Proceedings of Machine Learning Research 120:1–10, 2020 2nd Annual Conference on Learning for Dynamics and Control

Bayesian joint state and parameter tracking in autoregressive models

Ismail Senoz[1] I.SENOZ@TUE.NL
Albert Podusenko[1] A.PODUSENKO@TUE.NL
Wouter M. Kouw[1] W.M.KOUW@TUE.NL
Bert de Vries[1,2] BERT.DE.VRIES@TUE.NL
[1] *Eindhoven University of Technology, Eindhoven, the Netherlands*
[2] *GN Hearing, Eindhoven, the Netherlands*

Editors: A. Bayen, A. Jadbabaie, G.J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

We address the problem of online Bayesian state and parameter tracking in autoregressive (AR) models with time-varying process noise variance. The involved marginalization and expectation integrals cannot be analytically solved. Moreover, the online tracking constraint makes sampling and batch learning methods unsuitable for this problem. We propose a hybrid variational message passing algorithm that robustly tracks the time-varying dynamics of the latent states, AR coefficients and process noise variance. Since message passing in a factor graph is a highly modular inference approach, the proposed methods easily extend to other non-stationary dynamic modeling problems.

Keywords: Autoregressive models, hierarchical Gaussian filter, factor graphs, online learning, variational message passing

1. Introduction

Autoregressive (AR) models are of fundamental importance to problems in physics, economics and engineering. Although standard AR models have been successfully applied to various practical domains (Akaike, 1998; Hill et al., 2012), the underlying dynamics are often assumed to be stationary. Still, many applications involve modeling of signals where time-varying process statistics would lead to better performance. For example, good models for stock market prices contain time-varying variance parameters (Barber, 2012, Chapter 24). It is therefore important to be able to track slowly-varying parameters along with fast-changing latent states. Unfortunately, for many dynamic models, online Bayesian tracking leads to intractable equations.

Here, we introduce an AR model with a hierarchy of coupled Gaussian random walks (known by itself as a Hierarchical Gaussian Filter (HGF) (Şenöz and de Vries, 2018; Mathys, 2012)) to capture the slowly time-varying dynamics of process noise. The proposed (AR-HGF) model is flexible in terms of prior assumptions for the parameter dynamics. We present an online variational inference technique based on hybrid message passing and a quadrature rule. The proposed inference method allows for online tracking of states and parameters as well as tracking of variational free energy as a performance measure.

Classical alternatives for modeling this type of signal include time-varying AR and generalized AR conditional heteroskedasticity models (Bollerslev, 1986; Jiang and Kitagawa, 1993; Barber, 2012). These techniques are powerful, but lack the advantages of the Bayesian approach. Modern

© 2020 I. Senoz, A. Podusenko, W.M. Kouw & B. de Vries.

MCMC sampling-based methods generally track these types of dynamics well (Barnett et al., 1996; Andrieu et al., 2003; Gençağa et al., 2010), but are often too slow for online inference. Variational Bayes approaches are both fast and robust to overfitting. Roberts and Penny (2002) model process noise with a Gaussian mixture model, achieving time-varying variance via dynamic selection of components. In contrast, our approach is based on message passing-based inference on factor graphs, which makes inference extensible and automatable.

Our contributions include the following: first, in Sec. 2 we introduce our AR-HGF model for modeling non-stationary signals and present a corresponding Forney-style Factor Graph in Sec. 3.2. We derive message passing update rules that support online joint state and parameter tracking (Table 1). In Sec. 4 we experimentally validate the performance of an AR(2)-HGF model in an online state and parameter tracking task.

2. Model specification

We consider an M-th order autoregressive model (AR(M)) for an observed signal $\mathbf{y}_t = y_{1:t}$ with $y_t \in \mathbb{R}$, specified by

$$y_t = \sum_{m=1}^{M} \theta_m y_{t-m} + \epsilon_t \qquad \text{where } \epsilon_t \sim \mathcal{N}(0, \vartheta_t)\,. \tag{1}$$

This model is parameterized by autoregressive coefficients $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_M) \in \mathbb{R}^M$ and time-varying process noise variance $\vartheta_t \in \mathbb{R}^+$. Our goal is to infer both $\boldsymbol{\theta}$ and ϑ_t in an online fashion from an observed data stream $\mathbf{y}_{1:t}$.

