantly more complicated. In this paper we propose a variational inference and learning algorithm for a non-linear state-space model with state-dependent observation noise. The observation noise level of each sample depends on additional latent variables with a linear dependence on the latent state. The method yields significant improvements in predictive performance over regular non-linear state-space model as well as direct autoregressive prediction using Gaussian processes in a simulated Lorenz system with state-dependent noise and in stock price prediction.

1. INTRODUCTION

State-space models or Kalman filter models are a universal tool in analysis of time series data. The model provides a flexible description by modelling the dynamics in a latent state-space, which is usually observed indirectly through some observation mapping. Because of the importance of the model, several variational inference and learning methods have been proposed for linear [1, 2] as well as non-linear models [3]. A non-linear model based on approximate EM with Gaussian processes was recently proposed in [4].

All these previous methods assume a simple additive Gaussian noise model for both the dynamical and the observation mappings. Yet, in many applications such an assumption is highly restrictive and unrealistic. In this paper we release this assumption by introducing variational inference for a non-linear state-space model with state-dependent observation noise. This is accomplished by combining the non-linear state-space model [3] and the variance modelling techniques of [5]. This kind of modelling framework encompasses, among others, many GARCH stochastic volatility type models commonly applied in computational finance [6, 7, 8].

2. MODEL

2.1. Basic non-linear state-space model

In a general non-linear state-space model as presented in [3], the observed data vectors $\mathbf{X} = (\mathbf{x}(t))$ are modelled with the help of latent states $\mathbf{s}(t)$. The states are assumed to evolve according to

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

where $\mathbf{g}$ is some non-linear mapping with parameters $\boldsymbol{\theta}_g$, and $\mathbf{m}(t)$ denotes additive Gaussian noise. Typically the states cannot be observed directly but only through an observation mapping

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_f) + \mathbf{n}(t), \tag{2}$$

where $\mathbf{f}$ is again some non-linear mapping with parameters $\boldsymbol{\theta}_f$, and $\mathbf{n}(t)$ denotes additive Gaussian noise to accommodate the parts of the data not captured by the model.

In [3], the non-linear mappings $\mathbf{f}$ and $\mathbf{g}$ are modelled by multi-layer perceptron (MLP) networks [9] having one hidden layer with hyperbolic tangent non-linearity:

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

$$\mathbf{g}(\mathbf{s}(t-1)) = \mathbf{s}(t-1) + \mathbf{D} \tanh[\mathbf{C}\mathbf{s}(t-1) + \mathbf{c}] + \mathbf{d} \tag{4}$$

Often the values of sources $\mathbf{s}(t)$ do not change much from their previous values $\mathbf{s}(t-1)$. This makes it easier to model the non-linearities in the difference $\mathbf{s}(t) - \mathbf{s}(t-1)$ directly.

For the noise terms $\mathbf{n}(t)$ and $\mathbf{m}(t)$, we safely can make some simplifying assumptions without sacrificing too much generality, because with the help of some auxiliary sources, non-linearity $\mathbf{g}$ is in principle able represent arbitrarily complex noise models, if they are required. We will assume that $\mathbf{n}(t)$ and $\mathbf{m}(t)$ are normally distributed with diagonal covariance, and independent over different times t.

This work was supported by the Postdoctoral Researcher's Project No 121179 of the Academy of Finland.

2.2. State-dependent noise variance

Let us denote the variance of $\mathbf{n}(t)$ with $\exp(2\mathbf{u}(t))$.[1] In the original model [3] these variances were constant over time for each component. The purpose of this paper, however, is to study a variation of this model where the observation noise level depends on the source signals. For simplicity, we will use a linear mapping here, instead of the non-linear kind used for $\mathbf{f}$ and $\mathbf{g}$. We consider it unlikely that the data would contain enough information to justify the additional complexity caused by using a non-linearity here.

$$\mathbf{u}(t) = \mathbf{W}\mathbf{s}(t) + \mathbf{w} + \mathbf{o}(t) \tag{5}$$

Here matrix $\mathbf{W}$ is used for the linear mapping, vector $\mathbf{w}$ is a constant term, and $\mathbf{o}(t)$ is yet again Gaussian noise added to compensate for modeling imperfections.

