Gaussian Quadrature Based Expectation Propagation

Onno Zoeter **Tom Heskes**
Faculty of Science, University of Nijmegen
o.zoeter@science.ru.nl *tomh@cs.ru.nl*

Abstract

We present a general approximation method for Bayesian inference problems. The method is based on Expectation Propagation (EP). Projection steps in the EP iteration that cannot be done analytically are done using Gaussian quadrature. By identifying a general form in the projections, the only quadrature rules that are required are for exponential family weight functions. The corresponding cumulant and moment generating functions can then be used to automatically derive the necessary quadrature rules. In this article the approach is restricted to approximating families that factorize to a product of one-dimensional families.

The final algorithm has interesting similarities with particle filtering algorithms. We discuss these, and also discuss the relationship with variational Bayes and Laplace propagation. Experimental results are given for an interesting model from mathematical finance.

1 INTRODUCTION

Expectation Propagation (EP) [12] is a powerful deterministic approximate inference technique. It is briefly introduced in Section 3. In EP an initial approximation is iteratively refined by introducing interactions from the exact model and subsequently projecting the extended approximation back onto a chosen parametric family. These projections boil down to a matching of expected sufficient statistics. One of the problems that can stand in the way of a direct application of the EP technique, is that the required integrals implied by the computation of the expected sufficient statistics cannot be done analytically.

It is very natural to try to combine EP with an existing technique to approximate relatively low dimensional integrals. In [18] EP is combined with Laplace approximations. Here we expand upon [10] and [21] and define a general way of combining EP with Gaussian quadrature. Section 6 shows how every projection can be interpreted in a general form. Section 8 describes how, using Stieltjes procedure to construct orthogonal polynomials, the required quadrature rules can be derived automatically. This means that the entire quadrature EP routine could be computer generated for many chosen approximating families. Of course, making use of properties of specific exponential family forms can result in efficiency gains. And as it now stands, the chosen approximating family is required to factorize onto a product of one-dimensional families. However, the procedure as described in this article can form the basis for a method as general as the variational (mean-field) Bayes approach [1, 2].

To facilitate the description of the methods we first introduce our running example and briefly describe EP, the exponential family and Gaussian quadrature. Readers familiar with these basics may briefly glance at Figure 1 and jump to Section 6 right away.

2 STOCHASTIC VOLATILITY MODELS

Many of the, by now classic, results in mathematical finance assume that stocks follow a geometric Brownian motion. This model implies that equidistant log returns are independently, identically and normally distributed. Although the geometric Brownian motion gives a rough description of stock market behavior, the log returns tend to have fatter tails than a normal distribution and do not seem to be homoskedastic. This has led to the development of models where the standard deviation of the log returns, referred to as volatility in finance, is treated as a random variable itself. In this article we study the stochastic volatility

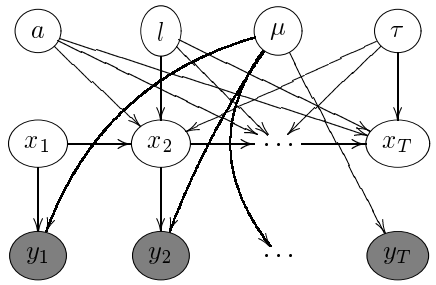

Figure 1: The dynamic Bayesian network that encodes the conditional independences in the stochastic volatility model. Shading emphasizes the fact that a particular variable is observed.

model from [7].

The model is defined in discrete time. The time-index $t = 1, 2, \ldots T$ ranges over equidistant points in time. We define $y_t = \log \frac{S_t}{S_{t-1}}$, the log return at t of stock S. As mentioned above, if the volatility would be constant, the log returns are independently, identically, and normally distributed. We keep the mean of their distributions fixed at μ, but treat the volatility as a random variable itself. The log of the volatility at t is denoted by x_t. The log volatility follows a mean reverting AR(1) process. The complete model reads

$$\begin{aligned} x_t &= a(x_t - l) + l + \epsilon_t, \quad \epsilon_t \sim N(0, \tau^{-1}) , \quad (1) \\ y_t &= \mathrm{e}^{x_t}\eta_t + \mu, \quad \eta_t \sim N(0,1) . \quad (2) \end{aligned}$$

In the above $N(m, v)$ denotes the Gaussian probability distribution with mean m and variance v. All disturbances ϵ_t and η_t are assumed to be independently drawn.

