ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

Novel approximations for inference and learning in nonlinear dynamical systems

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

Abstract. We formulate the problem of inference in nonlinear dynamical systems in the Expectation-Propagation framework, and propose two novel inference algorithms based on Laplace approximation and the Unscented transform. The algorithms are compared empirically and employed as an improved E-step in a conjugate gradient learning algorithm. We illustrate its use for data mining with two high-dimensional time series from marketing research.

1 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. In this paper we consider dynamical systems where we have nonlinearities in the state- and observation equations,

$$x_t = f(x_{t-1}) + v_t,\ v_t \sim \mathcal{N}(0, Q); \quad y_t = g(x_t) + w_t,\ w_t \sim \mathcal{N}(0, R) \qquad (1)$$

with conditionals $p(x_t|x_{t-1}) \sim \mathcal{N}(f(x_{t-1}), Q)$; $p(y_t|x_t) \sim \mathcal{N}(g(x_t), R)$. Here $f(\cdot)$ and $g(\cdot)$ are (known) nonlinear functions, see figure 1, and $\mathcal{N}(\mu, \Sigma)$ denotes the normal distribution with mean μ and covariance matrix Σ. In the familiar Kalman filter and smoother, 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, forward and backward messages cannot be computed exactly any more, so one has to resort to approximations. Two popular methods (e.g. see [3]) are the *extended Kalman filter* (EKF), which linearizes the nonlinearity so that Gaussian messages can be computed and the *unscented Kalman filter* (UKF), which again assumes Gaussian posterior beliefs but computes moments from a set of nonlinearly transformed points.

*Supported by Technology Foundation STW, project "Graphical models for data mining"

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

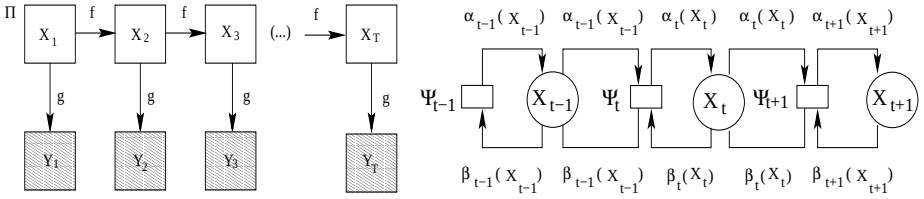

Figure 1: **left:** Nonlinear dynamical system. All nodes are continuous-valued, and f and g are arbitrary nonlinear functions. Shaded nodes are observed. π denotes the prior distribution on X. Time progresses from left to right; **right:** factor graph representation of the NLDS. Evidence is incorporated 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

2 Inference with Expectation-Propagation

Expectation-Propagation (EP, [2, 1]) is a message passing method to compute the beliefs in a graphical model, where all beliefs are approximated e.g. with Gaussians. Messages are propagated and recomputed iteratively until (possible) convergence. In the nonlinear dynamical system model (figure 1, left), we factorize $p(x_{1:T}, y_{1:T})$ as $\prod_{t=1}^{T} \Psi_t(x_{t-1}, x_t) = \prod_{t=1}^{T} \Psi_t^a(x_{t-1}, x_t) \cdot \Psi_t^b(x_t, y_t)$ and beliefs $p(x_t|y_{1:T})$ are computed by $\hat{\alpha}_t(x_t)\hat{\beta}_t(x_t)$, where the forward message $\hat{\alpha}_t(x_t)$ is the message from $\Psi_t \to x_t$ and the backward message $\hat{\beta}_{t-1}(x_{t-1})$ is the message from $\Psi_t \to x_{t-1}$ (figure 1, right). Hence we express a two-slice belief as a scaled product of a 2-slice potential and 'incoming messages', $\hat{p}_t(x_{t-1}, x_t) \propto \hat{\alpha}_{t-1}(x_{t-1})\Psi_t(x_{t-1}, x_t)\hat{\beta}_t(x_t)$ where $\Psi_t(x_{t-1}, x_t) = p(x_t|x_{t-1})p(y_t|x_t)$. Belief $q_t(x_t)$ is obtained by a marginalize-collapse, $q_t(x_t) = \text{collapse}_{x_{t-1}} p_t(x_{t-1}, x_t)$ 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 $q_t(x_{t-1})$ from the potential. The message passing algorithm then reads:

$$
\begin{aligned}
\textbf{1}.\ \hat{p}_t(x_{t-1}, x_t) &\propto \prod \text{incoming}(\Psi_t) \cdot \Psi_t(x_{t-1}, x_t) \\
&= \hat{\alpha}_{t-1}(x_{t-1}) \cdot \Psi_t(x_{t-1}, x_t) \cdot \hat{\beta}_t(x_t) \\
\textbf{2}.\ q_{t'}(x_{t'}) &= \text{collapse}\left[\int \hat{p}_t(x_{t-1}, x_t) d\backslash x_{t'}\right] \\
\textbf{3}.\ \text{message}(\Psi_t \to x_{t'}) &= \frac{q_{t'}(x_{t'})}{\text{message}(x_{t'} \to \Psi_t)}
\end{aligned}
$$