For innovation noise $\mathbf{m}(t)$ we will assume that each sources have constant noise variances $\exp(2\mathbf{v}_m)$. Noise $\mathbf{o}(t)$ will also have constant variance $\exp(2\mathbf{v}_u)$ for each component.

2.3. Probability model and priors

The model so far can be described with these three equations:

$$\begin{aligned}
\mathbf{s}(t) &\sim \mathrm{N}[\mathbf{g}(\mathbf{s}(t-1), \boldsymbol{\theta}_g), \mathrm{diag}(\exp(2\mathbf{v}_m))] && (6)\\
\mathbf{u}(t) &\sim \mathrm{N}[\mathbf{W}\mathbf{s}(t) + \mathbf{w}, \mathrm{diag}(\exp(2\mathbf{v}_u))] && (7)\\
\mathbf{x}(t) &\sim \mathrm{N}[\mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_f), \mathrm{diag}(\exp(2\mathbf{u}(t)))], && (8)
\end{aligned}$$

where $\mathrm{N}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]$ denotes a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Together they describe a model for data $\mathbf{x}$ using the unobserved variables

$$\boldsymbol{\Theta} = (\mathbf{s}(t), \mathbf{u}(t), \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b}, \mathbf{C}, \mathbf{c}, \mathbf{D}, \mathbf{d}, \mathbf{W}, \mathbf{w}, \mathbf{v}_m, \mathbf{v}_u)\,. \tag{9}$$

The priors of the variables are specified to fix the scaling ambiguity between $\mathbf{s}$ and $\mathbf{A}$ and to have a hierarchical prior allowing automatic relevance determination (ARD) [9] like decisions to inactivate parts of the model:

$$\begin{aligned}
A_{ij} &\sim \mathrm{N}[0, 1] && (10)\\
\Phi_{ij} &\sim \mathrm{N}[0, \exp(2v_{\Phi_j})] && (11)\\
\phi_i &\sim \mathrm{N}[m_\phi, \exp(2v_\phi)] && (12)\\
v_{u_i} &\sim \mathrm{N}[m_v, \exp(2v_v)] && (13)\\
v_{m_i} &\sim \mathrm{N}[m_{v_m}, \exp(2v_{v_m})] && (14)\\
v_{\Phi_j} &\sim \mathrm{N}[m_{v_\Phi}, \exp(2v_{v_\Phi})] && (15)
\end{aligned}$$

where $\phi \in \{a, b, c, d, w\}$ and $\Phi \in \{B, C, D, W\}$. All the hyperparameters have vague priors $\mathrm{N}[0, 100^2]$.

3. VARIATIONAL INFERENCE

In order to perform inference and learning on the model, we apply mean field type variational inference, also known as variational Bayes (VB) [1, 9]. In VB, the posterior $p(\boldsymbol{\Theta}|\mathbf{X})$ is approximated with a tractable distribution $q(\boldsymbol{\Theta})$ that is fitted by minimising the free energy

$$\begin{aligned}
\mathcal{F}(q(\boldsymbol{\Theta})) &= E_{q(\boldsymbol{\Theta})}\left\{\log \frac{q(\boldsymbol{\Theta})}{p(\mathbf{X}, \boldsymbol{\Theta})}\right\}\\
&= D_{\mathrm{KL}}(q(\boldsymbol{\Theta}) \| p(\boldsymbol{\Theta}|\mathbf{X})) - \log p(\mathbf{X}),
\end{aligned} \tag{16}$$

where $D_{\mathrm{KL}}(q\|p)$ is the Kullback–Leibler (KL) divergence between q and p. As the KL divergence is non-negative, the negative free energy provides a lower bound on model marginal likelihood $\log p(\mathbf{X})$.

Because of the non-linearities, the non-linear state-space model is not in the conjugate-exponential family and standard variational Bayesian expectation maximisation (VB EM) [9] is not applicable. Variational inference for the model basically follows the scheme introduced in [3, 10]: we derive a deterministic approximation of the free energy based on a fixed functional form of the posterior approximation and the apply gradient-based optimisation to minimise the free energy. The optimisation is made more efficient through the use of natural conjugate gradient (NCG) optimisation [11].