We take a Bayesian viewpoint on inference and proceed to introduce some priors. For the process noise variance ϑ_t, we impose a Gaussian random walk prior that is mapped to the positive domain to serve as a variance parameter. This prior model is known as a hierarchical Gaussian filter (Mathys, 2012; Şenöz and de Vries, 2018). Here we consider only a single layer, which is specified by

$$z_t = z_{t-1} + \eta_t \qquad \text{where } \eta_t \sim \mathcal{N}(0, \gamma^{-1})\,, \tag{2a}$$

$$\vartheta_t = \exp\left(\kappa z_t + \omega\right), \tag{2b}$$

with initial state $z_0 \sim \mathcal{N}(m_{z_0}, v_{z_0})$, which we usually choose as rather uninformative (e.g., zero mean and large variance). We will refer to $z_t \in \mathbb{R}$ as a *control* state since it controls the variance of ϵ_t. The control state is perturbed by Gaussian transition noise as well, but we initially assume that its variance γ^{-1} is fixed. The hierarchical extension assumes a similar Gaussian walk prior on variance γ^{-1}. Parameters κ and ω afford an affine transformation of the control state to the positive real domain via the exponential function.

With this dynamical prior on ϑ_t, the inference task now evaluates to online tracking of control states z_t and parameters $\boldsymbol{\Psi} = (\boldsymbol{\theta}, \gamma, \kappa, \omega)$. To complete the model, we use the following priors on the parameters:

$$\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}_\theta, \mathbf{V}_\theta),\ \gamma \sim \Gamma(\alpha, \beta),\ \kappa \sim \mathcal{N}(m_\kappa, v_\kappa),\ \omega \sim \mathcal{N}(m_\omega, v_\omega)\,. \tag{3}$$

We will execute online inference by message passing on a (Forney-style) factor graph (Loeliger et al., 2007). Factor graphs naturally take advantage of the (abundance of) conditional independencies between the variables in our model. Note that we can rewrite Eq. 1 as a state-space model

$$y_t = \boldsymbol{\theta}^T \mathbf{x}_{t-1} + \epsilon_t \tag{4a}$$

$$\mathbf{x}_t = \mathbf{S}\mathbf{x}_{t-1} + \mathbf{c}y_t \tag{4b}$$

where

$$\mathbf{x}_t \triangleq (y_t, y_{t-1}, \ldots, y_{t-M+1}), \;\; \mathbf{S} \triangleq \begin{bmatrix} \mathbf{0} & \\ \mathbf{I}_{M-1} & \mathbf{0} \end{bmatrix}, \;\; \mathbf{c} \triangleq (1, 0, \ldots, 0)^T . \tag{5}$$

We refer to $\mathbf{x}_t$ as a data buffer, since it retains M previous observations. Note that $\mathbf{x}_t$ is fully observed and therefore does not need to be estimated. The factor graph can now be constructed by rewriting the full model as the following factorized probability distribution:

$$p(\mathbf{y}, \mathbf{z}, \mathbf{x}, \boldsymbol{\Psi}) = \underbrace{p(z_0, \boldsymbol{\Psi})}_{\text{priors}} \prod_{t=1}^{T} \underbrace{p(y_t|z_t, \mathbf{x}_{t-1}, \boldsymbol{\Psi})}_{\text{observation}} \underbrace{p(z_t|z_{t-1}, \boldsymbol{\Psi})}_{\text{state transition}} \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1}, y_t)}_{\text{data buffer}} \tag{6}$$

where the priors are given by Eq. 3, the observation $p(y_t|z_t, \mathbf{x}_{t-1}, \boldsymbol{\Psi}) = \mathcal{N}(y_t|\boldsymbol{\theta}^\top \mathbf{x}_{t-1}, \vartheta_t)$, the state transition $p(z_t|z_{t-1}, \boldsymbol{\Psi}) = \mathcal{N}(z_t|z_{t-1}, \gamma^{-1})$ and the data buffer $p(\mathbf{x}_t|\mathbf{x}_{t-1}, y_t) = \delta(\mathbf{x}_t - (\mathbf{S}\mathbf{x}_{t-1} + \mathbf{c}y_t))$. Inference in the factor graph will be discussed in Sec. 3.2.

3. Online inference

We are interested in joint tracking of state z_t and parameters $\boldsymbol{\Psi}$ in model Eq. 6. Online inference can be achieved by sequential Bayesian updating which leads to a Chapman-Kolmogorov integral:

$$\underbrace{p(z_t, \boldsymbol{\Psi}|\mathbf{y}_{1:t})}_{\text{posterior}} \propto \int \underbrace{p(y_t|z_t, \mathbf{x}_{t-1}, \boldsymbol{\Psi})}_{\text{observation}} \underbrace{p(z_t|z_{t-1}, \boldsymbol{\Psi})}_{\text{state transition}} \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1}, y_t)}_{\text{data buffer}} \underbrace{p(z_{t-1}, \boldsymbol{\Psi}|\mathbf{y}_{1:t-1})}_{\text{prior}} \mathrm{d}z_{t-1} . \tag{7}$$

Due to non-linearities, this integral is not tractable. We choose a variational inference approach to approximate the integral.

