CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

THE EM ALGORITHM AND NEURAL NETWORKS FOR NONLINEAR STATE SPACE ESTIMATION

JFG de Freitas, M Niranjan and AH Gee

CUED/F-INFENG/TR 313

March 25, 1999

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

E-mail: jfgf@eng.cam.ac.uk
URL: http://www-svr.eng.cam.ac.uk/~jfgf

Abstract

In this paper, we derive an EM algorithm for nonlinear state space models. We use it to estimate jointly the neural network weights, the model uncertainty and the noise in the data. In the E-step we apply a forward-backward Rauch-Tung-Striebel smoother to compute the network weights. For the M-step, we derive expressions to compute the model uncertainty and the measurement noise. We find that the method is intrinsically very powerful, simple, elegant and stable.

Contents

1 Introduction **1**

2 Background **2**

3 Nonlinear State Space Modelling **2**

4 Inference with MLPs and extended Kalman smoothing **3**
4.1 The extended Kalman smoother 3
4.2 Training MLPs with the EKF 5

5 The EM algorithm **6**

6 The EM algorithm for nonlinear state space models **8**
6.1 Mathematical preliminaries 9
6.2 Computing the expectation of the log-likelihood 10
6.3 Differentiating the expected log-likelihood 11
6.3.1 Maximum with respect to A 11
6.3.2 Maximum with respect to R 11
6.3.3 Maximum with respect to Q 12
6.3.4 Maximum with respect to μ 12
6.3.5 Maximum with respect to Π 12
6.4 The E and M steps for nonlinear state space models 12

7 Experiments **13**
7.1 Simple regression example 13
7.2 Robot arm mapping 13
7.3 Classification with medical data 15

8 Conclusions **16**

9 Acknowledgements **18**

A An Important Inequality **19**

1 Introduction

In 1961, Kalman and Bucy (Kalman and Bucy 1961) developed an optimal linear-Gaussian state space filter that has become an essential component of any modern tracking and time series analysis tool-box or text. The Kalman-Bucy filter is a recurrent estimator of the hidden states of a state space model. The model consists of two linear matrix equations. The first one, known as the measurements equation, defines a set of measurements in terms of a linear combination of several hidden states and an additive Gaussian noise process. The second one, known as the process or dynamics equation, describes the evolution of the states in terms of a linear combination of previous values of the states and an additive Gaussian noise process. To estimate the hidden states optimally, the Kalman filter assumes that the measurements, the noise statistics and parameters governing the linear transformations are given.

Later, in 1965, Rauch, Tung and Striebel (Rauch, Tung and Striebel 1965) proposed a combination of forward and backward filtering to obtain improved estimates over stationary data segments. This *en bloc* estimation technique is widely know as linear Kalman smoothing. Other extensions to the original work on Kalman filtering include coloured noise filters and nonlinear estimators linearised about the current estimates. The latter are known as extended Kalman filters (EKFs).

A well known limitation of Kalman estimators is the assumption of known *a priori* statistics to describe the measurement and process noise. In many applications, it is not straightforward to choose the right noise covariance matrices (de Freitas, Niranjan and Gee 1997, Jazwinski 1970). Moreover, the matrices of parameters governing the linear transformations in the measurement and process equations are typically unknown. Unfortunately, the optimality of the Kalman filter often hinges on the designer's ability to formulate these matrices *a priori*. To circumvent this limitation and ensure optimality, it is important to design algorithms for estimating the noise covariances and parameter matrices without leading to a degradation in the performance of the Kalman estimator. In the remainder of this paper, we will refer to the problems of computing the model states and the model parameters, including the noise covariances, as inference and learning respectively.

Several researchers in the estimation, filtering and control fields have attempted to solve the problem of learning the noise covariances (de Freitas et al. 1997, Jazwinski 1969, Mehra 1970, Mehra 1971, Myers and Tapley 1976, Tenney, Hebbert and Sandell 1977). Mehra (Mehra 1972) and Li and Bar-Shalom (Li and Bar-Shalom 1994) have written brief surveys on this topic. In stationary environments where data is available in batches, it is possible to address the general problem of learning in a principled way, via the expectation maximisation (EM) algorithm (Dempster, Laird and Rubin 1977). This paper will focus on this learning approach and will aim at extending the current work on EM learning and linear Kalman smoothing to nonlinear Kalman smoothing.

In Section 2, we present a brief history of attempts to use the EM algorithm to learn linear state space models. Section 3 introduces the nonlinear state space modelling scheme adopted in this work. The approximation of nonlinear state space models with multi-layer perceptrons trained by extended Kalman smoothing is discussed in Section 4. Section 5 presents a brief derivation of the EM algorithm, which is used as a step towards the derivation of the EM algorithm for nonlinear state space models in Section 6. Section 7 examines some of the results obtained, while Section 8 provides a critical analysis of these results.

2 Background

The application of the EM algorithm to learning and inference in linear dynamical systems has occupied the attention of several researchers in the past. Chen (Chen 1981) was one of the pioneers in this field. In particular, he applied the EM algorithm to linear state space models known in the statistics literature as MIMIC models. In these models one observes multiple indicators and multiple causes of a single latent variable. Chen's MIMIC model was implemented in a simulation study relating social status and participation.

Watson and Engle (Watson and Engle 1983) have suggested using the EM algorithm, in conjunction with the method of scoring, for the estimation of linear dynamic factor, MIMIC and varying coefficient regression models. They evaluated their paradigm experimentally by estimating common factors in wage rage data from several industries in Los Angeles, USA.

In 1982, Shumway and Stoffer (Shumway and Stoffer 1982) proposed the use of the EM algorithm and linear state space models for time series smoothing and forecasting with missing observations. To demonstrate their method, they considered a health series representing total expenditures for physician services as measured by two different sources. The time series produced by each source have similar values but exhibit missing observations at different periods. In Shumway and Stoffer's approach, the two series are automatically merged into an overall expenditure series, which is then used for forecasting. Nine years later, Shumway and Stoffer (Shumway and Stoffer 1991) extended their work to switching linear dynamic models. In essence, they derived a state space representation with measurement matrices that switch according to a time varying independent random process. They illustrate their method on an application involving the tracking of multiple targets.

The method of learning and inference in linear state space models via the EM algorithm has also played a role in the fields of speech analysis and computer vision. Digalakis, Rohlicek and Ostendorf (Digalakis, Rohlicek and Ostendorf 1993) applied it to the speech recognition problem. They made a connection between this method and the Baum-Welch estimation algorithm for hidden Markov models (HMMs). North and Blake (North and Blake 1998) have implemented the method to learn linear dynamic state space models used for tracking contours in images. Rao and Ballard (Rao and Ballard 1997) have also explored the relevance of the EM algorithm together with state space estimation in the field of vision. They have developed a hierarchical network model of visual recognition that encapsulates those concepts.

Ghahramani (Ghahramani 1997) has embedded the EM method for learning dynamic linear systems in a graphical models framework. He treats computationally intractable models such as factorial HMMs and switching state space models by resorting to Gibbs sampling and variational approximations. In another paper, Roweis and Ghahramani (Roweis and Ghahramani 1997) make use of the EM algorithm and linear state space representations to present a unified view of linear Gaussian models including factor analysis, mixtures of Gaussians, standard and probabilistic versions of principal component analysis, vector quantisation, Kalman smoothing and linear hidden Markov models.

In this paper, we join the discourse on learning and inference with the EM and Kalman smoothing algorithms by extending the work to nonlinear estimators. In particular, we make use of a multi-layer perceptron (MLP) to model the nonlinear mapping between the hidden states and the output measurements.

3 Nonlinear State Space Modelling

To investigate the application of the EM algorithm to state space learning, we shall focus on the following nonlinear state space representation:

$$\mathbf{w}_{k+1} = A\mathbf{w}_k + \mathbf{d}_k \quad (1)$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{w}_k, \mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

where the output measurements of a system ($\mathbf{y}_k \in \Re^m$) depend on a nonlinear, multivariate function of the system inputs ($\mathbf{x}_k \in \Re^d$) and a set of states ($\mathbf{w}_k \in \Re^q$). A graphical representation of this model is depicted in Figure 1. In this work we assume that the states correspond to the weights of a neural network. However, it is possible to incorporate other variables, for example the model outputs, into the hidden state vector.

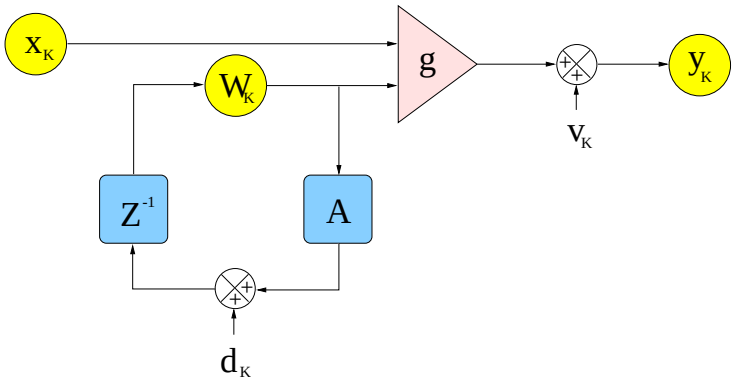

Figure 1: Nonlinear state space model. The symbol Z^{-1} denotes the delay operator.

The measurements nonlinear mapping $\mathbf{g}(.)$ is approximated by a multi-layer perceptron (MLP). Nonetheless, the work may be easily extended to encompass recurrent networks, radial basis networks and many other approximation techniques. The measurements are assumed to be corrupted by noise $\mathbf{v}_k$, which in our case we model as zero mean, uncorrelated Gaussian noise with covariance R.