3.1. Posterior approximation

In order to allow efficient learning, the posterior approximation $q(\boldsymbol{\Theta}) = N(\boldsymbol{\Theta};\ \boldsymbol{\mu}_{\boldsymbol{\Theta}}, \boldsymbol{\Lambda}_{\boldsymbol{\Theta}})$ is restricted to be Gaussian with mean $\boldsymbol{\mu}_{\boldsymbol{\Theta}}$ and precision (inverse covariance) $\boldsymbol{\Lambda}_{\boldsymbol{\Theta}}$. Furthermore, the precision of the approximation is restricted to be almost diagonal. The only allowed off-diagonal terms are in the approximation of $\mathbf{s}(t)$ which includes a correlation between $s_i(t)$ and $s_i(t+1)$. Different components of the state vector $\mathbf{s}(t)$ are still assumed independent, and the posterior approximation of the states is a product of independent chains.

Following the theory of Gaussian Markov random fields, this assumption translates to a tridiagonal precision (inverse covariance) matrix with non-zero elements only on the main diagonal and on the diagonal corresponding to the assumed links. The corresponding covariance matrix has full blocks for each component of the state.

3.2. The free energy

In order to derive the value of the free energy (16), we note that

$$\mathcal{F}(q(\boldsymbol{\Theta})) = E_{q(\boldsymbol{\Theta})}\{\log q(\boldsymbol{\Theta})\} + E_{q(\boldsymbol{\Theta})}\{-\log p(\mathbf{X}, \boldsymbol{\Theta})\}\,. \tag{17}$$

[1] In this model unknown variances of normal distributions will be parameterized as the natural logarithm of the standard deviation. If logarithm of standard deviation is v, variance will be e^{2v}.

The first term is the negative entropy of a Gaussian

$$E_{q(\boldsymbol{\Theta})}\left\{\log q(\boldsymbol{\Theta})\right\} = \frac{N}{2}\log(2\pi e) - \log\det \boldsymbol{\Lambda}_{\boldsymbol{\Theta}}, \quad (18)$$

where N is the dimensionality of $\boldsymbol{\Theta}$.

The second term splits to a sum of a number of terms according to Eqs. (6)–(15).

$$E_{q(\boldsymbol{\Theta})}\left\{-\log p(\mathbf{X}, \boldsymbol{\Theta})\right\} = \sum_{t,i} E_{q(\boldsymbol{\Theta})}\left\{-\log p(x_i(t)|\boldsymbol{\Theta})\right\} + \sum_{\gamma\in\boldsymbol{\Theta}} E_{q(\boldsymbol{\Theta})}\left\{-\log p(\gamma|\boldsymbol{\Theta}_{\setminus\gamma})\right\}, \quad (19)$$

where $\boldsymbol{\Theta}_{\setminus\gamma}$ denotes the parameters γ depends on.

The terms in the sum are expectations for parameters γ following a normal model $\mathrm{N}(m, e^{2v})$. The negative logarithm of the pdf is

$$-\log p(\gamma|\boldsymbol{\Theta}_{\setminus\gamma}) = \frac{1}{2}\ln(2\pi) + v + \frac{1}{2}(\gamma - m)^2\exp(-2v). \quad (20)$$

Assuming independent Gaussian approximations for γ, m and v with means $\bar{\gamma}, \bar{m}, \bar{v}$ and variances $\tilde{\gamma}, \tilde{m}, \tilde{v}^2$, the expectation is

$$E_{q(\boldsymbol{\Theta})}\left\{-\log p(\gamma|\boldsymbol{\Theta}_{\setminus\gamma})\right\} = \frac{1}{2}\ln(2\pi) + \bar{v} + \frac{1}{2}[(\bar{\gamma} - \bar{m})^2 + \tilde{\gamma} + \tilde{m}]\exp(2\tilde{v} - 2\bar{v}). \quad (21)$$

For the observations $x_i(t)$ we obtain similarly

$$E_{q(\boldsymbol{\Theta})}\left\{-\log p(x_i(t)|\boldsymbol{\Theta})\right\} = \frac{1}{2}\ln(2\pi) + \bar{u}_i(t) + \frac{1}{2}[(x - \bar{f}_i(t))^2 + \tilde{f}_i(t)]\exp(2\tilde{u}_i(t) - 2\bar{u}_i(t)), \quad (22)$$

where the means $\bar{f}_i(t)$ and variances $\tilde{f}_i(t)$ of $\mathbf{f}(t)$ are evaluated as explained in [10, 12].