3.1. Variational Inference

Variational methods approximate the intractable posterior distribution by a simpler distribution. The nature of approximation requires a measure of approximation error, which in variational inference is usually the Kullback-Leibler (KL) divergence. Let $q(\mathbf{z}, \boldsymbol{\Psi}) \approx p(\mathbf{z}, \boldsymbol{\Psi}|\mathbf{y})$ for the AR model. The KL divergence is defined as

$$D_{KL}\left[q(\mathbf{z}, \boldsymbol{\Psi})||p(\mathbf{z}, \boldsymbol{\Psi}|\mathbf{y})\right] \triangleq \mathbb{E}_{q(\mathbf{z}, \boldsymbol{\Psi})}\left[\log \frac{q(\mathbf{z}, \boldsymbol{\Psi})}{p(\mathbf{z}, \boldsymbol{\Psi}|\mathbf{y})}\right] . \tag{8}$$

Since the KL divergence is always non-negative, we can write

$$F[q(\mathbf{z}, \boldsymbol{\Psi})] \triangleq \mathbb{E}_{q(\mathbf{z}, \boldsymbol{\Psi})}\left[\log \frac{q(\mathbf{z}, \boldsymbol{\Psi})}{p(\mathbf{z}, \boldsymbol{\Psi}, \mathbf{y})}\right] \geq -\log p(\mathbf{y}) , \tag{9}$$

where $F[q(\mathbf{z}, \boldsymbol{\Psi})]$ is known as the (variational) *free-energy* functional, which is an upper bound to negative log-evidence, since the Kullback-Leibler divergence is always non-negative. In order

to simplify the inference computations, we assume a factorization $q(\mathbf{z}, \mathbf{\Psi}) = q(\mathbf{z})q(\mathbf{\Psi})$, where $q(\mathbf{\Psi}) = q(\boldsymbol{\theta})q(\kappa)q(\omega)q(\gamma)$. We will not further assume that the recognition distribution factorizes over time. With these assumptions, the free-energy functional can be minimized by variational calculus, leading to the following solution for the control states:

$$q(\mathbf{z}) \propto \exp\left(\mathbb{E}_{q(\mathbf{\Psi})q(\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Psi})p(\mathbf{z}|\mathbf{\Psi})\right]\right) . \tag{10}$$

Similarly, the solutions for the parameters are given by:

$$q(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \int p(\mathbf{x}) \exp\left(\mathbb{E}_{q(\omega)q(\kappa)q(\mathbf{z})}\left[\log p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Psi})\right]\right) \mathrm{d}\mathbf{x} \tag{11a}$$

$$q(\kappa) \propto p(\kappa) \exp\left(\mathbb{E}_{q(\omega)q(\mathbf{z})q(\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Psi})\right]\right) \tag{11b}$$

$$q(\omega) \propto p(\omega) \exp\left(\mathbb{E}_{q(\kappa)q(\mathbf{z})q(\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Psi})\right]\right) \tag{11c}$$

$$q(\gamma) \propto p(\gamma) \exp\left(\mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{z}|\gamma)\right]\right) . \tag{11d}$$

Derivations for these results follow from section 5 in (Dauwels, 2007). Next, in Sec. 3.2 we show how these equations can be used to derive message passing-based inference in a factor graph.

3.2. Inference by message passing on the factor graph

A Forney-style factor graph (FFG) is a graphical representation of a factorized probability distribution (Loeliger et al., 2007). Nodes in an FFG represent factors and edges represent variables. An edge is connected to a node if and only if the (edge) variable is part of the argument list of the (node) function. Fig. 1 is an FFG corresponding to one time-segment of the model in Eq. 6. The FFG contains Gaussian nodes $\boxed{\mathcal{N}}$, a dot product node $\boxed{\cdot}$, a data buffer update node $\boxed{\text{S,c}}$, equality nodes $\boxed{=}$ and a Gaussian-with-Controlled-Variance $\boxed{\text{GCV}}$ node. The Gaussian nodes describe stochastic relations between variables, with their perturbations driven by variance parameters. The GCV node is a generalization of the Gaussian node in that it takes an unconstrained variable as its variance parameter, which is mapped to the positive domain inside the node (see Eq.2b). Inference in the current GCV node deviates from the GCV node in Şenöz and de Vries (2018) in that it uses a structured factorization as opposed to the earlier mean-field factorization. Moreover, Şenöz and de Vries (2018) relies on Laplace approximation to tackle with non-linearities while in the current version quadrature methods are used. The other nodes implement deterministic relations between the variables. The dot product and data buffer nodes execute Eq.4 (without adding process noise). The equality node consists of the factor $f(x, y, z) = \delta(x - y)\delta(x - z)$ and constrains the marginals of associated variables to be equal (Loeliger et al., 2007).