For nonlinear systems, the 'difficult' quantity is $p_t(x_{t-1}, x_t)$ because of the nonlinearities in $\Psi_t(x_{t-1}, x_t)$ and the integral. In the *collapse* step, one projects the nongaussian marginal onto a suitable Gaussian approximation.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

2.1 Laplace approximation

In the *first approach* we collapse the nongaussian marginal onto a Gaussian by applying the Laplace approximation,

$$\begin{aligned} &\int\int \hat{\alpha}_{t-1}(x_{t-1})\Psi_t(x_{t-1},x_t)\hat{\beta}_t(x_t)h(x_{t-1},x_t)dx_{t-1}dx_t \\ \equiv &\int d\mathbf{x}_t \exp\{F(\mathbf{x}_t)\} \approx \int d\mathbf{x}_t \exp\{Q(\mathbf{x}_t)\} \end{aligned} \tag{2}$$

where $\exp\{Q(\mathbf{x_t})\} \sim \mathcal{N}(\mathbf{x}_t^*, -(F^{''})^{-1}(\mathbf{x}_t^*))$ and $Q(\mathbf{x_t})$ is the quadratic approximation of $F(\mathbf{x_t})$ around its extremum $\mathbf{x}_t^*$.

2.2 Unscented approximation

The *Unscented transform* (UT, e.g. [6, 3]) is a method for approximating the moments of a variable Y that is depending on a Gaussian variable X via a nonlinear transform f. E.g., the first moment of the distribution of Y is $\langle Y\rangle =^1 \int dX\mathcal{N}(X;\mu,V)f(X)$. The latter integral is approximated numerically as $\sum_i w_i\Upsilon_i$, where w_i are suitably chosen weights (in the UT, $\sum_i w_i = 1$) and $\Upsilon_i = f(\chi_i)$, i.e. nonlinearly transformed "sigma points" χ_i which are deterministically chosen samples from the Gaussian over X. In our *second approach*, we use the unscented transform to approximate the nongaussian two-slice joint $p_t(x_{t-1},x_t)$ with a Gaussian, in three steps: 1. *prediction*: approximate $\hat{\alpha}_{t-1}(x_{t-1})\Psi_t^a(x_{t-1},x_t)$ with a Gaussian $p_t^*(x_{t-1},x_t)$ using UT; 2. *correction*: compute $p_t^*(x_t)$ by marginalization; approximate $p_t^*(x_t)\Psi_t^b(x_t,y_t)$ with a Gaussian $p_t^*(x_t,y_t)$ using UT; incorporate evidence into $p_t^*(y_t|x_t) = p_t^*(x_t,y_t)/p_t^*(x_t)$, resulting in $p_t^{**}(y_t|x_t)$; 3. *combination*: compute $q_t(x_{t-1},x_t) = p_t^*(x_{t-1},x_t)p_t^{**}(y_t|x_t)\hat{\beta}_t(x_t)$, and obtain $q_t(x_{t-1})$ and $q_t(x_t)$ by marginalization. We use UT for computing moments of the joints $p_t^*(x_{t-1},x_t), p_t^*(x_t,y_t)$, e.g. by

$$\int\int \hat{\alpha}_{t-1}(x_{t-1})\Psi_t^a(x_{t-1},x_t)h(x_{t-1},x_t)dx_{t-1}dx_t \approx \sum_i w_i\mathcal{F}_h(\chi_i) \tag{3}$$

We remark that an Unscented smoother has been proposed before [6], but that our formulation does not require the dynamics to be inverted. Furthermore, we note that one forward-backward pass is already sufficient in this algorithm, since the $\hat{\beta}_t(x_t)$ message is not used inside the *collapse* operation.

3 Learning with radial basis functions

3.1 EM updates

It was proposed in [4] to parameterize the nonlinearities in (1) with radial basis functions ρ_f^i (dynamics) and ρ_g^i (observer), and include weighted inputs u_t:

[1] $\int\int dXdYp(X,Y)Y = \int dX\mathcal{N}(X;\mu,V)\int dY\mathcal{N}(Y;f(X),\Sigma)Y$; $\langle\cdot\rangle$ denotes expectation.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

$$\left\{ \begin{array}{rl} x_{t+1} = \sum_{i=1}^{I_f} h_f^i \rho_f^i(x_t) + A_f x_t + B_f u_t + b_f + v_t \equiv \theta_f \Phi_t^f + v_t \\ y_t = \sum_{i=1}^{I_g} h_g^i \rho_g^i(x_t) + A_g x_t + B_g u_t + b_g + w_t \equiv \theta_g \Phi_t^g + w_t \end{array} \right. \quad (4)$$

where $v_t \sim \mathcal{N}(0, Q), w_t \sim \mathcal{N}(0, R)$ and $\rho_f^i(x_t)$ and $\rho_g^i(x_t)$ are Gaussians in x_t space. In this model, *EM learning* can be done by alternating an E-step (e.g. using an inference algorithm from the previous section; an extended Kalman smoother was used in [4]) with an M-step (where parameters are updated). For the above model, one computes new parameters $\hat{\theta}_f, \hat{\theta}_g$ and covariances $\hat{Q}, \hat{R}$ as

$$\begin{array}{l} \hat{\theta}_f = \sum_{t=1}^{J} \langle x_{t+1} \Phi_t^{f,T} \rangle_t \left(\sum_{t=1}^{J} \langle \Phi_t^f \Phi_t^{f,T} \rangle_t \right)^{-1}; \hat{\theta}_g = \sum_{t=1}^{T} \langle y_t \Phi_t^{g,T} \rangle_t \left(\sum_{t=1}^{T} \langle \Phi_t^g \Phi_t^{g,T} \rangle_t \right)^{-1} \\ J\hat{Q} = \sum_{t=1}^{J} \langle x_{t+1} x_{t+1}^T \rangle_t - \hat{\theta}_f \sum_{t=1}^{J} \langle \Phi_t^f x_{t+1}^T \rangle_t; T\hat{R} = \sum_{t=1}^{T} \langle y_t y_t^T \rangle_t - \hat{\theta}_g \sum_{t=1}^{T} \langle \Phi_t^g y_t^T \rangle_t \end{array} \quad (5)$$

where $J = T-1$, superscripts $^T, ^{,T}$ denote transposition and y_t are instantiated when observed. Prediction of partially known outputs can be done by estimating the hidden state at the to be predicted time stamps (where known outputs are again instantiated and unknown outputs are integrated out), and the mean state estimates are then passed through the learned output nonlinearity.

3.2 ECG updates

In [5] it was shown that the actual gradient of the likelihood may be computed if the derivative of the complete data loglikelihood can be computed. Direct maximization of the gradient of the likelihood (e.g. using conjugate gradients, leading to an *Expectation-Conjugate-Gradient or ECG algorithm*) is beneficial when relatively many unobserved quantities are present in the model. If we define $S \equiv \sum_t \langle x_{t+1} x_{t+1}^T \rangle_t - \theta_f \sum_t \langle \Phi_t^f x_{t+1}^T \rangle_t - \sum_t \langle x_{t+1} \Phi_t^{f,T} \rangle_t \theta_f^T + \theta_f \sum_t \langle \Phi_t^f \Phi_t^{f,T} \rangle_t \theta_f^T$ we compute the gradient of the loglikelihood $\mathcal{L}$ with respect to Q and θ_f for the parameterized nonlinear model (4) as

$$\begin{array}{l} \nabla_Q(\mathcal{L}) = \frac{1}{2} Q^{-1} S Q^{-1} - \frac{J}{2} Q^{-1} \\ \nabla_{\theta_f}(\mathcal{L}) = Q^{-1} \left(\sum_t \langle x_{t+1} \Phi_t^{f,T} \rangle_t - \theta_f \sum_t \langle \Phi_t^f \Phi_t^{f,T} \rangle_t \right) \end{array}$$

Analogous expressions can be derived for $\nabla_R(\mathcal{L})$ and $\nabla_{\theta_g}(\mathcal{L})$. As an aside, we enforce positive semidefinite covariance matrices during learning by updating their Choleski factors P_{kl} rather than the matrix entries Q_{ij} themselves. By the chain rule of differentiation this requires for example to postmultiply the gradient with respect to Q with a factor $\frac{\partial Q_{(ij)}}{\partial P_{(kl)}}$.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

4 Comparison on artificial data

We analyze our EP-based inference algorithms with a 1-D NLDS: [6]

