Online Message Passing-based Inference in the Hierarchical Gaussian Filter

İsmail Şenöz
Electrical Engineering Dept.
Eindhoven University of Technology
Eindhoven, the Netherlands
i.senoz@tue.nl

Bert de Vries
Electrical Engineering Dept.
Eindhoven University of Technology & GN Hearing
Eindhoven, the Netherlands
bert.de.vries@tue.nl

***Abstract*—We address the problem of online state and parameter estimation in the Hierarchical Gaussian Filter (HGF), which is a multi-layer dynamic model with non-conjugate couplings between upper-layer hidden states and parameters of a lower layer. These non-conjugacies necessitate the approximation of marginalization and expectation integrals, while the online inference constraint renders batch learning and Monte Carlo sampling unsuitable. Here we formulate the problem as a message passing task on a factor graph and propose an online variational message passing-based state and parameter tracking algorithm, which uses Gaussian quadrature to deal with non-conjugacies. We present improved message update rules for all non-conjugate couplings, thus allowing a plug-in inference method for alternative models with equivalent non-conjugate layer couplings. The method is validated on a recorded time series of Bitcoin prices.**

***Index Terms*—dynamic modeling, variational message passing, hierarchical Gaussian filter, factor graphs, online learning**

I. Introduction

The hierarchical Gaussian filter (HGF) is a generative multi-layer random walk model for time series that is popular in the computational neuroscience literature [1]–[3]. Due to the non-conjugate coupling between layers, inference in the HGF is challenging. In [1], analytic update equations for online HGF state estimation are derived via variational Bayes with a mean-field assumption, and model parameters are offline estimated by a maximum a-posteriori (MAP) procedure. [4] builds upon this work by casting inference in the HGF as a message passing algorithm on a factor graph. In that work, results were derived by assuming a mean-field factorization and evaluating non-Gaussian messages with a Laplace approximation (Chapter 4.4 of [5]). These Laplace messages carry constant variances which lead to less accurate inference results in the higher layers of hierarchical models. In this paper, we extend and finesse the message passing procedure of [4]. Firstly, we remove the mean-field assumption in the temporal dimension, which results in messages whose functional forms contain temporal correlations that depend on the sufficient statistics of the states and parameters from upper and lower layers [6]. Secondly, instead of using a Laplace approximation for the non-conjugate messages, here we work out the application of more accurate Gaussian quadrature [7] to approximate the required marginal distributions. By changing to a structured factorization and approximating marginals rather than messages we propagate more informative messages to higher layers.

Specifically our contributions include the following:

- In Section II-D we show that online inference in the HGF model (1) can be carried out as a message passing algorithm and a structured factorization assumption results in more accurate message update rules than in [4].
- In Section II-E we isolate the challenging part of the model as a composite "Gaussian-with-Controlled-Variance" (GCV) node and derive message passing update rules for the GCV node that support joint tracking of states and parameters. All updates of the composite node are summarized in Table I.
- In Section II-F we show how the Gauss-Hermite quadrature can be used to approximate the multiplication of two non-conjugate messages by a Gaussian distribution.
- In Section III we validate the proposed inference algorithm on a real-world data set and provide free-energy performance tracks. The new HGF inference algorithm enjoys a higher accuracy compared to [4].

II. Methods

A. Model Specification

The hierarchical Gaussian filter is a Gaussian random walk model for a sequence of observations, where the variance of the random walk is itself modeled as a Gaussian random walk and so on [1]. Specifically, for an observation sequence $\mathbf{y} \triangleq \begin{bmatrix} y_1 \ y_2 \ \dots \ y_T \end{bmatrix}^\top$, an HGF model $p\left(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}\right)$ is specified as

$$\underbrace{p\left(\boldsymbol{\theta}\right) p(\mathbf{x}_0)}_{\text{prior}} \underbrace{\prod_{t=1}^{T} p\Big(y_t | x_t^{(1)}\Big)}_{\text{likelihood}} \underbrace{\prod_{i=1}^{N} p\left(x_t^{(i)} | x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)}\right)}_{\text{state transitions}} \quad (1)$$

with state transition model

$$p\left(x_t^{(i)} | x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)}\right) = \begin{cases} \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, g_t^{(i)}\right) & i < N \\ \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, \xi\right) & i = N, \end{cases} \quad (2)$$

$g_t^{(i)} = \exp\left(\kappa^{(i)} x_t^{(i+1)} + \omega^{(i)}\right)$ and ξ is a constant. Here, $x_t^{(i)}$ and $\boldsymbol{\theta}^{(i)} = \begin{bmatrix} \kappa^{(i)} \ \omega^{(i)} \end{bmatrix}^\top$ respectively denote the hidden state and parameters of layer i at time t.

