entropy

MDPI

Article

Message Passing-Based Inference for Time-Varying Autoregressive Models

Albert Podusenko [1,*], Wouter M. Kouw [1] and Bert de Vries [1,2]

[1] Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands; w.m.kouw@tue.nl (W.M.K.); bdevries@gnresound.com (B.d.V.)
[2] GN Hearing, JF Kennedylaan 2, 5612 AB Eindhoven, The Netherlands
* Correspondence: a.podusenko@tue.nl

Abstract: Time-varying autoregressive (TVAR) models are widely used for modeling of non-stationary signals. Unfortunately, online joint adaptation of both states and parameters in these models remains a challenge. In this paper, we represent the TVAR model by a factor graph and solve the inference problem by automated message passing-based inference for states and parameters. We derive structured variational update rules for a composite "AR node" with probabilistic observations that can be used as a plug-in module in hierarchical models, for example, to model the time-varying behavior of the hyper-parameters of a time-varying AR model. Our method includes tracking of variational free energy (FE) as a Bayesian measure of TVAR model performance. The proposed methods are verified on a synthetic data set and validated on real-world data from temperature modeling and speech enhancement tasks.

Keywords: Bayesian inference; free energy; factor graph; hybrid message passing; model selection; non-stationary systems; probabilistic graphical models

check for updates

Citation: Podusenko, A.; Kouw, W.M.; de Vries, B. Message Passing-Based Inference for Time-Varying Autoregressive Models. *Entropy* **2021**, *23*, 683. https://doi.org/10.3390/e23060683

Academic Editor: Antonio Salmerón and Rafael Rumí

Received: 18 April 2021
Accepted: 24 May 2021
Published: 28 May 2021

1. Introduction

Autoregressive (AR) models are capable of describing a wide range of time series patterns [1,2]. The extension to Time-Varying AR (TVAR) models, where the AR coefficients are allowed to vary over time, supports tracking of non-stationary signals. TVAR models have been successfully applied to a wide range of applications, including speech signal processing [3–5], signature verification [6], cardiovascular response modeling [7], acoustic signature recognition of vehicles [8], radar signal processing [9], and EEG analysis [10,11].

The realization of TVAR models in practice often poses some computational issues. For many applications, such as speech processing in a hearing aid, both a low computational load and high modeling accuracy are essential.

The problem of parameter tracking in TVAR models has been extensively explored in a non-Bayesian setting. For example, ref. [12] employs over-determined modified Yule-Walker equations and [13] applies the covariance method to track the parameters in a TVAR model. In [14], expressions for the mean vector and covariance matrix of TVAR model coefficients are derived and [15] uses wavelets for TVAR model identification. Essentially, all these approaches provide maximum likelihood estimates of coefficients for TVAR models without measurement noise. In [16], autoregressive parameters were estimated from noisy observations by using a recursive least-squares adaptive filter.

We take a Bayesian approach since we are also interested in tracking Bayesian evidence (or an approximation thereof) as a model performance measure. Bayesian evidence can be used to track the optimal AR model order or more generally, to compare the performance of a TVAR model to an alternative model. To date, Bayesian parameter tracking in AR models has mostly been achieved by Monte Carlo sampling methods [17–21]. Sampling-based inference is highly accurate, but it is very often computationally too expensive for real-time processing on wearable devices such as hearing aids, smartwatches, etc.

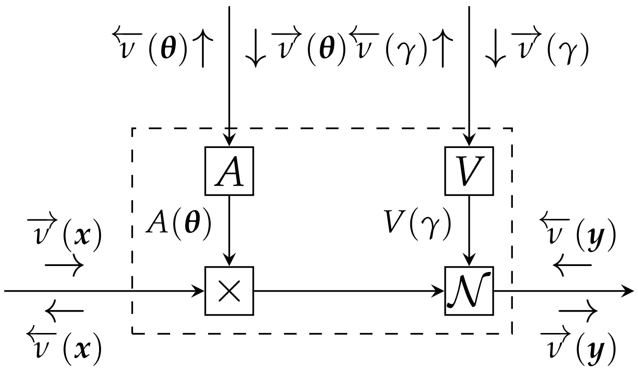
In this paper, we develop a low-complexity variational message passing-based (VMP) realization for tracking of states, parameters and free energy (an upper bound on Bayesian evidence) in TVAR models. All update formulas are closed-form and the complete inference process can easily be realized.

VMP is a low-complexity distributed message passing-based realization of variational Bayesian inference on a factor graph representation of the model [22,23]. Previous work on message passing-based inference for AR models include [24], but their work describes maximum likelihood estimation and therefore does not track proper posteriors and free energy. In [25], variational inference is employed to estimate the parameters of a multivariate AR model, but their work does not take advantage of the factor graph representation.

The factor graph representation that we employ in this paper provides some distinct advantages from other works on inference in TVAR models. First, a factor graph formulation is by definition completely modular and supports re-using the derived TVAR inference equations as a plug-in module in other factor graph-based models. In particular, since we allow for measurement noise in the TVAR model specification, the proposed TVAR factor can easily be used as a latent module at any level in hierarchical dynamical models. Moreover, due to the modularity, VMP update rules can easily be mixed with different update schemes such as belief propagation and expectation [26,27] in other modules, leading to hybrid message passing schemes for efficient inference in complex models. We have implemented the TVAR model in the open source and freely available factor graph toolbox ForneyLab [28].

The rest of this paper is organized as follows. In Section 2, we specify the TVAR model as a probabilistic state space model and define the inference tasks that relate to tracking of states, parameters, and Bayesian evidence. After a short discussion on the merits of using Bayesian evidence as a model performance criterion (Section 3.1), we formulate Bayesian inference in the TVAR model as a set of sequential prediction-correction processes (Section 3.2). We will realize these processes as VMP update rules and proceed with a short review of Forney-style factor graphs and message passing in Section 4. Then, in Section 5, the VMP equations are worked out for the TVAR model and summarized in Table 1. Section 6 discusses a verification experiment on a synthetic data set and applications of the proposed TVAR model to temperature prediction and speech enhancement problems. Full derivations of the closed-form VMP update rules are presented in Appendix A.

Table 1. Variational message update rules for the autoregressive (AR) node (dashed box) of Equation (30).

VMP for the Composite AR Node


$\overleftarrow{\nu}(\boldsymbol{\theta})\uparrow$
$\downarrow\overrightarrow{\nu}(\boldsymbol{\theta})$
$\overleftarrow{\nu}(\gamma)\uparrow$
$\downarrow\overrightarrow{\nu}(\gamma)$
A
V
$A(\boldsymbol{\theta})$
$V(\gamma)$
$\overrightarrow{\nu}(\boldsymbol{x})$
$\overleftarrow{\nu}(\boldsymbol{x})$
$\times$
$\mathcal{N}$
$\overleftarrow{\nu}(\boldsymbol{y})$
$\overrightarrow{\nu}(\boldsymbol{y})$

Table 1. *Cont.*

Outgoing messages	**Incoming messages**
$\overrightarrow{\nu}(\boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}\vert z_0, \Sigma)$	$\overleftarrow{\nu}(\boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}\vert m_{\boldsymbol{y}}, V_{\boldsymbol{y}})$
$\overleftarrow{\nu}(\boldsymbol{x}) \propto \mathcal{N}\Big(\boldsymbol{x}\vert \Lambda_1^{-1} z_1, \Lambda_1^{-1}\Big)$	$\overrightarrow{\nu}(\boldsymbol{x}) \propto \mathcal{N}\Big(\boldsymbol{x}\vert m_{\boldsymbol{x}}, V_{\boldsymbol{x}}\Big)$
$\overleftarrow{\nu}(\boldsymbol{\theta}) \propto \mathcal{N}\Big(\boldsymbol{\theta}\vert \Lambda_2^{-1} z_2, \Lambda_2^{-1}\Big)$	$\overrightarrow{\nu}(\boldsymbol{\theta}) \propto \mathcal{N}\Big(\boldsymbol{\theta}\vert m_{\boldsymbol{\theta}}, V_{\boldsymbol{\theta}}\Big)$
$\overleftarrow{\nu}(\gamma) \propto \Gamma\Big(\gamma\vert \frac{3}{2}, \frac{b}{2}\Big)$	$\overrightarrow{\nu}(\gamma) \propto \Gamma\Big(\gamma\vert \alpha, \beta\Big)$
Joint marginal $q(\boldsymbol{y}, \boldsymbol{x})$	
$q(\boldsymbol{y}, \boldsymbol{x}) \propto \overrightarrow{\nu}(\boldsymbol{x}) \exp\Big[\mathbb{E}_{q(\gamma)q(\boldsymbol{\theta})} \log f(\boldsymbol{y}\ \boldsymbol{x}, \boldsymbol{\theta}, \gamma)\Big] \overleftarrow{\nu}(\boldsymbol{y}) \propto \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \end{bmatrix} \middle\vert \begin{bmatrix} m_{\boldsymbol{y}}^* \\ m_{\boldsymbol{x}}^* \end{bmatrix}, \begin{bmatrix} V_{\boldsymbol{y}}^* & V_{\boldsymbol{yx}}^* \\ V_{\boldsymbol{xy}}^* & V_{\boldsymbol{x}}^* \end{bmatrix}\right)$ (Appendix A.7)	
Free energy $F[q]$	
$F[q] = u + \frac{1}{2}\log 2\pi + \frac{m_\gamma}{2}\Big(\sigma_y^2 + m_y^2 - 2\big[V_{yx^\intercal} + m_y m_{\boldsymbol{x}}^\intercal\big] m_{\boldsymbol{\theta}} + \mathrm{tr}\big[(V_{\boldsymbol{\theta}} + m_{\boldsymbol{\theta}} m_{\boldsymbol{\theta}}^\intercal) V_{\boldsymbol{x}}^*\big] + m_{\boldsymbol{\theta}}^\intercal (V_{\boldsymbol{x}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{x}}^*)^\intercal) m_{\boldsymbol{\theta}}\Big)$	
Auxiliary variables	
$\Sigma = m_{\boldsymbol{A}}(V_{\boldsymbol{x}}^{-1} + m_\gamma V_{\boldsymbol{\theta}})^{-1} m_{\boldsymbol{A}}^\intercal + m_{\boldsymbol{V}}$	$z_0 = m_{\boldsymbol{A}}(V_{\boldsymbol{x}}^{-1} + m_\gamma V_{\boldsymbol{\theta}})^{-1} V_{\boldsymbol{x}}^{-1} m_{\boldsymbol{x}}$
$\Lambda_1 = m_{\boldsymbol{A}}^\intercal \big(V_{\boldsymbol{y}} + m_{\boldsymbol{V}}\big)^{-1} m_{\boldsymbol{A}} + m_\gamma V_\theta$	$z_1 = m_{\boldsymbol{A}}^\intercal \big(V_{\boldsymbol{y}} + m_{\boldsymbol{V}}\big)^{-1} m_{\boldsymbol{y}}$
$\Lambda_2 = m_\gamma (V_{\boldsymbol{x}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{x}}^*)^\intercal)$	$z_2 = (V_{\boldsymbol{xy}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{y}}^*)^\intercal) \boldsymbol{c} m_\gamma$
$b = \boldsymbol{c}^\intercal \Big[V_{\boldsymbol{y}}^* + m_{\boldsymbol{y}}^* (m_{\boldsymbol{y}}^*)^\intercal - 2 m_{\boldsymbol{A}} (V_{\boldsymbol{xy}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{y}}^*)^\intercal) + m_{\boldsymbol{A}} (V_{\boldsymbol{x}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{x}}^*)^\intercal) m_{\boldsymbol{A}}^\intercal + \mathrm{tr}(V_\theta (V_{\boldsymbol{x}}^* + m_{\boldsymbol{x}}^* (m_{\boldsymbol{x}}^*)^\intercal))\Big] \boldsymbol{c}$	
$m_\gamma = \frac{\alpha}{\beta}$ $\quad m_{\boldsymbol{A}} = m_{A(\boldsymbol{\theta})}$	
$u = -\frac{1}{2}[\psi(\alpha) - \log\beta] + \frac{1}{2}\log 2\pi$ $\quad \sigma_y^2 = \boldsymbol{c}^\intercal V_{\boldsymbol{y}}^* \boldsymbol{c}$ $\quad m_y = \boldsymbol{c}^\intercal m_{\boldsymbol{y}}^* \boldsymbol{c}$ $\quad V_{yx} = V_{\boldsymbol{yx}}^* \boldsymbol{c}$	

2. Model Specification and Problem Definition

In this section, we first specify TVAR as a state-space model. This is followed by an inference problem formulation.

2.1. Model Specification

A TVAR model is specified as

$$\theta_{m,t} \sim \mathcal{N}(\theta_{m,t-1}, \omega) \tag{1a}$$

$$x_t \sim \mathcal{N}\Big(\sum_{m=1}^{M} \theta_{m,t} x_{t-m}, \gamma^{-1}\Big) \tag{1b}$$

$$y_t \sim \mathcal{N}(x_t, \tau)\,, \tag{1c}$$

where $y_t \in \mathbb{R}$, $x_t \in \mathbb{R}$ and $\theta_{m,t} \in \mathbb{R}$ represent the the observation, state and parameters at time t, respectively. M denotes the order of the AR model. As a notational convention, we use $\mathcal{N}(\mu, \Sigma)$ to denote a Gaussian distribution with mean μ and co-variance matrix Σ. We can re-write (1) in state-space form as

$$\boldsymbol{\theta}_t \sim \mathcal{N}(\boldsymbol{\theta}_{t-1}, \omega \mathrm{I}_M) \tag{2a}$$

$$\boldsymbol{x}_t \sim \mathcal{N}\Big(A(\boldsymbol{\theta}_t)\boldsymbol{x}_{t-1}, V(\gamma)\Big) \tag{2b}$$

$$y_t \sim \mathcal{N}(\boldsymbol{c}^\intercal \boldsymbol{x}_t, \tau)\,, \tag{2c}$$

where $\boldsymbol{\theta}_t = (\theta_{m,t}, \theta_{m-1,t}, \ldots, \theta_{m-M,t})^\intercal$, $\boldsymbol{x_t} = (x_t, x_{t-1}, \ldots, x_{t-M+1})^\intercal$, $\boldsymbol{c} = (1, 0, \ldots, 0)^\intercal$ is an M-dimensional unit vector, $V(\gamma) = (1/\gamma)\boldsymbol{c}\boldsymbol{c}^\intercal$, and

$$A(\boldsymbol{\theta}) = \begin{bmatrix} \multicolumn{2}{c}{\boldsymbol{\theta}^\intercal} \\ \mathrm{I}_{M-1} & \mathbf{0}. \end{bmatrix} \tag{3}$$

Technically, a TVAR model usually assumes $\tau = 0$ indicating that there is no measurement noise. Note that the presence of measurement noise in (2c) "hides" the states $\boldsymbol{x}_t$ in the generative model (2) from the observation sequence y_t, yielding a latent TVAR. We add measurement noise explicitly, so the model is able to accept information from likelihood functions that are not constrained to be delta functions with hard observations. As a result, the AR model that we define here can be used at any level in deep hierarchical structures such as [29] as a plug-in module.

In a time-invariant AR model, $\boldsymbol{\theta}$ are part of the parameters of the system. In a time-varying AR model, we consider $\boldsymbol{\theta}_t$ and $\boldsymbol{x}_t$ together the set of time-varying states. The parameters of the TVAR model are $\{\boldsymbol{\theta}_0, \boldsymbol{x}_0, \omega, \gamma, \tau\}$.

At the heart of the TVAR model is the transition model (2b), where $A(\boldsymbol{\theta}_t)$ is a companion matrix with AR coefficients. The multiplication $A(\boldsymbol{\theta})\boldsymbol{x}_{t-1}$ performs two operations: a dot product $\boldsymbol{\theta}_t^\intercal \boldsymbol{x}_{t-1}$ and a vector shift of $\boldsymbol{x}_{t-1}$ by one time step. The latter operation can be interpreted as bookkeeping, as it shifts each entry of $\boldsymbol{x}_{t-1}$ one position down and discards x_{t-M}.

2.2. Problem Definition

For a given time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$, we are firstly interested in recursively updating posteriors for the states $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:l})$ and $p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:l})$. In this context, prediction, filtering and smoothing are recovered for $l < t$, $l = t$ and $l > t$, respectively.

Furthermore, we are interested in computing posteriors for the parameters $p(\boldsymbol{\theta}_0|\boldsymbol{y})$, $p(\boldsymbol{x}_0|\boldsymbol{y})$, $p(\omega|\boldsymbol{y})$, $p(\gamma|\boldsymbol{y})$ and $p(\tau|\boldsymbol{y})$.

