ITERATED EXTENDED KALMAN SMOOTHING  
WITH EXPECTATION-PROPAGATION

Alexander Ypma and Tom Heskes
SNN Nijmegen, Geert Grootplein 21
6525 EZ Nijmegen, The Netherlands
Email: {ypma, tom}@snn.kun.nl, web: www.snn.kun.nl/~ypma

Abstract. We formulate extended Kalman smoothing in an expectation-propagation (EP) framework. The approximation involved (a local linearization) can be looked upon as a 'collapse' of a non-gaussian belief state onto a Gaussian form. This formulation allows us to come up with better approximations to the belief states, since we can iterate the algorithm until no further refinement of the beliefs is obtained. Compared to the standard extended Kalman smoother, we linearize around the mode of the actual two-slice belief state instead of the predicted mean of the one-slice belief. In initial experiments with a one-dimensional nonlinear dynamical system we found that our method improves over the extended Kalman filter and performs comparable to the unscented Kalman filter, whereas only second-order approximations are being made. The EP-formulation in principle allows for incorporation of higher-order approximations, possibly leading to further improvements.

INTRODUCTION

Many real-world systems are nonlinear, dynamical and stochastic in nature. Inference and learning of nonlinear system models with hidden dynamics is a difficult task, which requires approximations and simplifications to be made. A method that allows for accurate inference about the hidden states may be used to implement the E-step in an E-M algorithm for parameter learning with hidden variables. In this paper we consider the dynamic systems where we have nonlinearities in the state- and observation equations,

\begin{align*}
    x_t &= f(x_{t-1}) + v_t, \quad v_t \sim \mathcal{N}(0, \Gamma) \\
    y_t &= g(x_t) + w_t, \quad w_t \sim \mathcal{N}(0, \Sigma)
\end{align*}

with conditional distributions
\[ p(x_t|x_{t-1}) \sim \mathcal{N}(f(x_{t-1}), \Gamma) \]
\[ p(y_t|x_t) \sim \mathcal{N}(g(x_t), \Sigma) \]  
(2)

where \( f(\cdot) \) and \( g(\cdot) \) are (known) nonlinear functions, see figure 1, and \( \mathcal{N}(\mu, \Sigma) \) denotes the normal distribution with mean \( \mu \) and covariance matrix \( \Sigma \). In the familiar Kalman filter and smoother (see next section), all functions are assumed linear and so-called forward and backward messages (which serve as intermediate steps for computing the belief state at each time) can be computed exactly. In the nonlinear model, the forward and backward messages cannot be computed exactly any more (the integrals in (6) cannot be done analytically), one has to resort to approximations. Two popular methods are the extended Kalman filter (and smoother), which linearizes the nonlinearity so that Gaussian messages can be computed and the unscented Kalman filter, which again assumes Gaussian posterior beliefs and uses a set of carefully chosen points, propagates them through the nonlinearity and computes the moments from them.

We will briefly describe the basic idea behind these two approximative methods and will then show how linearization can be incorporated in the expectation-propagation framework. Experimental (preliminary) results and an outlook to future improvements conclude the paper.

**EXISTING METHODS FOR APPROXIMATE INFERENCE**

**Inference in linear dynamic systems**

A linear dynamical system is often modelled in state-space form as:

\[ x_t = Ax_{t-1} + v_t, \quad v_t \sim \mathcal{N}(0, \Gamma) \]
\[ y_t = Bx_t + w_t, \quad w_t \sim \mathcal{N}(0, \Sigma) \]  
(3)

Figure 1: Nonlinear dynamical system. All nodes are continuous-valued, and \( f \) and \( g \) are arbitrary nonlinear functions. Shaded nodes are observed. \( \pi \) denotes the prior distribution on \( X \). Time progresses from left to right.
implying that for states \( x \) a Markov property holds. The independence structure is such that the joint distribution over states and observations is given by

\[
p(x_1, \ldots, x_T, y_1, \ldots, y_T) = p(x_1)p(y_1|x_1)\prod_{t=2}^{T}p(x_t|x_{t-1})p(y_t|x_t)
\]  

(4)

One typically wants to infer the values of (some of the) hidden states \( x_1, \ldots, x_T \) when a series of observables \( y_1, \ldots, y_T \) has been given. One can write the one- and two-slice marginal distributions over the states as [5]:

\[
q_t(x_t) := p(x_t|y_1, \ldots, y_T) = \alpha_t(x_t)\beta_t(x_t)
\]  