We model the evolution of the model parameters by assuming that they depend on a deterministic component $A\mathbf{w}_k$ and a stochastic component $\mathbf{d}_k$. The process noise $\mathbf{d}_k$ may represent our uncertainty in how the parameters evolve, modelling errors or unknown inputs such as target manoeuvres. We assume the process noise to be a zero mean, uncorrelated Gaussian process with covariance Q. The matrix A contains information about how the states evolve. It is particularly useful in tracking applications. However, when the above model is employed merely for parameter estimation in neural network models, the matrix A should be viewed as a mechanism to achieve directed trajectories in state space. In other words, A allows for more general jumps than the simple random walk that would result by excluding A from the model. Although we derive analytical expressions for estimating A, we have found no obvious way of using it to improve the training of MLPs.

We need to estimate the model states $\hat{\mathbf{w}}_k$ and a set of parameters $\theta = \{R, Q, A\}$ given the measurements $Y_k = \{\mathbf{y}_1,\ \mathbf{y}_2,\ \cdots,\ \mathbf{y}_k\}$. The problem of estimating the states is known as inference, while the problem of estimating the parameters may be referred to as learning. The problem of estimating $\mathbf{w}_k$ given Y_τ is also called the smoothing problem if $k < \tau$; the filtering problem if $k = \tau$; or the prediction problem if $k > \tau$ (Gelb 1974, Jazwinski 1970). In this paper, we shall concentrate on the smoothing problem.

4 Inference with MLPs and extended Kalman smoothing

4.1 The extended Kalman smoother

The extended Kalman filter (EKF) is a versatile and computationally efficient algorithm for suboptimal nonlinear state estimation. The EKF is a minimum variance estimator based on a Taylor series expansion of the nonlinear function $\mathbf{g}(.)$ around the previous estimate

(Bar-Shalom and Li 1993, Gelb 1974). That is,

$$\mathbf{g}(\mathbf{w}) = \mathbf{g}(\hat{\mathbf{w}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{w}} \Big|_{(\mathbf{w}=\hat{\mathbf{w}})} (\mathbf{w} - \hat{\mathbf{w}}) + \cdots$$

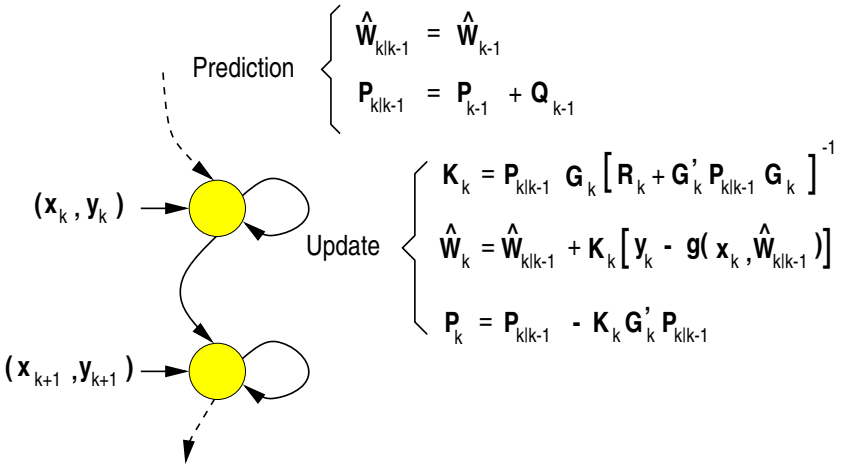

Figure 2: Extended Kalman filter predictor-corrector representation.

The EKF equations for the model of equations (1) and (2) and a linear expansion of $\mathbf{g}(.)$ are given by (see Figure 2):

$$\begin{aligned}
\hat{\mathbf{w}}_{k+1|k} &= A\hat{\mathbf{w}}_k \\
P_{k+1|k} &= AP_kA^T + Q \\
K_{k+1} &= P_{k+1|k}G_{k+1}[R + G_{k+1}^T P_{k+1|k} G_{k+1}]^{-1} \\
\hat{\mathbf{w}}_{k+1} &= \hat{\mathbf{w}}_{k+1|K} + K_{k+1}(\mathbf{y}_{k+1} - \mathbf{g}(\mathbf{x}_k, \hat{\mathbf{w}}_{k+1|k})) \\
P_{k+1} &= P_{k+1|k} - K_{k+1} G_{k+1}^T P_{k+1|k}
\end{aligned}$$

where K_{k+1} is known as the Kalman gain matrix. In the general multiple input, multiple output (MIMO) case, $\mathbf{g} \in \Re^m$ is a vector function and G represents the Jacobian matrix:

$$G = \frac{\partial \mathbf{g}}{\partial \mathbf{w}} \Big|_{(\mathbf{w}=\hat{\mathbf{w}})} = \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial \mathbf{w}_1} & \frac{\partial \mathbf{g}_2}{\partial \mathbf{w}_1} & \cdots & \frac{\partial \mathbf{g}_m}{\partial \mathbf{w}_1} \\ \frac{\partial \mathbf{g}_1}{\partial \mathbf{w}_2} & & & \\ \vdots & & & \vdots \\ \frac{\partial \mathbf{g}_1}{\partial \mathbf{w}_k} & \cdots & & \frac{\partial \mathbf{g}_m}{\partial \mathbf{w}_q} \end{bmatrix}^T$$

Since the EKF is a suboptimal estimator based on linearisation of a nonlinear mapping, $\hat{\mathbf{w}}$ is only an approximation to the expected value and, strictly speaking, P_k is an approximation to the covariance matrix. In mathematical terms:

$$\begin{aligned}
\hat{\mathbf{w}}_k &\approx \mathbf{E}[\mathbf{w}_k | Y_k] \\
P_k &\approx \mathbf{E}[(\mathbf{w}_k - \hat{\mathbf{w}}_k)^T (\mathbf{w}_k - \hat{\mathbf{w}}_k) | Y_k]
\end{aligned}$$

It is also important to point out that the EKF may diverge as a result of its inherent approximations. The consistency of the EKF may be evaluated by means of extensive Monte Carlo simulations (Bar-Shalom and Li 1993).

In this paper, we avoid the divergence problem by smoothing the EKF estimates and applying the EM algorithm in the learning phase. As shown in the following section, the EM algorithm will always grant convergence to a local maximum of the data likelihood.

Smoothing often entails forward and backward filtering over a segment of data so as to obtain improved averaged estimates. Various techniques have been proposed to accomplish this goal (Gelb 1974, Jazwinski 1970). In our work, we make use of the well known Rauch-Tung-Striebel smoother (Gelb 1974, Rauch et al. 1965). After computing the forward estimates $\hat{\mathbf{w}}_k$ and P_k with the EKF, over a segment of N samples, the Rauch-Tung-Striebel smoother makes use of the following backward recursions:

$$\begin{aligned} J_{k-1} &= P_{k-1} A^T P_{k|k-1}^{-1} \\ \hat{\mathbf{w}}_{k-1|N} &= \hat{\mathbf{w}}_{k-1} J_{k-1} (\hat{\mathbf{w}}_{k|N} - A\hat{\mathbf{w}}_{k-1}) \\ P_{k-1|N} &= P_{k-1} + J_{k-1}(P_{k|N} - P_{k|k-1}) J_{k-1}^T \\ P_{k,k-1|N} &= P_k J_{k-1}^T + J_k (P_{k+1,k|N} - AP_k) J_{k-1}^T \end{aligned}$$

where the parameters, covariance and cross-covariance are defined as follows:

$$\begin{aligned} \hat{\mathbf{w}}_{k|N} &\approx \mathbf{E}[\mathbf{w}_k | Y_N] \\ P_{k|N} &\approx \mathbf{E}[(\mathbf{w}_k - \hat{\mathbf{w}}_k)^T (\mathbf{w}_k - \hat{\mathbf{w}}_k) | Y_N] \\ P_{k,k-1|N} &\approx \mathbf{E}[(\mathbf{w}_k - \hat{\mathbf{w}}_k)^T (\mathbf{w}_{k-1} - \hat{\mathbf{w}}_{k-1}) | Y_N] \end{aligned}$$

They may be initialised with the following values:

$$\begin{aligned} \hat{\mathbf{w}}_{N|N} &= \hat{\mathbf{w}}_N \\ P_{N|N} &= P_N \\ P_{N,N-1|N} &= (I - K_N G_N^T) A P_{N-1} \end{aligned}$$

The extended Kalman smoother provides a minimum variance Gaussian approximation to the posterior probability density function $\mathrm{p}(\mathbf{w}|Y)$ (de Freitas et al. 1997). In many cases, $\mathrm{p}(\mathbf{w}|Y)$ is a multi-modal (several peaks) function. In this scenario, it is possible to use a committee of Kalman smoothers, where each individual smoother approximates a particular mode, to produce a more accurate approximation (Bar-Shalom and Li 1993, Blom and Bar-Shalom 1988, Kadirkamanathan and Kadirkamanathan 1995, Kadirkamanathan and Kadirkamanathan 1996). The parameter covariances of the individual estimators may be used to determine the contribution of each estimator to the committee. In addition, the parameter covariances serve the purpose of placing confidence intervals on the output prediction.

The immediate availability of confidence intervals and of mixing coefficients, required to generate mixtures of models, has motivated us to train neural networks with the extended Kalman smoothing algorithm.