978-1-7281-6432-8/20/$31.00 ©2020 IEEE

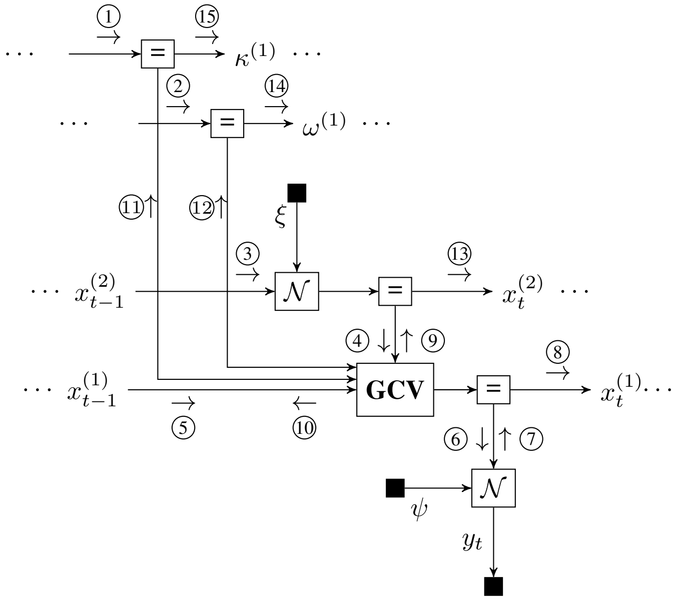


Fig. 1. A Forney-style factor graph (FFG) for one time-segment of the HGF model in Eq. 1. The arrow heads indicate the generative direction. Edges are named by the associated variable names. The triple dots indicate a graph continuation (replication) in both temporal directions. Dark small nodes indicate observations or set values for parameters. Equality nodes resolve the constraint that each variable can be connected to only two nodes. Circled numbers refer to the messages that are passed along the graph during inference. Details of the composite GCV node are provided in Table I.

The HGF model couples the state of a random walk layer to the variance parameter of the layer below through a positive non-linearity g_t. The model parameters $\kappa^{(i)}$ and $\omega^{(i)}$ determine the scale and bias of the random walks. The state vector at time t is denoted by $\mathbf{x}_t = \left[x_t^{(1)}\ x_t^{(2)}\ \ldots x_t^{(N)}\right]^\top$ and the collection of all states and parameters are written as $\mathbf{x} \triangleq \left[\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_T\right]^\top$ and $\boldsymbol{\theta} \triangleq \left[\boldsymbol{\theta}^{(1)}\ \boldsymbol{\theta}^{(2)} \ldots\ \boldsymbol{\theta}^{(N)}\right]^\top$.

In order to complete the model, the priors and likelihood should be specified. Technically any likelihood and prior can be combined with the HGF state transition model. In this paper we choose the priors $x_0^{(i)} \sim \mathcal{N}\left(m_{x_0}^{(i)}, v_{x_0}^{(i)}\right)$, $\kappa^{(i)} \sim \mathcal{N}\left(m_\kappa^{(i)}, v_\kappa^{(i)}\right)$ and $\omega^{(i)} \sim \mathcal{N}\left(m_\omega^{(i)}, v_\omega^{(i)}\right)$. For simplicity, we select a Gaussian likelihood $p\left(y_t|x_t^{(1)}\right) = \mathcal{N}\left(y_t|x_t^{(1)}, \psi\right)$ where ψ is a constant. A two-layer (Forney-style) factor graph of the HGF is given in Fig. 1.

B. Signal Processing as Inference

We are interested in joint tracking of states and parameters in model (1). This can be achieved by sequential Bayesian updating that leads to the Chapman-Kolmogorov integral [8]

$$p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{1:t}) = \frac{p(y_t|\mathbf{x}_t)}{p(y_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}) p(\mathbf{x}_{t-1}, \boldsymbol{\theta}|\mathbf{y}_{1:t-1}) \mathrm{d}\mathbf{x}_{t-1}\,, \quad (3)$$

where the denominator $p(y_t|\mathbf{y}_{1:t-1})$ is a running Bayesian evidence score, which can be evaluated as

$$\int p(y_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\theta}) p(\mathbf{x}_{t-1}, \boldsymbol{\theta}|\mathbf{y}_{1:t-1}) \mathrm{d}\boldsymbol{\theta} \mathrm{d}\mathbf{x}_{t-1} \mathrm{d}\mathbf{x}_t\,. \quad (4)$$

While (3) and (4) represent the exact solutions to joint tracking and evidence updating, due to the integration over states (and parameters) and non-conjugate prior-posterior pairing, the computation of these integrals is intractable in the HGF model.

In this paper, we propose an online hybrid message passing algorithm on a factor graph for the HGF model to perform approximate joint state and parameter tracking.