Let us define an augmented version of sources $\mathbf{s}$ and the matrix $\mathbf{W}$ by setting

$$\hat{\mathbf{W}}\cdot\hat{\mathbf{s}}(t) = (\mathbf{W}\ \mathbf{w})\cdot\begin{pmatrix}\mathbf{s}(t)\\ 1\end{pmatrix} = \mathbf{W}\mathbf{s}(t) + \mathbf{w}. \quad (23)$$

For the variance sources

$$u_i(t) \sim \mathrm{N}(\hat{\mathbf{w}}_i\hat{\mathbf{s}}(t), \exp(2v_{u_i})), \quad (24)$$

where $\hat{\mathbf{w}}_i$ is the ith row vector of $\hat{\mathbf{W}}$, we now obtain

$$E_{q(\boldsymbol{\Theta})}\left\{-\log p(u_i(t)|\boldsymbol{\Theta}_{\setminus\mathbf{U}})\right\} = \frac{1}{2}\ln(2\pi) + \bar{v}_{u_i}(t) + \frac{1}{2}[(\bar{u}_i(t) - \bar{h}_i(t))^2 + \tilde{u}_i(t) + \tilde{h}_i(t)]\exp(2\tilde{v}_{u_i}(t) - 2\bar{v}_{u_i}(t)), \quad (25)$$

where $\mathbf{h}(t) = \hat{\mathbf{W}}\hat{\mathbf{s}}(t)$ and its mean and variance are

$$\bar{h}_i(t) = \bar{\hat{\mathbf{w}}}_i\bar{\hat{\mathbf{s}}}(t) \quad (26)$$

$$\tilde{h}_i(t) = \sum_j \left[\tilde{\hat{w}}_{ij}(\tilde{\hat{s}}_j(t) + \bar{\hat{s}}_j(t)^2) + \bar{\hat{w}}_{ij}^2\tilde{\hat{s}}_j(t)\right]. \quad (27)$$

For the states $s_i(t)$ we can similarly derive [3]

$$E_{q(\boldsymbol{\Theta})}\left\{-\log p(s_i(t)|\boldsymbol{\Theta}_{\setminus\mathbf{s}(t)})\right\} = \frac{1}{2}\ln(2\pi) + \bar{v}_{m_i}(t) + \frac{1}{2}\Big[(\bar{s}_i(t) - \bar{g}_i(t))^2 + \tilde{s}_i(t) + \tilde{g}_i(t) - 2\breve{s}_i(t, t-1)\frac{g_i(t)}{s_i(t-1)}\tilde{s}_i(t-1)\Big]\exp(2\tilde{v}_{m_i}(t) - 2\bar{v}_{m_i}(t)), \quad (28)$$

where $\bar{g}_i(t)$ and $\tilde{g}_i(t)$ are the mean and variance of $\mathbf{g}(\mathbf{s}(t-1))$ evaluated similarly as those of $\mathbf{f}(\mathbf{s}(t))$ with the partial derivative arising as a byproduct of those, and $\breve{s}_i(t, t-1)$ is the linear correlation between $s_i(t-1)$ and $s_i(t)$ as explained in [3].

3.3. Update rules

The hyperparameters in Eqs. (13)–(15) are updated using a VB EM type scheme to find a global optimum, given current values of the other parameters [13].

For the states and the weights of the MLP networks, we apply a natural-conjugate-gradient-based (NCG) minimisation of the free energy, as described in [11, 3]. The update algorithm is summarised in Algorithm 1.