Finally, we are interested in scoring the performance of a proposed TVAR model m with specified priors for the parameters. In this paper, we take a full Bayesian approach and select Bayesian evidence $p(\boldsymbol{y}|m)$ as the performance criterion. Section 3.1 discusses the merits of Bayesian evidence as a model performance criterion.

3. Inference in TVAR Models

In this section, we first discuss some of the merits of using Bayesian evidence as a model performance criterion. This is followed by an exposition of how to compute Bayesian evidence and the desired posteriors in the TVAR model.

3.1. Bayesian Evidence as a Model Performance Criterion

Consider a model m with parameters θ and observations y. Bayesian evidence $p(\boldsymbol{y}|m)$ is considered an excellent model performance criterion. Note the following decomposition [30]:

$$\begin{aligned}
\log p(\boldsymbol{y}|m) &= \log \frac{p(\boldsymbol{y}|\boldsymbol{\theta}, m)p(\boldsymbol{\theta}|m)}{p(\boldsymbol{\theta}|\boldsymbol{y}, m)} \qquad \text{(use Bayes rule)} \\
&= \int p(\boldsymbol{\theta}|\boldsymbol{y}, m) \cdot \underbrace{\log \frac{p(\boldsymbol{y}|\boldsymbol{\theta}, m)p(\boldsymbol{\theta}|m)}{p(\boldsymbol{\theta}|\boldsymbol{y}, m)}}_{\log p(\boldsymbol{y}|m) \text{ is not a function of } \boldsymbol{\theta}} \mathrm{d}\boldsymbol{\theta} \\
&= \underbrace{\int p(\boldsymbol{\theta}|\boldsymbol{y}, m) \log p(\boldsymbol{y}|\boldsymbol{\theta}, m) \mathrm{d}\boldsymbol{\theta}}_{\text{data fit}} - \underbrace{\int p(\boldsymbol{\theta}|\boldsymbol{y}, m) \log \frac{p(\boldsymbol{\theta}|\boldsymbol{y}, m)}{p(\boldsymbol{\theta}|m)} \mathrm{d}\boldsymbol{\theta}}_{\text{complexity}}
\end{aligned} \tag{4}$$

The first term (data fit or sometimes called accuracy) measures how well the model predicts the data $\boldsymbol{y}$, after having learned from the data. We want this term to be large (although only focusing on this term could lead to over-fitting). The second term (complexity) quantifies the amount of information that the model absorbed through learning by moving parameter beliefs from $p(\boldsymbol{\theta}|m)$ to $p(\boldsymbol{\theta}|\boldsymbol{y}, m)$. To see this, note that the mutual information between two variables $\boldsymbol{\theta}$ and $\boldsymbol{y}$, which is defined as

$$I[\boldsymbol{\theta};\boldsymbol{y}] = \iint p(\boldsymbol{\theta},\boldsymbol{y}) \log \frac{p(\boldsymbol{\theta}|\boldsymbol{y})}{p(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{y}\,,$$

can be interpreted as expected complexity. The complexity term regularizes the Bayesian learning process automatically. Preference for models with high Bayesian evidence implies a preference for models that get a good data fit without the need to learn much from the data set. These types of models are said to *generalize* well, since they can be applied to different data sets without specific adaptations for each data set. Therefore, Bayesian learning automatically leads to models that tend to generalize well.

Note that Bayesian evidence for a model m, given a full times series $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$, can be computed by multiplication of the sample-based evidences:

$$p(\boldsymbol{y}|m) = \prod_{t=1}^{T} p(y_t|\boldsymbol{y}_{1:t-1}, m)\,. \tag{5}$$

3.2. Inference as a Prediction-Correction Process

To illustrate the type of calculations that are needed for computing Bayesian model evidence and the posteriors for states and parameters, we now proceed to write out the needed calculations for the TVAR model in a filtering context.

Assume that at the beginning of time step t, we are given the state posteriors $q(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$, $q(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})$. We will denote the inferred probabilities by $q(\cdot)$, in contrast to factors from the generative model that are written as $p(\cdot)$. We start the procedure by setting the state priors for the generative model at step t to the posteriors of the previous time step

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) := q(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) \tag{6}$$

$$p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1}) := q(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1}) \tag{7}$$

Given a new observation y_t, we are now interested inferring the evidence $q(y_t|\boldsymbol{y}_{t-1})$, and in inferring posteriors $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ and $q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$.

This involves a prediction (forward) pass through the system that leads to the evidence update, followed by a correction (backward) pass that updates the states. We work this out in detail below. For clarity of exposition, in this section we call $\boldsymbol{x}_t$ states and $\boldsymbol{\theta}_t$ parameters. Starting with the forward pass (from latent variables toward observation), we first compute a parameter prior predictive:

$$\underbrace{q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\\text{prior predictive}}} = \int \underbrace{p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})}_{\substack{\text{parameter}\\\text{transition}}} \underbrace{p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\\text{prior}}} \mathrm{d}\boldsymbol{\theta}_{t-1}\,. \tag{8}$$

Then the prior predictive for the state transition becomes:

$$\underbrace{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:t-1})}_{\substack{\text{state transition}\\\text{prior predictive}}} = \int \underbrace{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_t)}_{\text{state transition}} \underbrace{q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\\text{prior predictive}}} \mathrm{d}\boldsymbol{\theta}_t\,. \tag{9}$$

Note that the state transition prior predictive, due to its dependency on time-varying θ_t, is a function of the observed data sequence. The state transition prior predictive can be used together with the state prior to inferring the state prior predictive:

$$\underbrace{q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}_{\substack{\text{state}\\ \text{prior predictive}}} = \int \underbrace{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{1:t-1})}_{\substack{\text{state transition}\\ \text{prior predictive}}} \underbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})}_{\text{state prior}} \,\mathrm{d}\boldsymbol{x}_{t-1}\,. \tag{10}$$

The evidence for model m that is provided by observation y_t is then given by

$$\underbrace{q(y_t|\boldsymbol{y}_{1:t-1})}_{\text{evidence}} = \int \underbrace{p(y_t|\boldsymbol{x}_t)}_{\substack{\text{state}\\ \text{likelihood}}} \underbrace{q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}_{\substack{\text{state prior}\\ \text{predictive}}} \,\mathrm{d}\boldsymbol{x}_t\,. \tag{11}$$

When y_t has not yet been observed, $q(y_t|\boldsymbol{y}_{1:t-1})$ is a prediction for y_t. After plugging in the observed value for y_t, the evidence is a scalar that scores how well the model performed in predicting y_t. As discussed in (5), the results $q(y_t|\boldsymbol{y}_{1:t-1})$ for $t = 1, 2, \ldots, T$ in (11) can be used to score the model performance for a given time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$. Note that to update the evidence, we need to integrate over all latent variables $\boldsymbol{\theta}_{t-1}$, $\boldsymbol{\theta}_t$, $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ (by (8)–(11)). In principle, this scheme needs to be extended with integration over the parameters ω, γ and τ.

Once we have inferred the evidence, we proceed by a backward corrective pass through the model to update the posterior over the latent variables given the new observation y_t. The state posterior can be updated by Bayes rule:

$$\underbrace{q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})}_{\text{state posterior}} = \frac{\overbrace{p(y_t|\boldsymbol{x}_t)}^{\substack{\text{state}\\ \text{likelihood}}}\overbrace{q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}^{\substack{\text{state prior}\\ \text{predictive}}}}{\underbrace{q(y_t|\boldsymbol{y}_{1:t-1})}_{\text{evidence}}} \tag{12}$$

Next, we need to compute a likelihood function for the parameters. Fortunately, we can re-use some intermediate results from the forward pass. The likelihood for the parameters is given by

$$\underbrace{q(y_t|\boldsymbol{\theta}_t,\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\ \text{likelihood}}} = \int \underbrace{p(y_t|\boldsymbol{x}_t)}_{\substack{\text{state}\\ \text{likelihood}}} \underbrace{q(\boldsymbol{x}_t|\boldsymbol{\theta}_t,\boldsymbol{y}_{1:t-1})}_{\substack{\text{state prior}\\ \text{predictive}}} \,\mathrm{d}\boldsymbol{x}_t \tag{13}$$

The parameter posterior then follows from Bayes rule:

$$\underbrace{q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})}_{\text{parameter posterior}} = \frac{\overbrace{q(y_t|\boldsymbol{\theta}_t,\boldsymbol{y}_{1:t-1})}^{\substack{\text{parameter}\\ \text{likelihood}}}\overbrace{q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1})}^{\substack{\text{parameter}\\ \text{prior predictive}}}}{\underbrace{q(y_t|\boldsymbol{y}_{1:t-1})}_{\text{evidence}}} \tag{14}$$

Equations (11), (12) and (14) contain the solutions to our inference task. Note that the evidence $q(y_t|\boldsymbol{y}_{1:t-1})$ is needed to normalize the latent variable posteriors in (12) and (14). Moreover, while we integrate over the states by (11) to compute the evidence, (14) reveals that the evidence can alternatively be computed by integrating over the parameters through

$$\underbrace{q(y_t|\boldsymbol{y}_{1:t-1})}_{\text{evidence}} = \int \underbrace{q(y_t|\boldsymbol{\theta}_t,\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\ \text{likelihood}}} \underbrace{q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1})}_{\substack{\text{parameter}\\ \text{prior predictive}}} \,\mathrm{d}\boldsymbol{\theta}_t\,. \tag{15}$$

This latter method of evidence computation may be useful if re-using (11) in (14) leads to numerical rounding issues.

Unfortunately, many of Equations (8) through (14) are not analytically tractable for the TVAR model. This happens due to (1) integration over large state spaces, (2) non-conjugate prior-posterior pairing, and (3) the absence of a closed-form solution for the evidence factor.

To overcome this challenge, we will perform inference by a hybrid message passing scheme in a factor graph. In the next section, we give a short review of Forney-Style Factor Graphs (FFG) and Message-Passing (MP) based inference techniques.

4. Factor Graphs and Message Passing-Based Inference

In this section, we make a brief introduction of Forney-Style Factor graph (FFG) and sum-product (SP) algorithm. After that we review the minimization of variational free energy and Variational Message Passing (VMP) algorithm.

4.1. Forney-Style Factor Graphs

A Forney-style Factor graph is a representation of a factorized function where the factors and variables are represented by nodes and edges, respectively. An edge is connected to a node if and only if the (edge) variable is an argument of the node function. In our work, we use FFGs to represent factorized probability distributions. FFGs provide both an attractive visualization of the model and a highly efficient and modular inference method based on message passing. An important component of the FFG representation is the equality node. If a variable x is shared between more than two nodes, then we introduce two auxiliary variables x' and x'' and use an equality node

$$f_{=}(x, x', x'') = \delta(x - x')\delta(x - x'') \tag{16}$$

to constrain the marginal beliefs over x, x', x'' to be equal. With this mechanism, any factorized function can be represented as an FFG.

An FFG visualization of the TVAR model is depicted in Figure 3, but for illustrative purposes, we first consider an example factorized distribution

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3) \tag{17}$$

This distribution can be visualized by an FFG shown in Figure 1. An FFG is in principle an undirected graph but we often draw arrows on the edges in the "generative" direction, which is the direction that describes how the observed data is generated. Assume that we are interested in computing the marginal for x_2, given by

$$p(x_2) = \iiint p(x_1, x_2, x_3, x_4)\mathrm{d}x_1\mathrm{d}x_3\mathrm{d}x_4 \tag{18}$$

We can reduce the complexity of computing this integral by rearranging the factors over the integration signs as

$$\begin{aligned} p(x_2) &= \underbrace{\int \underbrace{p(x_1)}_{\overrightarrow{\mu}_1(x_1)} p(x_2|x_1)\mathrm{d}x_1}_{\overrightarrow{\mu}_2(x_2)} \cdot \underbrace{\Big(\int p(x_3|x_2) \underbrace{\big(\int p(x_4|x_3)\mathrm{d}x_3\big)}_{\overleftarrow{\mu}_3(x_3)} \mathrm{d}x_3\Big)}_{\overleftarrow{\mu}_2(x_2)} && \text{(19a)} \\ &= \overrightarrow{\mu}_2(x_2) \cdot \overleftarrow{\mu}_2(x_2)\,. && \text{(19b)} \end{aligned}$$

These re-arranged integrals can be interpreted as messages that are passed over the edges, see Figure 1. It is a notational convention to call a message $\overrightarrow{\mu}(\cdot)$ that aligns with the direction of the edge arrow a forward message and similarly, a message $\overleftarrow{\mu}(\cdot)$ that opposes the direction of the edge is called a backward message.

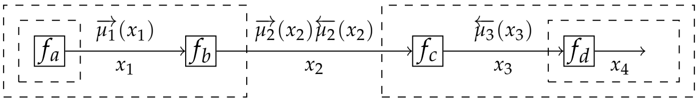

f_a
$\overrightarrow{\mu_1}(x_1)$
x_1
f_b
$\overrightarrow{\mu_2}(x_2)\overleftarrow{\mu_2}(x_2)$
x_2
f_c
$\overleftarrow{\mu_3}(x_3)$
x_3
f_d
x_4

Figure 1. An FFG corresponding to model (17), including messages as per (19). For graphical clarity, we defined $f_a(x_1) = p(x_1)$, $f_b(x_1, x_2) = p(x_2|x_1)$, $f_c(x_2, x_3) = p(x_3|x_2)$ and $f_d(x_3, x_4) = p(x_4|x_3)$.

This message passed-based algorithm of computing the marginal is called belief propagation (BP) or the sum-product algorithm. As can be verified in (19), for a node with factor $f(y, x_1, \ldots, x_n)$, the outgoing BP message $\overrightarrow{\mu}(y)$ to variable y can be expressed as

$$\overrightarrow{\mu}_y(y) = \int \cdots \int f(y, x_1, \ldots, x_n) \prod_{i=1}^{n} \overrightarrow{\mu}_i(x_i) \mathrm{d}x_i \,. \tag{20}$$

where $\overrightarrow{\mu}_i(x_i)$ is an incoming message over edge x_i. If the factor graph is a tree, meaning that the graph contains no cycles, then BP leads to exact Bayesian inference. A more detailed explanation of belief propagation message passing in FFGs can be found in [26].

4.2. Free Energy and Variational Message Passing

Technically, BP is a message passing algorithm that belongs to a family of message passing algorithms that minimize a constrained variational free energy functional [31]. Unfortunately, the sum-product rule (20) only has a closed-form solution for Gaussian incoming messages $\overrightarrow{\mu}_i(x_i)$ and linear variable relations in $f(y, x_1, \ldots, x_n)$. Another important member of the free energy minimizing algorithms is the Variational Message Passing (VMP) algorithm [22]. VMP enjoys a wider range of analytically computable message update rules.

We shortly review variational Bayesian inference and VMP next. Consider a model $p(\boldsymbol{y}, \boldsymbol{x})$ with observations $\boldsymbol{y}$ and unobserved (latent) variables $\boldsymbol{x}$. We are interested in inferring the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$. In variational inference we introduce an approximate posterior $q(\boldsymbol{x})$ and define a variational free energy functional as

$$F[q] \triangleq \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{y}, \boldsymbol{x})} \mathrm{d}\boldsymbol{x} = \underbrace{\int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})} \mathrm{d}\boldsymbol{x}}_{\text{KL divergence } D_{\mathrm{KL}}(q,p)} - \underbrace{\log p(\boldsymbol{y})}_{\text{log-evidence}} \,. \tag{21}$$

The second term in (21) (log-evidence) is not a function of the argument of F. The first term is a KL-divergence, which is by definition non-negative and only equals zero for $q(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$. As a result, variational inference by minimization of $F[q]$ provides

$$q^*(\boldsymbol{x}) = \arg\min_q F[q] \tag{22}$$

which is an approximation to the Bayesian posterior $p(\boldsymbol{x}|\boldsymbol{y})$. Moreover, the minimized free energy $F[q^*]$ is an upper bound for minus log-evidence and in practice is used a model performance criterion. Similarly to (4), the free energy can be decomposed as

$$F[q] = \underbrace{\int q(\boldsymbol{x}) \log p(\boldsymbol{y}|\boldsymbol{x}, m) \mathrm{d}\boldsymbol{x}}_{\text{accuracy}} - \underbrace{\int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{\underbrace{p(\boldsymbol{x}|m)}_{\text{prior}}} \mathrm{d}\boldsymbol{x}}_{\text{complexity}} \tag{23}$$

which underwrites its usage as a performance criterion for model m, given observations $\boldsymbol{y}$.