C. Forney-style Factor Graphs

A factor graph is a bipartite graph representing a factorization of a global function [9]. A *Forney-style* factor graph (FFG) is a specific type of factor graph, where nodes represent factors and edges correspond to variables [10]. In an FFG, an edge is connected to a node if and only if the (edge) variable is part of the argument list of the (node) function (see Chapter 3 of [11]). Since an edge cannot be connected to more than two nodes, auxiliary *equality* nodes are used to resolve this problem through the creation of copies of variables (see [11]–[13] for full explanation).

Fig. 1 is a time segment of an FFG corresponding to the model in (1). We call the state transition function $f_t^{(i)} \triangleq p\left(x_t^{(i)}|x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)}\right)$ a Gaussian-with-Controlled-Variance (GCV), which can be represented as a composite node in an FFG. The internal structure of GCV is given in Table I. Next, we show how variational inference can be implemented as a message passing algorithm in an FFG and then discuss message passing-based inference in the HGF.

D. Free Energy Minimization and Variational Message Passing

In the variational Bayes approach to inference, the intractable posterior $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ is approximated by a simpler distribution $q(\mathbf{x}, \boldsymbol{\theta})$ that is obtained by minimizing a divergence, usually the Kullback-Leibler (KL) divergence. Since the KL divergence is always non-negative, we can write

$$F[q] \triangleq \mathbb{E}_{q(\mathbf{x},\boldsymbol{\theta})}\left[\log \frac{q(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})}\right] \geq -\log p(\mathbf{y})\,, \quad (5)$$

where $F[q]$ is known as the (variational) *free-energy* functional, which is an upper bound to negative log-evidence.

We are interested in the recognition distribution that minimizes the free energy functional. To simplify computations we will assume a factorization across the layers $q(\mathbf{x}, \boldsymbol{\theta}) = \prod_i q\left(\mathbf{x}^{(i)}\right) q\left(\boldsymbol{\theta}^{(i)}\right)$ where $\mathbf{x}^{(i)} = \left[x_1^{(i)}\ x_2^{(i)}\ \ldots\ x_T^{(i)}\right]^\top$. However, we make no mean-field assumptions for the posterior across the temporal dimension. Following the generic recipe for structural variational message passing given in section 5 in [6], we can now compute the optimizing distributions as

$$q\left(\mathbf{x}^{(i)}\right) \propto \exp\left(\mathbb{E}_{q(\boldsymbol{\theta})q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})\right]\right) \quad (6a)$$

$$q\left(\boldsymbol{\theta}^{(i)}\right) \propto \exp\left(\mathbb{E}_{q\left(\boldsymbol{\theta}^{\setminus i}\right)q(\mathbf{x})}\left[\log p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})\right]\right) \quad (6b)$$

where $q\left(\mathbf{x}^{\setminus i}\right) \triangleq \prod_{j \neq i} q\left(\mathbf{x}^{(j)}\right)$ and similarly for $q\left(\boldsymbol{\theta}^{\setminus i}\right)$.

Next, we will show how marginalization of (6a) leads to the computation of $q\left(x_t^{(i)}\right)$ that approximates the marginal distribution for states $p\left(x_t^{(i)} | \mathbf{y}_{1:t}\right)$. Then, we will rewrite the approximate marginal as a multiplication of two variational messages (Eq. 10).

Suppose we are interested in the marginal

$$q\left(x_t^{(i)}\right) \propto \int q\left(\mathbf{x}^{(i)}\right) \mathrm{d}\mathbf{x}_{\setminus t}^{(i)} . \tag{7}$$

Substituting (6a) into (7) leads to

$$q\left(x_t^{(i)}\right) \propto \int \prod_{\tau=1}^{T} \exp\left(\mathbb{E}_{q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p\left(y_\tau | x_\tau^{(1)}\right)\right]\right) \cdot \tag{8}$$

$$\prod_{j=1}^{N} \exp\left(\mathbb{E}_{q(\boldsymbol{\theta}) q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p\left(x_\tau^{(j)} | x_{\tau-1}^{(j)}, x_\tau^{(j+1)}, \boldsymbol{\theta}^{(j)}\right)\right]\right) \mathrm{d}\mathbf{x}_{\setminus t}^{(i)}$$

where we omit the constant term (that is due to integration of $p(\boldsymbol{\theta})$ with $q(\boldsymbol{\theta})$). Note that the term $\mathbb{E}_{q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p\left(y_\tau | x_\tau^{(1)}\right)\right]$ is constant for $1 < i \leq N$.