We will consider factorized subjective priors of the form

$$\begin{aligned} p(a) &= Beta(\alpha_a, \beta_a) \\ p(l) &= N(m_l, v_l) \\ p(\tau) &= Gamma(\alpha_\tau, \beta_\tau) \\ p(\mu) &= N(m_\mu, v_\mu) \\ p(x_1) &= N(m_{x_1}, v_{x_1}) . \end{aligned}$$

The full model is depicted as a dynamic Bayesian network [15] in Figure 1. The notational conventions for the various exponential families are introduced in Section 4.

The interest is in smoothed posteriors: $p(x_t|s_{1:T})$, with $t < T$, and in posteriors over the log volatility and drift parameters $p(\theta|s_{1:T})$, with $\theta = \{a, l, \tau, \mu\}$. Unfortunately, as in most Bayesian inference problems, the required integrals cannot be solved analytically.

3 EXPECTATION PROPAGATION

We assume that the joint over all variables in our model factorizes as a product of factors Ψ_t. For the stochastic volatility model this becomes:

$$\begin{aligned} p(a, l, \tau, \mu, x_{1:T}, y_{1:T}) &= \prod \Psi_t(h_t), \quad \text{with} \\ \Psi_1(h_1) &\equiv p(a)p(l)p(\tau)p(\mu)p(x_1) \times \\ &\quad p(y_1|x_1, \mu)p(y_2|x_2, \mu) \times \\ &\quad p(x_2|x_1, a, l, \tau) , \\ \Psi_t(h_t) &\equiv p(y_{t+1}|x_{t+1}, \mu) \times \\ &\quad p(x_{t+1}|x_t, a, l, \tau) , \end{aligned}$$

for $t = 1, 2, \ldots, T-1$. We denote the hidden variables in the domain of Ψ_t jointly as h_t.

The required posteriors are proportional to this joint

$$p(h|y_{1:T}) \propto \prod \Psi_t(h_t) . \quad (3)$$

To derive an expectation propagation algorithm we choose a tractable approximating exponential family $\mathcal{Q}$. For the stochastic volatility model we take a fully factorized approximation with every marginal in the same exponential family as its corresponding prior. I.e. every element $q(h) \in \mathcal{Q}$ satisfies

$$q(h) = q_a(a)q_l(l)q_\tau(\tau)q_\mu(\mu)\prod_{t=1}^{T} q_t(x_t) , \quad (4)$$

with $q_a, q_l, q_\tau, q_\mu, q_t$, a Beta, Normal, Gamma, Normal and Normal distribution respectively.

The exact posterior in (3) is approximated by a product of approximate factors $\tilde{\Psi}_t$:

$$p(h|y_{1:T}) \approx q(h) = \prod_{t=1}^{T-1} \tilde{\Psi}_t(h_t) . \quad (5)$$

The product is restricted to be in the chosen approximating family $q(h)$. Since this family is taken fully factorized the individual terms can be written without loss of generality as a product of terms over the individual variables in its domain:

$$\begin{aligned} \tilde{\Psi}_t(h_t) &\equiv m_{t\to a}(a)m_{t\to l}(l)m_{t\to\tau}(\tau) \times \\ &\quad m_{t\to\mu}(\mu)m_{t\to x_t}(x_t)m_{t\to x_{t+1}}(x_{t+1}). \end{aligned}$$

The final algorithm can be interpreted as a message passing algorithm, we will therefore refer to the $m_{t\to\cdot}$ terms as the messages going out from factor Ψ_t.

By definition the approximation of a marginal over h_i is now the product of all the messages from factors coming into h_i. E.g.

$$q(a) = \prod_{t=2}^{T-1} m_{t\to a}(a) .$$

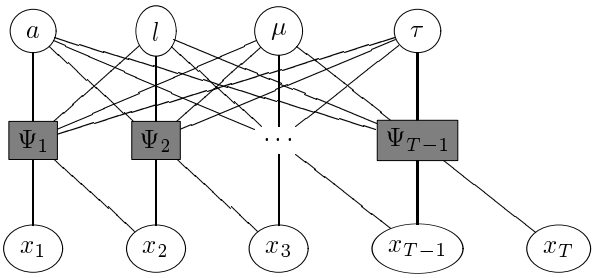

Figure 2: The factor graph corresponding to the choice of factors for the stochastic volatility model. Shaded squares represent factors.

Figure 2 gives a factor graph [8] interpretation of the chosen approximation.

To find an approximation in the family $q(h)$ that is close to the exact posterior (3) EP proceeds as follows:

1. Initialize the approximate factors $\tilde{\Psi}_t$ (and hence the outgoing messages).

2. Compute the initial approximation $q(h)$ from the product of the approximating factors:

$$q(h) = \prod_{t=1}^{T-1} \tilde{\Psi}_t(h_t) \ . \tag{6}$$

   Here we assume without loss of generality that the $\tilde{\Psi}_t$ are initialized such that the initial approximation (6) is normalized.