In an FFG context, the model $p(\boldsymbol{y}, \boldsymbol{x})$ is represented by a set of connected nodes. Consider a generic node of the FFG, given by $f(y, x_1, \ldots, x_n)$ where in the case of VMP, the incoming messages are approximations to the marginals $q_i(x_i)$, $i = 1, \ldots, n$, see Figure 2.

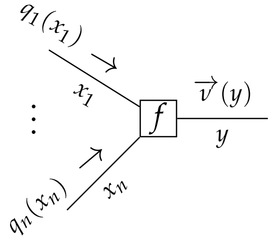


Figure 2. A generic node $f(y, x_1, \ldots, x_n)$ with incoming variational messages $q_i(x_i)$ and outgoing variational message $\overrightarrow{\nu}(y)$, see Equation (24). Note that the marginals $q(\cdot)$ propagate in the graph as messages.

It can be shown that the outgoing VMP message of f towards edge y is given by [32]

$$\overrightarrow{\nu}(y) \propto \exp\left(\int \cdots \int \log f(y, x_1, \ldots, x_n) \prod_{i=1} q(x_i) \mathrm{d}x_i\right). \tag{24}$$

In this paper, we adopt the notational convention to denote belief propagation messages (computed by (20)) by μ and VMP messages (computed by (24)) by ν. The approximate marginal $q(y)$ can be obtained by multiplying incoming and outgoing messages on the edge for y

$$q(y) \propto \overrightarrow{\nu}(y)\overleftarrow{\nu}(y). \tag{25}$$

This process (compute forward and backward messages for an edge and update the marginal) is executed sequentially and repeatedly for all edges in the graph until convergence. In contrast to BP-based inference, the VMP and marginal update rules (24) and (25) lead to closed-form expressions for a large set of conjugate node pairs from the exponential family of distributions. For instance, updating the variance parameter of a Gaussian node with a connected inverse-gamma distribution node results in closed-form VMP updates.

In short, both BP- and VMP-based message passing can be interpreted as minimizing variational free energy, albeit under a different set of local constraints [31]. Typical constraints include factorization and form constraints on the posterior such as $q(\boldsymbol{x}) = \prod_i q_i(x_i)$ and $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$, respectively. Since the constraints are local, BP and VMP can be combined in a factor graph to create hybrid message passing-based variational inference algorithms. For a more detailed explanation of VMP in FFGs, we refer to [32]. Note that hybrid message passing does in general not guarantee to minimize variational free energy [33]. However, in our experiments in Section 6 we will show that iterating our stationary solutions by message passing does lead to free energy minimization.

5. Variational Message Passing for TVAR Models

In this section, we focus on deriving message passing-based inference in the TVAR model. We develop a TVAR composite factor for the FFG framework and specify the intractable BP messages around the TVAR node. Then we present a message passing-based inference solution.

5.1. Message Passing-Based Inference in the TVAR Model

The TVAR model at time step t can be represented by an FFG as shown in Figure 3. We are interested in providing a message passing solution to the inference tasks as specified by Equations (8)–(14). At the left-hand side of Figure 3, the incoming messages are the priors $p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})$ and $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$. At the bottom of the graph, there is a new observation y_t. The goal is to pass messages in the graph to compute posteriors $q(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$ (message ⑯) and $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ (message ⑪). In order to support smoothing algorithms, we also want to be able to pass incoming prior messages from the right-hand side to outgoing messages ⓭ and ⓲ at the left-hand side. Forward and backward messages are drawn as open and closed circles respectively.

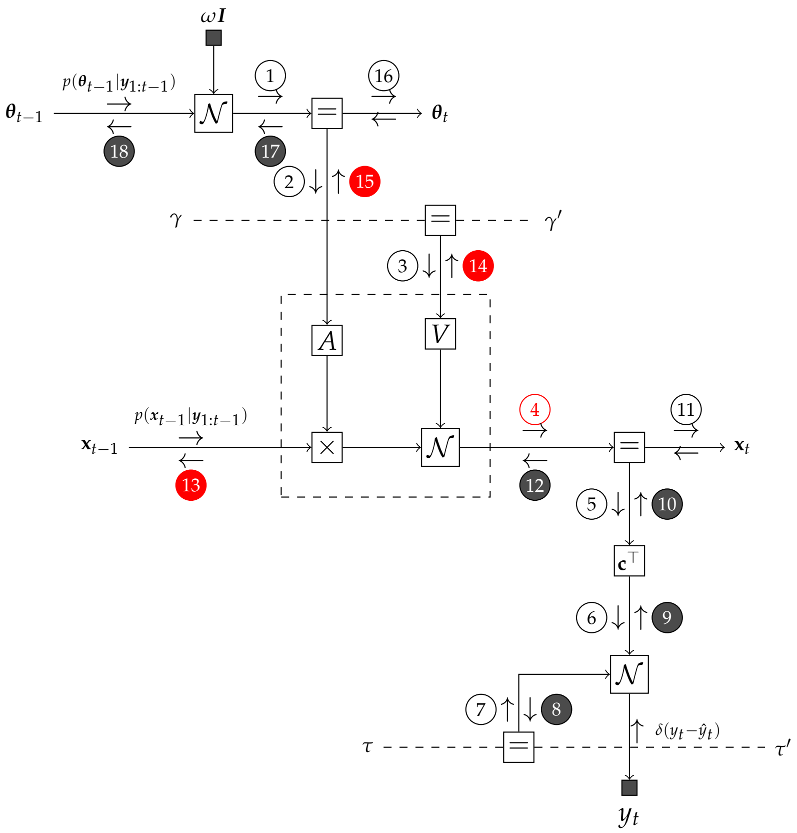


Figure 3. One time segment of an FFG corresponding to the TVAR model. We use small black nodes to denote observations and fixed given parameter values. The observation node for y_t sends a message $\delta(y_t - \hat{y}_t)$ into the graph to indicate that $y_t = \hat{y}_t$ has been observed. Dashed undirected edges denote time-invariant variables. Circled numbers indicate a selected computation schedule. Backward messages are marked by black circles. The intractable messages are labeled with red. The dashed box represents a composite AR node as specified by (30).

Technically, the generative model (2) at time step t for the TVAR model can shortly be written as $p(y_t|z_t)p(z_t|z_{t-1})$, where $z_t = \{x_t, \boldsymbol{\theta}_t, \omega, \gamma, \tau\}$ are the latent variables. On this view, we can write the free energy functional for the TVAR model at time step t as

$$F[q(z_{t-1}, z_t|\boldsymbol{y}_{1:t})] = \iint q(z_{t-1}, z_t|\boldsymbol{y}_{1:t}) \log \frac{\overbrace{q(z_{t-1}, z_t|\boldsymbol{y}_{1:t})}^{\text{posterior}}}{\underbrace{p(y_t|z_t)p(z_t|z_{t-1})}_{\text{generative model}}\,\underbrace{p(z_{t-1}|\boldsymbol{y}_{1:t-1})}_{\text{prior from past}}} \mathrm{d}z_{t-1}\mathrm{d}z_t\,. \tag{26}$$

and minimize $F[q]$ by message passing. In a smoothing context, we would include a prior from the future $p(z_t|\boldsymbol{y}_{t+1:t+T}) := q(z_t|\boldsymbol{y}_{t+1:t+T})$, yielding a free energy functional

$$F[q(z_{t-1}, z_t|\boldsymbol{y}_{1:T})] = \iint q(z_{t-1}, z_t|\boldsymbol{y}_{1:T}) \log \frac{\overbrace{q(z_{t-1}, z_t|\boldsymbol{y}_{1:T})}^{\text{posterior}}}{\underbrace{p(y_t|z_t)p(z_t|z_{t-1})}_{\text{generative model}}\,\underbrace{p(z_{t-1}|\boldsymbol{y}_{1:t-1})}_{\substack{\text{prior}\\ \text{from past}}}\,\underbrace{p(z_t|\boldsymbol{y}_{t+1:t+T})}_{\substack{\text{prior}\\ \text{from future}}}} \mathrm{d}z_{t-1}\mathrm{d}z_t. \tag{27}$$

In a filtering context, $q(\boldsymbol{z}_t|\boldsymbol{y}_{t+1:t+T}) \propto 1$ and the functional (27) simplifies to (26).

5.2. Intractable Messages and the Composite AR Node

The modularity of message passing in FFGs allows us to focus on only the intractable message and marginal updates. For instance, while there is no problem with the analytical computation of the backward message ⓬, the corresponding forward message ④,

$$\overrightarrow{\mu}(\boldsymbol{x}_t) = \int \mathcal{N}\Big(\boldsymbol{x}_t|A(\boldsymbol{\theta}_t)\boldsymbol{x}_{t-1}, V(\gamma)\Big)\underbrace{\overrightarrow{\mu}(\boldsymbol{x}_{t-1})\overrightarrow{\mu}(\boldsymbol{\theta}_t)\overrightarrow{\mu}(\gamma)}_{\text{Gaussian messages}}\mathrm{d}\gamma\mathrm{d}\boldsymbol{\theta}_t\boldsymbol{x}_{t-1} \tag{28}$$

cannot be analytically solved [34]. Similarly, some other messages ⓭, ⓮ and ⓯ do not have a closed-form solution in the constrained free energy minimization framework. For purpose of identification, in Figure 3 intractable messages are marked in red color.

In an FFG framework, we can isolate the problematic part of the TVAR model (Figure 3) by introducing a "composite" AR node. Composite nodes conceal their internal operations from the rest of the graph. As a result, inference can proceed as long as each composite node follows proper message passing communication rules at its interfaces to the rest of the graph. The composite AR node

$$f_{\mathrm{AR}}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{\theta}_t, \gamma) = \mathcal{N}(\boldsymbol{x}_t|A(\boldsymbol{\theta}_t)\boldsymbol{x}_{t-1}, V(\gamma)) \tag{29}$$

is indicated in Figure 3 by a dashed box. Note that the internal shuffling of the parameters $\boldsymbol{\theta}_t$ and γ, respectively by means of $A(\boldsymbol{\theta}_t)$ and $V(\gamma)$, is hidden from the network outside the composite AR node.

5.3. VMP Update Rules for the Composite AR Node

We isolate the composite AR node by the specification

$$f_{\mathrm{AR}}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}, \gamma) = \mathcal{N}(\boldsymbol{y}|A(\boldsymbol{\theta})\boldsymbol{x}, V(\gamma))\,, \tag{30}$$

where, relative to (29), we used substitutions $\boldsymbol{y} = \boldsymbol{x}_t, \boldsymbol{x} = \boldsymbol{x}_{t-1}, \boldsymbol{\theta} = \boldsymbol{\theta}_t$.

Under the structural factorization constraint (See Appendix A.1 for more on structural VMP).

$$q(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}, \gamma) = q(\boldsymbol{y}, \boldsymbol{x})q(\boldsymbol{\theta})q(\gamma)\,, \tag{31}$$

and consistency constraints

$$q(\boldsymbol{y}) = \int q(\boldsymbol{y}, \boldsymbol{x})\mathrm{d}\boldsymbol{x}\,, \quad q(\boldsymbol{x}) = \int q(\boldsymbol{y}, \boldsymbol{x})\mathrm{d}\boldsymbol{y} \tag{32}$$

the marginals $q(\boldsymbol{\theta})$, $q(\boldsymbol{x})$, $q(\boldsymbol{y})$ and $q(\gamma)$ can be obtained from the minimisation of the composite-AR free energy functional

$$F_{\mathrm{AR}}[q] = \int q(\boldsymbol{y}, \boldsymbol{x})q(\boldsymbol{\theta})q(\gamma)\log\frac{\overbrace{q(\boldsymbol{y}, \boldsymbol{x})q(\boldsymbol{\theta})q(\gamma)}^{\text{posterior}}}{\underbrace{f_{\mathrm{AR}}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}, \gamma)}_{\text{AR node}}}\mathrm{d}\boldsymbol{y}\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\theta}\mathrm{d}\gamma\,. \tag{33}$$

Recalling (25), we can write the minimizer of FE functional (33) with respect to $\boldsymbol{\theta}$ as

$$q(\boldsymbol{\theta}) \propto \overrightarrow{\nu}(\boldsymbol{\theta})\overleftarrow{\nu}(\boldsymbol{\theta}) \tag{34}$$

where $q(\boldsymbol{\theta})$ is associated with the incoming message to AR node and $\overrightarrow{\nu}(\boldsymbol{\theta})$ is a variational outgoing message. Hence, the outgoing message from the AR node toward $\boldsymbol{\theta}$ can be written as

$$\overrightarrow{\nu}(\boldsymbol{\theta}) \propto \exp\Big(\mathbb{E}_{q(\boldsymbol{y},\boldsymbol{x})q(\boldsymbol{\theta}_t)q(\gamma)} \log\Big[\mathcal{N}\Big(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x},\boldsymbol{V}\Big)\Big]\Big) \tag{35}$$

In Appendix A we work out a closed-form solution for this and all other update rules plus an evaluation of free energy for the composite AR node. The results are reported in Table 1. With these rules in hand, the composite AR node can be plugged into any factor graph and take part in a message passing-based free energy minimization process.

6. Experiments

In this section, we first verify the proposed methodology by a simulation of the proposed TVAR model on synthetic data, followed by validation experiments on two real world problems. We implemented all derived message passing rules in the open source Julia package `ForneyLab.jl` [28]. The code for the experiments and for the AR node can be found in public Github repositories. (https://github.com/biaslab/TVAR_FFG, accessed on 27 May 2021, https://github.com/biaslab/LAR, accessed on 27 May 2021) We used the following computer configuration to run the experiments. *Operation system*: macOS Big Sur, *Processor*: 2,7 GHz Quad-Core Intel Core *i*7, *RAM*: 16 GB.

6.1. Verification on a Synthetic Data Set

To verify the proposed TVAR inference methods, we synthesized data from two generative models m_1 and m_2, as follows:

$$\boldsymbol{\theta}_t \sim \begin{cases} \delta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) & \text{if } m = m_1 \\ \mathcal{N}(\boldsymbol{\theta}_{t-1}, \omega \mathrm{I}_M) & \text{if } m = m_2 \end{cases} \tag{36a}$$

$$\boldsymbol{x}_t \sim \mathcal{N}\Big(A(\boldsymbol{\theta}_t)\boldsymbol{x}_{t-1}, V(\gamma)\Big) \tag{36b}$$

$$y_t \sim \mathcal{N}(\boldsymbol{c}^T\boldsymbol{x}_t, \tau) \tag{36c}$$

with priors

$$p(M = k) = \prod_{k=1}^{10} 0.1^{[M=k]} \tag{37a}$$

$$\boldsymbol{\theta}_0 \sim \begin{cases} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) & \text{if } m = m_1 \\ \mathcal{N}(\boldsymbol{0}, 1e12\boldsymbol{I}) & \text{if } m = m_2 \end{cases} \tag{37b}$$

$$\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, 1e12\boldsymbol{I}) \tag{37c}$$

$$\gamma \sim \Gamma(1.0, 1e-5) \tag{37d}$$

$$\tau = 1.0 \tag{37e}$$

$$\omega = 0.01 \tag{37f}$$

where M is the number of AR coefficients. Although these models differ only with respect to the properties of the AR coefficients $\boldsymbol{\theta}$, this variation has an important influence on the data generative process. The first model m_1 specifies a stationary AR process, since $\delta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$ in (36a) indicates that θ is not time-varying in m_1. The second model m_2 represents a proper TVAR process as the prior evolution of the AR coefficients follows a random walk. One-time segment FFGs corresponding to the Equation (36) are depicted in Figure 4.