The expectation inside the exponent in the second line of (8) produces results that depend on $x_t^{(i)}$ only when $j = i$ or $j = i - 1$. Using this property we can write the right hand side of (8) as

$$\int \prod_{\tau=1}^{T} \exp\left(\mathbb{E}_{q(\boldsymbol{\theta}) q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p\left(x_\tau^{(i)} | x_{\tau-1}^{(i)}, x_\tau^{(i+1)}, \boldsymbol{\theta}^{(i)}\right)\right]\right) \tag{9}$$

$$\exp\left(\mathbb{E}_{q(\boldsymbol{\theta}) q\left(\mathbf{x}^{\setminus i}\right)}\left[\log p\left(x_\tau^{(i-1)} | x_{\tau-1}^{(i-1)}, x_\tau^{(i)}, \boldsymbol{\theta}^{(i-1)}\right)\right]\right) \mathrm{d}\mathbf{x}_{\setminus t}^{(i)} .$$

Equation (9) can be further simplified by rearranging the order of integral. As a result we can write the marginal as a product

$$q\left(x_t^{(i)}\right) \propto \overrightarrow{\nu}\left(x_t^{(i)}\right) \overleftarrow{\nu}\left(x_t^{(i)}\right) \tag{10}$$

where we define forward and backward messages as

$$\overrightarrow{\nu}\left(x_t^{(i)}\right) \propto \nu\!\uparrow\!\left(x_t^{(i)}\right) \int \overrightarrow{\nu}\left(x_{t-1}^{(i)}\right) \tilde{f}\left(x_{t-1}^{(i)}, x_t^{(i)}\right) \mathrm{d}x_{t-1}^{(i)} \tag{11a}$$

$$\overleftarrow{\nu}\left(x_t^{(i)}\right) \propto \int \nu\!\uparrow\!\left(x_{t+1}^{(i)}\right) \overleftarrow{\nu}\left(x_{t+1}^{(i)}\right) \tilde{f}\left(x_{t+1}^{(i)}, x_t^{(i)}\right) \mathrm{d}x_{t+1}^{(i)} \tag{11b}$$

and auxiliary functions

$$\nu\!\uparrow\!\left(x_t^{(i)}\right) \propto \exp\left(\mathbb{E}_{q\left(\boldsymbol{\theta}^{(i-1)}\right) q\left(x_{t-1}^{(i-1)}, x_t^{(i-1)}\right)}\left[\log f_t^{(i)}\right]\right) \tag{12a}$$

$$\tilde{f}\left(x_{t-1}^{(i)}, x_t^{(i)}\right) = \exp\left(\mathbb{E}_{q\left(\boldsymbol{\theta}^{(i)}\right) q\left(x_t^{(i+1)}\right)}\left[\log f_t^{(i)}\right]\right) . \tag{12b}$$

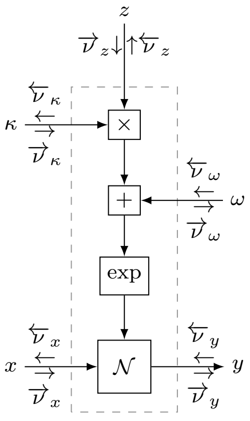
For $i = 1$, the upward message (12a) becomes the likelihood message, i.e., $\nu\!\uparrow\!\left(x_t^{(1)}\right) = p\left(y_t | x_t^{(1)}\right)$.

The results for parameter updates follow from the same line of reasoning. The only difference is that we do not assume a transition model between time steps for the parameters, which leads to a forward message that is equal to the prior, i.e., $\overrightarrow{\nu}(\kappa^{(i)}) \propto p(\kappa^{(i})$ and $\overrightarrow{\nu}(\omega^{(i)}) \propto p(\omega^{(i})$. Backward messages evaluate to

$$\overleftarrow{\nu}\left(\kappa^{(i)}\right) \propto \exp\left(\mathbb{E}_{q\left(\omega^{(i)}\right) q\left(x_t^{(i+1)}\right) q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)}\left[\log f_t^{(i)}\right]\right) \tag{13a}$$

$$\overleftarrow{\nu}\left(\omega^{(i)}\right) \propto \exp\left(\mathbb{E}_{q\left(\kappa^{(i)}\right) q\left(x_t^{(i+1)}\right) q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)}\left[\log f_t^{(i)}\right]\right) \tag{13b}$$

TABLE I
MESSAGE PASSING UPDATE RULES FOR THE GCV NODE.