Algorithm 1 An overview of the learning algorithm

```
initialize $\mathbf{s}$ using PCA with embedded data
initialize MLP network weights $\boldsymbol{\theta}_f, \boldsymbol{\theta}_g$ randomly
repeat
    apply VB EM type update for $\mathbf{u}(t)$
    update the parameters (including $\mathbf{W}$, $\mathbf{w}$) with VB EM
    calculate $\mathcal{F}$ for the current iteration
    calculate the gradient of $\mathcal{F}$
    update $\mathbf{s}(t)$ and the weights $\boldsymbol{\theta}_f, \boldsymbol{\theta}_g$ using NCG
until $\mathcal{F}$ decreases less than $\varepsilon$ on one iteration
```

3.3.1. *Updating the linear noise mapping*

When we update the means $\bar{\hat{w}}_{ij}$, we must consider the each row vector $\hat{\mathbf{w}}_i$ as one unit. The free energy terms depending on vector $\hat{\mathbf{w}}_i$ over the whole duration are

$$\sum_t C(u_i(t), \hat{\mathbf{w}}_i\,\hat{\mathbf{s}}(t), v_{u_i}) + \sum_j C(\hat{w}_{ij}, m_j, z_j) + E(\hat{w}_{ij}). \quad (29)$$

[2] We will use the notation $\bar{\gamma}$ for the mean and $\tilde{\gamma}$ for the variance of γ in the approximation for all variables.

Mean of $\hat{\mathbf{w}}_i\, \hat{\mathbf{s}}(t)$ is simply $\bar{\hat{\mathbf{w}}}_i\, \bar{\hat{\mathbf{s}}}(t)$ and its variance is

$$\mathrm{Var}(\hat{\mathbf{w}}_i \hat{\mathbf{s}}(t)) = \sum_j \tilde{\hat{w}}_{ij} \cdot (\tilde{\hat{s}}_j(t) + \bar{\hat{s}}_j(t)^2) + \bar{\hat{w}}_{ij}^2 \cdot \tilde{\hat{s}}_j(t). \quad (30)$$

Now the free energy as a function of $\bar{\hat{\mathbf{w}}}_i$ can be written as

$$\frac{1}{2} \bar{\hat{\mathbf{w}}}_i \mathbf{A} \bar{\hat{\mathbf{w}}}_i^{\mathrm{T}} - \bar{\hat{\mathbf{w}}}_i \mathbf{b} + c \quad (31)$$

where $\mathbf{A}$ is the symmetric positive-definite matrix

$$\begin{aligned}\mathbf{A} = \sum_t \left[\bar{\hat{\mathbf{s}}}(t) \cdot \bar{\hat{\mathbf{s}}}(t)^{\mathrm{T}} + \mathrm{diag}(\tilde{\hat{\mathbf{s}}}(t)) \right] \exp(2\tilde{v}_i - 2\bar{v}_i) \\ + \mathrm{diag}\left[\exp(2\tilde{\mathbf{z}} - 2\bar{\mathbf{z}}) \right]\end{aligned} \quad (32)$$

and $\mathbf{b}$ is the vector

$$\mathbf{b} = \sum_t \left[\bar{\hat{\mathbf{s}}}(t)\, \bar{u}_i(t) \right] \exp(2\tilde{v} - 2\bar{v}) + \left[m_j \exp(2\tilde{z}_j - 2\bar{z}_j) \right]_j . \quad (33)$$

Exact value of the constant c isn't relevant; the free energy is minimized by setting

$$\bar{\hat{\mathbf{w}}}_i = (\mathbf{A}^{-1}\mathbf{b})^{\mathrm{T}}. \quad (34)$$

Variance updates can be calculated separately for each component $\hat{w}_{ij}$. The free energy as a function of $\tilde{\hat{w}}_{ij}$ can be written as

$$\frac{1}{2}\, a\, \tilde{\hat{w}}_{ij} - \frac{1}{2} \ln \tilde{\hat{w}}_{ij} + d \quad (35)$$

where

$$a = \sum_t \left[\tilde{\hat{s}}_j(t) + \bar{\hat{s}}_j(t)^2 \right] \exp(2\tilde{v}_i - 2\bar{v}_i) + \exp(2\tilde{z}_j - 2\bar{z}_j). \quad (36)$$