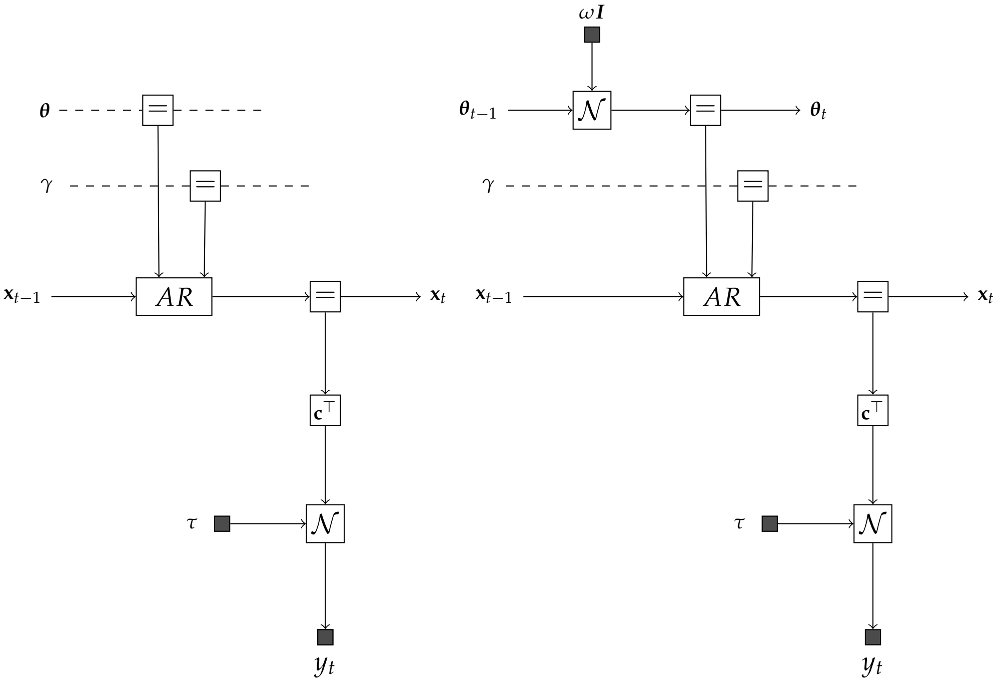


Figure 4. Forney-style Factor Graphs corresponding to Equation (36). (**Left**) model m_1. (**Right**) model m_2.

For each model, we generated a data set of 100 different time series, each of length 100 (so we have 2 × 100 × 100 data points). For each time series, as indicated by (37a), the AR order M of the generative process was randomly drawn from the set $\{1, 2, \ldots, 10\}$. We used rather non-informative/broad priors for states and parameters for both models, see (37). This was done to ensure that the effect of the prior distributions is negligible relative to the information in the data set.

These time series were used in the following experiments. We selected two recognition models m_1 and m_2 with the same specifications as were used for generating the data set. The recognition models were trained on time series that were generated by models with the same AR order.

We proceeded by computing the quantities $q(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$, $q(\boldsymbol{\theta}_{1:T}|\boldsymbol{y}_{1:T})$, $q(\gamma|\boldsymbol{y}_{1:T})$ and $F[q(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t|\boldsymbol{y}_{1:T})]$ (where $\boldsymbol{z}$ comprises all latent states and parameters) for both models, following the proposed rules from Table 1.

As a verification check, we first want to ensure that inference recovers the hidden states x_t for each $t \in (1, 2, \ldots, 100)$. Secondly, we want to verify the convergence of FE. As we have not used any approximations along the derivations of variational messages, we expect a smoothly decreasing curve for FE until convergence. The results of the verification stage are highlighted for a typical case in Figure 5. The figure confirms that states x_t are accurately tracked and that a sliding average of the AR coefficients $\boldsymbol{\theta}_t$ is also nicely tracked. Figure 5 also indicates that the FE uniformly decreases towards lower values as we spend more computational power.

We note that the FE score by itself does not explain whether the model is good or not, but it serves as a good measure for model comparison. In the following subsection, we demonstrate how FE scores can be used for model selection.

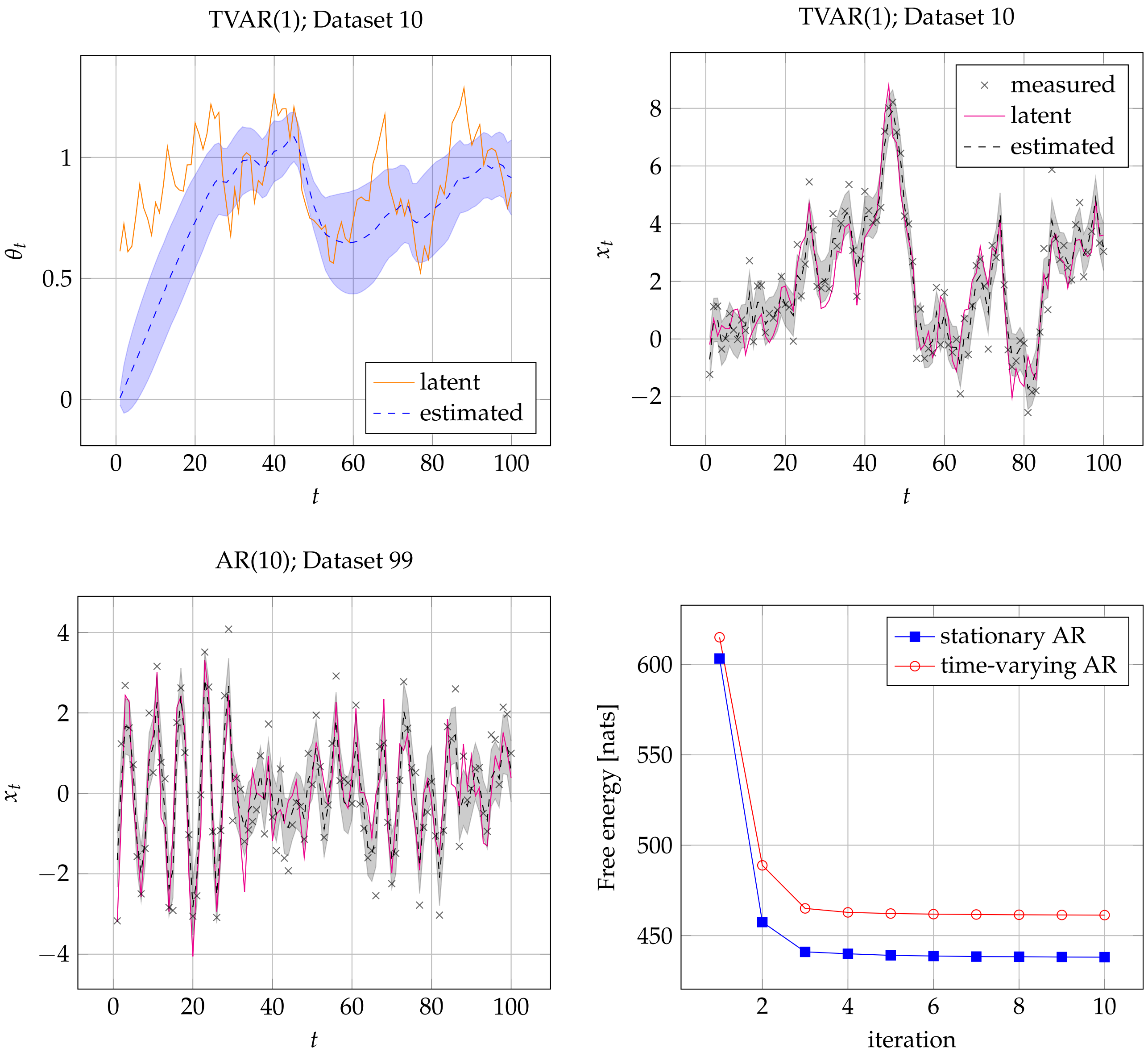


Figure 5. Verification results. The solid line corresponds to the value of the latent (hidden) states in the generative processes. The dashed line corresponds to the expected mean value of the posterior estimates of hidden states $q(\cdot|\boldsymbol{y}_{1:100})$ in the recognition models. The shadowed regions corresponds to one standard deviation of the posteriors in the recognition models below and above the estimated mean. The top two plots show inference results for the coefficients $\boldsymbol{\theta}_t$ (top-left) and states x_t (top-right) of TVAR(1) (model m_2, AR order $M = 1$) for time series ♯10. (bottom-left) State trajectory $q(x_t|\boldsymbol{y}_{1:100})$ model m_1, AR order $M = 1$ on time series ♯99. (Bottom-right) Evolution of FE for m_1 (AR) and m_2 (TVAR), averaged over their corresponding time series. The iteration number at the abscissa steps through a single marginal update for all edges in the graph.

6.2. Temperature Modeling

AR models are well-known for predicting different weather conditions such as wind, temperature, precipitation, etc. Here, we will revisit the problem of modeling daily temperature. We used a data set of daily minimum temperatures (in °C) in Melbourne, Australia, 1981–1990 (3287 days) (https://www.kaggle.com/paulbrabban/daily-minimum-temperatures-in-melbourne, accessed on 27 May 2021). We then corrupted the data set by adding random noise sampled from $\mathcal{N}(0, 10.0)$ to the actual temperatures. A fragment of the time-series is depicted in Figure 6.

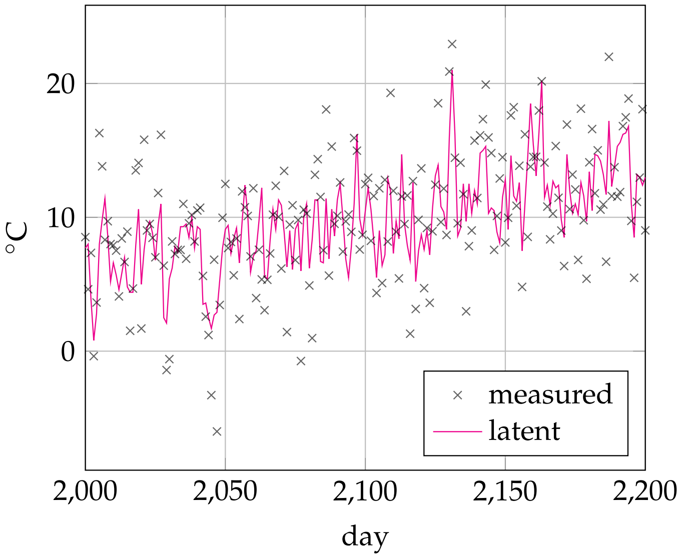


Figure 6. Temperature time-series from days 2000 to 2200. Crosses denote the thermometer readings plus added noise. The solid line corresponds to the latent (hidden) daily temperature.

To estimate the actual temperature based on past noisy observations by computing $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, we use a TVAR model with measurement noise to simulate uncertainty about corrupted observations. The model is specified by the following equation set

$$\boldsymbol{\theta}_t \sim \mathcal{N}(\boldsymbol{\theta}_{t-1}, \mathrm{I}_M) \tag{38a}$$

$$\boldsymbol{x}_t \sim \mathcal{N}\Big(A(\boldsymbol{\theta}_t)\boldsymbol{x}_{t-1} + \boldsymbol{c}\eta, V(\gamma)\Big) \tag{38b}$$

$$y_t \sim \mathcal{N}(\boldsymbol{c}^\intercal \boldsymbol{x}_t, \tau) \tag{38c}$$

with priors

$$\boldsymbol{\theta}_0 \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I}) \qquad \boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \qquad \eta \sim \mathcal{N}(0.0, 10.0) \tag{39a}$$

$$\gamma \sim \Gamma(1.0, 1.0) \qquad \tau \sim \Gamma(0.1, 1.0) \tag{39b}$$

Since the temperature data is not centered around 0 °C, we added a bias term η to the state $\boldsymbol{x}_t$. The corresponding FFG is depicted in Figure 7.

Note that we put a Gamma prior on the measurement noise precision τ, meaning that we are uncertain about the size of the error of the thermometer reading. The inference task for the model is computing $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, in other words, we track the states based only on past data. Of course, after training, we could use the model for temperature prediction by tracking $q(\boldsymbol{x}_{t+k}|\boldsymbol{y}_{1:t})$ for $k \geq 1$. We compare the performance of four TVAR models with AR orders $M = \{1, 2, 3, 4\}$. To choose the best model, we computed the average FE score for each TVAR(M) model.

Figure 8 shows that on average TVAR(3) outperforms its competitors. The complexity vs accuracy decomposition (23) of FE explains why a lower order model may outperform higher order models. TVAR(4) maybe as accurate or more accurate than TVAR(3) but the increase in accuracy is more than offset by the increase in complexity. For the lower order models, it is the other way around: they are less complex and involve fewer computations than TVAR(3), but the loss in model complexity leads to too much loss in data modeling accuracy. Overall, TVAR(3) is the best model for this data set. Practically, we always favor the model that features the lowest FE score. In the next subsection we will use this technique (scoring FE) for online model selection.

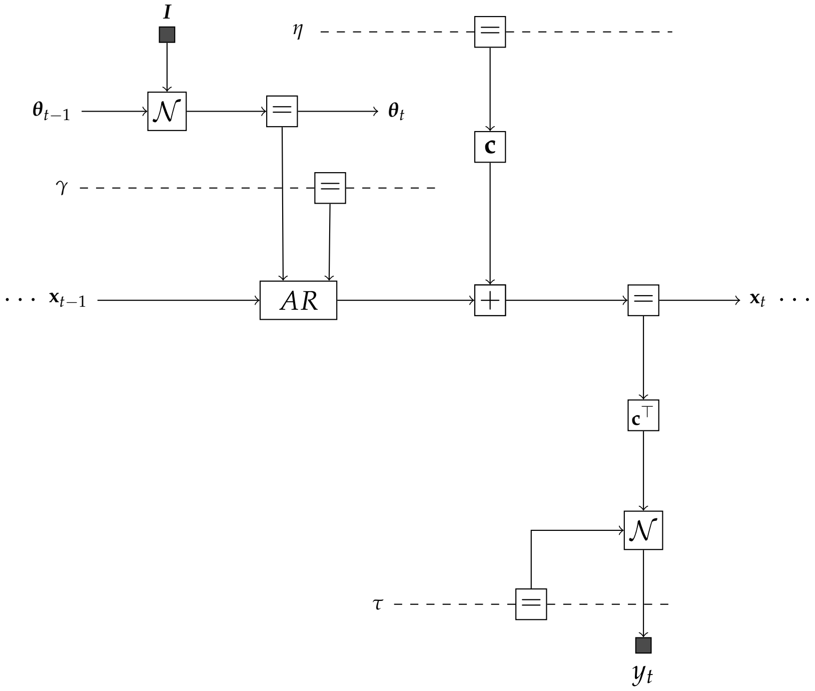


Figure 7. One time segment of a Forney-style factor graph (FFG) for the TVAR model in the temperature modeling task (38).

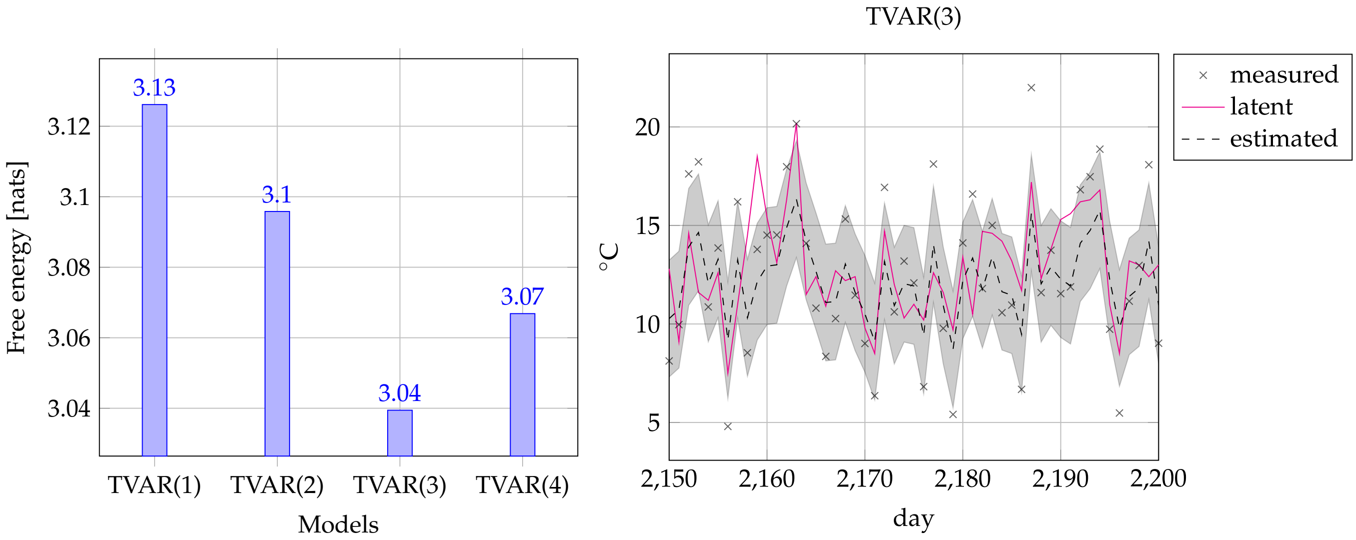


Figure 8. (**Left**) Comparison of four TVAR(M) models for the temperature filtering problem. Bars correspond to the averaged (over 3287 days) FE score for each model. (**Right**) Inference example of the best performing model (TVAR(3)). Crosses denote the thermometer reading plus added noise. The solid line corresponds to the latent (hidden) daily temperature. The dashed line corresponds to the mean of the posterior estimates of hidden temperature and the shadowed region corresponds to one standard deviation below and above the estimated temperature.