GCV Node	Auxilary	
 z $\overrightarrow{\nu}_z \downarrow$ $\uparrow \overleftarrow{\nu}_z$ $\overleftarrow{\nu}_\kappa$ κ $\overrightarrow{\nu}_\kappa$ $\times$ $\overleftarrow{\nu}_\omega$ $+$ ω $\overrightarrow{\nu}_\omega$ exp $\overleftarrow{\nu}_x$ x $\overrightarrow{\nu}_x$ $\mathcal{N}$ $\overleftarrow{\nu}_y$ y $\overrightarrow{\nu}_y$ $\mathcal{N}(y \mid x, \exp(\kappa z + \omega))$	γ_1	$m_z^2 v_\kappa + m_\kappa^2 v_z + v_z v_\kappa$
	γ_2	$\exp\left(-m_\kappa m_z + 0.5\gamma_1\right)$
	γ_3	$\exp\left(-m_\omega + 0.5 v_\omega\right)$
	γ_4	$(m_1 - m_2)^2 + \Lambda_{11} + \Lambda_{22} - \Lambda_{12} - \Lambda_{21}$
	γ_5	$\gamma_4 \gamma_3 \exp\left(-m_\kappa z + 0.5 z^2 v_\kappa\right)$
	γ_6	$\gamma_4 \gamma_2 \exp\left(-\omega\right)$
	γ_7	$\gamma_4 \gamma_3 \exp\left(-m_z \kappa + 0.5 \kappa^2 v_z\right)$
	m	$\Lambda^{-1} \begin{bmatrix} \overrightarrow{m}_x / \overrightarrow{v}_x \\ \overleftarrow{m}_y / \overleftarrow{v}_y \end{bmatrix}$
	Λ	$\begin{bmatrix} 1/\overrightarrow{v}_x + \gamma_2\gamma_3 & -\gamma_2\gamma_3 \\ -\gamma_2\gamma_3 & 1/\overleftarrow{v}_y + \gamma_2\gamma_3 \end{bmatrix}$
	Messages	
	$\overleftarrow{\nu}(y)$	$\mathcal{N}\left(\overleftarrow{m}_y, \overleftarrow{v}_y\right)$
	$\overrightarrow{\nu}(y)$	$\mathcal{N}\left(\overrightarrow{m}_x, \overrightarrow{v}_x + \gamma_2\gamma_3\right)$
	$\overrightarrow{\nu}(x)$	$\mathcal{N}\left(\overrightarrow{m}_x, \overrightarrow{v}_x\right)$
	$\overleftarrow{\nu}(x)$	$\mathcal{N}\left(\overleftarrow{m}_y, \overleftarrow{v}_y + \gamma_2\gamma_3\right)$
	$\overrightarrow{\nu}(z)$	$\mathcal{N}(\overrightarrow{m}_z, \overrightarrow{v}_z)$
Marginals	$\overleftarrow{\nu}(z)$	$\exp\left(-0.5\left(m_\kappa z + \gamma_5\right)\right)$
$q(x, y) = \mathcal{N}(m, \Lambda)$	$\overrightarrow{\nu}(\kappa)$	$\mathcal{N}(\overrightarrow{m}_\kappa, \overrightarrow{v}_\kappa)$
$q(z) = \mathcal{N}(m_z, v_z)$	$\overleftarrow{\nu}(\kappa)$	$\exp\left(-0.5\left(m_z \kappa + \gamma_6\right)\right)$
$q(\kappa) = \mathcal{N}(m_\kappa, v_\kappa)$	$\overrightarrow{\nu}(\omega)$	$\mathcal{N}(\overrightarrow{m}_\omega, \overrightarrow{v}_\omega)$
$q(\omega) = \mathcal{N}(m_\omega, v_\omega)$	$\overleftarrow{\nu}(\omega)$	$\exp\left(-0.5\left(m_\omega + \gamma_7\right)\right)$
Entropy	**Average Energy**	
$0.5 \log(2\pi e)^5 \lvert\Lambda^{-1}\rvert v_z v_\kappa v_\omega$	$0.5\left(\log 2\pi + m_\kappa m_z + m_\omega + \gamma_4\gamma_3\gamma_2\right)$	

The integration scheme specified by (11a) and (11b) is called message passing because all computations can be evaluated locally in space and time (even though the posterior distributions were derived from a global objective). Note that in (12a) and (12b), the expectations require only $q\left(x_{t-1}^{(i-1)}, x_t^{(i-1)}\right)$ and $q\left(x_t^{(i)}\right)$ as opposed to the whole joint distributions. This means that the model induces a factorization.

E. Messages for the GCV Node

In order to realize message passing in the model, we need to compute the messages for states and parameters according to rules that are defined in Section II-D and implement a schedule as illustrated in Fig. 1.

Messages around Gaussian and equality nodes have already been tabulated in Chapter 4 of [12]. What remains is to derive messages around the GCV node. In Table I, we present update rules for all outgoing messages for the GVC node. For notational clarity, in these formulas we have renamed the variable names for the GVC node as indicated in the figure in Table I.[1]

F. Marginal Approximation with Gaussian Quadrature

After having derived the messages, we now address the computation of marginals in (10). Here, we approximate the

[1] Derivation of these rules can be found in http://biaslab.github.io/pdf/isit2020/i_senoz_derivations.pdf.

multiplication with a Gaussian through application of the Gauss-Hermite quadrature. For instance, (11b) requires the multiplication $\nu\!\uparrow\!\left(x_t^{(i)}\right)\overleftarrow{\nu}\left(x_t^{(i)}\right)$. From Table I we see that $\overleftarrow{\nu}\left(x_t^{(i)}\right) \propto \mathcal{N}\left(\overleftarrow{m}_t^{(i)}, \overleftarrow{v}_t^{(i)}\right)$ is a Gaussian message. However $\nu\!\uparrow\!\left(x_t^{(i)}\right)$ is neither a Gaussian nor conjugate to a Gaussian. Still, $\nu\!\uparrow\!\left(x_t^{(i)}\right)\overleftarrow{\nu}\left(x_t^{(i)}\right)$ is an integrable function, which means that it can be normalized.