4.2 Training MLPs with the EKF

One of the earliest implementations of EKF trained MLPs is due to Singhal and Wu (Singhal and Wu 1988). In their method, the network weights are grouped into a single vector $\mathbf{w}$ that is updated in accordance with the EKF equations. The entries of the Jacobian matrix are calculated by back-propagating the m output values $\{y_1(t), y_2(t), \cdots, y_m(t)\}$ through the network. An example of how to do this for a simple MLP is presented in (de Freitas et al. 1997).

The algorithm proposed by Singhal and Wu requires a considerable computational effort. The complexity is of the order mq^2 multiplications per estimation step. Shah, Palmieri and Datum (Shah, Palmieri and Datum 1992) and Puskorius and Feldkamp (Puskorius and Feldkamp 1991) have proposed strategies for decoupling the global EKF estimation algorithm

into local EKF estimation sub-problems. For example, they suggest that the weights of each neuron could be updated independently. The assumption in the local updating strategies is that the weights are decoupled and, consequently, P is a block-diagonal matrix.

The EKF is an improvement over conventional MLP estimation techniques, such as back-propagation, in that it makes use of second order statistics (covariances). These statistics are essential for placing error bars on the predictions and for combining separate networks into committees of networks. It has been proven elsewhere that the back-propagation algorithm is simply a degenerate of the EKF algorithm (de Freitas et al. 1997, Ruck, Rogers, Kabrisky, Maybeck and Oxley 1992). That is, the back-propagation algorithm, in its basic form, makes no use of second order statistics and adaptive, distributed learning rates.

However, the EKF algorithm for training MLPs suffers from serious difficulties, namely choosing the initial conditions $(\mathbf{w}_0, P_0)$, the noise covariance matrices R and Q and the state dynamics matrix A. In our work, we make use of smoothing and parameter estimation via the EM algorithm to obtain estimates for the initial conditions, noise covariances and state dynamics matrix. To understand how this is done, we need to review the EM algorithm briefly.

5 The EM algorithm

So far, we have shown that given a set of parameters $\theta = \{R, Q, A\}$ and a matrix of N measurements Y, it is possible to compute the expected values of the states with an extended Kalman smoother. In this section, we present a treatment of the EM algorithm that will allow us to learn the parameters θ of nonlinear state space models.

The EM algorithm is an iterative method for finding a mode of the likelihood function $\mathrm{p}(Y|\theta)$. Its most remarkable attribute is that it ensures an increase in the likelihood function at each iteration. Roughly speaking, the EM algorithm proceeds as follows: (1) estimate the states $\mathbf{w}$ given a set of parameters θ, (2) estimate the parameters given the new states, (3) re-estimate the states with the new parameters, and so forth.

It is convenient to think of $\mathbf{w}$ either as missing observations or as latent variables. EM is particularly useful because many models, such as mixtures and hierarchical models, may be re-expressed in augmented parameter spaces, where the extra parameters $\mathbf{w}$ can be thought of as missing data. That is, in situations where it is hard to maximise $\mathrm{p}(Y|\theta)$, EM will allow us to accomplish this by working with $\mathrm{p}(Y, \mathbf{w}|\theta)$.

In some pattern recognition scenarios, it is useful to interpret $\mathbf{w}$ as the latent causes responsible for generating a particular density model (Bishop, Svensén and Williams 1996). In such applications, the number of hidden variables is typically much smaller than the number of observed variables. Therefore, complex idiosyncrasies in the data may be mapped into much lower dimensions, where patterns, such as causes, are clearly manifest. Figure 3 shows a schematic representation of this data visualisation technique proposed by Bishop (Bishop et al. 1996).

To gain more insight into the EM method, let us express the likelihood function as follows:

$$\mathrm{p}(Y|\theta) = \mathrm{p}(Y|\theta)\frac{\mathrm{p}(\mathbf{w}|Y,\theta)}{\mathrm{p}(\mathbf{w}|Y,\theta)} = \frac{\mathrm{p}(\mathbf{w},Y|\theta)}{\mathrm{p}(\mathbf{w}|Y,\theta)}$$

Taking the logarithms of both sides, yields the following identity:

$$\ln \mathrm{p}(Y|\theta) = \ln \mathrm{p}(\mathbf{w},Y|\theta) - \ln \mathrm{p}(\mathbf{w}|Y,\theta)$$

Let us treat $\mathbf{w}$ as a random variable with the distribution $\mathrm{p}(\mathbf{w}|Y,\theta^{\mathrm{old}})$, where θ^{old} is the current guess. If we then take expectations on both sides of the previous identity, while remembering that the left hand side does not depend on $\mathbf{w}$, we get:

$$\ln \mathrm{p}(Y|\theta) = \mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] - \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y,\theta)] \tag{3}$$

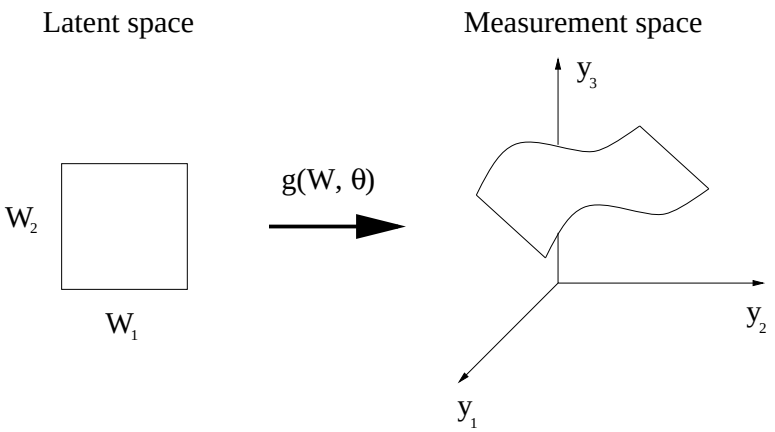

Figure 3: Data visualisation with latent models.

where the expectations involve averaging over $\mathbf{w}$ under the distribution $\mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})$. For example:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)] = \int [\ln \mathrm{p}(\mathbf{w}, Y|\theta)] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w}$$

A key result for the EM algorithm (see Appendix A for a proof) is that the second term on the right side of equation (3) is maximised for θ^{old}. That is:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \geq \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta)]$$

for any θ.

To apply the EM algorithm, we need to compute the first term on the right side of equation (3) repeatedly. The aim is to maximise this term at each iteration. One method of maximising it is discussed in detail in the next section. For the time being, let us assume that we can maximise it, that is:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta^{\mathrm{new}})] \geq \mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta^{\mathrm{old}})]$$

Then, it follows that the likelihood function also increases at every iteration. To demonstrate this important result, consider the change in likelihood for a single iteration:

$$\begin{aligned} \ln \mathrm{p}(Y|\theta^{\mathrm{new}}) - \ln \mathrm{p}(Y|\theta^{\mathrm{old}}) &= \left(\mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta^{\mathrm{new}})] - \mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta^{\mathrm{old}})] \right) \\ &\quad - \left(\mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{new}})] - \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \right) \end{aligned}$$

The right hand side of the above equation is positive because we are averaging under $\mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})$. Consequently, the likelihood function is guaranteed to increase at each iteration.

The EM algorithm's name originates from the steps that are required to increase $\mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)]$, namely compute the Expectation and then Maximise. The EM algorithm, thus involves the following steps:

Initialisation : Start with a guess for θ^0.

E-step : Determine the expected log-likelihood density function of the complete data given the current estimate θ^{old}:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)] = \int [\ln \mathrm{p}(\mathbf{w}, Y|\theta)] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w}$$

M-step : Compute a new value of θ that maximises the expected log-likelihood of the complete data. The maximum can be found by simple differentiation of the expected log-likelihood with respect to θ.

Note that at the M-step, we only require an increase in the expected log-likelihood of the complete data. That is, we do not need to find the maximum. This is the basis for generalised EM (GEM) algorithms (Dempster et al. 1977, Gelman, Carlin, Stern and Rubin 1995).

The price to pay for the simplicity and stability of the EM algorithm is slow convergence, especially in the presence of large amounts of missing information (Gelman et al. 1995, Meng and Rubin 1992). Although several methods have been proposed to mitigate the convergence problem, they often jeopardise the simplicity and stability of the algorithm (Meng and Rubin 1992).

It is instructive to consider an alternative view of the EM algorithm, suggested by Neal and Hinton (Neal and Hinton 1998). By marginalising the log-likelihood of the complete data, one obtains the log-likelihood of the observed data:

$$\ln \mathrm{p}(Y|\theta) = \ln \int \mathrm{p}(\mathbf{w}, Y|\theta) \mathrm{d}\mathbf{w}$$

If we now consider any distribution Q over the latent variables $\mathbf{w}$, we may expand the log-likelihood of the observed data as follows:

$$\begin{aligned} \ln \int \mathrm{p}(\mathbf{w}, Y|\theta) \mathrm{d}\mathbf{w} &= \ln \int Q(\mathbf{w}) \frac{\mathrm{p}(\mathbf{w}, Y|\theta)}{Q(\mathbf{w})} \mathrm{d}\mathbf{w} \\ &\geq \int Q(\mathbf{w}) \ln \frac{\mathrm{p}(\mathbf{w}, Y|\theta)}{Q(\mathbf{w})} \mathrm{d}\mathbf{w} \\ &= \int Q(\mathbf{w}) \ln \mathrm{p}(\mathbf{w}, Y|\theta) \mathrm{d}\mathbf{w} - \int Q(\mathbf{w}) \ln Q(\mathbf{w}) \mathrm{d}\mathbf{w} \\ &= \mathcal{L}(Q, \theta) \end{aligned}$$

where we have made use of Jensen's inequality (Bishop 1995). The lower bound is the negative of the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951). The KL divergence is a measure of the distance between two distributions. Therefore, by minimising the KL divergence (increasing the lower bound), we increase the log-likelihood of the observed data. If we define the energy of a configuration to be $-\ln \mathrm{p}(\mathbf{w}, Y|\theta)$, the lower bound is known in statistical physics as the free energy. It is given by the expected energy under Q minus the entropy of Q. Note that, from the results in this section, the inequality becomes an equality when $Q(\mathbf{w}) = \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})$.