6.3. Single-Channel Speech Enhancement

Single-channel speech enhancement (SCSE) is a well-known challenging task that aims to enhance noisy speech signals that were recorded by a single microphone. In single microphone recordings, we cannot use any spatial information that is commonly used in beamforming applications. Much work has been done to solve the SCSE task, ranging from Wiener filter-inspired signal processing techniques [35,36] to deep learning neural networks [37]. In this paper, we use data from the speech corpus (NOIZEUS) (https://ecs.utdallas.edu/loizou/speech/noizeus/, accessed on 27 May 2021) [38] and corrupted clean speech signals with white Gaussian noise, leading to a signal-to-noise ratio (SNR)

$$\mathrm{SNR}(\boldsymbol{s}_{1:T}, \boldsymbol{y}_{1:T}) = 10 \log_{10}\left[\frac{\sum_t^T s_t^2}{\sum_t^T (s_t - y_t)^2}\right] \approx 13.36\,\mathrm{dB} \tag{40}$$

where $\boldsymbol{s}_{1:T} = (s_1, \ldots, s_T)$ and $\boldsymbol{y}_{1:T} = (y_1, \ldots, y_T)$ are clean and corrupted speech signals. s_t is a speech signal at time t and T is the length of the signal.

Historically, AR models have shown to perform well for modeling speech signals in the time (waveform) domain [39,40]. Despite the fact that speech is a highly nonstationary signal, we may assume it to be stationary within short time intervals (frames) of about 10 [ms] each [41]. Since voiced, unvoiced and silence frames have very different characteristics, we used 5 different models (a random walk model (RW), AR(1), AR(2), TVAR(1) and TVAR(2)) for each frame of 10 [ms] with 2.5 [ms] overlap. Given a sampling frequency of 8 [kHZ], each frame results into 80 samples with 20 samples overlap. The AR and TVAR models were specified by Equation (36). For each frame, we evaluated the model performance by minimized FE and selected the model with minimal FE score. We used identical prior parameters for all models where possible. To recover the speech signal we computed the mean values of $q(\boldsymbol{x}_t|y_{1:T})$ of the selected model for each frame. The SNR gain of this SCSE system was

$$\mathrm{SNR}(\boldsymbol{s}_{1:T}, \boldsymbol{x}_{1:T}) - \mathrm{SNR}(\boldsymbol{s}_{1:T}, \boldsymbol{y}_{1:T}) \approx 3.7\,\mathrm{dB}. \tag{41}$$

Figure 9 show the spectrograms of the clean, noisy and filtered signal respectively.

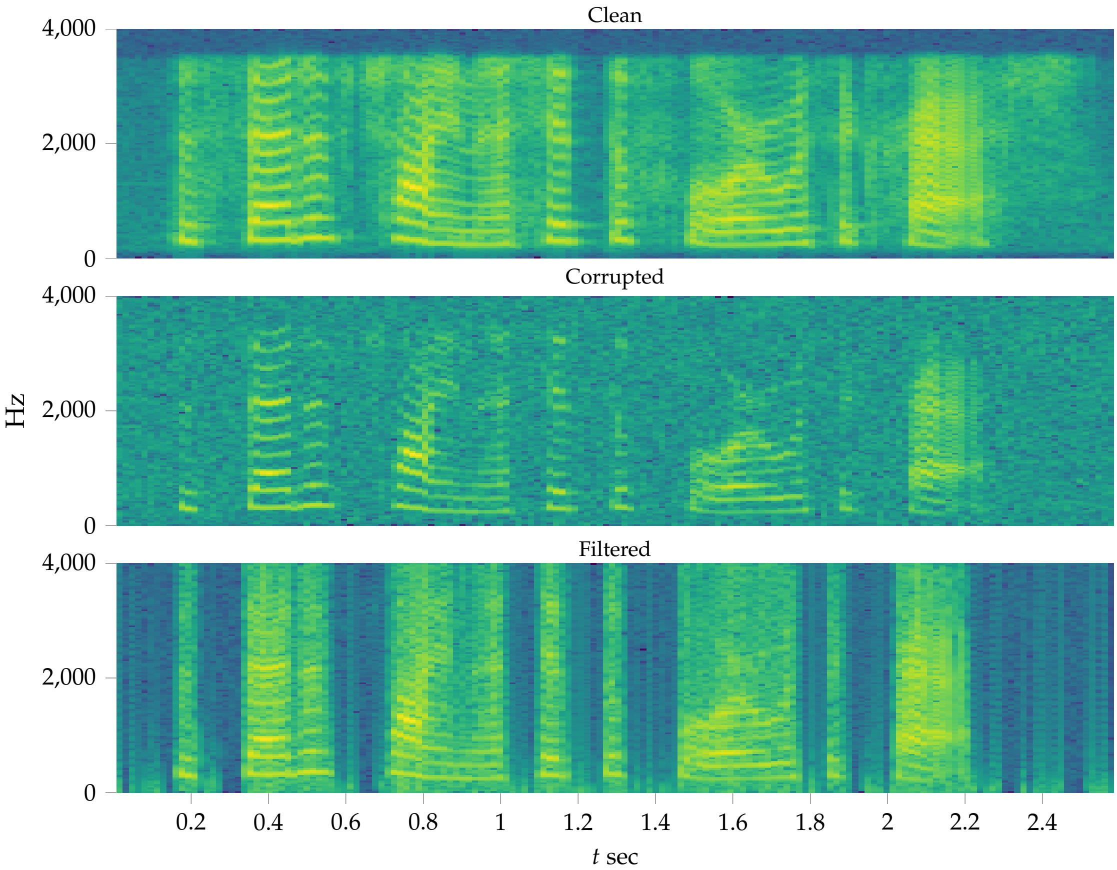


Figure 9. Spectrogram of recovered speech signal in the experiment of Section 6.3.

Next, we analyze the inference results in a bit more detail. Table 2 shows the percentage of winning models for each frame based on the free energy score.

Table 2. Percentage of preferred models (based on FE scores) for all frames on the speech enhancement task.

	RW	AR(1)	AR(2)	TVAR(1)	TVAR(2)
Ratio	32.2%	54.3%	10.7%	1.2%	0.5%

As we can see, for more than 30% of all frames, the random walk model performs best. This happens mostly because for a silent frame the AR model gets penalized by its complexity term. We recognize that in about 90% of the frames the best models are AR(1) and RW. On the other hand, for the frames where the speech signal transitions from silent or unvoiced to voiced, these fixed models start to fail and the time-varying AR models perform better. This effect can be seen in Figure 10.

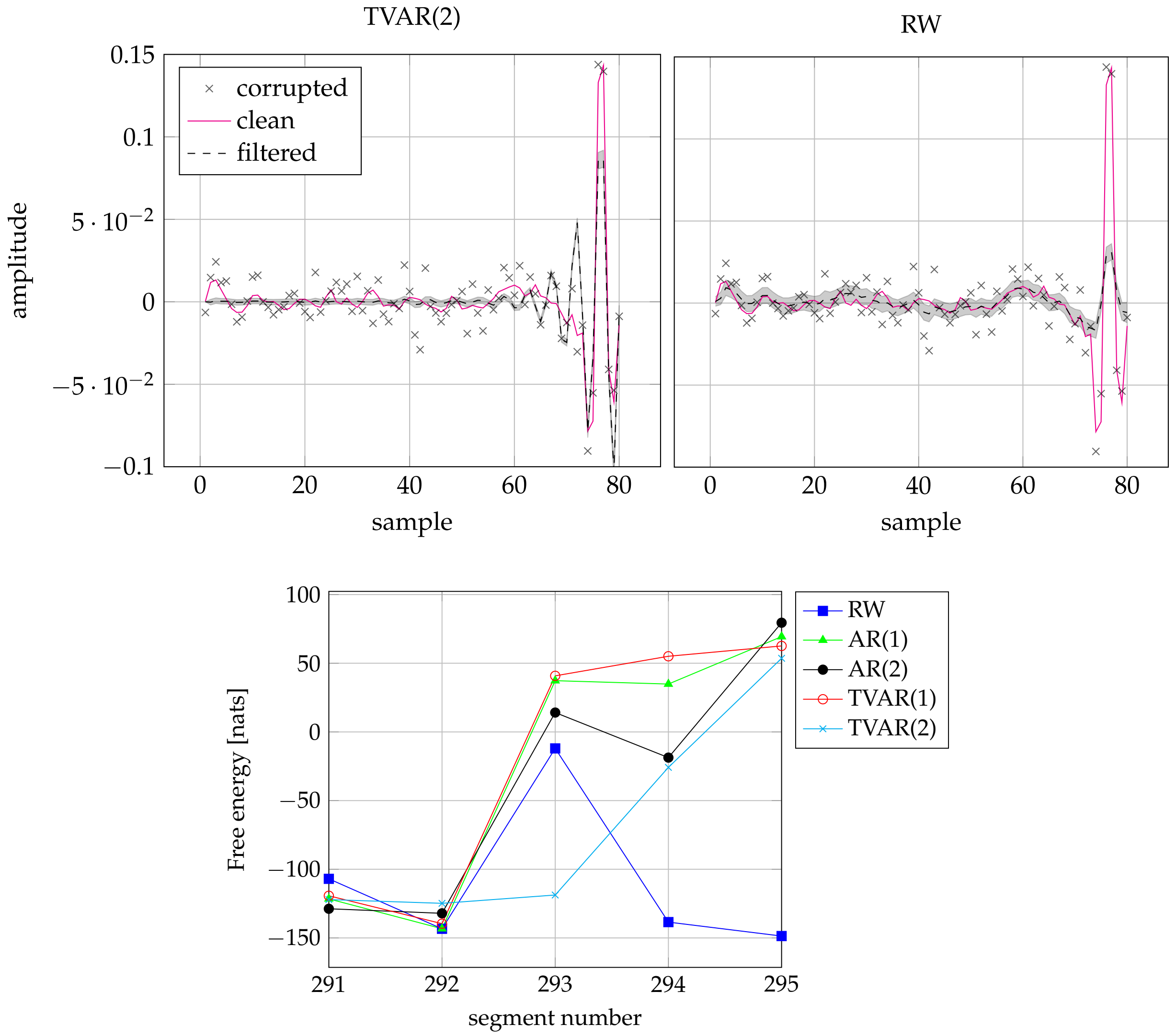


Figure 10. (Top) (**Top-left**) Inference by TVAR(2) for the segment 293. (**Top-right**) Inference by RW for the segment 293. Note how the TVAR model is able to follow the transitions at the end of the frame, while the RW cannot adapt within one frame. (**Bottom**) FE scores from segment 291 to 295. TVAR(2) wins frame 293 as it has the lowest FE score.

Figure 11 shows the performance of the AR(2) and RW models on a frame with a voiced speech signal. For this case, the AR(2) model performs better.

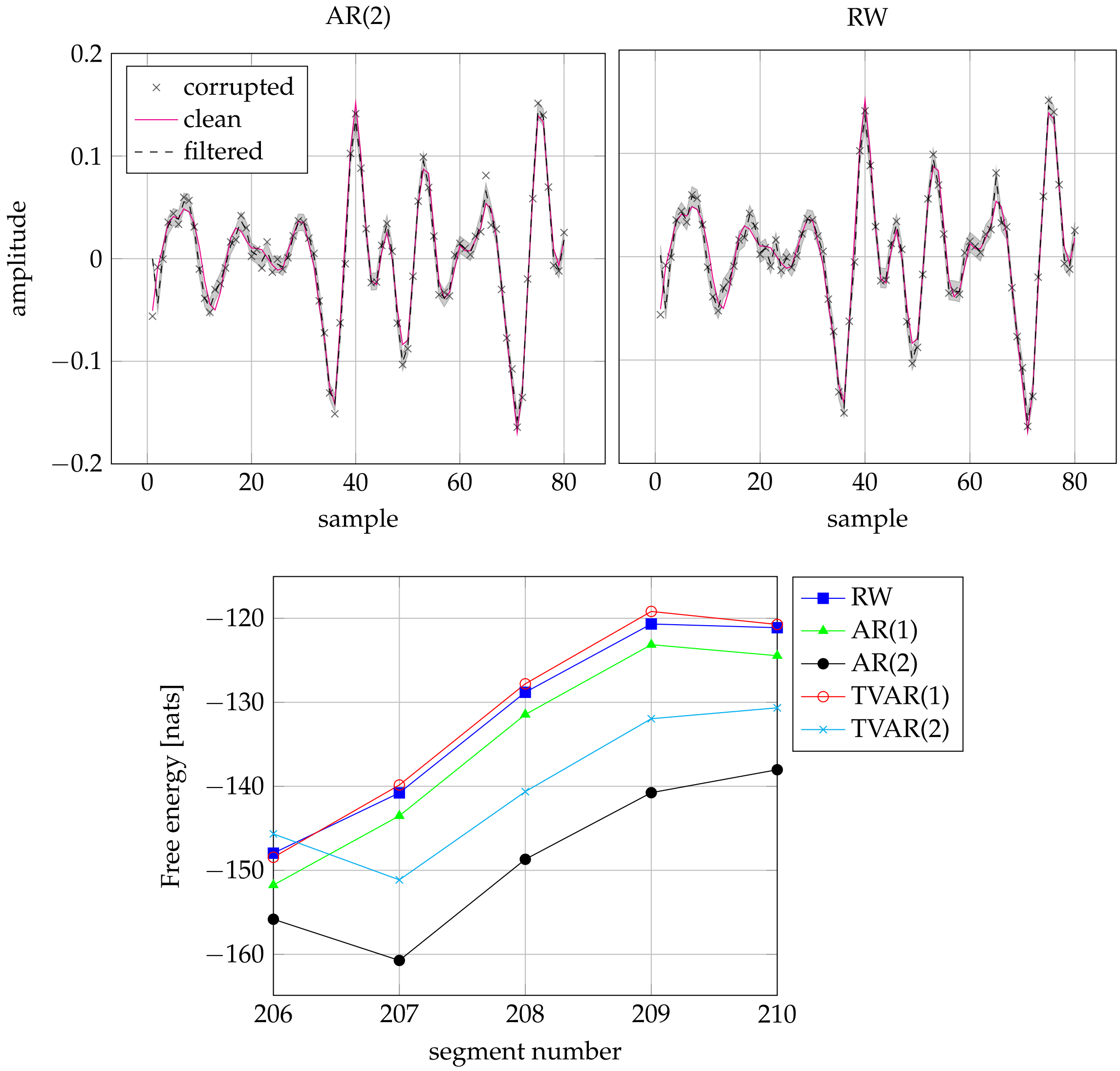


Figure 11. Comparison of AR(2) and RW models for a voiced signal frame. (**Top-left**) Inference by AR(2) for the segment 208. (**Top-right**) Inference by RW for the segment 208. (**Bottom**) FE scores from segment 206 to 210. The AR(2) model wins frame 208.

Finally, Figure 12 shows how the TVAR(2) model compares to the RW model on one of the unvoiced/silence frames. While the estimates of TVAR(2) appear to be more accurate, it pays a bigger "price" for the model complexity term in the FE score and the RW model wins the frame.