⑧	$\mathcal{N}(y_t \mid \mathbb{E}[\boldsymbol{\theta}^\top]\mathbf{x}_t, \xi_2\xi_3)$	ξ_1	$\mathbb{E}[z_t]^2\mathbb{V}[\kappa]+\mathbb{E}[\kappa]^2\mathbb{V}[z_t]+\mathbb{V}[z_t]\mathbb{V}[x_t]$
⑩	$\mathcal{N}(\boldsymbol{\theta}^\top\mathbf{x}_t \mid y_t, \xi_2\xi_3)$	ξ_2	$\exp(-\mathbb{E}[z_t]\mathbb{E}[\kappa] + \frac{1}{2}\xi_1)$
⑫	$\exp\left(-\frac{1}{2}\left(\omega + \xi_1(y_t - \mathbb{E}[\boldsymbol{\theta}^\top]\mathbf{x}_t)^2 \exp(-\omega)\right)\right)$	ξ_3	$\exp(-\mathbb{E}[\omega] + \frac{1}{2}\mathbb{V}[\omega])$
⑬	$\exp\left(-\frac{1}{2}\left(\mathbb{E}[z_t]\kappa + \xi_3(y_t - \mathbb{E}[\boldsymbol{\theta}^\top]\mathbf{x}_t)^2\xi_4\right)\right)$	ξ_4	$\exp\left(-\mathbb{E}[z_t]\kappa + \frac{1}{2}\kappa_t^2\mathbb{V}[z_t]\right)$
⑭	$\exp\left(-\frac{1}{2}\left(\mathbb{E}[\kappa]z_t + \xi_3(y_t - \mathbb{E}[\boldsymbol{\theta}^\top]\mathbf{x}_t)^2\xi_5\right)\right)$	ξ_5	$\exp\left(-\mathbb{E}[\kappa]z_t + \frac{1}{2}z_t^2\mathbb{V}[\kappa]\right)$

Table 1: Messages associated with GCV node, used in Fig. 1.