3. Until all $\tilde{\Psi}_t$ converge:
   (a) Choose a $\tilde{\Psi}_t$ to refine.
   (b) Remove $\tilde{\Psi}_t(h_t)$ from the approximation $q(h)$ by division:

$$q^{\setminus t}(h_t) = \frac{q(h_t)}{\tilde{\Psi}_t(h_t)} \ .$$

   (c) Combine $q^{\setminus t}(h_t)$ with the exact factor $\Psi_t(h_t)$:

$$\tilde{p}(h_t) = \frac{\Psi_t(h_t) q^{\setminus t}(h_t)}{Z_t} \ .$$

   The normalizing constant is defined as $Z_t \equiv \int \Psi_t(h_t) q^{\setminus t}(h_t) dh_t$.
   (d) Since $\tilde{p}(h_t)$ is not in chosen family $\mathcal{Q}$, project:

$$q^{\text{new}}(h_t) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \, \mathrm{KL}\left(\tilde{p}(h_t) || q(h_t)\right) \ . \tag{7}$$

   (e) Infer the new approximating factor (and hence messages) by division:

$$\tilde{\Psi}_t^{\text{new}}(h_t) = \frac{q^{\text{new}}(h_t)}{q^{\setminus t}(h_t)} \ .$$

4. Use the normalizing constant of $q(h)$ as an approximation of $p(y_{1:T})$:

$$p(y_{1:T}) \approx \prod_{t=1}^{T-1} Z_t \ .$$

This algorithm is closely related to loopy belief propagation [13]. Just as for loopy belief propagation convergence is not guaranteed.

4 EXPONENTIAL FAMILY MODELS

The approximating family $\mathcal{Q}$ is restricted to be in the exponential family. The exponential family has some pleasant properties: an exponential family is closed under product and division, and the minimum of (7) is determined by a finite number of statistics from $\tilde{p}(h)$. In this section we introduce our notation for exponential family models and give the basic results that are required in the rest of the text.

Exponential family models can be represented as

$$p(y|\theta) = \mathrm{e}^{\theta^\top u(y) - \phi(\theta)} \ . \tag{8}$$

We say that a class $\mathcal{F}$ of models is an exponential family if all its members can be written in the form (8). We refer to θ as the vector of natural parameters or canonical parameters, to $u(y)$ as the vector of sufficient statistics, and to $\phi(\theta)$ as the log partition function. We assume that the family is represented minimally, i.e. that no elements of $u(y)$ are linear combinations of others. From the exponential form in (8) we immediately see that the family is closed under product and division.

The expected values of the sufficient statistics $\langle u(y) \rangle_{p(y|\theta)}$, will be important in our further description of exponential family models. We will refer to them as *natural* moments to contrast them with $\langle y^i \rangle_{p(y|\theta)}$, the i-th moment of $p(y)$. The so-called link-function $g(\cdot)$ maps canonical parameters to natural moments

$$g(\theta) \equiv \langle u(y) \rangle_{p(y|\theta)} = \int u(y) p(y|\theta) dy \ .$$

The link function can also be derived as the first derivative of $\phi(\theta)$ [9]

$$\frac{\partial \phi(\theta)}{\partial \theta} = g(\theta) \ .$$

The KL minimization in the EP projection step (7) boils down to a matching of the natural moments [9]. I.e. $q^{\text{new}}(h_t) = \mathrm{e}^{\gamma^\top u(h_t)}$, with $\gamma = g^{-1}(\langle u(h_t) \rangle_{\tilde{p}(h_t)})$, is the distribution in $\mathcal{Q}$ that matches the natural moments of the distribution $\tilde{p}(h_t)$.

5 GAUSSIAN QUADRATURE

Gaussian quadrature is a general technique to approximate integrals of the form $\int_a^b f(x)K(x)dx$, where $K(x)$ is a known non-negative function. In the inference algorithms $K(x)$ will be a (normalized) exponential family distribution, not necessarily Gaussian (the method is due to Gauss, which explains the name of the quadrature procedure).

Based on $K(x)$, n points $\mathcal{X}_1, \ldots, \mathcal{X}_n$ and n corresponding weights $w_1, \ldots, w_n$ are chosen such that

$$\int_a^b K(x)f(x)dx \approx \sum_{i=1}^{n} f(\mathcal{X}_i)w_i \ ,$$

is exact if $f(x)$ is a polynomial of degree at most $2n-1$. General procedures to determine $\mathcal{X}_i$ and w_i are based on sets of n polynomials which are orthogonal w.r.t. $K(x)$ and the interval $[a, b]$. See e.g. [17] for an introduction to Gaussian quadrature.