(5)

\[
q_{t-1,t}(x_{t-1}, x_t) := p(x_{t-1}, x_t|y_1, \ldots, y_T) = \frac{1}{c_t} \alpha_{t-1}(x_{t-1})p(x_t|x_{t-1})p(y_t|x_t)\beta_t(x_t)
\]

if one defines:

\[
c_t := p(y_1, \ldots
\]

\[
\alpha_t(x_t) := \frac{1}{c_t}p(y_t|x_t)\int p(x_t|x_{t-1})\alpha_{t-1}(x_{t-1})dx_{t-1}
\]

\[
\beta_{t-1}(x_{t-1}) := \frac{1}{c_t}\int p(x_t|x_{t-1})p(y_t|x_t)\beta_t(x_t)dx_t
\]

(6)

By interpreting \( \alpha_{t-1}(x_{t-1}) \) and \( \beta_t(x_t) \) as incoming messages to the factor over nodes \( x_{t-1}, x_t \) and defining this factor to be the two-slice potential \( \Psi_{t-1,t} := p(x_t|x_{t-1})p(y_t|x_t) \), one recognizes the sum-product algorithm for inference in graphical models [3]. The algorithm for filtering the beliefs and estimating the most likely parameters in equation (3) is the well-known Kalman filter.

**Extended Kalman filtering and smoothing**

In the extended Kalman filter, one linearizes the nonlinearities \( f \) and \( g \) around the predicted mean state at time \( t-1 \), \( m_{t|t-1} \),

\[
\dot{A}_t = \left. \frac{\partial f(x_t)}{\partial x_t} \right|_{(x_t = m_{t|t-1})}
\]

\[
\dot{B}_t = \left. \frac{\partial g(x_t)}{\partial x_t} \right|_{(x_t = m_{t|t-1})}
\]

(7)

If one assumes zero-mean dynamic and observation noise, the extended Kalman filter update equations are found by replacing in the Kalman filter update equations the linear dynamics \( A \) and observer \( B \) by the above (time-dependent) linearizations. The predicted mean at time \( t-1 \) is now obtained by propagating the updated mean at time \( t-1 \) through the nonlinearity. For computing the predicted variance at time \( t-1 \) one uses the linearization of \( f \).
and $g$ around the predicted mean at time $t - 1$. Finally, the updated mean and variance at time $t$ are computed from the predicted values, the Kalman gain, the linearization of $g$ and evaluation of $f$ at the predicted mean. The smoothing can be done by adapting the Kalman filter smoothing equations accordingly. However, note that in this formulation the linearization in the smoothing phase does not incorporate 'future' information, i.e. information from the observables $y_{t+1}, \ldots, y_T$.

**Unscented Kalman filtering and smoothing**

The *Unscented transform* ([2]) (UT) is a method for approximating the moments of a variable $Y$ that is depending on a Gaussian variable $X$ via a nonlinear transform $f$ (see figure 2.3). More specifically, the first moment of the distribution of $Y$ is $E(Y) = \int \int dX dY p(X, Y) Y = \int dX \mathcal{N}(X; \mu, V) \mathcal{N}(Y; f(X), \Sigma) Y = \int dX \mathcal{N}(X; \mu, V) f(X)$ \( (8) \)

The latter integral is approximated numerically as $\sum_i w_i Y_i$, where $w_i$ are suitably chosen weights (in the UT, $\sum_i w_i = 1$) and $Y_i = f(X_i)$, i.e. nonlinearly transformed “sigma points” $X_i$ which are deterministically chosen samples from the Gaussian over $X$. The Unscented transform leads to improved estimates of the moments of $p(Y)$ compared to a linearization approach (cf. extended Kalman filtering), [2, 4]. The Unscented transform is applied in the Kalman filter context to approximate the beliefs on $x_t$ given the belief on $x_{t-1}$ and nonlinear dynamics $f(\cdot)$. A Kalman smoother based on the UT has been proposed in [9], where the dynamics is inverted by training an MLP to model the inverse dynamics.

**ITERATED KALMAN SMOOTHING WITH EP**

**Expectation-Propagation**

One can express inference in a graphical model as a sequence of multiplications and a summarisation (an integral) of local factors and messages (which
is equivalent to belief propagation in a graphical model) [3], see also figure 3. Here, factor refers to the conditional probabilities relevant to a local node and message refers to evidence that is being propagated to a local node from another part of the network. If there are no loops in the graph and if the summarization can be done exactly, this procedure results in the exact posterior beliefs.