6 The EM algorithm for nonlinear state space models

To derive the EM algorithm for nonlinear state space models we need to develop a probabilistic model for equations (1) and (2). We assume the likelihood of the data given the states, initial conditions and evolution of the states to be represented by Gaussian distributions. That is, if the initial guess of the states and covariance is given by μ and Π, then:

$$\mathrm{p}(\mathbf{w}_1) = \frac{1}{(2\pi)^{q/2}|\Pi|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{w}_1 - \mu)^T \Pi^{-1} (\mathbf{w}_1 - \mu) \right]$$

$$\mathrm{p}(\mathbf{w}_k|\mathbf{w}_{k-1}) = \frac{1}{(2\pi)^{q/2}|Q|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{w}_k - A\mathbf{w}_{k-1})^T Q^{-1} (\mathbf{w}_k - A\mathbf{w}_{k-1}) \right]$$

$$\mathrm{p}(\mathbf{y}_k|\mathbf{w}_k) = \frac{1}{(2\pi)^{m/2}|R|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}_k - \mathbf{g}(\mathbf{w}_k, \mathbf{x}_k))^{\mathrm{T}} \mathrm{R}^{-1} (\mathbf{y}_k - \mathbf{g}(\mathbf{w}_k, \mathbf{x}_k)) \right]$$

Let us use Y to denote the N measurements $\{\mathbf{y}_1 \cdots \mathbf{y}_N\}$ and $\mathbf{w}$ to denote $\{\mathbf{w}_1 \cdots \mathbf{w}_N\}$. Under the initial model assumptions of uncorrelated noises and Gauss-Markov state evolution, the likelihood of the complete data is given by:

$$\mathrm{p}(\mathbf{w}, Y|\theta) = \mathrm{p}(\mathbf{w}_1) \prod_{k=2}^{N} \mathrm{p}(\mathbf{w}_k|\mathbf{w}_{k-1}) \prod_{k=1}^{N} \mathrm{p}(\mathbf{y}_k|\mathbf{w}_k)$$

Hence, the log-likelihood of the complete data is given by the following expression:

$$\begin{aligned}\ln \mathrm{p}(\mathbf{w}, Y|\theta) &= -\sum_{k=1}^{N}\left[\ \frac{1}{2}(\mathbf{y}_k - \mathbf{g}(\mathbf{w}_\mathrm{k}, \mathbf{x}_\mathrm{k}))^\mathrm{T}\mathrm{R}^{-1}(\mathbf{y}_\mathrm{k} - \mathbf{g}(\mathbf{w}_\mathrm{k}, \mathbf{x}_\mathrm{k}))\ \right] - \frac{\mathrm{N}}{2}\ln|\mathrm{R}| \\ &\quad -\sum_{k=2}^{N}\left[\ \frac{1}{2}(\mathbf{w}_k - A\mathbf{w}_{k-1})^T Q^{-1}(\mathbf{w}_k - A\mathbf{w}_{k-1})\ \right] - \frac{N-1}{2}\ln|Q| \\ &\quad -\frac{1}{2}(\mathbf{w}_1 - \mu)^T\Pi^{-1}(\mathbf{w}_1 - \mu) - \frac{1}{2}\ln|\Pi| - \frac{N(m+q)}{2}\ln(2\pi) \qquad (4)\end{aligned}$$

As discussed in the previous section, all we need to do now is to compute the expectation of $\ln \mathrm{p}(\mathbf{w}, Y|\theta)$ and then differentiate the result with respect to the parameters θ so as to maximise it. The EM algorithm for nonlinear state space models will thus involve computing the expected values of the states and covariances with the extended Kalman smoother and then maximising the parameters θ with the formulae obtained by differentiating the expected log-likelihood. Before we proceed to derive these formulas we require to revise a few basic mathematical preliminaries.

6.1 Mathematical preliminaries

The following results are required to understand the subsequent derivations:

1. The trace of a matrix A, denoted by $\mathrm{tr}(A)$, is the sum of the diagonal entries of A. The trace is invariant under circular permutations in its argument, consequently:

   $$\mathrm{tr}(ABC) = \mathrm{tr}(BCA) = \mathrm{tr}(CAB)$$

   In addition, the trace is a linear operator:

   $$\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B) \qquad \text{and} \qquad \mathrm{tr}(\alpha A) = \alpha\mathrm{tr}(A)$$

2. The following expectation for a vector $\mathbf{x}$ and a matrix A holds:

   $$\begin{aligned}\mathbf{E}[\mathbf{x}^T A\mathbf{x}] &= \mathbf{E}[\mathrm{tr}(\mathbf{x}^T A\mathbf{x})] \\ &= \mathbf{E}[\mathrm{tr}(A\mathbf{x}\mathbf{x}^T)] \\ &= \mathrm{tr}(A\mathbf{E}[\mathbf{x}\mathbf{x}^T])\end{aligned}$$

3. The following sufficient statistics, which follow from elementary probability theory, will be required:

   $$\begin{aligned}\mathbf{E}[\mathbf{w}_k|Y] &= \hat{\mathbf{w}}_{k|N} \\ \mathbf{E}[\mathbf{w}_k\mathbf{w}_k^T|Y] &= P_{k|N} + \hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k|N}^T \\ \mathbf{E}[\mathbf{w}_k\mathbf{w}_{k-1}^T|Y] &= P_{k,k-1|N} + \hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k-1|N}^T\end{aligned}$$

4. We shall need the following results for matrix differentiation (see for example (Graham 1981)):

   $$\begin{aligned}\frac{\partial \ln|A|}{\partial A} &= (A^{-1})^T \\ \frac{\partial \mathrm{tr}(BA)}{\partial A} &= B^T \\ \frac{\partial \mathrm{tr}(A^T BA)}{\partial A} &= BA + B^T A\end{aligned}$$

6.2 Computing the expectation of the log-likelihood

If we take the expectation of the log-likelihood for the complete data, by averaging over $\mathbf{w}$ under the distribution $\mathrm{p}(\mathbf{w}|Y,\theta^{\mathrm{old}})$, we get the following expression:

$$\begin{aligned}
\mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] &= -\frac{N}{2}\ln|R| - \frac{N-1}{2}\ln|Q| - \frac{1}{2}\ln|\Pi| - \frac{N(m+q)}{2}\ln(2\pi) \\
&\quad -\sum_{k=1}^{N}\frac{1}{2}\mathbf{E}\left[\, \mathbf{y}_k^T R^{-1}\mathbf{y}_k - \mathbf{y}_k^T R^{-1}\mathbf{g}(\mathbf{w}_\mathrm{k},\mathbf{x}_\mathrm{k}) \right. \\
&\qquad \left. - \mathbf{g}(\mathbf{w}_\mathrm{k},\mathbf{x}_\mathrm{k})^{\mathrm{T}}\mathrm{R}^{-1}\mathbf{y}_\mathrm{k} + \mathbf{g}(\mathbf{w}_\mathrm{k},\mathbf{x}_\mathrm{k})^{\mathrm{T}}\mathrm{R}^{-1}\mathbf{g}(\mathbf{w}_\mathrm{k},\mathbf{x}_\mathrm{k}) \,\right] \\
&\quad -\sum_{k=2}^{N}\frac{1}{2}\mathbf{E}\left[\, \mathbf{w}_k^T Q^{-1}\mathbf{w}_k - \mathbf{w}_k^T Q^{-1}A\mathbf{w}_{k-1} - \mathbf{w}_{k-1}^T A^T Q^{-1}\mathbf{w}_k \right. \\
&\qquad \left. + \mathbf{w}_{k-1}^T A^T Q^{-1}A\mathbf{w}_{k-1} \,\right] \\
&\quad -\frac{1}{2}\mathbf{E}\left[\, \mathbf{w}_1^T\Pi^{-1}\mathbf{w}_1 - \mathbf{w}_1^T\Pi^{-1}\mu - \mu^T\Pi^{-1}\mathbf{w}_1 + \mu^T\Pi^{-1}\mu \,\right]
\end{aligned}$$

We need to digress briefly to compute the expectation of the measurements mapping $\mathbf{g}(\mathbf{w}_\mathrm{k},\mathbf{x}_\mathrm{k})$. We should recall that the EKF approximation to this mapping is given by:

$$\mathbf{g}(\mathbf{w}) = \mathbf{g}(\hat{\mathbf{w}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\hat{\mathbf{w}})}(\mathbf{w}-\hat{\mathbf{w}}) + \cdots$$