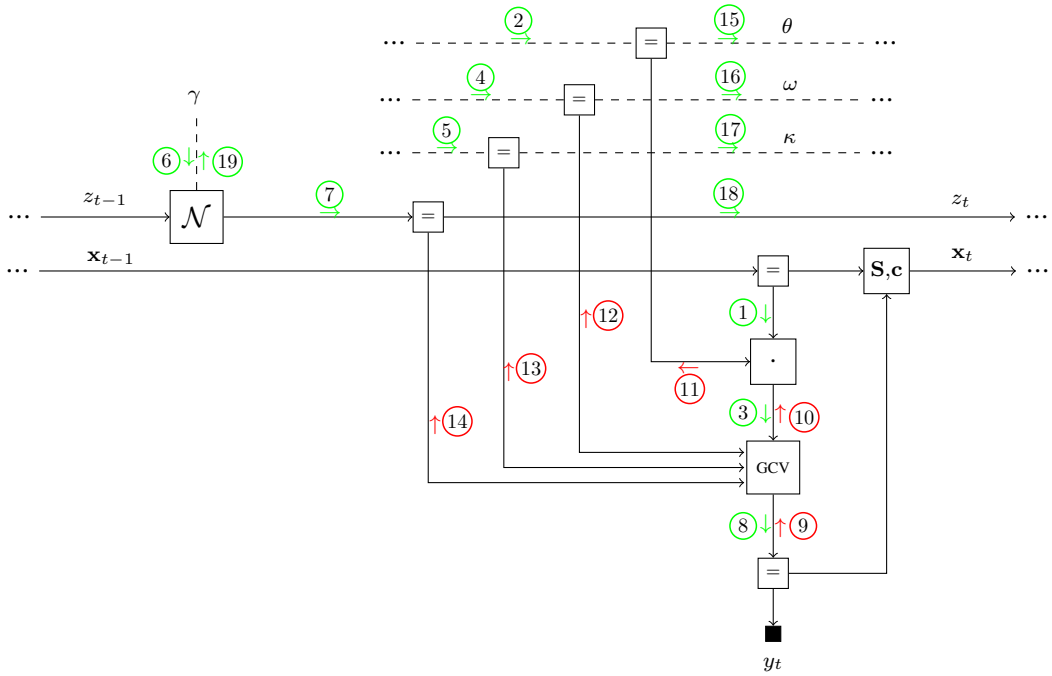


Figure 1: One time segment of a Forney-style factor graph (FFG) for AR(M)-HGF defined by Eq. 6. The $\boxed{\mathbf{S},\mathbf{c}}$ node denotes the data buffering operation (Eq. 4b), $\boxed{\text{GCV}}$ represents the observation $\mathcal{N}(y_t|, \boldsymbol{\theta}^\top \mathbf{x}_t, \vartheta_t)$ and $\boxed{\cdot}$ a dot product. The small black node corresponds to an observation y_t. Medium-sized nodes represent deterministic factors and larger nodes denote stochastic factors. Solid and dashed edges are associated with states and parameters respectively. The arrowheads are not used in any computations but visually indicate the "generative" direction (towards observations).

Variational inference in an FFG can be made efficient by recognizing that the conditional independence structure of the model is hard-coded in the graph structure. For instance, updating the joint $q(\mathbf{z})$ at an arbitrary time point t' involves:

$$q(z_{t'}) = \int q(\mathbf{z}) \mathrm{d}\mathbf{z}_{\setminus t'} \propto \int \underbrace{\exp\left(\mathbb{E}_{q(\boldsymbol{\Psi})q(\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Psi}) p(\mathbf{z}|\boldsymbol{\Psi})\right]\right)}_{\text{from eq. 10}} \mathrm{d}\mathbf{z}_{\setminus t'} \tag{12a}$$

$$\propto \int \prod_t \exp\left(\mathbb{E}_{q(\boldsymbol{\Psi})q(\mathbf{x},\mathbf{y})}\left[\log p(y_t|\mathbf{x}_t, z_t, \boldsymbol{\theta}, \kappa, \omega) p(z_t|z_{t-1}, \gamma)\right]\right) \mathrm{d}\mathbf{z}_{\setminus t'} \tag{12b}$$

$$\propto \underbrace{\int \exp\left(\mathbb{E}_{q(\gamma)}\left[\log p(z_{t'}|z_{t'-1}, \gamma)\right]\right) \overrightarrow{\nu}(z_{t'-1}) \mathrm{d}z_{t'-1}}_{\overrightarrow{\nu}(z_{t'}) \quad 7}$$

$$\times \underbrace{\exp\left(\mathbb{E}_{q(\kappa)q(\omega)q(y_{t'}, \boldsymbol{\theta}^\top \mathbf{x}_{t'})}\left[\log p(y_{t'}|\mathbf{x}_{t'}, z_{t'}, \boldsymbol{\theta}, \kappa, \omega)\right]\right)}_{\overleftarrow{\nu}(z_{t'}) \quad 14} . \tag{12c}$$

where Eq. 12c can be recognized as a multiplication of two colliding *messages* ⑦ and ⑭ that are called forward and backward message respectively.

Online inference in the model (Eq. 6) consists of computing the messages in the order indicated in Fig. 1. Update rules for messages ①, ②, ③, ④, ⑤, ⑥, ⑦, ⑪, ⑮, ⑯, ⑰, ⑲ are tabulated in (Korl, 2005, Table 4.2 of Chapter 4). The remaining computational issues are (1) the computation of outgoing messages ⑧, ⑩, ⑫, ⑬ and ⑭ for the GCV node, and (2) getting a closed-form solution to the multiplication of messages to approximate the marginals $q(z_t)$ and $q(\mathbf{\Psi})$. With respect to the first issue, Table 1 presents the derived variational message update rules.[1] Next, we discuss how to compute marginals.

3.3. Computation of marginals through Gaussian quadrature

When a colliding forward and backward message are not conjugate, the product needs to be explicitly normalized to obtain a proper marginal. For example, consider the posterior $q(z_t)$ that is represented by message ⑱, which is proportional to the multiplication of messages ⑦ and ⑭. Message ⑦ is a Gaussian and message ⑭ is neither a Gaussian nor conjugate to a Gaussian. We denote messages ⑦ and ⑭ with $\overrightarrow{\nu}(z_t)$ and $\overleftarrow{\nu}(z_t)$, respectively. The marginal of message ⑱ is

$$q(z_t) = \frac{\overrightarrow{\nu}(z_t)\overleftarrow{\nu}(z_t)}{\int \overrightarrow{\nu}(z_t)\overleftarrow{\nu}(z_t)\mathrm{d}z_t} \tag{13a}$$