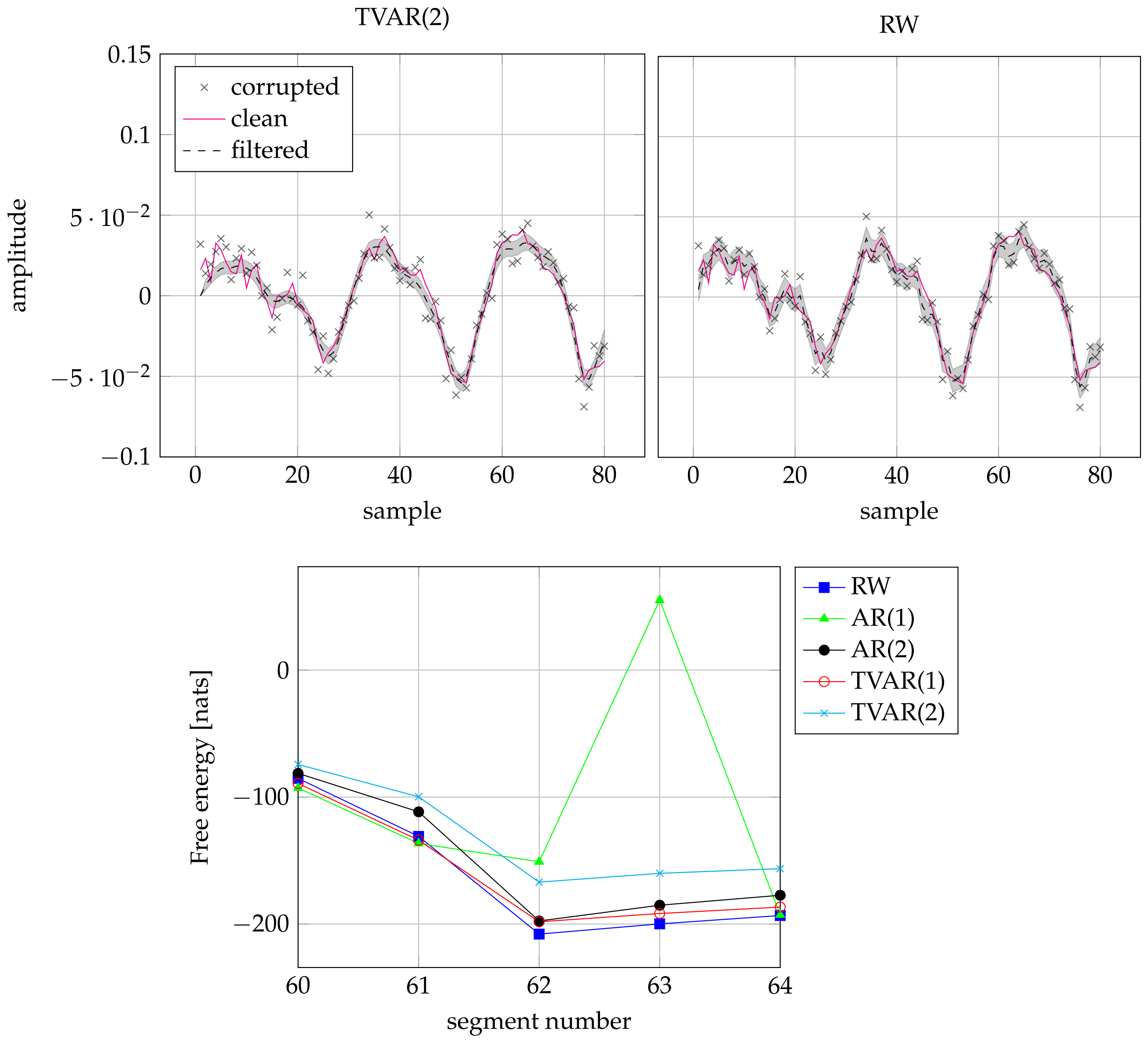


Figure 12. Comparison of TVAR(2) and RW models for an unvoiced/silence frame. (**Top-left**) Inference by TVAR(2) for the frame 62. (**Top-right**) Inference by RW for the frame 62. **(Bottom)** FE scores from segment 60 to 64. The RW model scores best on frame 62 due to its low complexity.

7. Discussion

We have introduced a TVAR model that includes efficient joint variational Bayesian tracking of states, parameters and free energy. The system can be used as a plug-in module in factor graph-based representations of other models. At several points in this paper, we have made some design decisions that we shortly review here.

While FE computation for the AR node provides a convenient performance criterion for model selection, we noticed in the speech enhancement simulation that separate FE tracking for each candidate model leads to a large computational overhead. There are ways to improve the model selection process that we used in the speech enhancement simulation. One way is to consider a mixture model of candidate models and track the posterior over the mixture coefficients [42]. Alternatively, a very cheap method for online Bayesian model selection may be the recently developed Bayesian Model Reduction (BMR) method [43]. The BMR method is based on a generalization of the Savage-Dickey Density Ratio and supports tracking of free energy of multiple nested models with almost no computational overhead. Both methods seem to integrate well with a factor graph representation and we plan to study this issue in future work.

In this paper, the posterior factorization (31) supports the modeling of temporal dependencies between input and output of the AR node in the posterior. Technically, (31) corresponds to a structural VMP assumption, in contrast to the more constrained mean-field VMP algorithm that would be based on $q(\boldsymbol{z}) = \prod_i q_i(z_i)$, where $\boldsymbol{z}$ is the set of all latent variables [44]. We could have also worked out alternative update rules for the assumption of a joint factorization of precision γ and AR coefficients $\boldsymbol{\theta}$. In that case, the prior (incoming message $\overrightarrow{\nu}(\boldsymbol{\theta}, \gamma)$ to AR node) would be in the form of a Normal-Gamma distribution. While any of these these assumptions are technically valid, each choice accepts a different trade-off in the accuracy vs. complexity space. We review structural VMP in Appendix A.1.

In the temperature modelling task, we added some additional random variables (bias, measurement noise precision). To avoid identifiability issues, in (38a) we fixed the covariance matrix of the time-varying AR coefficient to the identity matrix. In principle, this constraint can be relaxed. For example, an Inverse-Wishart prior distribution can be added to the covariance matrix.

In our speech enhancement experiments in Section 6.3, we assume that the measurement noise variance is known. In a real-world scenario, this information is usually not accessible. However, online tracking of measurement noise or other (hyper-)parameters is usually not a difficult extension when the process is simulated in a factor graph toolbox such as ForneyLab [28]. If so desired, we could add a prior on the measurement noise variance and track the posterior. The online free energy criterion (23) can be used to determine if the additional computational load (complexity) of Bayesian tracking of the variance parameter has been compensated by the increase in modeling accuracy.

The realization of the TVAR model in ForneyLab comes with some limitations. For large smoothing problems (say, >1000 data points), the computational load of message passing in ForneyLab becomes too heavy for a standard laptop (as was used in the paper). Consequently, in the current implementation it is difficult to employ the AR node for processing large time series on a standard laptop. To circumvent this issue, when using ForneyLab, one can combine filtering and smoothing solutions into a batch learning procedure. In future work we plan to remedy this issue by some ForneyLab refactoring work. Additionally, the implemented AR node does not provide a closed-form update rule for the marginal distribution when the probability distribution types of the incoming messages (priors) are different from the ones used in our work. Fortunately, ForneyLab supports resorting to (slower) sampling-based update rules when closed-form update rules are not available.

8. Conclusions

We presented a variational message passing approach to tracking states and parameters in latent TVAR models. The required update rules have been summarized and implemented in the factor graph package ForneyLab.jl, thus making transparent usage of TVAR factors available in freely definable stochastic dynamical systems. Aside from VMP update rules, we derived a closed-form expression for the variational free energy (FE) of an AR factor. Free Energy can be used as a proxy for Bayesian model evidence and as such allows for model performance comparisons between the TVAR models and alternative structures. Owing to the locality and modularity of the FFG framework, we demonstrated how AR nodes can be applied as plug-in modules in various dynamic models. We verified the correctness of the rules on a synthetic data set and applied the proposed TVAR model to a few relatively simple but different real-world problems. In future work, we plan to extend the current factor graph-based framework to efficient and transparent tracking of AR model order and to online model comparison and selection with alternative models.

Author Contributions: Conceptualization, A.P., W.M.K. and B.d.V.; methodology, A.P., W.M.K.; software, A.P.; validation, A.P.; formal analysis, A.P., W.M.K.; investigation, A.P., W.M.K. and B.d.V.; resources, A.P., W.M.K. and B.d.V.; data curation, A.P., W.M.K. and B.d.V.; writing—original draft preparation, A.P., W.M.K.; writing—review and editing, W.M.K. and B.d.V.; visualization, A.P., W.M.K.

and B.d.V.; supervision, W.M.K. and B.d.V.; project administration, B.d.V.; funding acquisition, B.d.V. All authors have read and agreed to the published version of the manuscript.

Funding: 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).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the BIASlab team members, in particular Thijs van de Laar, Ismail Senoz and Semih Akbayrak for insightful discussions on various topics related to this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivations

Figure A1 represents a composite AR node.

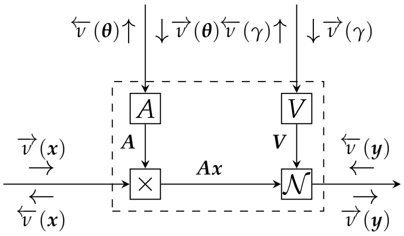

$\overleftarrow{\nu}(\boldsymbol{\theta})\uparrow$
$\downarrow\overrightarrow{\nu}(\boldsymbol{\theta})$
$\overleftarrow{\nu}(\gamma)\uparrow$
$\downarrow\overrightarrow{\nu}(\gamma)$
A
V
$\boldsymbol{A}$
$\boldsymbol{V}$
$\boldsymbol{Ax}$
$\times$
$\mathcal{N}$
$\overrightarrow{\nu}(\boldsymbol{x})$
$\overleftarrow{\nu}(\boldsymbol{x})$
$\overleftarrow{\nu}(\boldsymbol{y})$
$\overrightarrow{\nu}(\boldsymbol{y})$

Figure A1. Autoregressive (AR) node.

The corresponding node function of Figure A1 $f(\boldsymbol{y}\ \boldsymbol{x}, \boldsymbol{\theta}, \gamma)$:

$$f(\boldsymbol{y}\ \boldsymbol{x}, \boldsymbol{\theta}, \gamma) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{Ax}, \boldsymbol{V})$$

where

$$\boldsymbol{A} = A(\boldsymbol{\theta}) \qquad \boldsymbol{V} = V(\gamma) = \begin{bmatrix} \gamma^{-1} & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \vdots \\ 0 & 0 & 0 & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \end{bmatrix}.$$

Appendix A.1. Structural Variational Message Passing

The message update rule (24) implies a mean-field factorization, meaning that all variables represented by edges around the factor node f are independent. In this paper, we impose a structural dependence between states. To illustrate how structured VMP works, let us consider the example depicted in Figure A2.

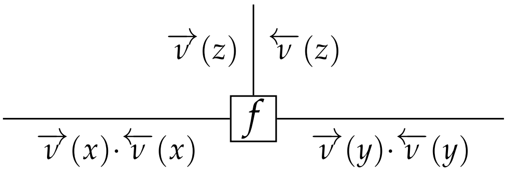

$\overrightarrow{\nu}(z)$
$\overleftarrow{\nu}(z)$
f
$\overrightarrow{\nu}(x) \cdot \overleftarrow{\nu}(x)$
$\overrightarrow{\nu}(y) \cdot \overleftarrow{\nu}(y)$

Figure A2. A node $f(x, y, z)$ representing an arbitrary joint distribution. Arrows above the messages $\nu(\cdot)$ indicate the direction (incoming or outgoing).

Suppose that we constrain the joint posterior (A1) as

$$q(x,y,z) = q(x,y)q(z) \tag{A1}$$

The message passing algorithm for updating the marginal posteriors $q^*(x,y)$ and $q^*(z)$ can now be executed as follows:

(1) compute outgoing messages $\overrightarrow{\nu}(y)$, $\overleftarrow{\nu}(x)$:

$$\overrightarrow{\nu}(y) \propto \int \overrightarrow{\nu}(x) \exp\left(\int q(z) \log[f(x,y,z)]\mathrm{d}z\right)\mathrm{d}x \tag{A2a}$$

$$\overleftarrow{\nu}(x) \propto \int \overleftarrow{\nu}(y) \exp\left(\int q(z) \log[f(x,y,z)]\mathrm{d}z\right)\mathrm{d}y \tag{A2b}$$

(2) update joint posterior $q^*(x,y)$:

$$q^*(x,y) \propto \overrightarrow{\nu}(x) \exp\left(\int q(z) \log f(x,y,z)\mathrm{d}z\right)\overleftarrow{\nu}(y) \tag{A3}$$

(3) compute the outgoing message $\overrightarrow{\nu}(z)$:

$$\overrightarrow{\nu}(z) \propto \exp\left(\int q^*(x,y) \log f(x,y,z)\mathrm{d}x\mathrm{d}y\right), \tag{A4}$$

(4) update posterior $q^*(z)$:

$$q^*(z) \propto \overrightarrow{\nu}(z)\overleftarrow{\nu}(z) \tag{A5}$$

Every marginal update rule (Equations (25), (A3) and (A5)) corresponds to a coordinate descent step on the variational free energy, and therefore the free energy is guaranteed to converge to a local minimum.

Appendix A.2. Auxiliary Node Function

Before obtaining the update messages for TVAR we need to evaluate the auxiliary node function $\tilde{f}(\boldsymbol{x},\boldsymbol{y}) \propto \exp\left\{\mathbb{E}_{q(\gamma)q(\boldsymbol{\theta})} \log[f(\boldsymbol{y}\ \boldsymbol{x},\boldsymbol{\theta},\gamma)]\right\}$. We also need to address the issue of invertability of the covariance matrix V. To tackle this problem, we assume $\epsilon > 0, \epsilon^2 \approx 0$ which allows us to introduce matrix $\boldsymbol{W} = \boldsymbol{V}^{-1}$ ($\boldsymbol{W}^{-1}\boldsymbol{V} = \boldsymbol{V}^{-1}\boldsymbol{W} = I$).

$$\boldsymbol{V} = \begin{bmatrix} \gamma^{-1} & 0 & 0 & \dots & 0 \\ 0 & \epsilon & 0 & \dots & \vdots \\ 0 & 0 & \epsilon & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \end{bmatrix}$$

$$\begin{aligned}
\log\tilde{f}(\boldsymbol{x},\boldsymbol{y}) &= \mathbb{E}_{q(\gamma)q(\boldsymbol{\theta})} \log f(\boldsymbol{y}\ \boldsymbol{x},\boldsymbol{\theta},\gamma) + const \\
&= \frac{1}{2}\mathbb{E}_{q(\gamma)}[\log|\boldsymbol{W}|] - \frac{1}{2}\mathbb{E}_{q(\gamma)q(\boldsymbol{\theta})}\left[(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^\top\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})\right] + const \\
&= -\frac{1}{2}\mathbb{E}_{q(\gamma)q(\boldsymbol{\theta})}\left[\mathrm{tr}\left(\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^\top\right)\right] + const \\
&= -\frac{1}{2}\mathrm{tr}\left(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{\theta})}\left[(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^\top\right]\right) + const \\
&= -\frac{1}{2}\mathrm{tr}\left(m_{\boldsymbol{W}}\left(\boldsymbol{y}\boldsymbol{y}^\top - m_{\boldsymbol{A}}\boldsymbol{x}\boldsymbol{y}^\top - \boldsymbol{y}\boldsymbol{x}^\top m_{\boldsymbol{A}} + \mathbb{E}_{q(\boldsymbol{\theta})}\left[\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{A}^\top\right]\right)\right) \\
&\quad + const
\end{aligned}$$

We work out the expectation term inside the trace separately. To do this, we notice, that the product $\boldsymbol{Ax}$ can be separated in the shifting operator $\boldsymbol{Sx}$ and the inner vector product $\boldsymbol{cx}^\top\boldsymbol{\theta}$ in the following way:

$$\boldsymbol{Ax} = \boldsymbol{Sx} + \boldsymbol{cx}^\top\boldsymbol{\theta} = \underbrace{(\boldsymbol{S} + \boldsymbol{c\theta}^\top)}_{\boldsymbol{Ax}} \tag{A6}$$

where

$$\boldsymbol{S} = \begin{bmatrix} & \boldsymbol{0}^\top & \\ \boldsymbol{I}_{M-1} & & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{c} = (1, 0, \dots, 0)^\top$$

$$\begin{aligned}
\mathbb{E}_{q(\boldsymbol{\theta})}\left[\boldsymbol{Axx}^\top\boldsymbol{A}^\top\right] &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\left(\boldsymbol{Sx} + \boldsymbol{cx}^\top\boldsymbol{\theta}\right)\left(\boldsymbol{Sx} + \boldsymbol{cx}^\top\boldsymbol{\theta}\right)^\top\right] \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\boldsymbol{Sx}(\boldsymbol{Sx})^\top + \boldsymbol{cx}^\top\boldsymbol{\theta}(\boldsymbol{Sx})^\top + \boldsymbol{Sx}(\boldsymbol{cx}^\top\boldsymbol{\theta})^\top + \boldsymbol{cx}^\top\boldsymbol{\theta\theta}^\top\boldsymbol{xc}^\top\right] \\
&= \boldsymbol{Sx}(\boldsymbol{Sx})^\top + \boldsymbol{cx}^\top m_{\boldsymbol{\theta}}(\boldsymbol{Sx})^\top + \boldsymbol{Sx}(\boldsymbol{cx}^\top m_{\boldsymbol{\theta}})^\top + \boldsymbol{cx}^\top\left[V_{\boldsymbol{\theta}} + m_{\boldsymbol{\theta}}m_{\boldsymbol{\theta}}^\top\right]\boldsymbol{xc}^\top \\
&= \left(\boldsymbol{Sx} + \boldsymbol{cx}^\top m_{\boldsymbol{\theta}}\right)\left(\boldsymbol{Sx} + \boldsymbol{cx}^\top m_{\boldsymbol{\theta}}\right)^\top + \boldsymbol{cx}^\top V_{\boldsymbol{\theta}}\boldsymbol{xc}^\top \\
&= m_{\boldsymbol{A}}x(m_{\boldsymbol{A}}x)^\top + \boldsymbol{cx}^\top V_{\boldsymbol{\theta}}\boldsymbol{xc}^\top
\end{aligned}$$