Consequently, if we take expectations on both sides of the equation, we get:

$$\mathbf{E}[\mathbf{g}(\mathbf{w}_\mathrm{k})] = \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}})$$

and

$$\begin{aligned}
\mathbf{E}[(\mathbf{g}(\mathbf{w})-\mathbf{g}(\hat{\mathbf{w}}))^{\mathrm{T}}(\mathbf{g}(\mathbf{w})-\mathbf{g}(\hat{\mathbf{w}}))] &= \mathbf{E}\left[\, \left(\, \mathbf{g}(\hat{\mathbf{w}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\hat{\mathbf{w}})}(\mathbf{w}-\hat{\mathbf{w}}) - \mathbf{g}(\hat{\mathbf{w}}) \,\right)^{\mathrm{T}} \right. \\
&\qquad \left. \left(\, \mathbf{g}(\hat{\mathbf{w}}) + \frac{\partial \mathbf{g}}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\hat{\mathbf{w}})}(\mathbf{w}-\hat{\mathbf{w}}) - \mathbf{g}(\hat{\mathbf{w}}) \,\right) \,\right] \\
&= \mathbf{E}\left[\, \left(\, \frac{\partial \mathbf{g}}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\hat{\mathbf{w}})}(\mathbf{w}-\hat{\mathbf{w}}) \,\right)^{T} \right. \\
&\qquad \left. \left(\, \frac{\partial \mathbf{g}}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\hat{\mathbf{w}})}(\mathbf{w}-\hat{\mathbf{w}}) \,\right) \,\right] \\
&= \mathbf{E}\left[\, \left(\, (\mathbf{w}-\hat{\mathbf{w}})G \,\right)^{T} \left(\, (\mathbf{w}-\hat{\mathbf{w}})G \,\right) \,\right] \\
&= \mathbf{E}\left[\, G^T(\mathbf{w}-\hat{\mathbf{w}})^T(\mathbf{w}-\hat{\mathbf{w}})G \,\right] \\
&= G^T P G
\end{aligned}$$

Hence, under the distribution $\mathrm{p}(\mathbf{w}|Y,\theta^{\mathrm{old}})$, it follows that:

$$\mathbf{E}[\mathbf{g}(\mathbf{w}_\mathrm{k})^{\mathrm{T}}\mathbf{g}(\mathbf{w}_\mathrm{k})] = \mathrm{G}_\mathrm{k}^{\mathrm{T}}\mathrm{P}_{\mathrm{k|N}}\mathrm{G}_\mathrm{k} + \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}})^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}})$$

Using these results for the expectation of the measurements mapping, together with the results 1, 2 and 3 of Section 6.1, the expectation of the log-likelihood becomes:

$$\begin{aligned}
\mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] &= -\frac{N}{2}\ln|R| - \frac{N-1}{2}\ln|Q| - \frac{1}{2}\ln|\Pi| - \frac{N(m+q)}{2}\ln(2\pi) \\
&\quad -\sum_{k=1}^{N}\frac{1}{2}\mathrm{tr}\left(\, R^{-1}\left[\, \mathbf{y}_k^T\mathbf{y}_k - \mathbf{y}_k^T\mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_\mathrm{k}) - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_\mathrm{k})^{\mathrm{T}}\mathbf{y}_\mathrm{k} \right.\right. \\
&\qquad \left.\left. + \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_\mathrm{k})^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_\mathrm{k}) + \mathrm{G}_\mathrm{k}^{\mathrm{T}}\mathrm{P}_{\mathrm{k|N}}\mathrm{G}_\mathrm{k} \,\right] \,\right)
\end{aligned}$$

$$
\begin{aligned}
&-\sum_{k=2}^{N}\frac{1}{2}\mathrm{tr}\,\big(\,Q^{-1}\,\big[\,\hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k|N}^{T}+P_{k|N}-2A(\hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k-1|N}^{T}+P_{k,k-1|N})^{T}\\
&\qquad\qquad+A(\hat{\mathbf{w}}_{k-1}\hat{\mathbf{w}}_{k-1|N}^{T}+P_{k-1|N})A^{T}\,\big]\,\big)\\
&-\frac{1}{2}\mathrm{tr}\,\big(\,\Pi^{-1}\,\big[\,\hat{\mathbf{w}}_{1|N}\hat{\mathbf{w}}_{1|N}^{T}+P_{1|N}-2\hat{\mathbf{w}}_{1|N}\mu^{T}+\mu\mu^{T}\,\big]\,\big)
\end{aligned}
$$

Completing squares and using the following abbreviations:

$$
\begin{aligned}
\Gamma &= \sum_{k=2}^{N}\hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k|N}^{T}+P_{k|N}\\
\Delta &= \sum_{k=2}^{N}\hat{\mathbf{w}}_{k-1|N}\hat{\mathbf{w}}_{k-1|N}^{T}+P_{k-1|N}\\
\Upsilon &= \sum_{k=2}^{N}\hat{\mathbf{w}}_{k|N}\hat{\mathbf{w}}_{k-1|N}^{T}+P_{k,k-1|N}
\end{aligned}
$$

we get our final expression for the expectation of the log-likelihood:

$$
\begin{aligned}
\mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] &= -\frac{N}{2}\ln|R|-\frac{N-1}{2}\ln|Q|-\frac{1}{2}\ln|\Pi|-\frac{N(m+q)}{2}\ln(2\pi)\\
&\quad-\sum_{k=1}^{N}\frac{1}{2}\mathrm{tr}\,\big(\,R^{-1}\,\big[\,(\mathbf{y}_k-\mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_{\mathrm{k}}))(\mathbf{y}_{\mathrm{k}}-\mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}},\mathbf{x}_{\mathrm{k}}))^{\mathrm{T}}\\
&\qquad\qquad+G_k^{T}P_{k|N}G_k\,\big]\,\big)\\
&\quad-\frac{1}{2}\mathrm{tr}\,\big(\,Q^{-1}\,\big[\,\Gamma-2A\Upsilon^{T}+A\Delta A^{T}\,\big]\,\big)\\
&\quad-\frac{1}{2}\mathrm{tr}\,\big(\,\Pi^{-1}\,\big[\,(\hat{\mathbf{w}}_{1|N}-\mu)(\hat{\mathbf{w}}_{1|N}-\mu)^{T}+P_{1|N}\,\big]\,\big) \qquad (5)
\end{aligned}
$$

6.3 Differentiating the expected log-likelihood

To maximise the expected value of the log-likelihood with respect to the parameters θ, we need to compute the derivatives with respect to each parameter individually. This is done in the subsequent sections, where we make use of the results from point 4 in Section 6.1.

6.3.1 Maximum with respect to A

Differentiating the expected log-likelihood with respect to A yields:

$$
\begin{aligned}
\frac{\partial}{\partial A}\mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] &= -\frac{1}{2}\frac{\partial}{\partial A}\mathrm{tr}\,\big(\,Q^{-1}\,\big[\,\Gamma-2A\Upsilon^{T}+A\Delta A^{T}\,\big]\,\big)\\
&= -\frac{1}{2}\,\big(\,-2Q^{-1}\Upsilon^{T}+2Q^{-1}A\Delta\,\big)
\end{aligned}
$$

Equating this result to zero, yields the value of A that maximises the log-likelihood:

$$
A=\Upsilon\Delta^{-1} \qquad (6)
$$

6.3.2 Maximum with respect to R

Differentiating the expected log-likelihood with respect to R^{-1} gives:

$$
\frac{\partial}{\partial R^{-1}}\mathbf{E}[\ln \mathrm{p}(\mathbf{w},Y|\theta)] = \frac{\partial}{\partial R^{-1}}\,\big(\,\frac{N}{2}\ln|R^{-1}|-\sum_{k=1}^{N}\frac{1}{2}\mathrm{tr}\,\big(\,R^{-1}\,\big[\,G_k^{T}P_{k|N}G_k
$$

$$+ (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))(\mathbf{y}_\mathrm{k} - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))^\mathrm{T} \rceil \;) \;)$$

$$= \frac{N}{2} R - \sum_{k=1}^{N} \frac{1}{2} \left(\; G_k^T P_{k|N} G_k + (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))(\mathbf{y}_\mathrm{k} - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))^\mathrm{T} \; \right)$$

Hence, by equating the above result to zero, the maximum of the log-likelihood with respect to R is given by:

$$R = \frac{1}{N} \sum_{k=1}^{N} \left(\; G_k^T P_{k|N} G_k + (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))(\mathbf{y}_\mathrm{k} - \mathbf{g}(\hat{\mathbf{w}}_{\mathrm{k|N}}, \mathbf{x}_\mathrm{k}))^\mathrm{T} \; \right) \tag{7}$$

6.3.3 Maximum with respect to Q

Following the same steps, the derivative of the expected log-likelihood with respect to Q^{-1} is given by:

$$\frac{\partial}{\partial Q^{-1}} \mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)] = \frac{N-1}{2} Q - \frac{1}{2} \left(\; \Gamma - 2A\Upsilon^T + A\Delta A^T \; \right)$$

Hence, equating to zero and using the result that $A = \Upsilon\Delta^{-1}$, the maximum of the log-likelihood with respect to Q is given by:

$$Q = \frac{1}{N-1} \left(\; \Gamma - \Upsilon\Delta^{-1}\Upsilon^T \; \right) \tag{8}$$

6.3.4 Maximum with respect to μ

It is also possible to treat the initial conditions as parameters and improve their estimates in the M-step of the EM algorithm. Finding the derivative of the expected log-likelihood with respect to the initial states gives:

$$\frac{\partial}{\partial \mu} \mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)] = \frac{1}{2} \Pi^{-1} \left(\; -2\hat{\mathbf{w}}_{1|N} + 2\mu \right)$$

Hence, the initial value for the states should be:

$$\mu = \hat{\mathbf{w}}_{1|N} \tag{9}$$

6.3.5 Maximum with respect to Π

The derivative of the expected log-likelihood with respect to the inverse of the initial covariance gives:

$$\frac{\partial}{\partial \Pi^{-1}} \mathbf{E}[\ln \mathrm{p}(\mathbf{w}, Y|\theta)] = \frac{\Pi}{2} - \frac{1}{2} \left(\; (\hat{\mathbf{w}}_{1|N} - \mu)(\hat{\mathbf{w}}_{1|N} - \mu)^T + P_{1|N} \right)$$

Therefore, the initial covariance should be updated as follows:

$$\Pi = P_{1|N} \tag{10}$$

6.4 The E and M steps for nonlinear state space models

We can now prescribe the EM algorithm for nonlinear state space models as follows:

Initialisation : Start with a guess for $\theta = \{R, Q, A, \Pi, \mu\}$.