$$= \frac{\exp\left(-\frac{(z_t-\overrightarrow{m}_t^{(z)})^2}{2\overrightarrow{v}_t^{(z)}}\right)\exp\left(-\frac{1}{2}\left(m_t^{(\kappa)}z+\gamma_4\gamma_3\exp\left(-m_t^{(\kappa)}z+z^2v_t^{(\kappa)}/2\right)\right)\right)}{\int\exp\left(-\frac{(z_t-\overrightarrow{m}_t^{(z)})^2}{2\overrightarrow{v}_t^{(z)}}\right)\exp\left(-\frac{1}{2}\left(m_t^{(\kappa)}z+\gamma_4\gamma_3\exp\left(-m_t^{(\kappa)}z+z^2v_t^{(\kappa)}/2\right)\right)\right)\mathrm{d}z_t}. \tag{13b}$$

Since the integral is one-dimensional, a quadrature method can be used to obtain the normalization constant $Z_t = \int \overrightarrow{\nu}(z_t)\overleftarrow{\nu}(z_t)\mathrm{d}z_t$ (Särkkä, 2013), leading to

$$Z_t = \int \overleftarrow{\nu}(z_t)\mathcal{N}(\overrightarrow{m}_t^{(z)}, \overrightarrow{v}_t^{(z)})\mathrm{d}z_t \approx \frac{1}{\sqrt{\pi}}\sum_{i=1}^{r}\mathcal{W}^{(i)}\overleftarrow{\nu}\left(\overrightarrow{m}_t^{(z)} + \xi^{(i)}\sqrt{2\overrightarrow{v}_t^{(z)}}\right), \tag{14}$$

where $\mathcal{W}$ are the quadrature weights and ξ are the sigma points (Eqs. 6.17, 6.18, Särkkä, 2013). Here, r is the order of Gauss-Hermite polynomial that is used to obtain the weights and points. Once Z_t has been obtained we can determine the moments of $q(z_t)$ by computing:

$$\mathbb{E}_{q(z_t)}[z_t^n] = \frac{1}{Z_t}\int z_t^n\overleftarrow{\nu}(z_t)\mathcal{N}(\overrightarrow{m}_t^{(z)}, \overrightarrow{v}_t^{(z)})\mathrm{d}z_t$$
$$\approx \frac{1}{Z_t\sqrt{\pi}}\sum_{i=1}^{r}\mathcal{W}^{(i)}\left(\overrightarrow{m}_t^{(z)} + \xi^{(i)}\sqrt{2\overrightarrow{v}_t^{(z)}}\right)^n\overleftarrow{\nu}\left(\overrightarrow{m}_t^{(z)} + \xi^{(i)}\sqrt{2\overrightarrow{v}_t^{(z)}}\right). \tag{15}$$

We approximate $q(z_t)$ with a Gaussian distribution by matching the first two moments. Note that the unscented Kalman filter is a special case of Eq. 15 (Ch. 5 Särkkä, 2013; Meyer et al., 2013). In the present application, we have used this quadrature-based normalization procedure to propagate non-conjugate incoming pairs through equality nodes. This procedure can applied generally to message-passing-based inference in factor graphs.

1. Derivations of update rules can be found at https://biaslab.github.io/pdf/l4dc2020/i_senoz_a_podusenko_L4DC_online_learning_AR_HGF.pdf

4. Experimental validation

To validate the proposed model and inference algorithm, we modeled a synthesized data stream of 1000 points. The data was generated by the model of Eq. 6, with AR order $M = 2$ (see Fig. 2 top left). In the data generating process, we used (non-informative) priors $\gamma \sim \Gamma(10^{-4}, 10^{-4})$, $\theta \sim \mathcal{N}([0,0], 10I)$ and (informative) priors $\kappa \sim \mathcal{N}(1.5, 0.1)$ and $\omega \sim \mathcal{N}(-3, 0.1)$, along with initial state $z_0 \sim \mathcal{N}(0, 10)$. For each time step, we iterate the full message passing schedule (messages 1 through 19) 10 times in order to drive the free energy to convergence. The simulations[2] were implemented in the open source Julia package `ForneyLab` (Cox et al., 2019).

Figure 2 (second and third row left) shows the inference tracks for the control state z_t and variance ϑ_t. The table in Figure 2 (top right) presents the inferred posterior parameters after 1000 samples. Note that the posterior over ϑ_t captures the process noise variance of the autoregressive model, even though the inferred means of κ and ω are underestimated. Figure 2 (second and third row right) shows the evolution from $t = 10$ to $t = 500$ of the marginal posterior for $q(\boldsymbol{\theta})$. We observe that the inferred parameter values converge to the true values. Figure 2 (bottom row) shows the free energy tracks, both as a function of time after ten iterations (left) and as a function of iterations, averaged over time (right). The free energy is an upper bound on negative log-evidence and is used as a performance measure. New observations add free energy temporarily in the form of prediction errors, which are subsequently squashed by message passing in between observations (left). Note that each message passing iteration drives the free energy further down, thus improving model fit (right).