To approximate multi-dimensional integrals over factoring weight functions, grids can be used. The location of the grid points can be determined from the position of the points determined for the individual marginal weight functions. The weights are simply multiplied:

$$\begin{aligned}\int_a^b \int_c^d f(x,y)K_x(x)K_y(y)dxdy \\ \approx \int_a^b \sum_i w_i f(\mathcal{X}_i, y)K_y(y)dy \\ \approx \sum_j \sum_i w_j w_i f(\mathcal{X}_i, \mathcal{Y}_i) \ .\end{aligned}$$

Throughout this paper we will assume that elements in $\mathcal{Q}$ factorize as products of univariate marginals. More advanced rules derived directly from the exactness of integrals over multinomials form an interesting extension.

6 QUADRATURE EP

If we combine the EP steps (3.b) to (3.d), we can identify a general way of using Gaussian quadrature as a numerical projection method. Combining the steps, an update from q to $\tilde{p}$ is defined as:

$$\tilde{p}(h_t) \equiv \frac{\Psi_t(h_t)}{Z_t \tilde{\Psi}_t(h_t)} q(h_t) \ , \tag{9}$$

with $Z_t \equiv \int \frac{\Psi_t(h_t)}{\tilde{\Psi}_t(h_t)} q(h_t)dh_t$. If we identify $q(h_t)$ as the weight function, we can approximate the normalization constant using Gaussian quadrature. Since the weight function is by construction a normalized exponential family distribution, we can make use of this fact in deriving the quadrature rules.

Integrals involving $\tilde{p}(h)$ can now be approximated by reweighted function evaluations. To project $\tilde{p}$ back onto the chosen family, we first approximate the natural moments $\langle u(y) \rangle_{\tilde{p}}$ using the reweighted points. Then, the inverse of the link function is used to find the parameters of q^{new} given the approximate moments.

Summarizing, quadrature EP updates an approximation of factor Ψ_t as follows:

1. Compute weights w_i and points $\mathcal{X}_i$ for the (factorized) old posterior $q(h_t)$.

2. Reweight every point:

$$\tilde{w}_i = \frac{w_i \frac{\Psi_t(\mathcal{X}_i)}{\tilde{\Psi}_t(\mathcal{X}_i)}}{\sum_j w_j \frac{\Psi_t(\mathcal{X}_j)}{\tilde{\Psi}_t(\mathcal{X}_j)}} \ . \tag{10}$$

3. Approximate the natural moments using the reweighted points

$$\langle u(h_t) \rangle_{\tilde{p}} \approx \sum_i \tilde{w}_i u(\mathcal{X}_i) \ .$$

4. The parameters for the new approximation are found by inverting the link function:

$$\begin{aligned} q^{\text{new}}(h_t) &= \mathrm{e}^{\nu_t^\top u(h_t)}, \quad \text{with,} \\ \nu_t &= g^{-1}(\sum_i \tilde{w}_i u(\mathcal{X}_i)) \ . \end{aligned}$$

5. Infer new messages by division

$$\tilde{\Psi}_t^{\text{new}}(h_t) = \frac{q^{\text{new}}(h_t)}{q^{\setminus t}(h_t)} = \frac{q^{\text{new}}(h_t)}{q(h_t)} \tilde{\Psi}_t(h_t) \ .$$

Just as for EP itself, the iterative refinement of factors is not guaranteed to converge.

There is a strong resemblance between the above algorithm and particle filtering algorithms (see e.g. [4]). Just as in the particle filtering algorithm, points are used twice: once to approximate a normalization constant and once to approximate a posterior distribution. Also, if q^{new} has its center of mass in a very different area from q, only a few points $\mathcal{X}_i$ will get non-negligible weight. As a result the approximation of q^{new} is very poor. We compare the algorithm in more detail to particle filtering and other approaches in Section 9.

7 APPROXIMATE INFERENCE IN THE SV MODEL

7.1 QUADRATURE EP FOR THE SV MODEL

The specific quadrature EP algorithm for the stochastic volatility model only depends on our choice of the approximating family $\mathcal{Q}$. We will assume that elements in $\mathcal{Q}$ factorize as a product of univariate marginals. For the stochastic volatility model, $\mathcal{Q}$ consists of all elements of the form (4).