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

This system has unstable fixed points at $-\pi, \pi \pmod{2\pi}$ and a stable fixed point at $0 \pmod{2\pi}$. The squaring nonlinearity in the observer gives rise to ambiguity in the polarity of the underlying state. We compared the performance of our EP-based algorithms (Laplace, denoted by EPEKS; Unscented, by EPUKS) with two benchmark algorithms (EKF, UKF) at different noise levels. We measured algorithm performance with the statistic NMAD $= \text{mean}_t |x_t - \hat{x}_t| / \text{var}(\{y_t\})$. We repeated 25 runs with different noise realisations (for varying noise levels Q, R) and fixed the data length T to 40. In each trial we used 2 iterations for our Laplace algorithm (EPEKS) and 1 iteration for our Unscented algorithm (EPUKS). In figure 2 the results are plotted for a nonlinear (left) and linear (right) observer, resp. In all cases, both EP algorithms outperform EKF and UKF (further signified by the fact that the distribution of performance *differences*[2] has mean larger than zero), except for the case $[Q, R] = [0.01, 1]$ where EPEKS suffers from the emergence of non-positive definite covariance matrices. In turn, the inferred state at these nodes becomes incorrect since the search for the function optimum in the Laplace algorithm diverges. To our knowledge, no remedies have yet been devised in the literature to deal with this (technical, yet important) problem. This effect is even more pronounced in the linear observer case: apparently, the linear observer causes the same 'high observation noise' behaviour as in the nonlinear case for already small magnitudes of R. On the other hand, when EPEKS does not suffer from this phenomenon, it performs better than all other methods. Finally, our Unscented algorithm is better than EKF and UKF in all cases, making it the more robust choice; in experiments with a two-dimensional system, this was further confirmed.

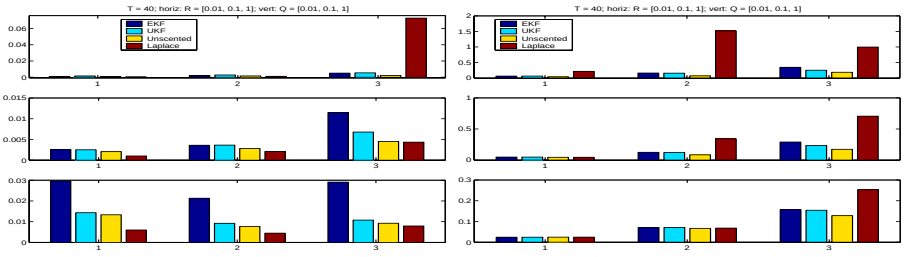

Figure 2: Median NMAD for algorithms EKF, UKF, Unscented, Laplace over 25 runs, for **(left)** nonlinear and **(right)** linear observer

[2] $\text{NMAD(COMP)} - \text{NMAD(EP)}$, where $\text{COMP} = \{\text{EKF, UKF}\}, \text{EP} = \{\text{EPEKS, EPUKS}\}$

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 361-366

5 Data mining of marketing time series

We applied the ECG algorithm[3] to the task of data mining of marketing time series[4]. Here the underlying assumption is that a marketing steering variable has both an immediate influence on the output (via the observer) and a delayed influence via the dynamics (e.g. when 'the general opinion' about a brand gradually changes as a result of PR activities). Two time series were 'compressed' into a 2-D hidden representation. First: 12 inputs (marketing mix, exogenous), 21 outputs (market shares, consumer perceptions), length 64 weeks. Second: 10 inputs (marketing mix, exogenous), 46 outputs (sales figures, consumer perceptions), length 24 weeks. The market-shares time series has periodicities in the

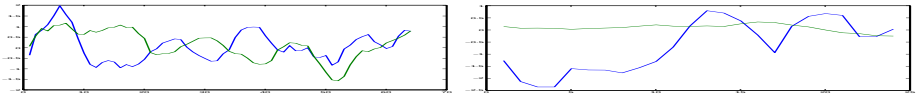

Figure 3: Compressed 2-D representation of marketing time series using ECG

order of 16 weeks (figure 3, left), indicating more global trends. The sales time series shows underlying bursts (figure 3, right) that appear to be correlated with some of the inputs, indicating stronger dependence on steering variables. In the sequel we will study ways to incorporate prior knowledge on the process in the NLDS model and evaluate the predictive power of our method.

References

[1] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In A. Darwiche and N. Friedman, editors, *Proceedings UAI-2002*, pages 216–223, 2002.

[2] T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, 2001.

[3] K. Murphy. *Dynamic Bayesian networks*, chapter in Probabilistic Graphical Models, Jordan (ed.). publisher unknown, 2002.

[4] S. Roweis and Z. Ghahramani. *An EM algorithm for identification of nonlinear dynamical systems*, chapter in Kalman Filtering and Neural Networks, Haykin (ed.). Wiley, 2001.

[5] R. Salakhutdinov, S. Roweis, and Z. Ghahramani. Optimization with EM and Expectation-Conjugate-Gradient. In *Proc. of ICML-2003*. Washington.

[6] E. Wan and R. van der Merwe. *The unscented Kalman filter*, chapter in Kalman Filtering and Neural Networks, Haykin (ed.). Wiley, 2001.

[3] EM training resulted in non-positive definite covariance matrices during updating. We initialized ECG with a linear Kalman smoother with PCA-initialized observation matrix

[4] The data was kindly provided by BrandmarC b.v., Leusden, The Netherlands