E-step : Determine the expected values $\hat{\mathbf{w}}_{k|N}$, $P_{k|N}$ and $P_{k,k-1|N}$, given the current parameter estimate θ^{old}, using the extended Kalman smoothing equations described in Section 4.1.

M-step : Compute new values of the parameters $\theta = \{R, Q, A, \Pi, \mu\}$ using equations (6) to (10).

7 Experiments

7.1 Simple regression example

For the purposes of demonstrating the method, we address the problem of learning the following nonlinear mapping from (x_1, x_2) to y:

$$y = 4\sin(x_1 - 2) + 2x_2 + 5 + \eta$$

where x_1 and x_2 were chosen to be two normal random sequences of 700 samples each . The noise process η was sampled from a zero mean Gaussian distribution with variance $R = 0.5$.

An MLP with 4 sigmoidal neurons in the hidden layer and a linear neuron in the output

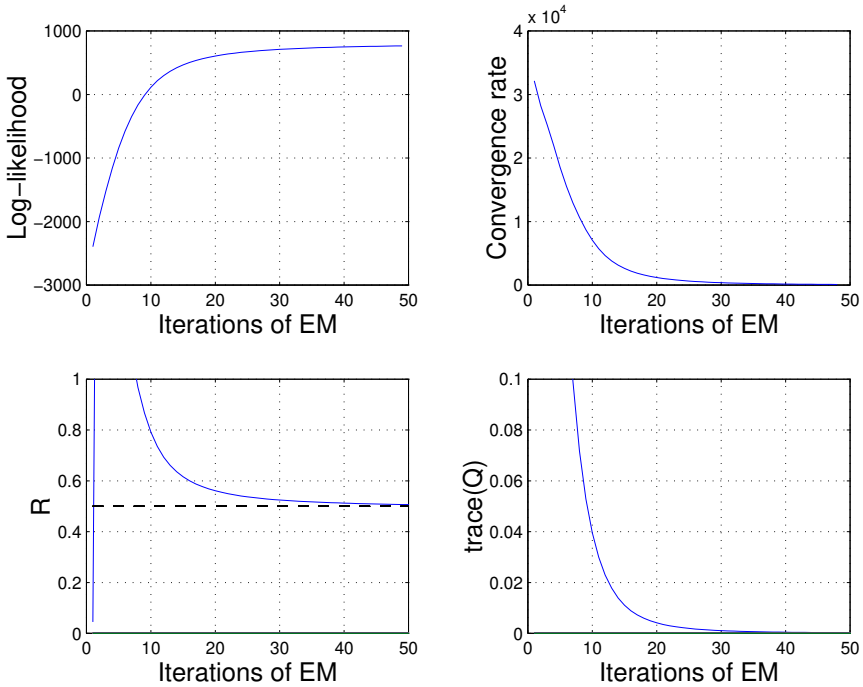

Figure 4: The top plots show the log-likelihood function and the convergence rate (log-likelihood slope) for the simple regression problem. The bottom plots show the convergence of the measurements noise covariance R and the trace of the process noise covariance Q.

layer was used to approximate the measurements mapping. After 50 iterations, as shown in Figure 4, the estimate of observation variance R converges to the true value. In addition, the uncertainty of the model Q goes to zero. As a result, the innovations covariance (variance of the evidence function $p(\mathbf{y}_k|Y_{k-1}, \hat{\mathbf{w}}_{k|k-1}, Q_{k-1}, R_{k-1})$) tends to R over the entire data set, as shown in Figure 5. The top plot of this figure shows that the MLP approximates the true function without fitting the noise. That is, it generalises well. Figure 4 also shows how the log-likelihood increases at each step, thereby demonstrating that the algorithm converges well.

7.2 Robot arm mapping

This data set is often used as a benchmark to compare neural network algorithms[1]. It involves implementing a model to map the joint angle of a robot arm (x_1, x_2) to the position of the

[1]The data set can be found in David Mackay's home page: `http://wol.ra.phy.cam.ac.uk/mackay/`

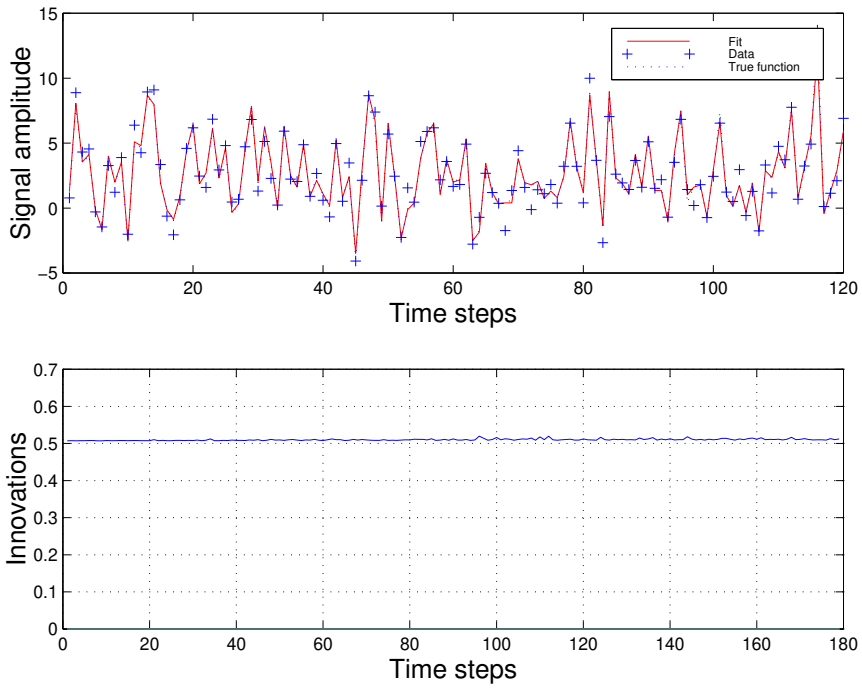

Figure 5: The top plot shows that the MLP fit, for the regression example, approximates the true function; it does not fit the noise. As a result the network exhibits good generalisation performance. The bottom plot shows that the uncertainty in the predictions (innovations) converges to the uncertainty engendered by the measurement noise.

end of the arm (y_1, y_2). The data were generated from the following model:

$$\begin{aligned} y_1 &= 2.0\cos(x_1) + 1.3\cos(x_1 + x_2) + \epsilon_1 \\ y_2 &= 2.0\sin(x_1) + 1.3\sin(x_1 + x_2) + \epsilon_2 \end{aligned}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$; $\sigma = 0.05$. We use the first 200 observations of the data set to train our models and the last 200 observations to test them.

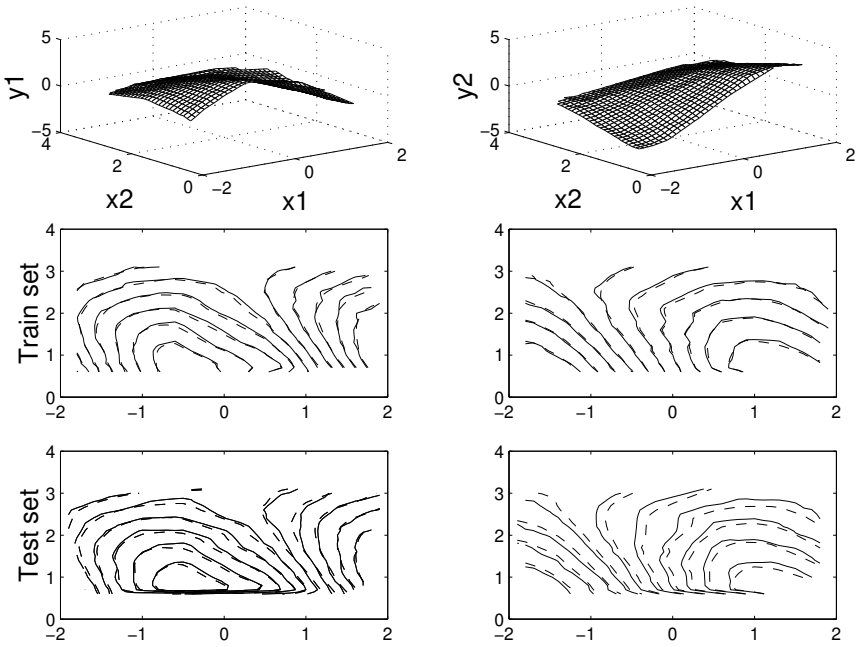

Figure 6: The top plots show the training data surfaces corresponding to each coordinate of the robot arm's position. The Middle and bottom plots show the training and validation data [- -] and the respective MLP mappings [—].