We define the normalized distribution corresponding to the multiplication of the messages as

$$\tilde{q}\left(x_t^{(i)}\right) = \frac{\nu\!\uparrow\!\left(x_t^{(i)}\right)\overleftarrow{\nu}\left(x_t^{(i)}\right)}{Z_t^{(i)}}, \tag{14}$$

where

$$Z_t^{(i)} = \int \nu\!\uparrow\!\left(x_t^{(i)}\right)\overleftarrow{\nu}\left(x_t^{(i)}\right) \mathrm{d}x_t^{(i)} \tag{15}$$

is a (Gaussian integral-based) normalization constant that can be solved by a Gaussian quadrature as described in Chapter 6 of [8]. This means we approximate $Z_t^{(i)}$ as

$$Z_t^{(i)} \approx \frac{1}{\sqrt{\pi}} \sum_k \mathcal{W}^{(k)} \nu\!\uparrow\!\left(\overleftarrow{m}_{x_t}^{(i)} + \phi^{(k)}\sqrt{2\overleftarrow{v}_{x_t}^{(i)}}\right), \tag{16}$$

where $\mathcal{W}$ and ϕ are quadrature weights and abscissas respectively that can be calculated by the Golub-Welsch algorithm [7]. The order of quadrature, k, is fixed to 10 in the simulations.

We can now evaluate the moments of $x_t^{(i)}$ through quadrature, i.e.,

$$\mathbb{E}_{\tilde{q}\left(x_t^{(i)}\right)}\left[\left(x_t^{(i)}\right)^n\right] \approx \frac{1}{Z_t^{(i)}\sqrt{\pi}}.$$

$$\sum_k \mathcal{W}^{(k)}\left(\overleftarrow{m}_{x_t}^{(i)}+\phi^{(k)}\sqrt{2\overleftarrow{v}_{x_t}^{(i)}}\right)^n \nu\!\uparrow\!\left(\overleftarrow{m}_{x_t}^{(i)}+\phi^{(k)}\sqrt{2\overleftarrow{v}_{x_t}^{(i)}}\right) \tag{17}$$

Finally, we approximate the marginals $\tilde{q}\left(x_t^{(i)}\right)$ by a Gaussian distribution by computing the mean and variance parameters using (17). As a result, using a quadrature approximation, we can keep propagating Gaussian messages even after receiving a non-conjugate message from a layer below.

III. Experimental Validation

We validate the presented message passing methods by measuring the predictive performance of Bitcoin prices between 25/10/2010 and 29/11/2011. We measure the predictive performance by the free-energy. The top row in Fig. 2 shows the recorded prices. We employed a 3-layer HGF model to predict this sequence and discuss here the estimation and evaluation results.

We implemented all message passing algorithms in `ForneyLab` [14], which is an open source FFG Package for Julia [15] that is under development in our research group.[2] For comparison we include the results that are obtained by [4]. In the following, we refer to message passing algorithm described in this paper as the "Struct-Quad" algorithm and compare it to the "MF-Laplace" algorithm of [4].