In the dynamic Bayesian network considered here (figure 1), no loops occur, but the inability to do an exact summarization and to represent the non-Gaussian one-slice belief prohibits exact inference. However, the exact summarization can be replaced in the former procedure by a ‘collapse’ step, where one-slice beliefs is approximated by Gaussians [1]. The exact posterior belief is then represented by a Gaussian that matches the first two moments of the exact posterior. This leads to an approximate inference method based on local message passing called expectation-propagation [6]. A nice property of this scheme is that both beliefs and terms will stay within the chosen family of distributions (e.g. normal/Gaussian densities). In the context of this paper, the factors are the terms, which are iteratively refined from the refined posterior approximation. No convergence guarantees can be given for expectation-propagation, but when it converges it ends up in a minimum of the so-called Bethe free energy (that takes into account two-point correlations between neighbouring nodes in the network). Algorithms that aim at minimizing the Bethe free energy usually result in better approximations than e.g. mean field methods (which ignore correlations between neighbouring nodes).

Figure 3: Factor graph representation of the dynamic Bayesian network of figure 1. Evidence is incorporated into the factor nodes, which span two consecutive hidden nodes. Messages are sent between hidden and factor nodes. A hidden node’s outgoing message in a certain direction equals the incoming message in this direction. The product of incoming messages to a hidden node equals its belief. In the EP-framework, we express a two-slice belief as a scaled product of a 2-slice potential and ‘incoming messages’,

$$q_t(x_{t-1}, x_t) = \alpha_{t-1}(X_{t-1}) \beta_t(x_t)$$

Figure 3: Factor graph representation of the dynamic Bayesian network of figure 1. Evidence is incorporated into the factor nodes, which span two consecutive hidden nodes. Messages are sent between hidden and factor nodes. A hidden node’s outgoing message in a certain direction equals the incoming message in this direction. The product of incoming messages to a hidden node equals its belief.

In the EP-framework, we express a two-slice belief as a scaled product of a 2-slice potential and ‘incoming messages’,

$$q_t(x_{t-1}, x_t) = \frac{1}{c_t} \alpha_{t-1}(x_{t-1}) \Psi_{t-1,t}(x_{t-1}, x_t) \beta_t(x_t)$$

where the 2-slice potential is $\Psi_{t-1,t} = p(x_t|x_{t-1})p(y_t|x_t)$. The one-slice belief
\( q_t(x_t) \) is obtained by a marginalize-collapse:

\[
\tilde{q}_t(x_t) = \text{collapse}_{x_{t-1}} q_t(x_{t-1}, x_t)
\]

(10)

where \( \text{collapse}_{x_{t-1}} \) involves projection to a Gaussian and marginalization over \( x_{t-1} \). A similar expression can be given for obtaining \( \tilde{q}_t(x_{t-1}) \) from the potential.

Because of the nonlinearities \( f \) and \( g \), the potential \( \Psi_{t-1,t} \) is nongaussian, and the exact one-slice beliefs would also be nongaussian. As stated before, we assume (as in the extended and unscented Kalman filter and smoother) that each one-slice belief is Gaussian. Its moments cannot be found analytically, again because of the nonlinearities (unlike e.g. in EP for switching linear dynamical systems, where a deterministic moment match exists [1]).

We propose an \textit{approximate} moment match by linearizing the nonlinear argument of the nongaussian two-slice potential using a Taylor approximation. In this sense, our collapse step resembles the extended Kalman filter and smoother, which also linearize a nonlinear function.

\textbf{Algorithm for iterated extended Kalman smoothing}

The moments of either \( \tilde{q}_t(x_t) \) (forward pass) or \( \tilde{q}_t(x_{t-1}) \) (backward pass) are given by

\[
\text{mom}(i, \tau) = \frac{1}{c} \int \int \alpha_{t-1}(x_{t-1}) \cdot \Psi_{t-1,t} \cdot \beta_t(x_t) \cdot G^t(x_t) dx_{t-1} dx_t
\]