Again, exact value of d isn't relevant; (36) is minimized by setting

$$\tilde{\hat{w}}_{ij} = 1/a. \quad (37)$$

3.3.2. Updating $\mathbf{u}(t)$

Other variance parameters in the model, for example $\mathbf{v}_m$ and $\mathbf{v}_u$ have a simple Gaussian prior mean. The free energy associated with these parameters is minimized by Newton's iteration [3]. The free energy terms associated with the variance nodes $\mathbf{u}(t)$ in Eq. (25) and from the prior have the same form as the other variance parameters and are updated similarly.

3.4. Predictions

Predictions from the model can be obtained by sampling. Possible values for all variables $\mathbf{\Theta}$ in (9) are drawn directly from the posterior approximation $q(\mathbf{\Theta})$. For $\mathbf{s}(t)$, only the latest time step is needed. Given these values, it is straightforward to simulate one sequence of predictions of the model. We repeat this for new values of $\mathbf{\Theta}$ to get as many samples as required.

The quantiles needed for plotting can be directly calculated from these samples. The most important metric used for evaluating the quality of the predictions was simple loglikelihood of the test data under the predictive posterior distribution for future observations. The predictions obtained by sampling were often multimodal, so the predictive distributions needed for log-likelihood computation were estimated by using a Dirichlet process Gaussian mixture model [14].

4. EXPERIMENTS

The method was tested on two data sets: a synthetic data set based on Lorenz processes and a stock market data set. On both data sets, the method was compared to the original NSSM, which has constant variance observation noise.

Additionally on the synthetic data set, the method was compared against a direct non-linear auto-regressive prediction using Gaussian Processes (GP) [15]. The values of $\mathbf{x}(t)$ were estimated directly from the values of $\mathbf{x}(t-1), \mathbf{x}(t-2), \ldots, \mathbf{x}(t-k)$ for some chosen history length k. Several values of k were tested and $k = 12$ was selected, because it resulted in the best predictions in the test set. This selection method gives GP a slight advantage, as normally this would need to be selected using a separate test set. Squared Exponential covariance function with Automatic Relevance Determination (ARD) distance measure was used for the GP [16].

4.1. Lorenz data set

The Lorenz data set was generated by using two independent Lorenz processes and a harmonic oscillator. The (σ, ρ, β) parameter vectors for the Lorenz processes were $(3, 26.5, 1)$ and $(4, 30, 1)$. The harmonic oscillator had an angular velocity of 1/3. 1600 time steps were used for learning and 50 for evaluation.

To make the problem more challenging, these 8 signals were first projected to a 5 dimensions using a random linear mapping. This means that the method has to reconstruct the original state-space indirectly, because there are no direct observations of all the states. Finally, the 10 dimensional data vectors were generated from the 5 reduced signals by a random MLP network with $\sinh^{-1}$ non-linearity. Gaussian noise was added, and its noise level (logarithm of the standard deviation) was generated by a random linear mapping from the 5 reduced signals. Apart from the noise, the data set is similar to the one used in [3].

Some predictions on the Lorenz data using the HNSSM method are illustrated in Fig. 1. The figure shows the chaotic nature of the data: short-term prediction is possi-

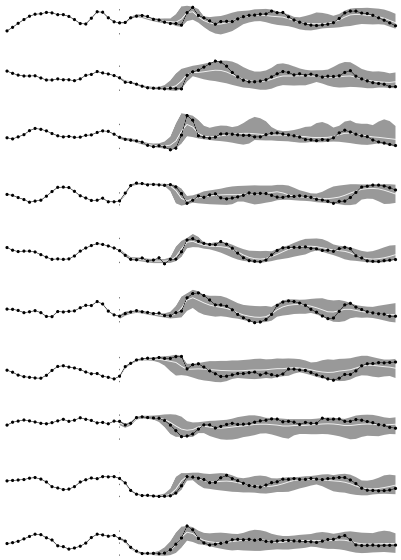

Fig. 1. Lorenz data predictions using the HNSSM method. The dotted vertical line denotes the end of training data. The shaded region denotes 95% posterior credible region of the predictions, and the thin white line in the middle the posterior mean prediction.

ble, but detailed long-term prediction is not. This is also reflected in the results of cumulative root mean squared prediction errors in Fig. 2 and predictive log-likelihood in Fig. 3. All these comparisons show that both NSSM and HNSSM are clearly superior to the GP method, with HNSSM being slightly better than regular NSSM.