Hence

$$\begin{aligned}
&\log \tilde{f}(\boldsymbol{y}, \boldsymbol{x}) \\
&\quad= -\frac{1}{2}\operatorname{tr}\left(m_{\boldsymbol{W}}\left[\boldsymbol{yy}^\top - m_{\boldsymbol{A}}\boldsymbol{xy}^\top - \boldsymbol{yx}^\top m_{\boldsymbol{A}} + m_{\boldsymbol{A}}\boldsymbol{x}(m_{\boldsymbol{A}}\boldsymbol{x})^\top + \boldsymbol{cx}^\top V_{\boldsymbol{\theta}}\boldsymbol{xc}^\top\right]\right) \\
&\qquad + const \\
&\quad= -\frac{1}{2}\left(\boldsymbol{y}^\top m_{\boldsymbol{W}}\boldsymbol{y} - \boldsymbol{y}^\top m_{\boldsymbol{W}}m_{\boldsymbol{A}}\boldsymbol{x} - (m_{\boldsymbol{A}}\boldsymbol{x})^\top m_{\boldsymbol{W}}\boldsymbol{y} + (m_{\boldsymbol{A}}\boldsymbol{x})^\top m_{\boldsymbol{W}}m_{\boldsymbol{A}}\boldsymbol{x}\right) \\
&\qquad - \frac{m_\gamma}{2}\boldsymbol{x}^\top V_{\boldsymbol{\theta}}\boldsymbol{x} + const \\
&\quad= -\frac{1}{2}(\boldsymbol{y} - m_{\boldsymbol{A}}\boldsymbol{x})^\top m_{\boldsymbol{W}}(\boldsymbol{y} - m_{\boldsymbol{A}}\boldsymbol{x}) - \frac{m_\gamma}{2}\boldsymbol{x}^\top V_{\boldsymbol{\theta}}\boldsymbol{x} + const
\end{aligned}$$

We can write the auxiliary node function as

$$\tilde{f}(\boldsymbol{x}, \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, (m_\gamma V_\theta)^{-1}) \tag{A7}$$

Appendix A.3. Update of Message to $\boldsymbol{y}$

Owing Equation (A7),

$$\begin{aligned}
\overrightarrow{\nu}(\boldsymbol{y}) &\propto \int \overrightarrow{\nu}(\boldsymbol{x})\tilde{f}(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{x} \\
&\propto \int \mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{x}}, V_{\boldsymbol{x}})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, (m_\gamma V_\theta)^{-1})\mathrm{d}\boldsymbol{x} \\
&\propto \int \mathcal{N}(\boldsymbol{x}|\Lambda^{-1}\boldsymbol{z}, \Lambda^{-1})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})\mathrm{d}\boldsymbol{x}
\end{aligned}$$

where

$$\begin{aligned}
\Lambda &= V_{\boldsymbol{x}}^{-1} + m_\gamma V_{\boldsymbol{\theta}} \\
\boldsymbol{z} &= V_{\boldsymbol{x}}^{-1}m_{\boldsymbol{x}}
\end{aligned}$$

In this way, the message $\overrightarrow{\nu}(\boldsymbol{y})$

$$\begin{aligned}\overrightarrow{\nu}(\boldsymbol{y}) &\propto \int \mathcal{N}(\boldsymbol{x}|\Lambda^{-1}\boldsymbol{z},\Lambda^{-1})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x},m_{\boldsymbol{W}}^{-1})\mathrm{d}\boldsymbol{x}\\ &\propto \mathcal{N}\Big(\boldsymbol{y}|m_{\boldsymbol{A}}(V_{\boldsymbol{x}}^{-1}+m_{\gamma}V_{\boldsymbol{\theta}})^{-1}V_{\boldsymbol{x}}^{-1}m_{\boldsymbol{x}},m_{\boldsymbol{A}}(V_{\boldsymbol{x}}^{-1}+m_{\gamma}V_{\boldsymbol{\theta}})^{-1}m_{\boldsymbol{A}}^{\top}+m_{\boldsymbol{V}}\Big)\end{aligned}$$

Appendix A.4. Update of Message to $\boldsymbol{x}$

Owing Equation (A7),

$$\begin{aligned}\overleftarrow{\nu}(\boldsymbol{x}) &\propto \int \overleftarrow{\nu}(\boldsymbol{y})\tilde{f}(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{y}\\ &\propto \int \mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{y}},V_{\boldsymbol{y}})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x},m_{\boldsymbol{W}}^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(m_{\gamma}V_{\theta})^{-1})\mathrm{d}\boldsymbol{y}\end{aligned}$$

Let us consider the log of $\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x},m_{\boldsymbol{W}}^{-1})$:

$$\begin{aligned}\log\Big[\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x},m_{\boldsymbol{W}}^{-1})\Big] &= (\boldsymbol{y}-m_{\boldsymbol{A}}\boldsymbol{x})^{\top}m_{\boldsymbol{W}}(\boldsymbol{y}-m_{\boldsymbol{A}}\boldsymbol{x})+const\\ &= (-m_{\boldsymbol{A}}^{-1}\boldsymbol{y}+\boldsymbol{x})^{\top}m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}m_{\boldsymbol{A}}(-m_{\boldsymbol{A}}^{-1}\boldsymbol{y}+\boldsymbol{x})+const\end{aligned}$$

Which yields,

$$\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x},m_{\boldsymbol{W}}^{-1}) \propto \mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{A}}^{-1}\boldsymbol{y},(m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}m_{\boldsymbol{A}})^{-1}) \tag{A8}$$

Therefore,

$$\begin{aligned}\overleftarrow{\nu}(\boldsymbol{x}) &\propto \int \mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{y}},V_{\boldsymbol{y}})\mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{A}}^{-1}\boldsymbol{y},(m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}m_{\boldsymbol{A}})^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(m_{\gamma}V_{\theta})^{-1})\mathrm{d}\boldsymbol{y}\\ &\propto \mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(m_{\gamma}V_{\theta})^{-1})\int \mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{y}},V_{\boldsymbol{y}})\mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{A}}^{-1}\boldsymbol{y},(m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}m_{\boldsymbol{A}})^{-1})\mathrm{d}\boldsymbol{y}\\ &\propto \mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(m_{\gamma}V_{\theta})^{-1})\mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{y}},m_{\boldsymbol{A}}^{-1}V_{\boldsymbol{y}}m_{\boldsymbol{A}}^{-\top}+(m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}m_{\boldsymbol{A}})^{-1})\\ &\propto \mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(m_{\gamma}V_{\theta})^{-1})\mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{y}},m_{\boldsymbol{A}}^{-1}(V_{\boldsymbol{y}}+m_{\boldsymbol{V}})m_{\boldsymbol{A}}^{-\top})\\ &\propto \mathcal{N}\Big(\boldsymbol{x}|\Lambda^{-1}\boldsymbol{z},\Lambda^{-1}\Big)\end{aligned}$$

where

$$\begin{aligned}\Lambda &= m_{\boldsymbol{A}}^{\top}(V_{\boldsymbol{y}}+m_{\boldsymbol{V}})^{-1}m_{\boldsymbol{A}}+m_{\gamma}V_{\theta}\\ \boldsymbol{z} &= m_{\boldsymbol{A}}^{\top}(V_{\boldsymbol{y}}+m_{\boldsymbol{V}})^{-1}m_{\boldsymbol{y}}\end{aligned}$$

Appendix A.5. Update of Message to $\boldsymbol{\theta}$

The outgoing variational message to $\boldsymbol{\theta}$ is defined as

$$\overleftarrow{\nu}(\boldsymbol{\theta}) \propto \exp\Big\{\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\gamma)}\log f(\boldsymbol{y}\ \boldsymbol{x},\boldsymbol{\theta},\gamma)\Big\}$$

Instead of working out $\overleftarrow{\nu}(\boldsymbol{\theta})$, we will work with corresponding log message

$$\begin{aligned}\log\overleftarrow{\nu}(\boldsymbol{\theta}) &= \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\gamma)}\Big[\log|\boldsymbol{W}|^{\frac{1}{2}}-\frac{1}{2}\Big((\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})^{\top}\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x})\Big)\Big]+const\\ &= -\frac{1}{2}\operatorname{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[\boldsymbol{y}\boldsymbol{y}^{\top}-\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^{\top}-\boldsymbol{y}(\boldsymbol{A}\boldsymbol{x})^{\top}+\boldsymbol{A}\boldsymbol{x}(\boldsymbol{A}\boldsymbol{x})^{\top}\Big]\Big)+const\\ &= -\frac{1}{2}\operatorname{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[-\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^{\top}-\boldsymbol{y}(\boldsymbol{A}\boldsymbol{x})^{\top}+\boldsymbol{A}\boldsymbol{x}(\boldsymbol{A}\boldsymbol{x})^{\top}\Big]\Big)+const\\ &= -\frac{1}{2}\operatorname{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[-(\boldsymbol{S}\boldsymbol{x}+\boldsymbol{c}\boldsymbol{x}^{\top}\boldsymbol{\theta})\boldsymbol{y}^{\top}-\boldsymbol{y}(\boldsymbol{S}\boldsymbol{x}+\boldsymbol{c}\boldsymbol{x}^{\top}\boldsymbol{\theta})^{\top}\Big]\Big)\\ &\quad -\frac{1}{2}\operatorname{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}(\boldsymbol{S}\boldsymbol{x}+\boldsymbol{c}\boldsymbol{x}^{\top}\boldsymbol{\theta})(\boldsymbol{S}\boldsymbol{x}+\boldsymbol{c}\boldsymbol{x}^{\top}\boldsymbol{\theta})^{\top}\Big)+const\end{aligned}$$

To proceed further, we recall one useful property

$$\boldsymbol{S}^\top \boldsymbol{\Sigma} \boldsymbol{c} = \boldsymbol{0} \quad \boldsymbol{c}^\top \boldsymbol{\Sigma} \boldsymbol{S} = \boldsymbol{0}^\top$$

where $\boldsymbol{\Sigma}$ is an arbitrary diagonal matrix. Now, let us work out the following term

$$\begin{aligned}
&\mathrm{tr}\Big(m_{\boldsymbol{W}}(\boldsymbol{S}\boldsymbol{x} + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta})(\boldsymbol{S}\boldsymbol{x} + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta})^\top\Big) \\
&\quad = \mathrm{tr}\Big(m_{\boldsymbol{W}}\Big[\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{S}\boldsymbol{x}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top\Big]\Big) \\
&\quad = \left[(\boldsymbol{S}\boldsymbol{x})^\top m_{\boldsymbol{W}}\boldsymbol{S}\boldsymbol{x} + \underbrace{\boldsymbol{c}^\top m_{\boldsymbol{W}}\boldsymbol{S}\boldsymbol{x}\boldsymbol{\theta}^\top\boldsymbol{x}}_{\boldsymbol{0}^\top} + \underbrace{\boldsymbol{S}^\top m_{\boldsymbol{W}}\boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{x}^\top}_{\boldsymbol{0}} + \boldsymbol{c}^\top m_{\boldsymbol{W}}\boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top\boldsymbol{x}\right] \\
&\quad = \mathrm{tr}\Big(m_{\boldsymbol{W}}\Big[\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top\Big]\Big)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\log \overleftarrow{\nu}(\boldsymbol{\theta}) = &-\frac{1}{2}\,\mathrm{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[-\boldsymbol{S}\boldsymbol{x}\boldsymbol{y}^\top - \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{y}^\top - \boldsymbol{y}\boldsymbol{x}^\top\boldsymbol{S}^\top - \boldsymbol{y}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top\Big]\Big) \\
&-\frac{1}{2}\,\mathrm{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top\Big]\Big) + const
\end{aligned}$$

We move terms which do not depend on $\boldsymbol{\theta}$ to the *const*, hence

$$\begin{aligned}
\log \overleftarrow{\nu}(\boldsymbol{\theta}) &= -\frac{1}{2}\,\mathrm{tr}\Big(m_{\boldsymbol{W}}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[-\boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{y}^\top - \boldsymbol{y}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top + \boldsymbol{c}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{c}^\top\Big]\Big) + const \\
&= -\frac{1}{2}\,\mathrm{tr}\Big(m_{\boldsymbol{W}}\Big[-\boldsymbol{c}\boldsymbol{\theta}^\top(V_{\boldsymbol{x},\boldsymbol{y}^\top} + m_{\boldsymbol{x}}m_{\boldsymbol{y}}^\top) - (V_{\boldsymbol{x},\boldsymbol{y}^\top} + m_{\boldsymbol{x}}m_{\boldsymbol{y}}^\top)\boldsymbol{\theta}\boldsymbol{c}^\top\Big]\Big) \\
&\quad -\frac{1}{2}\,\mathrm{tr}\Big(m_{\boldsymbol{W}}\Big[\boldsymbol{c}\Big(\mathrm{tr}(\boldsymbol{\theta}\boldsymbol{\theta}^\top V_{\boldsymbol{x}}) + m_{\boldsymbol{x}}^\top\boldsymbol{\theta}\boldsymbol{\theta}^\top m_{\boldsymbol{x}}\Big)\boldsymbol{c}^\top\Big]\Big) + const \\
&= -\frac{1}{2}\left[-\underbrace{\boldsymbol{c}^\top m_{\boldsymbol{W}}(V_{\boldsymbol{x},\boldsymbol{y}} + m_{\boldsymbol{y}}m_{\boldsymbol{x}}^\top)}_{\boldsymbol{z}^\top}\boldsymbol{\theta} - \boldsymbol{\theta}^\top\underbrace{(V_{\boldsymbol{x},\boldsymbol{y}} + m_{\boldsymbol{x}}m_{\boldsymbol{y}}^\top)m_{\boldsymbol{W}}\boldsymbol{c}}_{\boldsymbol{z}}\right] \\
&\quad -\frac{1}{2}\left[\boldsymbol{\theta}^\top\underbrace{m_\gamma(V_{\boldsymbol{x}} + m_{\boldsymbol{x}}m_{\boldsymbol{x}}^\top)}_{D}\boldsymbol{\theta}\right] + const \\
&= -\frac{1}{2}\Big[\boldsymbol{\theta}^\top D\boldsymbol{\theta} - \boldsymbol{z}^\top\boldsymbol{\theta} - \boldsymbol{\theta}^\top\boldsymbol{z}\Big] + const
\end{aligned}$$

Hence,

$$\overleftarrow{\nu}(\boldsymbol{\theta}) \propto \mathcal{N}(\Lambda^{-1}\boldsymbol{z}, \Lambda^{-1})$$

where

$$\begin{aligned}
\Lambda &= m_\gamma(V_{\boldsymbol{x}} + m_{\boldsymbol{x}}m_{\boldsymbol{x}}^\top) \\
\boldsymbol{z} &= (V_{\boldsymbol{x}\boldsymbol{y}} + m_{\boldsymbol{x}}m_{\boldsymbol{y}}^\top)\boldsymbol{c}m_\gamma
\end{aligned}$$

Appendix A.6. Update of Message to γ

$$\begin{aligned}
\log \overleftarrow{\nu}(\gamma) &= \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\log f(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta},\gamma) + const \\
&= \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\log|\boldsymbol{W}|^{\frac{1}{2}} - \frac{1}{2}\Big((\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})^\top\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})\Big)\Big] + const \\
&= \log|\boldsymbol{W}|^{\frac{1}{2}} - \frac{1}{2}\,\mathrm{tr}\Big(\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{y}\boldsymbol{y}^\top - \boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^\top + \boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{A}^\top - \boldsymbol{y}\boldsymbol{x}^\top\boldsymbol{A}^\top\Big]\Big) \\
&\quad + const
\end{aligned}$$