We compared the performance of AR-HGF to an AR model with gamma prior on process noise precision (i.e., $\vartheta_t^{-1} = \vartheta^{-1} \sim \Gamma(10^{-4}, 1.0)$). Averaged over time, the static variant (AR-static) has a higher free energy than the AR-HGF (see Figure 2 bottom right). The AR-static model is less able to track the signal in volatile periods, which can be seen by the blue spikes in free energy around t = 240, t = 320 and t = 550. It also cannot cope with a sudden drop in process noise variance (see t = 740 and t = 750 in top left): the AR-HGF's free energy drops to 0 and below, while that of the AR-static stays almost flat around 2.

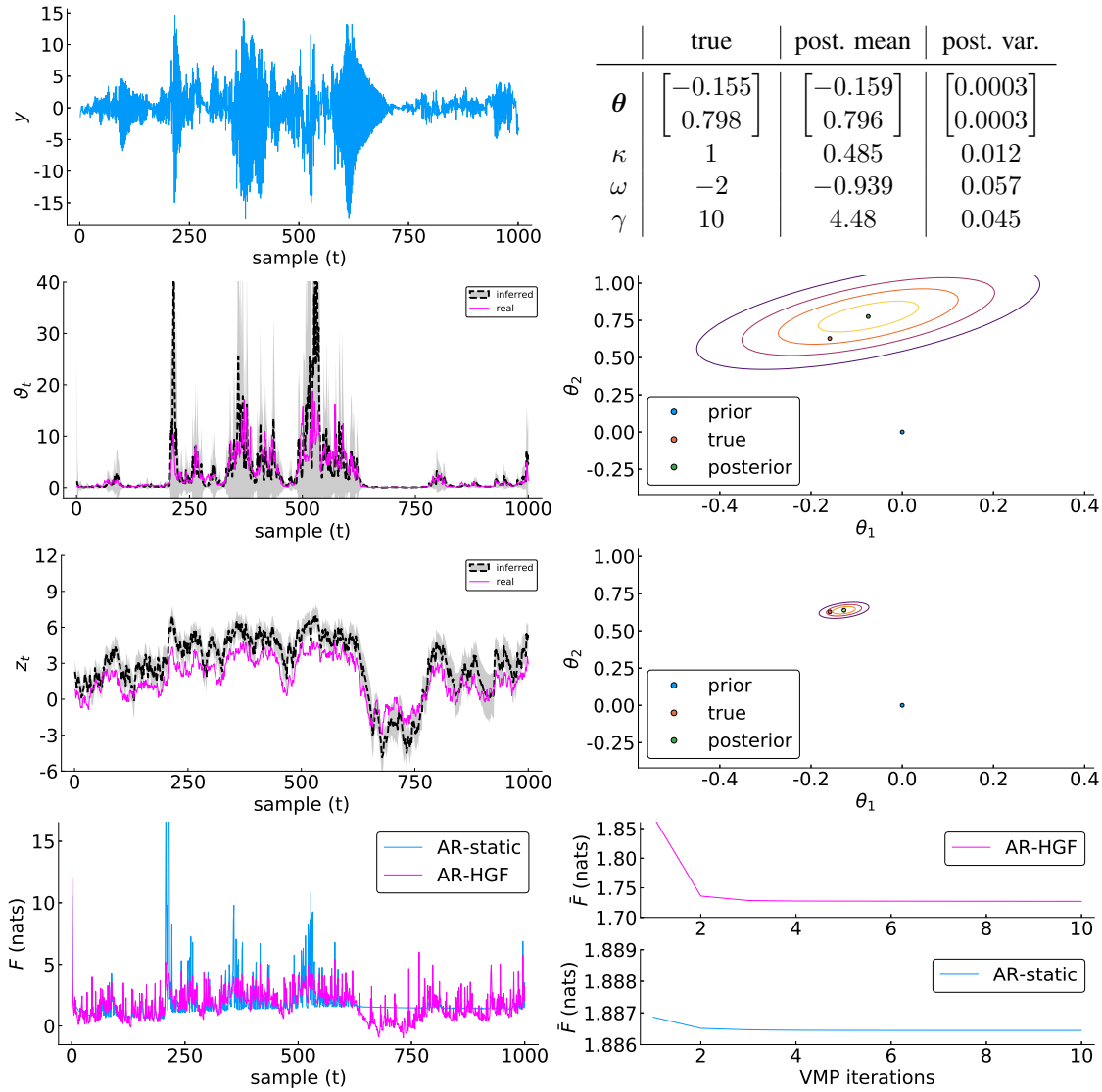
5. Conclusions

We presented a message passing-based online inference method for joint state and parameter tracking in autoregressive models. Our model includes a dynamical prior on process noise variance, which can be extended hierarchically. New update rules for the Gaussian node with externally controlled variance parameter were derived. In principle, as message passing in factor graphs is a highly modular inference approach, the presented methods are re-usable in a wider application context.