Figure 6 shows the 3D plots of the training data and the contours of the training and test data. The contour plots also include the typical approximations that were obtained using our algorithm and an MLP with 2 linear output neurons and 20 sigmoidal hidden neurons. Figure 7, shows the convergence of the algorithm. In this particular run the training and test mean square errors were 0.0057 and 0.0081 (the minimum bound being $2\sigma^2 = 0.005$). Our mean square errors are of the same magnitude as the ones reported by other researchers (Andrieu, de Freitas and Doucet 1999, Holmes and Mallick 1998, Mackay 1992, Neal 1996, Rios Insua and Müller 1998). Figure 7 also shows the two diagonal entries of the measurements noise covariance and the trace of the process noise covariance. They behave as expected.

7.3 Classification with medical data

Here, we consider an interesting nonlinear classification data set[2] collected as part of a study to identify patients with muscle tremor (Roberts, Penny and Pillot 1996, Spyers-Ashby, Bain and Roberts 1998). The data was gathered from a group of patients (9 with, primarily, Parkinson's disease or multiple sclerosis) and from a control group (not exhibiting the disease). Arm muscle tremor was measured with a 3-D mouse and a movement tracker in three linear and three angular directions. The time series of the measurements were parameterised using

[2]The data is available at Stephen Roberts' home page: `http://www.ee.ic.ac.uk/hp/staff/sroberts.html`

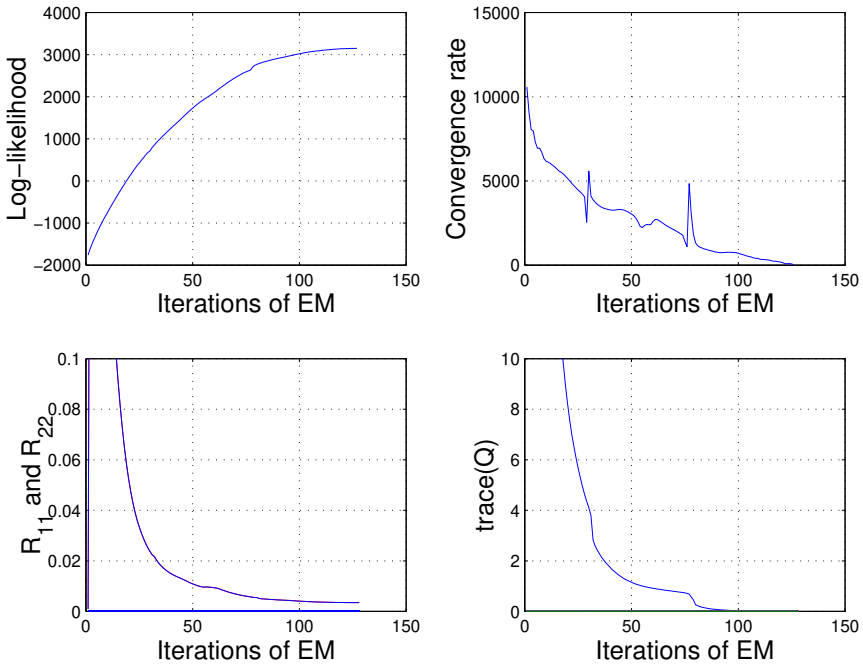

Figure 7: The top plots show the log-likelihood function and the convergence rate (log-likelihood slope) for the robot arm problem. The bottom plots show the convergence of the diagonal entries of the measurements noise covariance R (almost identical) and the trace of the process noise covariance Q.

a set of autoregressive models. The number of features was then reduced to two (Roberts et al. 1996). Figure 9 shows a plot of these features for patient ($\circ$) and control groups ($+$). The figure also shows the decision boundaries (solid lines) and confidence intervals (dashed lines) obtained with an MLP, consisting of 10 sigmoidal hidden neurons and an output linear neuron.

The size of the confidence intervals is given by the innovations covariance. That is, our confidence of correctly classifying a sample occurring within these intervals should be very low. The receiver operating characteristic (ROC) curve, shown in Figure 10, indicates that we can expect to detect patients with a 70% confidence without making any mistakes. The percentage of classification errors in the test set was found to be 15.17. This error is of the same magnitude as previous results (Roberts and Penny 1998). Finally, the convergence properties of the EM algorithm for this application are illustrated in Figure 8.

8 Conclusions

In this paper, we derived an EM algorithm to estimate the neural network weights, measurement noise and model uncertainty jointly. We applied the method to regression and classification tasks. In both cases, we found that it performs well in terms of model accuracy and generalisation ability.

The method is able to estimate the measurement and model uncertainty via the noise covariances R and Q. As a result, it does not overfit the data and does not, necessarily, require an additional test set to cross-validate the data.

Further research avenues include extending the method to other types of noise processes, testing on additional data sets and investigating ways of efficiently initialising the algorithm

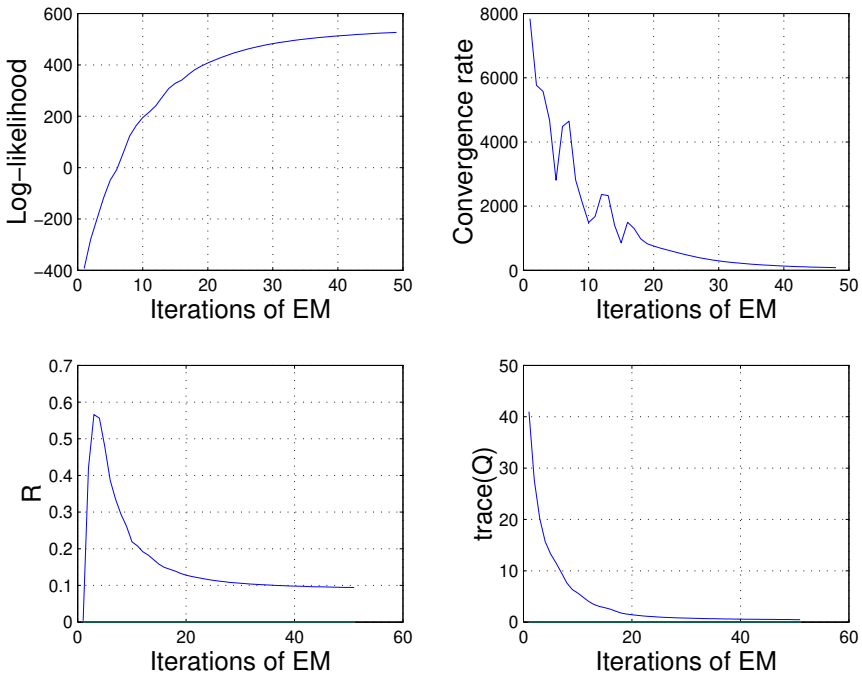

Figure 8: The top plots show the log-likelihood function and the convergence rate (log-likelihood slope) for the tremor data classification problem. The bottom plots show the convergence of the measurements noise covariance R and the trace of the process noise covariance Q.

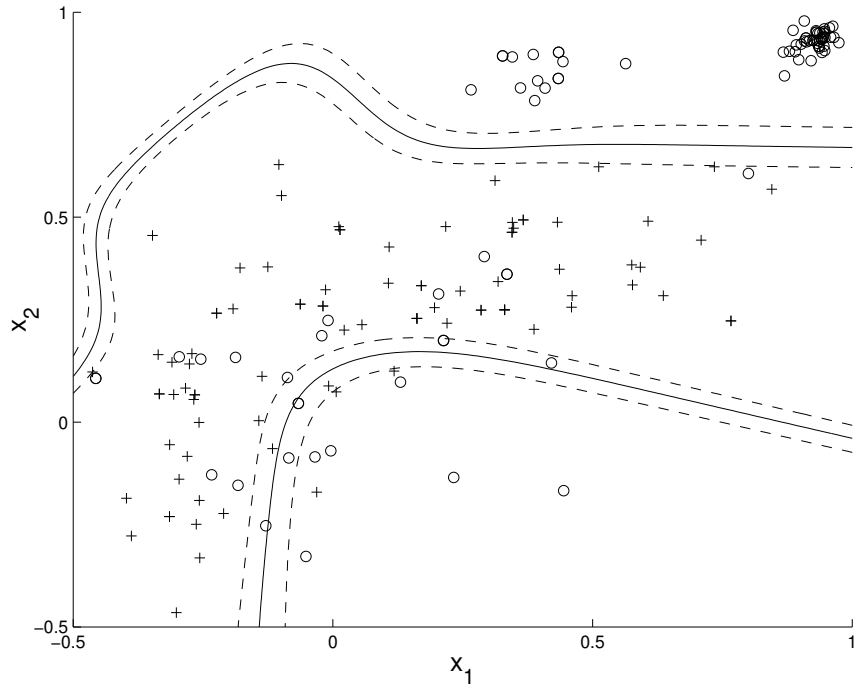

Figure 9: Classification boundaries (–) and confidence intervals (- -) for the MLP classifier. The circles indicate patients, while the crosses represent the control group.