(11)

where \( \text{mom}(i, \tau) \) is the \( i \)th moment of \( \tilde{q}_t(x_t) \), \( i = 1, 2, \tau = t - 1, t \) and \( G^t(x_t) \) is \( x_t \) for \( i = 1 \) and \((x_t - m_t)^2\) for \( i = 2 \). The term \( c = \int \alpha_{t-1}(x_{t-1}) \cdot \Psi(t - 1, t) \cdot \beta_t(x_t) dx_{t-1} dx_t \) normalizes the potentials in the numerator. We now 'collapse' the (non-Gaussian) posterior two-slice belief \( q_t(x_{t-1}, x_t) \) to a Gaussian form by Laplace approximation. Using the convention \( x_t = [x_{t-1}^T, x_t^T]^T \), this integral can be cast into the form

\[
\frac{1}{c} \int dx_t \exp(F(x_t)) G(x_t)
\]

(12)

with the definition of \( G(x_t) \) analogous to (11) and

\[
F(x_t) = -\frac{1}{2} ((x_{t-1} - m_{t-1})^T V^{-1}_{t-1} (x_{t-1} - m_{t-1})
+ (x_t - f(x_{t-1}))^T \Gamma^{-1} (x_t - f(x_{t-1}))
+ (Y_t - g(x_t))^T \Sigma^{-1} (Y_t - g(x_t))
+ (x_t - \hat{m}_t)^T \hat{V}_t^{-1} (x_t - \hat{m}_t))
\]

(13)
We use a Laplace approximation of $F(x_t)$ and $G(X_t)$ around the extremum $x^*_t$ of $F(x_t)$ to arrive at the approximation

$$\exp[F(x_t)] \approx \exp(Q(x_t)) \sim \mathcal{N}(x_t; x^*_t - (F')^{-1}(x^*_t))$$ (14)

where $(F')^{-1}(x^*_t)$ is the inverse Hessian of $F(x^*_t)$. From this approximate two-slice belief, the approximate one-slice beliefs can be derived by marginalization:

$$\hat{q}_t(x_{t-1}) = \frac{1}{c} \int dx_t \exp(Q(x_t))dx_t$$ (15)

$$\hat{q}_t(x_t) = \frac{1}{c} \int dx_t \exp(Q(x_t))dx_{t-1}$$ (16)

Note that extended Kalman filtering resembles the forward pass of EPEKS; all beta messages are set to 1 in this pass, so in our EP-formulation, we can refine the 'terms' $\Psi_f(x_{t-1}, x_t)$ iteratively, which may improve the belief estimates obtained with the EKF. Furthermore, note that in our scheme we linearize around the mode of the two-slice posterior (involving both past and future evidence to a node) and take second-order information about the nonlinearity into account (the width of the approximating Gaussian), whereas in the extended Kalman filter one linearizes around the predicted mean (involving only past evidence to the node) and uses only first-order information (the gradient).

**EXPERIMENT**

We analyze the performance of our algorithm with the following one-dimensional nonlinear dynamical system

$$\begin{cases} x_t = x_{t-1} + \sin(x_{t-1}) \cdot x_{t-1} + u_t, & u_t \sim \mathcal{N}(0,Q) \\ y_t = x_t^2 + n_t, & n_t \sim \mathcal{N}(0,R) \end{cases}$$

This system has unstable fixed points at $-\pi, \pi$ (modulo $2\pi$) and a stable fixed point at 0 (modulo $2\pi$). The squaring nonlinearity in the observer gives rise to ambiguity in the polarity of the underlying state. Time series from the system are shown in figure 4. To study the performance of the three different algorithms (EKF, UKF and our EPEKS algorithm) quantitatively, we performed several runs with different noise levels. We measured algorithm performance with the normalized mean absolute deviation statistic:

$$\text{NMAD} = \text{mean} \frac{|x_t - \hat{x}_t|}{\text{var}(Y)}$$ (17)

where $Y = \{Y_t\}$. We repeat 25 runs with different noise realisations (for particular dynamic and observation noise levels $Q, R$ and data length $T$) and compute the performance for each method. We then analyze the distribution.
Figure 4: The three overlapping plots in the middle of the figure are the original trajectory of the hiddens, along with the UKF and EPEKS reconstruction. The trajectory that is lying below these plots is the typical trajectory when EKF performs badly in estimation (it seems to make an error in estimating the polarity of the hiddens, possibly due to the ambiguity induced by the squaring-operation in the observer). The '+'-trajectory is the observed time series. The numbers in the caption refer to the NMAD error achieved by different algorithms of the performance differences (for EKF: $NMAD_{EKF} - NMAD_{EPEKS}$, analogous for the comparison with UKF). In each trial we used 2 iterations for EPEKS. The results are shown in table 1, from which it can be seen that very often the means of the distribution of performance differences is $> 0$, i.e. the error of the competing method (EKF, UKF) is on average larger than the error of EPEKS, even for large noise levels.