Acknowledgements

This work was partly financed by research program ZERO with project number P15-06, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

2. The Jupyter notebook with the experiments can be found at https://github.com/biaslab/L4DC-2020



	true	post. mean	post. var.
$\boldsymbol{\theta}$	$\begin{bmatrix}-0.155\\0.798\end{bmatrix}$	$\begin{bmatrix}-0.159\\0.796\end{bmatrix}$	$\begin{bmatrix}0.0003\\0.0003\end{bmatrix}$
κ	1	0.485	0.012
ω	-2	-0.939	0.057
γ	10	4.48	0.045

Figure 2: Experimental results. (First row left). Example of a synthetic signal of an AR process with time-varying process noise variance. (First row right) True values for $\boldsymbol{\theta}, \kappa, \omega, \gamma$ used for generating the synthetic signal (top left), and inferred values (posterior mean and variance). (Second row left) Real (pink solid) vs inferred (black dashed) tracks for variance parameter ϑ_t. (Second row right) Posterior for $\boldsymbol{\theta}$ at $t = 10$. (Third row left) Real (pink solid) vs inferred (black dashed) control states z_t. (Third row right) Posterior for $\boldsymbol{\theta}$ at $t = 500$. (Last row left) Free energy of final estimates, for each time-step. (Last row right) Free energy averaged over time as a function of the number of message-passing iterations for AR-HGF and AR-static models. AR-HGF outperforms AR-static with a lower-free energy profile over iterations.

References

Hirotugu Akaike. *Autoregressive Model Fitting for Control*, pages 153–170. Springer New York, New York, NY, 1998.

Christophe Andrieu, Manuel Davy, and Arnaud Doucet. Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, 51(7):1762–1770, 2003.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Glen Barnett, Robert Kohn, and Simon Sheather. Bayesian estimation of an autoregressive model using Markov chain Monte Carlo. *Journal of Econometrics*, 74(2):237–254, 1996.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Marco Cox, Thijs van de Laar, and Bert de Vries. A factor graph approach to automated design of Bayesian signal processing algorithms. *International Journal of Approximate Reasoning*, 104: 185–204, 2019.

Justin Dauwels. On variational message passing on factor graphs. In *IEEE International Symposium on Information Theory*, pages 2546–2550, 2007.

Deniz Gençağa, Ercan E Kuruoğlu, and Ayşın Ertüzün. Modeling non-Gaussian time-varying vector autoregressive processes by particle filtering. *Multidimensional Systems and Signal Processing*, 21(1):73, 2010.

David C. Hill, David McMillan, Keith R. W. Bell, and David Infield. Application of auto-regressive models to U.K. wind speed data for power system impact studies. *IEEE Transactions on Sustainable Energy*, 3(1):134–141, 2012.

Xing-Qi Jiang and Genshiro Kitagawa. A time varying coefficient vector AR modeling of nonstationary covariance time series. *Signal Processing*, 33(3):315–331, 1993.

Sascha Korl. *A factor graph approach to signal modelling, system identification and filtering*. PhD thesis, Swiss Federal Institute of Technology, Zürich, 2005.

Hans-Andrea Loeliger, Justin Dauwels, Junli Hu, Sascha Korl, Li Ping, and Frank R Kschischang. The factor graph approach to model-based signal processing. *Proceedings of the IEEE*, 95(6): 1295–1322, 2007.

Christoph D Mathys. *Hierarchical Gaussian filtering*. PhD thesis, ETH Zürich, Zürich, 2012.

Florian Meyer, Ondrej Hlinka, and Franz Hlawatsch. Sigma point belief propagation. *IEEE Signal Processing Letters*, 21(2):145–149, 2013.

Stephen J Roberts and Will D Penny. Variational Bayes for generalized autoregressive models. *IEEE Transactions on Signal Processing*, 50(9):2245–2257, 2002.

Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

İsmail Şenöz and Bert de Vries. Online variational message passing in the hierarchical Gaussian filter. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2018.