4.2. Stock market data

The stock market data set consists of daily closing prices for 33 large and widely held public companies in the United States during years 1992–2008. The prices are adjusted for splits and dividends. The selected companies were the ones that had been used to calculate the Dow Jones Industrial Average (DJIA) during that period. For simplicity in implementation of the algorithm and also in interpretation of the results, the companies that did not have data for the full period were not included in the data set.

Before using the algorithm, a logarithm was taken of the prices. No other preprocessing was used. For both HNSSM and NSSM, the number of hidden sources used was 50, and the number of hidden neurons in both the observation and temporal MLP was 70. The GP method was not applicable, because the data set has trends making it non-stationary.

When the length of the training data was chosen to be 1000, the linear observation variance was effectively pruned out, and HNSSM and NSSM produced identical results. However, when using a training data of length 2000 or 3000, some variance sources were found.

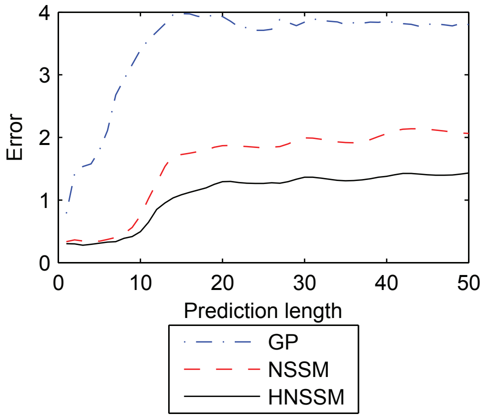

Fig. 2. Cumulative root mean squared prediction errors in the Lorenz data using different tested methods.

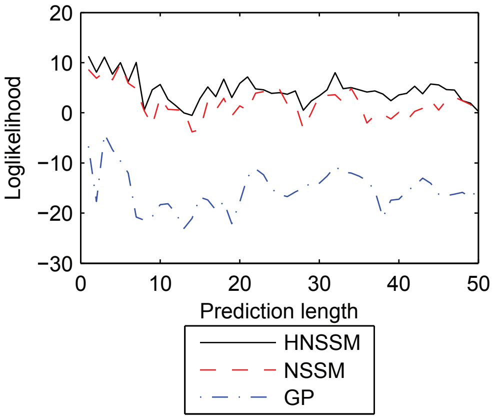

Fig. 3. Predictive log-likelihoods in the Lorenz data using different tested methods.

The results of the stock market prediction are shown in Fig. 4. The HNSSM method yields a significantly lower free energy (-238714 vs. -222612), but the the prediction performance is only slightly superior to the regular NSSM method.

5. DISCUSSION

Our proposed method would be easily applicable to continuous-time models following [17].

Our comparison results are qualitatively quite different from those presented in [4], where auto-regressive GP similar to one used in our comparison was very close to the best. Most likely this is due to the use of a scalar time series in [4].

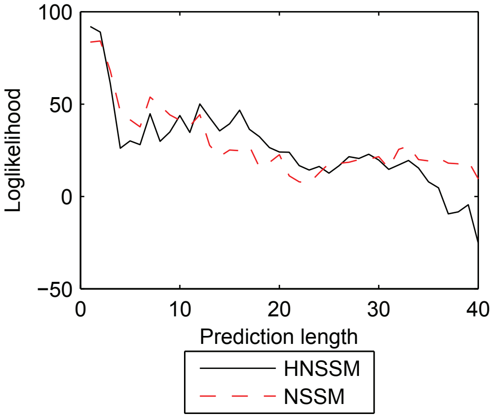

Fig. 4. Predictive log-likelihoods in the stock market data using different tested methods on a training set of 3000 samples.