To complete the general description of Section 6, we need to define how the points and weights are chosen for $q(h)$, and how new parameters are found given approximate moments.

For the Gaussian components $q_l(l), q_\mu(\mu), q_t(x_t)$, the points and weights can be determined using Gauss-Hermite polynomials. See e.g. [17] for a description.

Since

$$\int_{-\infty}^{\infty} f(x)N(x|m,v)dx = \int_{-\infty}^{\infty} f\left(y\sqrt{v}+m\right)N(y|0,1)dy\ , \tag{11}$$

we can determine points once for $N(x|0,1)$, and transform these when we encounter an integral for $N(x|m,v)$ with $m \neq 0$ or $v \neq 1$.

In exponential form we write the normal distribution as

$$\begin{aligned} N(x|m,v) &= \frac{1}{\sqrt{2\pi v}}\mathrm{e}^{\frac{-(x-m)^2}{2v}} \\ &= \mathrm{e}^{\theta_N^\top u_N(x)-\phi_N(\theta_N)}, \text{with} \\ \theta_N &\equiv \begin{bmatrix} h \equiv \frac{m}{v} \\ L \equiv -\frac{1}{2v} \end{bmatrix} \\ u_N(x) &= \begin{bmatrix} x \\ x^2 \end{bmatrix} \\ \phi_N(\theta_N) &= \frac{1}{2}\log\left(-\frac{\pi}{L}\right) - \frac{h^2}{4L}\ . \end{aligned}$$

The link function is

$$g_N(\theta_N) = \frac{\partial\phi_N(\theta_N)}{\partial\theta_N} = \begin{bmatrix} -\frac{h}{2L} & -\frac{1}{2L} - \frac{h^2}{4L^4} \end{bmatrix}^\top\ .$$

which we can invert analytically

$$g_N^{-1}(\langle u_N(u)\rangle) = \begin{bmatrix} \frac{\langle x\rangle}{\langle x^2\rangle - \langle x\rangle^2} & \frac{-1}{2\left(\langle x^2\rangle - \langle x\rangle^2\right)} \end{bmatrix}^\top\ .$$

Unfortunately, there is no result analogous to (11) for the Beta and Gamma distributions. We can, however, rewrite integrals involving Beta and Gamma distributions as integrals over some well-studied weight functions.

We write the Beta distribution as

$$\begin{aligned} Beta(x|\alpha,\beta) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} \\ &= \mathrm{e}^{\theta_B^\top u_B(x)-\phi_B(\theta_B)}, \text{with} \\ \theta_B &\equiv \begin{bmatrix} \alpha-1 \\ \beta-1 \end{bmatrix} \\ u_B(x) &= \begin{bmatrix} \log x \\ \log(1-x) \end{bmatrix} \\ \phi_B(\theta_B) &= \log\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\ . \end{aligned}$$

Any integral with a Beta weight function can be transformed to a Gauss-Jacobi form as follows

$$\begin{aligned} &\int_0^1 f(x)\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}dx \\ &= c(\alpha,\beta)\int_{-1}^{1} f\left(\frac{y+1}{2}\right)(1+y)^{\alpha-1}(1-y)^{\beta-1}dy\ . \end{aligned}$$

with $c(\alpha,\beta) = \frac{1}{2^{\alpha+\beta-1}}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$. Just as for the Gauss-Hermite case, the coefficients of the polynomials are known functions of the parameters in the weight function, and very good approximations of the roots exist [17].

The link function is given by

$$g_B(\theta_B) = \frac{\partial\phi_B(\theta_B)}{\partial\theta_B} = \begin{bmatrix} \psi(\alpha)-\psi(\alpha+\beta) \\ \psi(\beta)-\psi(\alpha+\beta) \end{bmatrix}^\top\ ,$$

where $\psi(x) \equiv \frac{\partial}{\partial x}\Gamma(x)$, is the digamma function. There exists no analytical inverse of this link function (in fact, depending on definitions, the link function itself is not analytic due to the digamma function). Minimizing the squared distance $\left(g_B(\theta_B) - \langle u_B(y)\rangle_{\tilde{p}}\right)^\top\left(g_B(\theta_B) - \langle u_B(y)\rangle_{\tilde{p}}\right)$ w.r.t. θ_B gives a numerical inverse.