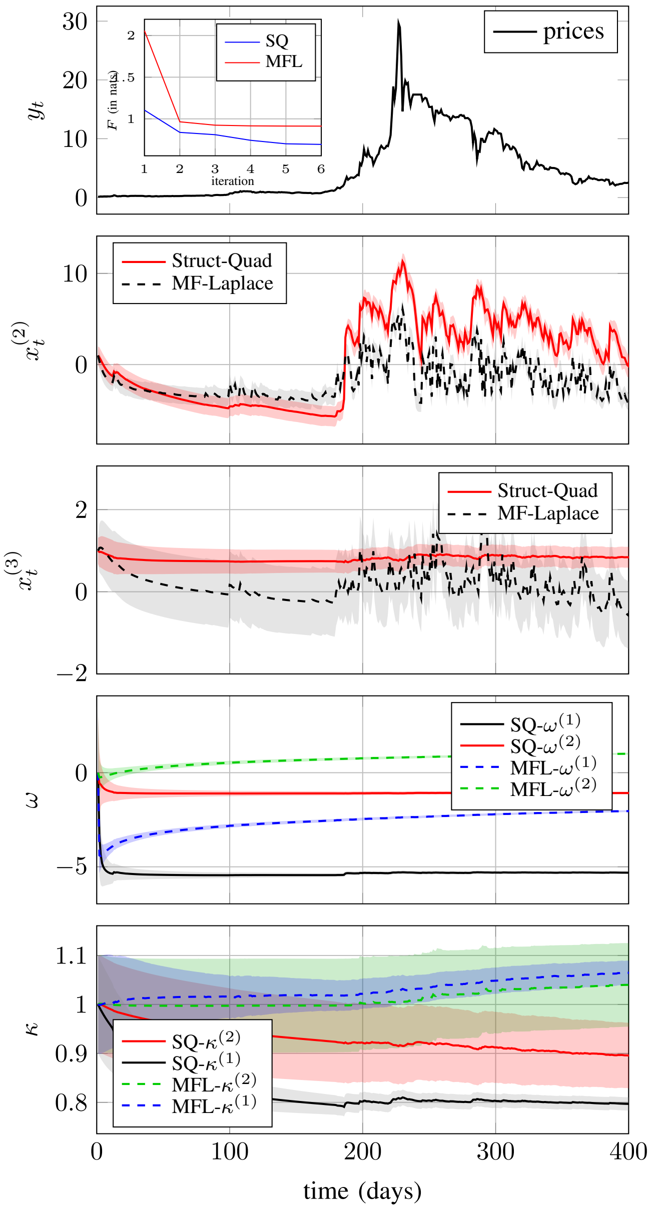


Fig. 2. Experimental validation results. Estimates by the inference method described in this paper are labeled as "Struct-Quad (SQ)" and estimates returned by [4] as "MF-Laplace (MFL)". The top row shows Bitcoin prices, i.e., the observations y_t. The inset plot in the top row shows time-averaged free-energy (in *nats*) over iterations. Both algorithms converge, but the proposed Strut-Quad message passing algorithm converges to a lower value. The second and third subplots show state estimates $x_t^{(2)}$ (the "volatility") and $x_t^{(3)}$ respectively. The solid and dashed lines represent the mean estimates for SQ and MFL respectively, and shaded regions represent one standard deviation (mean ± standard deviation). The fourth and fifth subplots depict estimates for the ω and κ parameters over time, respectively.

A. Experimental Setup: Choice of Priors

In order to perform inference in the model, the sufficient statistics of the priors should be specified. For the "tonic"

[2] The Jupyter notebook with the experiments can be found at https://github.com/biaslab/ISIT_2020_HGF.

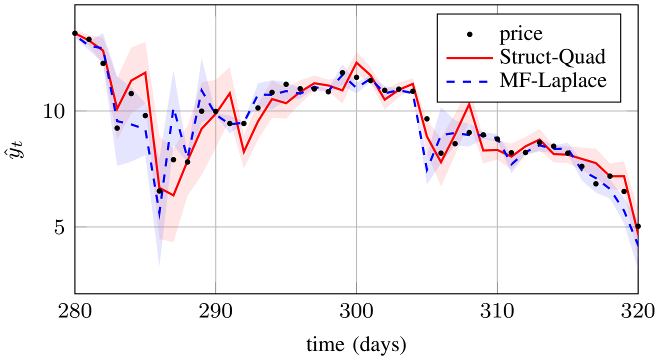


Fig. 3. Predictions for Bitcoin prices. For clarity, we only plot 40 days. The dotted line shows the actual prices and the solid and dashed lines represent the mean of Struct-Quad and MF-Laplace predictions. The width of the shaded areas indicate two standard deviations.

parameters $\omega^{(i)}$ we choose uninformative priors $\omega^{(i)} \sim \mathcal{N}(0.0, 10.0)$ and for the "scale" parameters $\kappa^{(i)}$ we choose $\kappa^{(i)} \sim \mathcal{N}(1, 0.01)$, where $i = 1, 2$. In [1], the $\kappa^{(i)}$ values are fixed to 1 and the justification behind this choice is to ensure parameter identifiability. However, we suspect that allowing κ to vary relatively slowly relative to states $x_t^{(i)}$ might have benefits in terms of adapting to changing market dynamics. Hence we set the mean of the κ priors to 1 and add a small variance. Finally, we choose $\xi \sim \Gamma(0.001, 0.001)$ as the state transition precision in the third layer and $\psi \sim \Gamma(0.0001, 0.0001)$ for the precision of the observation model. These choices are mainly motivated by the conjugacy.

B. Analysis of Results

We will examine the state estimates first, see second and third subplots in Fig. 2. At the second layer, both algorithms return estimates that share similar patterns. The tracks of $x_t^{(2)}$ capture the trends of increase and decrease in volatility. Nevertheless, the Struct-Quad algorithm tracks more smoothly compared to MF-Laplace. While there are still some salient events, the third-layer state of Struct-Quad evolve smoothly. On the other hand the third-layer state for MF-Laplace is quite active. Smoothness of the Struct-Quad estimates is due to structured assumption which keeps track of temporal correlations.