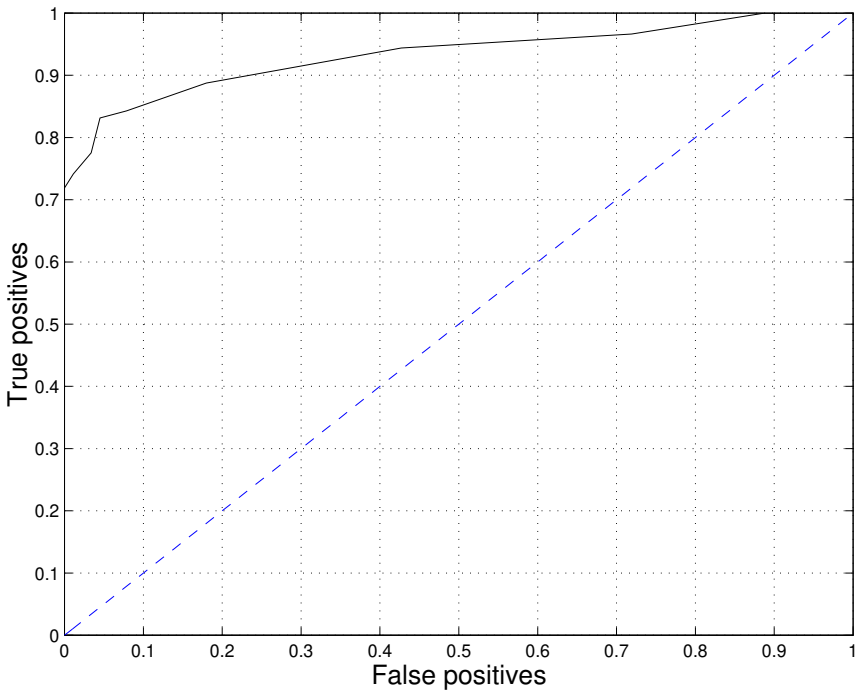

Figure 10: Receiver operating characteristic (ROC) of the classifier for the tremor data.

so as to avoid local minima. Variational approximations might provide an answer for the latter issue.

9 Acknowledgements

We would like to thank Mari Ostendorf (Boston University), Vassilis Digalakis (Technical University of Crete), David Melvin (Cambridge Clinical School) and Ben North (Oxford University) for their valuable comments. We are very grateful to Zoubin Ghahramani (University of Toronto) for his help. We would also like to thank David Stoffer (University of Pittsburgh) for making available some of his technical reports and Stephen Roberts and Will Penny (Imperial College of London) for the tremor data.

João FG de Freitas is financially supported by two University of the Witwatersrand Merit Scholarships, a Foundation for Research Development Scholarship (South Africa), an ORS Award (UK) and a Trinity College External Research Studentship (Cambridge).

A An Important Inequality

In this section, we prove an important inequality which arises in the derivation of the EM algorithm:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \geq \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta)]$$

where the expectations are taken as follows:

$$\int [\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w} \geq \int [\ln \mathrm{p}(\mathbf{w}|Y, \theta)] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w} \tag{11}$$

We begin by noticing that the function $x - 1$ is tangent to the function $\ln x$ at $x = 1$. In addition, the logarithmic function is concave, hence the following inequality holds:

$$\ln x \leq x - 1$$

As a result, it follows that:

$$\begin{aligned}
\mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta)] - \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] &= \int [\ln \mathrm{p}(\mathbf{w}|Y, \theta) - \ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w} \\
&= \int \ln \frac{\mathrm{p}(\mathbf{w}|Y, \theta)}{\mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})} \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w} \\
&\leq \int \left[\frac{\mathrm{p}(\mathbf{w}|Y, \theta)}{\mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})} - 1 \right] \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}}) \mathrm{d}\mathbf{w} \\
&= 0
\end{aligned}$$

Hence:

$$\mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta^{\mathrm{old}})] \geq \mathbf{E}[\ln \mathrm{p}(\mathbf{w}|Y, \theta)]$$

References

Andrieu, C., de Freitas, J. F. G. and Doucet, A. (1999). Robust full Bayesian learning for neural networks, *Technical Report CUED/F-INFENG/TR 343*, Cambridge University, http://svr-www.eng.cam.ac.uk/.

Bar-Shalom, Y. and Li, X. R. (1993). *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Boston.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

Bishop, C. M., Svensén, M. and Williams, C. K. (1996). EM optimization of latent-variable density models, *in* D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (eds), *Advances in Neural Information Processing Systems*, Vol. 8, pp. 465–471.

Blom, H. A. P. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients, *IEEE Transactions on Automatic Control* **33**(8): 780–783.

Chen, C. F. (1981). The EM algorithm to the multiple indicators and multiple causes model via the estimation of the latent variable, *Journal of the American Statistical Association* **76**(375): 704–708.

de Freitas, J. F. G., Niranjan, M. and Gee, A. H. (1997). Hierarchichal Bayesian-Kalman models for regularisation and ARD in sequential learning, *Technical Report CUED/F-INFENG/TR 307*, Cambridge University, http://svr-www.eng.cam.ac.uk/~jfgf.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Series B* **39**: 1–38.

Digalakis, V., Rohlicek, J. R. and Ostendorf, M. (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition, *IEEE Transactions on Speech and Audio Processing* **1**(4): 431–442.

Gelb, A. (ed.) (1974). *Applied Optimal Estimation*, MIT Press.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995). *Bayesian Data Analysis*, Chapman and Hall.

Ghahramani, Z. (1997). Learning dynamic Bayesian networks, *in* C. L. Giles and M. Gori (eds), *Adaptive Processing of Temporal Information*, Lecture Notes in Artificial Intelligence, Springer-Verlag.

Graham, A. (1981). *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood Limited.

Holmes, C. C. and Mallick, B. K. (1998). Bayesian radial basis functions of variable dimension, *Neural Computation* **10**: 1217–1233.

Jazwinski, A. H. (1969). Adaptive filtering, *Automatica* **5**: 475–485.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*, Academic Press.

Kadirkamanathan, V. and Kadirkamanathan, M. (1995). Recursive estimation of dynamic modular RBF networks, *in* D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (eds), *Advances in Neural Information Processing Systems 8*, pp. 239–245.

Kadirkamanathan, V. and Kadirkamanathan, M. (1996). Kalman filter based estimation of dynamic modular networks, *in* S. Usui, Y. Tohkura, S. Katagiri and E. Wilson (eds), *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing VI*, pp. 180–189.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory, *Transactions of the ASME (Jounal of Basic Engineering)* **83D**: 95–108.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency, *Annals of Mathematical Statistics* **22**: 79–86.

Li, X. R. and Bar-Shalom, Y. (1994). A recursive multiple model approach to noise identification, *IEEE Transactions on Aerospace and Electronic Systems* **30**(3): 671–684.

Mackay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks, *Neural Computation* **4**: 448–472.

Mehra, R. K. (1970). On the identification of variances and adaptive Kalman filtering, *IEEE Transactions on Automatic Control* **AC-15**(2): 175–184.

Mehra, R. K. (1971). On-line identification of linear dynamic systems with applications to Kalman filtering, *IEEE Transactions on Automatic Control* **AC-16**(1): 12–21.

Mehra, R. K. (1972). Approaches to adaptive filtering, *IEEE Transactions on Automatic Control* pp. 693–698.

Meng, X. L. and Rubin, D. B. (1992). Recent extensions to the EM algorithm, *in* J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. Smith (eds), *Bayesian Statistics*, Vol. 4, Oxford University Press, pp. 307–320.

Myers, K. A. and Tapley, B. D. (1976). Adaptive sequential estimation of unknown noise statistics, *IEEE Transactions on Automatic Control* pp. 520–523.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics No. 118, Springer-Verlag, New York.

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants, *in* M. I. Jordan (ed.), *Learning in Graphical Models*, Kluwer Academic Press.

North, B. and Blake, A. (1998). Learning dynamical models using expectation-maximisation, *Proceedings of the 6th International Conference on Computer Vision*, Mumbai, India.

Puskorius, G. V. and Feldkamp, L. A. (1991). Decoupled extended Kalman filter training of feedforward layered networks, *International Joint Conference on Neural Networks*, Seattle, pp. 307–312.

Rao, R. P. and Ballard, D. H. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex, *Neural Computation* **9**: 721–763.

Rauch, H. E., Tung, F. and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems, *AIAA Journal* **3**(8): 1445–1450.

Rios Insua, D. and Müller, P. (1998). Feedforward neural networks for nonparametric regression, *Technical Report 98–02*, Institute of Statistics and Decision Sciences, Duke University. Available at http://www.stat.duke.edu.

Roberts, S. J. and Penny, W. D. (1998). Bayesian neural networks for classification: How useful is the evidence framework?, To appear in Neural Networks.

Roberts, S. J., Penny, W. D. and Pillot, D. (1996). Novelty, confidence and errors in connectionist systems, *IEE Colloquium on Intelligent Sensors and Fault Detection*, pp. 1–10.

Roweis, S. and Ghahramani, Z. (1997). A unifying review of linear Gaussian models, Submitted for publication. Available at http://www.cs.toronto.edu/~zoubin/.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Maybeck, P. S. and Oxley, M. E. (1992). Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(6): 686–690.

Shah, S., Palmieri, F. and Datum, M. (1992). Optimal filtering algorithms for fast learning in feedforward neural networks, *Neural Networks* **5**: 779–787.

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm, *Journal of Time Series Analysis* **3**(4): 253–264.

Shumway, R. H. and Stoffer, D. S. (1991). Dynamic linear models with switching, *Journal of the American Statistical Association* **86**(415): 763–769.

Singhal, S. and Wu, L. (1988). Training multilayer perceptrons with the extended Kalman algorithm, *in* D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems*, Vol. 1, San Mateo, CA, pp. 133–140.

Spyers-Ashby, J. M., Bain, P. and Roberts, S. J. (1998). A comparison of fast Fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data, *Journal of Neuroscience Methods* **83**: 35–43.

Tenney, R. R., Hebbert, R. S. and Sandell, N. S. (1977). A tracking filter for maneuvering sources, *IEEE Transactions on Automatic Control* pp. 246–251.

Watson, M. W. and Engle, R. F. (1983). Alternative algorithms for the estimation of dynamic factor, MIMIC and varying coefficient regression models, *Journal of Econometrics* **23**: 385–400.