The Gamma distribution is given by

$$\begin{aligned} Gamma(x|\alpha,\beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\mathrm{e}^{-\beta x} \\ &= \mathrm{e}^{\theta_G^\top u_G(x)-\phi_G(\theta_G)}, \text{with} \\ \theta_G &\equiv \begin{bmatrix} -\beta \\ \alpha-1 \end{bmatrix} \\ u_G(x) &= \begin{bmatrix} x \\ \log x \end{bmatrix} \\ \phi_G(\theta_G) &= \log\frac{\Gamma(\alpha)}{\beta^\alpha}\ . \end{aligned}$$

We can rewrite Gamma integrals into a Gauss-Laguerre form by noting that

$$\begin{aligned} &\int_0^\infty f(x)\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\mathrm{e}^{-\beta x}dx \\ &= \frac{1}{\Gamma(\alpha)}\int_0^\infty f\left(\frac{y}{\beta}\right)y^{\alpha-1}\mathrm{e}^{-y}dy \end{aligned}$$

The link function is

$$g_G(\theta_G) = \frac{\partial \phi_G(\theta_G)}{\partial \theta_G} = \begin{bmatrix} \frac{\alpha}{\beta} \\ \psi(\alpha) - \log \beta \end{bmatrix}^\top .$$

Which also has no analytic inverse.

There is a dynamical aspect in the model, so it seems most logical to update the approximate factors $\tilde{\Psi}_t$ in a forward-backward fashion. The next section presents a useful initialization of the approximation $q(h)$ and of the messages. This completes the description of the quadrature EP algorithm for the stochastic volatility model. Section 7.3 gives results of experiments.

7.2 A FIRST FORWARD PASS

In principle, many initializations of $q(h)$ and $\tilde{\Psi}_t$ will do. Although, as mentioned in Section 6 we want to take care that $q(h)$ 'has mass' wherever $q^{\text{new}}(h)$ has. Otherwise q^{new}, and hence its sufficient statistics, are poorly approximated by points computed from $q(h)$.

We can initialize $q(h)$ dynamically by constructing $q(h_t)$ during the first forward pass. We initialize $q_a(a), q_l(l), q_\tau(\tau), q_\mu(\mu)$, and $q_1(x_1)$ with the priors from the model. All messages are initialized as 1. The consecutive $q_{t+1}(x_{t+1})$ are initialized by drawing points from $q(a, l, \tau, x_t)$ and propagating these through the deterministic part of the transition model (1). These propagated points are used to construct a Gaussian approximation of $q_{t+1}(x_{t+1})$. To introduce the stochastic part of (1) we simply add the corresponding Gaussian disturbance to the preliminary estimate of $q_{t+1}(x_{t+1})$. Now a regular EP update can be performed.

7.3 EXPERIMENTS

The procedure from Section 6 and the initialization scheme from Section 7.2 allow us to compute approximate posteriors of the form (4). Figure 3 presents approximate posteriors over a, l, $v \equiv \tau^{-1}$, and μ for a very small artificially generated five-slice problem. The solid curves show posteriors computed using quadrature EP. The histograms present approximations based on 100.000 Gibbs samples. Since the problem is so small, we expect the Gibbs approximation to be near exact. Despite the restrictive form of the approximating family (4), the EP approximation is reasonable. Code for the stochastic volatility model is available from `www.snn.ru.nl/~orzoeter`.

Figure 4 demonstrates the risk of an ill-matched initial estimate of $q(h)$. As mentioned in Section 6, if the initial estimate of $q(h)$ has low weight in a significant part of $p(h|y_{1:T})$ the resulting approximation is poor. The example is constructed such that the

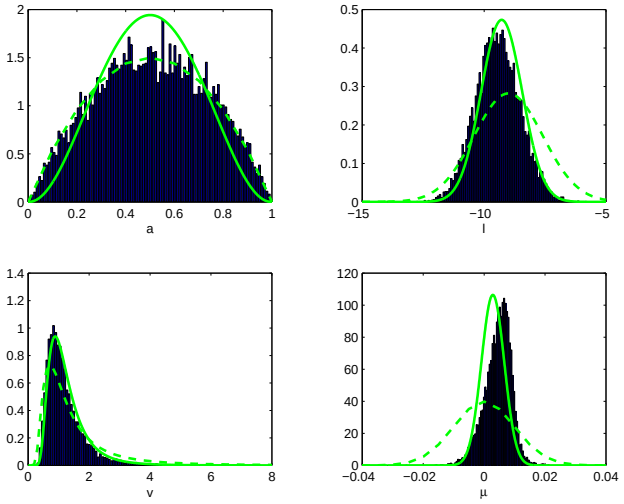

Figure 3: Approximate posteriors for a small problem with five observations. A Gibbs approximation is presented by the histograms, the solid curves show a quadrature EP approximation, the dashed curves are the subjective priors.

prior $p(v)$ (represented by a dotted line) and posterior $p(v|y_{1:T})$ (approximated by Gibbs samples in the histogram) have their mass in different areas. If $q(v)$ is initialized with the prior, the quadrature points and weights that are drawn from $p(v)$ (presented as circles) result in a poor approximation of $p(v|y_{1:T})$ (represented by a solid line). A related problem would have occurred if the prior for μ would have been very flat in the example in Figure 3. The exact posterior is very peaked close to zero. The quadrature points from a very flat distribution centered at zero would have one point at zero which effectively takes all the weight in (10). The result would be an approximation of $p(v|y_{1:T})$ by a delta-peak.