Experiments with more EP-iterations in the one-dimensional system and experiments with a two-dimensional nonlinear system were less successful: in the course of the algorithm, the eigenvalues of the covariance matrix of a Gaussian one-slice belief approximation sometimes becomes negative (which is the same as stating that the covariance matrix is not positive semidefinite any more). In turn, the inferred state at these nodes becomes incorrect and the algorithm diverges. A possible solution to this problem has been sug-

Table 1: Average differences in performance between EPEKS and competing methods. The upper subtable refers to a comparison of EPEKS with EKF, the lower subtable compares EPEKS to UKF. Here, $[T_1, T_2, T_3] = [15, 40, 100]$ and $[Q_1, Q_2, Q_3] = [0.1, 0.5, 1.0]$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$Q_1, T_1$</th>
<th>$Q_2, T_1$</th>
<th>$Q_3, T_1$</th>
<th>$Q_1, T_2$</th>
<th>$Q_2, T_2$</th>
<th>$Q_3, T_2$</th>
<th>$Q_1, T_3$</th>
<th>$Q_2, T_3$</th>
<th>$Q_3, T_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>.01</td>
<td>.001</td>
<td>.012</td>
<td>.074</td>
<td>.005</td>
<td>.035</td>
<td>.071</td>
<td>.024</td>
<td>.050</td>
<td>.041</td>
</tr>
<tr>
<td>.1</td>
<td>.002</td>
<td>.017</td>
<td>.057</td>
<td>.018</td>
<td>.023</td>
<td>.033</td>
<td>.026</td>
<td>.050</td>
<td>.042</td>
</tr>
<tr>
<td>.5</td>
<td>.004</td>
<td>.011</td>
<td>.010</td>
<td>.013</td>
<td>.024</td>
<td>.037</td>
<td>.030</td>
<td>.034</td>
<td>.043</td>
</tr>
<tr>
<td>1.5</td>
<td>.007</td>
<td>.022</td>
<td>.036</td>
<td>.012</td>
<td>.017</td>
<td>.022</td>
<td>.0149</td>
<td>.037</td>
<td>.034</td>
</tr>
<tr>
<td>.01</td>
<td>.001</td>
<td>.002</td>
<td>.002</td>
<td>.000</td>
<td>.003</td>
<td>.007</td>
<td>.000</td>
<td>.004</td>
<td>.007</td>
</tr>
<tr>
<td>.1</td>
<td>.000</td>
<td>.001</td>
<td>-.006</td>
<td>.000</td>
<td>.002</td>
<td>.006</td>
<td>-.000</td>
<td>.003</td>
<td>.006</td>
</tr>
<tr>
<td>.5</td>
<td>.002</td>
<td>.001</td>
<td>.004</td>
<td>.002</td>
<td>.003</td>
<td>.005</td>
<td>.003</td>
<td>.003</td>
<td>.006</td>
</tr>
<tr>
<td>1.5</td>
<td>.004</td>
<td>.005</td>
<td>-.003</td>
<td>.002</td>
<td>.004</td>
<td>.005</td>
<td>-.001</td>
<td>.002</td>
<td>.003</td>
</tr>
</tbody>
</table>
gested in [4] where an approximating Gaussian is constrained to have positive semi-definite covariance matrix (and at the same time is being updated in the direction of an optimum of the performance criterion of the approximate inference method).

CONCLUSION

We formulated extended Kalman smoothing in an expectation-propagation (EP) framework, which resulted in improved inference of the hidden states in a one-dimensional nonlinear dynamical system compared to extended Kalman filtering and comparable performance to unscented Kalman filtering. The latter method is regarded as a significant improvement over extended Kalman filtering (since the unscented transform allows for approximating the nonlinearities to higher orders). Apparently, our iterative scheme allows for a comparable improvement while using only second-order approximations. The hope is that a future extension of our method that incorporates the unscented transform into a collapse step, allows for further improvements. Furthermore, it will be necessary to look for ways to retain positive semi-definite covariance matrices. Our final goal is the learning of parameters of a nonlinear dynamical system, where an inference algorithm is one (important) step in an E-M or variational framework [7, 8].

ACKNOWLEDGEMENTS

This work is supported by the Dutch Technology Foundation STW, project NNN.5321 “Graphical models for data mining”.

REFERENCES