The fourth and fifth subplots show the estimation tracks for tonic and phasic parameters $\omega^{(i)}$ and $\kappa^{(i)}$ respectively. Estimates for $\omega^{(i)}$ vary at slower time scale than the states, but they do exhibit certain variation and adaptation. The κ tracks in the last subplot show a decreasing trend for Struct-Quad that leads to reduced impact of changes in superior layer states on the inferior layer parameters.

We measure the performance of the two algorithms by the free energy functional which can be interpreted as accuracy plus model complexity cost function. The inset plot in the top subplot in Fig. 2 shows the free-energy averaged over number of iterations per time step. Smaller free energy values represents a better fit. While both algorithms converge, the proposed Struct-Quad algorithm converges to a lower free energy value in comparison to the MF-Laplace algorithm of [4]. To visualize the predictive power of the algorithms we plot the prediction results in Fig 3. In the first 10 days, the prediction and confidence interval of MF-Laplace algorithm is not as accurate as the predictions of Struct-Quad. This advantage is due to the improved accuracy of the message passing updates.

IV. Conclusions

In this paper, we extended previous work [4] by introducing an improved online variational message passing algorithm for the HGF model. In order to propagate more accurate backward messages to higher levels we assume a structured factorization over time and apply Gaussian quadrature to obtain the distribution corresponding to multiplication of two non-conjugate messages. By exploiting the modularity of the FFG framework, we obtained local update equations for the posterior distributions of parameters and states. The presented method supports plug-in online parameter and state estimation in alternative models with equivalent layer couplings. We showed on a real-world Bitcoin time series that online variational tracking with structured factorization of states and slowly-varying parameters in a 3-layer HGF with quadrature resulted in convergence to a lower final free energy value in comparison to [4].

Acknowledgements

We like to thank the BIASlab team members (http://biaslab.org) and in particular Wouter M. Kouw and Thijs van de Laar for their help and insightful discussions.

References

[1] C. D. Mathys, "Hierarchical Gaussian filtering," Ph.D. dissertation, Diss., Eidgenoessische Technische Hochschule ETH Zuerich, Nr. 20909, 2012. [Online]. Available: http://e-collection.library.ethz.ch/view/eth:6419

[2] S. Iglesias, C. Mathys, K. H. Brodersen, L. Kasper, M. Piccirelli, H. E. M. den Ouden, and K. E. Stephan, "Hierarchical Prediction Errors in Midbrain and Basal Forebrain during Sensory Learning," *Neuron*, vol. 80, no. 2, pp. 519–530, Oct. 2013. [Online]. Available: https://www.cell.com/neuron/abstract/S0896-6273(13)00807-6

[3] C. D. Mathys, J. Daunizeau, K. J. Friston, and S. E. Klaas, "A Bayesian foundation for individual learning under uncertainty," *Frontiers in Human Neuroscience*, vol. 5, 2011. [Online]. Available: http://www.frontiersin.org/Journal/10.3389/fnhum.2011.00039/full

[4] Şenöz and B. de Vries, "Online Variational Message Passing in the Hierarchical Gaussian Filter," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2018, pp. 1–6.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. [Online]. Available: http://www.springer.com/computer/image+processing/book/978-0-387-31073-2

[6] J. Dauwels, "On Variational Message Passing on Factor Graphs," in *IEEE International Symposium on Information Theory*, Jun. 2007, pp. 2546–2550. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4557602

[7] G. H. Golub and J. H. Welsch, "Calculation of Gauss Quadrature Rules," *Mathematics of Computation*, vol. 23, Apr. 1969. [Online]. Available: https://www.jstor.org/stable/2004418

[8] S. Särkkä, *Bayesian Filtering and Smoothing*. London ; New York: Cambridge University Press, Oct. 2013.

[9] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.

[10] G. Forney, “Codes on graphs: normal realizations,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/910573

[11] J. Dauwels, “On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation,” Ph.D. dissertation, ETH Zurich, 2006. [Online]. Available: http://www.dauwels.com/Justin/thesis.pdf

[12] S. Korl, “A factor graph approach to signal modelling, system identification and filtering,” Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, 2005.

[13] J. Forney and P. O. Vontobel, “Partition Functions of Normal Factor Graphs,” *arXiv:1102.0316 [cs, math]*, Feb. 2011, arXiv: 1102.0316. [Online]. Available: http://arxiv.org/abs/1102.0316

[14] M. Cox, T. van de Laar, and B. de Vries, “A factor graph approach to automated design of Bayesian signal processing algorithms,” *International Journal of Approximate Reasoning*, vol. 104, pp. 185–204, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X18304298

[15] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, “Julia: A Fast Dynamic Language for Technical Computing,” *arXiv:1209.5145 [cs]*, Sep. 2012, arXiv: 1209.5145. [Online]. Available: http://arxiv.org/abs/1209.5145