Simple experiments seem to be encouraging. However, the strong influence of the initial estimate $q(h)$ requires extra care. Either better ways of initializing $q(h)$, a different proposal distribution, or smart adaptive ways to position the quadrature points are needed to construct a reliable approximation. Note that particle filters suffer from related problems.

8 COMPUTER GENERATED RULES

Ideally we would like to have a very general class of approximation techniques where code for specific models can easily be computer generated. BUGS [19] is a very successful example for Gibbs sampling, and VIBES [20] for mean-field based approaches.

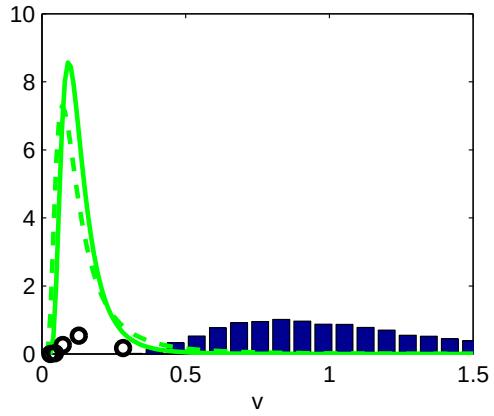

Figure 4: A quadrature EP approximation of $p(v|y_{1:T})$ based on an ill-matched initial approximation $q(h)$. See text for details.

The general form (9) of the quadrature EP algorithm ensures that the weight functions are always (products of) exponential family models. There are several techniques of recursively constructing orthogonal polynomials that only require the evaluation of moments. Stieltjes procedure [17] is among the best known, but perhaps not the most stable.

If the moment generating function

$$M(t) = \int \mathrm{e}^{tx} p(x) dx \ ,$$

of an exponential distribution is known, the computation of the moments follows from differentiating the moment generating function:

$$\left\langle x^i \right\rangle_{p(x)} = \left. \frac{d^i M(t)}{dt^i} \right|_0 \ .$$

This is mechanical and can be automated. Extensions to multi-dimensional quadrature rules is a topic for future research.

9 RELATED METHODS

In this section we give a, by no means complete, comparison to related methods.

Perhaps the method closest to the one presented here is particle filtering [4]. Instead of determining points using Gaussian quadrature, particle filtering algorithms draw points from a proposal distribution. And, instead of using reweighted points to project the posterior onto a chosen parametric form, the reweighted points (particles) are kept as a non-parametric approximation of the posterior. Quadrature based filters in the fully Gaussian case are a.o. described in [16] and [3]. Tangent to our approach are lattice particle filters [14] that stay within the particle filter class, but generate proposal points in a clever way.

In Laplace propagation [18] the KL projection in the EP algorithm is replaced by a Laplace approximation. This may form a good alternative in many settings, especially if there is a relatively large number of observations and posteriors are well approximated by Gaussians.

Note that, for the current model, any approximation method that approximates x_t and y_t jointly as a Gaussian, will result in very poor results. Since x_t and y_t are uncorrelated in (2), a Gaussian approximation will treat x_t and y_t as independent. Hence a prior for x_t will not be updated in the light of observing y_t. The unscented Kalman filter [6] will therefore, for this model, only propagate the prior, i.e. break down completely. See [21] for more details.

EP and mean-field approaches are closely related. However they are not as closely related as the factored form (4) of $\mathcal{Q}$ may lead us to assume. Both methods can be derived starting from the following variational objective

$$-\log p(y_{1:T}) = \min_{q \in \mathcal{P}} -\log p(y_{1:T}) + \mathrm{KL}\left(q(h) || p(h|y_{1:T})\right) \ . \tag{12}$$

If $\mathcal{P}$ is the set of all valid distributions on h the KL term in (12) vanishes at the minimum, and the equality is indeed an equality. Both mean-field and EP approaches arrive at an approximation by replacing $\mathcal{P}$ with a tractable set. For mean-field approaches $\mathcal{P}$ is replaced by the set $\mathcal{Q}$ with factored elements (4). Perhaps confusingly, the EP approach is not based on the same set $\mathcal{Q}$. Instead of replacing $\mathcal{P}$ by a set of simpler, but proper distributions q, the minimization is over sets of overlapping pseudo marginals with certain constistency constraints (see e.g. [11, 5] for more details). The choice of $\mathcal{Q}$ as a family on which $\tilde{p}$ is projected, determines the overlaps of these pseudo marginals. Since the approximation retains more structure of the original model, the hope is that the approximation is better than a fully factorized approximation of $\mathcal{P}$.