The NSSM and HNSSM methods are much better suited for higher dimensional problems where the state-space can be utilised to model correlations between observed variables.

It might seem tempting to obtain long-term predictions using the same propagation rules with the VB posterior that are used in learning. Unfortunately this leads to incorrect predictions, as the parameters would incorrectly be assumed independent at each step.

The applied standard VB method may not be optimal for long-term prediction performance. The method is essentially based on balancing short-term training set prediction accuracy with the complexity of the model. In this one can sometimes observe that the lowest free energy is attained by a simple model that attains a reasonable short-term prediction performance, but not as good performance as a more complex model. When evaluating long-term prediction performance, it might be optimal to make a slightly different trade-off between these aspects. Finding ways to correct this is an important item for future research.

We have presented a method for incorporating a model of varying variance into our non-linear state-space model framework. The relatively straightforward extension leads to significant improvements in predictive performance in situations where the observation noise level is not constant.

6. REFERENCES

[1] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, Ph.D. thesis, University College London, UK, 2003.

[2] D. Barber and S. Chiappa, “Unified inference for variational Bayesian linear Gaussian state-space models,” in *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2007, The MIT Press.

[3] 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.

[4] R. Turner, M. P. Deisenroth, and C. Rasmussen, “System identification in Gaussian process dynamical systems,” in *Nonparametric Bayes Workshop (NIPS 2009)*, Whistler, BC, Canada, 2009.

[5] H. Valpola, M. Harva, and J. Karhunen, “Hierarchical models of variance sources,” *Signal Processing*, vol. 84, no. 2, pp. 267–282, 2004.

[6] E. Ghysels, A. C. Harvey, and E. Renault, “Stochastic volatility,” in *Statistical Methods in Finance*, C. R. Rao and G. S. Maddala, Eds., pp. 119–191. North-Holland, Amsterdam, 1996.

[7] N. Shepard, “Statistical aspects of ARCH and stochastic volatility,” in *Time series models in econometrics, finance and other fields*, D. R. Cox, D. V. Hinkley, and O. E. Barndorff-Nielson, Eds., pp. 1–67. Chapman & Hall, London, 1996.

[8] E. Jacquier, N. G. Polson, and P. E. Rossi, “Bayesian analysis of stochastic volatility models,” *Journal of Business & Economic Statistics*, vol. 12, no. 4, pp. 371–389, 1994.

[9] C. Bishop, *Pattern Recognition and Machince Learning*, Springer, New York, 2006.

[10] A. Honkela and H. Valpola, “Unsupervised variational Bayesian learning of nonlinear models,” in *Advances in Neural Information Processing Systems 17*, pp. 593–600. MIT Press, Cambridge, MA, 2005.

[11] A. Honkela, M. Tornio, T. Raiko, and J. Karhunen, “Natural conjugate gradient in variational inference,” in *Proc. 14th Int. Conf. on Neural Information Processing (ICONIP 2007)*, Kitakyushu, Japan, 2008, vol. 4985 of *Lecture Notes in Computer Science*, pp. 305–314, Springer-Verlag, Berlin.

[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. 5, pp. 914–934, 2007.

[13] H. Lappalainen and J. Miskin, “Ensemble learning,” in *Advances in Independent Component Analysis*, M. Girolami, Ed., pp. 75–92. Springer-Verlag, Berlin, 2000.

[14] K. Kurihara, M. Welling, and N. Vlassis, “Accelerated variational Dirichlet process mixtures,” in *Advances in Neural Information Processing Systems 19*, pp. 761–768. MIT Press, Cambridge, MA, 2007.

[15] A. Girard, C. E. Rasmussen, J. Quiñonero-Candela, and R. Murray-Smith, “Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting,” in *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003, pp. 529–536, MIT Press.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[17] A. Honkela, M. Tornio, and T. Raiko, “Variational Bayes for continuous-time nonlinear state-space models,” in *NIPS*2006 Workshop on Dynamical Systems, Stochastic Processes and Bayesian Inference*, Whistler, B.C., 2006.