10 DISCUSSION

We have shown how general quadrature approximations can be identified in the standard EP scheme. The approach appears to be rather flexible and is closely related to particle filtering algorithms. The projections onto a chosen family allows iterative improvements of approximations. This is in contrast to particle filtering algorithms that can select the position of points (particles) only once and can only reweight in the light of extra information.

The running time of the quadrature EP approach is exponential in the number of variables in h_t. This is because we have approximated integrals over h_t by a grid over all variables in h_t. This complexity is identical to Kikuchi and junction tree algorithms in fully discrete networks. More advanced quadrature rules may result in a running time sub-exponential in the largest clique size. The cost of determining the grid points depends on the particular choice of $\mathcal{Q}$, the exponential family on which the posterior is projected. For Gaussians, grid points can be computed once and rescaled whenever needed (11). In the worst case, a set of orthogonal polynomials has to be constructed, of which the roots must be found numerically.

When the quadrature based method is computationally too intensive, replacing steps 1 to 3 from the algorithm in Section 6 by importance sampling may form an interesting alternative.

The Beta and Gamma components in the choice of $\mathcal{Q}$ in Section 7 imply extra computational effort. After seeing many observations, the posterior over θ will tend to be Gaussian. It is interesting to establish for what observation sizes the extra effort is worthwhile.

The procedure to generate orthogonal polynomials and quadrature rules described in this article is among the best studied in the literature. But it is unlikely that it is the optimal one for the EP framework. We would at least require rules for multi-dimensional weight functions, taking the posterior of parameters factorized is probably relatively coarse. Also, traditional Gaussian quadrature is designed to achieve zero error for a class of polynomials. For the current application it may be interesting to require good performance for different classes of functions.

Acknowledgments

We would like to thank Alexander Ypma, Wim Wiegerinck, Tjeerd Dijkstra and Ali Taylan Cemgil for helpful discussions. The Gibbs approximations were produced using WinBUGS which is available from `www.mrc-bsu.cam.ac.uk/bugs`.

References

[1] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings UAI*, 1999.

[2] M. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.

[3] E. Bolviken and G. Storvik. Deterministic and stochastic particle filters in state space models. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

[4] A. Doucet, N. D. Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer–Verlag, 2001.

[5] T. Heskes and O. Zoeter. Generalized belief propagation for approximate inference in hybrid Bayesian networks. In *Proceedings AISTATS*, 2003.

[6] S. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense*, 1997.

[7] S. Kim, N. Shephard, and S. Chib. Stochastic volatilty: Likelihood inference and comparison with ARCH models. *Review of Economic Studies*, 65, 1998.

[8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), 2001.

[9] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[10] U. Lerner. *Hybrid Bayesian Networks for Reasoning about Complex Systems*. PhD thesis, Stanford University, 2002.

[11] T. Minka. The EP energy function and minimization schemes. Technical report, 2001.

[12] T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings UAI*, 2001.

[13] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings UAI*, 1999.

[14] D. Ormoneit, C. Lemieux, and D. J. Fleet. Lattice particle filters. In *Proceedings UAI*, 2001.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufmann, 1988.

[16] A. Pole and M. West. Efficient Bayesian learning in non-linear dynamic models. *Journal of Forecasting*, 9(2), 1990.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Programming*. Cambridge University Press, 2nd edition, 1992.

[18] A. Smola, V. Vishwanathan, and E. Eskin. Laplace propagation. In *Proceedings NIPS*. 2004.

[19] D. J. Spiegelhalter, A. Thomas, and N. G. Best. Computation on Bayesian graphical models. In *Bayesian Statistics 5*, pages 407–425, 1996.

[20] J. Winn and C. Bishop. Structured variational distributions in VIBES. In *Proceedings AISTATS*, 2003.

[21] O. Zoeter, A. Ypma, and T. Heskes. Improved unscented Kalman smoothing for stock volatility estimation. In *Proceedings MLSP*, 2004.