First of all, let us work out the term $\log|\boldsymbol{W}|^{\frac{1}{2}}$

$$\log|\boldsymbol{W}|^{\frac{1}{2}} = \frac{1}{2}\log\begin{vmatrix} \gamma & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\epsilon} & 0 & \dots & \vdots \\ 0 & 0 & \frac{1}{\epsilon} & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \end{vmatrix}$$

$$= \frac{1}{2}\log\gamma + \frac{1}{2}(1-M)\log(\epsilon) = \log\gamma^{\frac{1}{2}} + const$$

We split the expression under the expectation into four terms:

I: $\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\left[\boldsymbol{y}\boldsymbol{y}^\top\right]$
II: $\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\left[\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^\top\right]$
III: $\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\left[\boldsymbol{y}\boldsymbol{x}^\top\boldsymbol{A}^\top\right]$ and
IV: $\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\left[\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{A}^\top\right]$

Term I:

$$\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{y}\boldsymbol{y}^\top\Big] = \boldsymbol{W}\Big(V_{\boldsymbol{y}} + m_{\boldsymbol{y}}m_{\boldsymbol{y}}^\top\Big)$$

Recalling Equation (A6), term II:

$$\begin{aligned}\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^\top\Big] &= \boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big((\boldsymbol{S}+\boldsymbol{c}\boldsymbol{\theta}^\top)\boldsymbol{x}\boldsymbol{y}^\top\Big) \\ &= \boldsymbol{W}\Big(m_{\boldsymbol{A}}(V_{\boldsymbol{x}\boldsymbol{y}^\top} + m_{\boldsymbol{x}}m_{\boldsymbol{y}}^\top)\Big)\end{aligned}$$

Term III:

$$\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{y}\boldsymbol{x}^\top\boldsymbol{A}^\top\Big] = \boldsymbol{W}\Big((V_{\boldsymbol{y}\boldsymbol{x}^\top} + m_{\boldsymbol{y}}m_{\boldsymbol{x}}^\top)m_{\boldsymbol{A}}^\top\Big)$$

Term IV:

$$\begin{aligned}\boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{A}^\top\Big] &= \boldsymbol{W}\,\mathbb{E}\Big[(\boldsymbol{S}+\boldsymbol{c}\boldsymbol{\theta}^\top)\boldsymbol{x}\boldsymbol{x}^\top(\boldsymbol{S}+\boldsymbol{c}\boldsymbol{\theta}^\top)^\top\Big] \\ &= \boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{\theta})}\Big[\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{c}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{c}^\top + \boldsymbol{c}\boldsymbol{\theta}^\top\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{\theta}\boldsymbol{c}^\top\Big] \\ &= \boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{c}m_{\boldsymbol{\theta}}^\top\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{S}^\top + \boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top m_{\boldsymbol{\theta}}\boldsymbol{c}^\top\Big] \\ &\quad + \boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[\boldsymbol{c}(\boldsymbol{x}^\top V_\theta\boldsymbol{x} + m_{\boldsymbol{\theta}}^\top\boldsymbol{x}\boldsymbol{x}^\top m_{\boldsymbol{\theta}})\boldsymbol{c}^\top\Big] \\ &= \boldsymbol{W}\,\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[m_{\boldsymbol{A}}\boldsymbol{x}\boldsymbol{x}^\top m_{\boldsymbol{A}}^\top + \boldsymbol{c}\boldsymbol{x}^\top V_\theta\boldsymbol{x}\boldsymbol{c}^\top\Big] \\ &= \boldsymbol{W}\Big[m_{\boldsymbol{A}}(V_{\boldsymbol{x}} + m_{\boldsymbol{x}}m_{\boldsymbol{x}}^\top)m_{\boldsymbol{A}}^\top + \boldsymbol{c}(\mathrm{tr}(V_\theta V_{\boldsymbol{x}}) + m_{\boldsymbol{x}}^\top V_\theta m_{\boldsymbol{x}})\boldsymbol{c}^\top\Big]\end{aligned}$$

As the resulting message should depend solely on γ we need to get rid of all terms which incorporate matrix $\boldsymbol{W}$. We notice that

$$\mathrm{tr}(\boldsymbol{W}\boldsymbol{M}) = \mathrm{tr}\left(\boldsymbol{M}\cdot\begin{pmatrix} \gamma & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\epsilon} & 0 & \dots & \vdots \\ 0 & 0 & \frac{1}{\epsilon} & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \end{pmatrix}\right) = \boldsymbol{c}^\top\gamma\boldsymbol{M}\boldsymbol{c} + const$$

where $\boldsymbol{M}$ is an arbitrary matrix of the same dimensionality as the matrix $\boldsymbol{W}$ ($\boldsymbol{M}$ denote the first element of the matrix). In this way

$$
\begin{aligned}
\log \overleftarrow{\nu}(\gamma) = \log \gamma^{\frac{1}{2}} &- \frac{\gamma}{2}\boldsymbol{c}^{\top}\Big[V_{\boldsymbol{y}} + m_{\boldsymbol{y}} m_{\boldsymbol{y}}^{\top} - 2m_{\boldsymbol{A}}(V_{\boldsymbol{x}\boldsymbol{y}^{\top}} + m_{\boldsymbol{x}} m_{\boldsymbol{y}}^{\top})\Big]\boldsymbol{c} \\
&- \frac{\gamma}{2}\boldsymbol{c}^{\top}\Big[m_{\boldsymbol{A}}(V_{\boldsymbol{x}} + m_{\boldsymbol{x}} m_{\boldsymbol{x}}^{\top})m_{\boldsymbol{A}}^{\top} + \mathrm{tr}(V_{\theta} V_{\boldsymbol{x}}) + m_{\boldsymbol{x}}^{\top} V_{\theta} m_{\boldsymbol{x}})\Big]\boldsymbol{c}
\end{aligned}
$$

After exponentiating $\log \overleftarrow{\nu}(\gamma)$ it yields the gamma distribution:

$$
\overleftarrow{\nu}(\gamma) \propto \gamma^{\frac{1}{2}} \exp\Big\{-\frac{\gamma}{2}b\Big\}
$$

or

$$
\overleftarrow{\nu}(\gamma) \propto \Gamma\bigg(\frac{3}{2}, \frac{b}{2}\bigg)
$$

where

$$
\begin{aligned}
b = \Big(V_{\boldsymbol{y}} + m_{\boldsymbol{y}} m_{\boldsymbol{y}}^{\top}\Big) &- 2\Big(m_{\boldsymbol{A}}(V_{\boldsymbol{x}\boldsymbol{y}^{\top}} + m_{\boldsymbol{x}} m_{\boldsymbol{y}}^{\top})\Big) \\
&+ \Big(m_{\boldsymbol{A}}(V_{\boldsymbol{x}} + m_{\boldsymbol{x}} m_{\boldsymbol{x}}^{\top})m_{\boldsymbol{A}}^{\top}\Big) + \mathrm{tr}(V_{\theta}\Big(V_{\boldsymbol{x}} + m_{\boldsymbol{x}} m_{\boldsymbol{x}}^{\top}\Big))
\end{aligned}
$$

Appendix A.7. Derivation of $q(\boldsymbol{x}, \boldsymbol{y})$

The joint recognition distribution is given by

$$
\begin{aligned}
q(\boldsymbol{x}, \boldsymbol{y}) &\propto \overrightarrow{\nu}(\boldsymbol{x})\tilde{f}(\boldsymbol{x}, \boldsymbol{y})\overleftarrow{\nu}(\boldsymbol{y}) \\
&= \mathcal{N}(\boldsymbol{x}|m_{\boldsymbol{x}}, V_{\boldsymbol{x}})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, (m_{\gamma} V_{\theta})^{-1})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{y}}, V_{\boldsymbol{y}}) \\
&= \mathcal{N}\Big(\boldsymbol{x}|\Lambda^{-1}\boldsymbol{z}, \Lambda^{-1}\Big)\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{y}}, V_{\boldsymbol{y}})\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})
\end{aligned}
$$

where

$$
\begin{aligned}
\Lambda &= V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta} \\
\boldsymbol{z} &= V_{\boldsymbol{x}}^{-1} m_{\boldsymbol{x}}
\end{aligned}
$$

$$
q(\boldsymbol{x}, \boldsymbol{y}) \propto \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{x}\end{bmatrix}\middle|\begin{bmatrix}m_{\boldsymbol{y}} \\ \Lambda^{-1}\boldsymbol{z}\end{bmatrix}, \begin{bmatrix}V_{\boldsymbol{y}}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \Lambda\end{bmatrix}^{-1}\right)\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})
$$

Let us rearrange the terms in the Gaussian $\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1})$

$$
\begin{aligned}
\mathcal{N}(\boldsymbol{y}|m_{\boldsymbol{A}}\boldsymbol{x}, m_{\boldsymbol{W}}^{-1}) &\propto \exp\bigg(-\frac{1}{2}(\boldsymbol{y} - m_{\boldsymbol{A}}\boldsymbol{x})^{\top} m_{\boldsymbol{W}}(\boldsymbol{y} - m_{\boldsymbol{A}}\boldsymbol{x})\bigg) \\
&\propto \exp\bigg(-\frac{1}{2}\Big[\boldsymbol{y}^{\top} m_{\boldsymbol{W}}\boldsymbol{y} - \boldsymbol{y}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}}\boldsymbol{x} + \boldsymbol{x}^{\top} m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{x}^{\top} m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}}\boldsymbol{y}\Big]\bigg) \\
&\propto \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{x}\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{0}\end{bmatrix}, \begin{bmatrix}m_{\boldsymbol{W}} & -m_{\boldsymbol{W}} m_{\boldsymbol{A}} \\ -m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} & m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}}\end{bmatrix}^{-1}\right)
\end{aligned}
$$

$$
\begin{aligned}
q(\boldsymbol{x}, \boldsymbol{y}) \propto \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{x}\end{bmatrix}\middle|\begin{bmatrix}m_{\boldsymbol{y}} \\ \Lambda^{-1}\boldsymbol{z}\end{bmatrix}, \begin{bmatrix}V_{\boldsymbol{y}}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \Lambda\end{bmatrix}^{-1}\right) \qquad \text{(A9a)} \\
\cdot \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{x}\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{0}\end{bmatrix}, \begin{bmatrix}m_{\boldsymbol{W}} & -m_{\boldsymbol{W}} m_{\boldsymbol{A}} \\ -m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} & m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}}\end{bmatrix}^{-1}\right) \qquad \text{(A9b)} \\
= \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{x}\end{bmatrix}\middle| W_q^{-1}\begin{bmatrix}V_{\boldsymbol{y}}^{-1} m_{\boldsymbol{y}} \\ V_{\boldsymbol{x}}^{-1} m_{\boldsymbol{x}}\end{bmatrix}, W_q^{-1}\right) \qquad \text{(A9c)}
\end{aligned}
$$

where

$$W_q = \begin{bmatrix} m_{\boldsymbol{W}} + V_{\boldsymbol{y}}^{-1} & -m_{\boldsymbol{W}} m_{\boldsymbol{A}} \\ -m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} & m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}} + \Lambda \end{bmatrix}$$

The precision matrix W_q, to put it mildly, is quite far from a nice shape as it contains "unpleasant" matrix $m_{\boldsymbol{W}}$ with ϵ^{-1} on the diagonal. Let us workout the covariance matrix $V_q = W_q^{-1}$. To do this, we recall two important matrix identities:

$$(A+B)^{-1} = A^{-1} - A^{-1}(B^{-1} + A^{-1})^{-1}A^{-1} \tag{A10}$$

and

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \\ = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}$$

Let us denote the block elements of W_q as follows:

$$\begin{aligned} A &= m_{\boldsymbol{W}} + V_{\boldsymbol{y}}^{-1} & B &= -m_{\boldsymbol{W}} m_{\boldsymbol{A}} \\ C &= -m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} & D\, &\underbrace{m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}} + V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta}}_{D^*} \end{aligned}$$

In this way,

$$\begin{aligned} (A - BD^{-1}C)^{-1} &= (\underbrace{m_{\boldsymbol{W}} + V_{\boldsymbol{y}}^{-1}}_{A} - m_{\boldsymbol{W}} m_{\boldsymbol{A}} D^{-*} m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}})^{-1} \\ D &= A^{-1} - A^{-1}(A^{-1} - (m_{\boldsymbol{W}} m_{\boldsymbol{A}} D^{-*} m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}})^{-1})^{-1}A^{-1} \end{aligned}$$

Let us work out the auxiliary terms

$$\begin{aligned} A^{-1} = (m_{\boldsymbol{W}} + V_{\boldsymbol{y}}^{-1})^{-1} &= V_{\boldsymbol{y}} - V_{\boldsymbol{y}}(m_{\boldsymbol{V}} + V_{\boldsymbol{y}})^{-1}V_{\boldsymbol{y}} \\ &= \underbrace{m_{\boldsymbol{V}} - m_{\boldsymbol{V}}(V_{\boldsymbol{y}} + m_{\boldsymbol{V}})^{-1} m_{\boldsymbol{V}}}_{E} \end{aligned}$$

$$\begin{aligned} &(m_{\boldsymbol{W}} m_{\boldsymbol{A}} D^{-*} m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}})^{-1} = m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top} D^* m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} \\ &= m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top}(m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}} + V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta}) m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} \\ &= \underbrace{m_{\boldsymbol{V}} + m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top}(V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta}) m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}}}_{F} \end{aligned}$$

Hence,

$$(A - BD^{-1}C)^{-1} = E - E(F+E)^{-1}E$$

Next, let us consider D^{-1}, $D^{-1}C$ and BD^{-1}:

$$\begin{aligned} D^{-1} &= D^{-*} = \left(m_{\boldsymbol{A}}^{\top} m_{\boldsymbol{W}} m_{\boldsymbol{A}} + (V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta})\right)^{-1} \\ &= m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top} \\ &\quad - m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top} \left[m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top} + (V_{\boldsymbol{x}}^{-1} + m_{\gamma} V_{\theta})^{-1}\right]^{-1} m_{\boldsymbol{A}}^{-1} m_{\boldsymbol{V}} m_{\boldsymbol{A}}^{-\top} \end{aligned}$$

$$\begin{aligned}
D^{-1}C &= D^{-1}(-m_{\boldsymbol{A}}^{\top}m_{\boldsymbol{W}}) \\
&= -m_{\boldsymbol{A}}^{-1} + m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{V}}m_{\boldsymbol{A}}^{-\top}\left[m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{V}}m_{\boldsymbol{A}}^{-\top} + (V_{\boldsymbol{x}}^{-1} + m_{\gamma}V_{\theta})^{-1}\right]^{-1}m_{\boldsymbol{A}}^{-1} \\
BD^{-1} &= (-m_{\boldsymbol{W}}m_{\boldsymbol{A}})D^{-1} \\
&= -m_{\boldsymbol{A}}^{-\top} + m_{\boldsymbol{A}}^{-\top}\left[m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{V}}m_{\boldsymbol{A}}^{-\top} + (V_{\boldsymbol{x}}^{-1} + m_{\gamma}V_{\theta})^{-1}\right]^{-1}m_{\boldsymbol{A}}^{-1}m_{\boldsymbol{V}}m_{\boldsymbol{A}}^{-\top}
\end{aligned}$$

Although the resulting expressions do not have a nice form, we got rid of "unpleasant" matrix $m_{\boldsymbol{W}}$.

Appendix A.8. Free Energy Derivations

In this section we describe how to compute the variational free energy of AR node $f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}, \gamma)$. Note that essentially AR node implements the univariate Gaussian $f(y, \boldsymbol{x}, \boldsymbol{\theta}, \gamma) = \mathcal{N}\left(y \mid \boldsymbol{\theta}^{\top}\boldsymbol{x}, \gamma^{-1}\right)$ (Multivariate formulation is needed for bookkeeping previous states). The free energy functional is defined as

$$\begin{aligned}
F[q] &\triangleq U[q] - H[q] \\
U[q] &\triangleq -\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})q(\gamma)} \log f \\
H[q] &\triangleq -\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})q(\gamma)} \log q
\end{aligned}$$

At first, let us work out the entropy term $H[q]$.

$$\begin{aligned}
H[q] &= -\mathbb{E}_{q(\boldsymbol{x},y)} \log q(\boldsymbol{x}, y) - \mathbb{E}_{q(\boldsymbol{\theta})} \log q(\boldsymbol{\theta}) - \mathbb{E}_{q(\gamma)} \log q(\gamma) \\
&= \frac{1}{2}\left(\log|2\pi e V_{xy}| + \log|2\pi e V_{\boldsymbol{\theta}}|\right) \\
&\quad - \alpha - \log\beta + \log\Gamma(\alpha) + (1-\alpha)\psi(\alpha)
\end{aligned}$$

where $\psi(\alpha)$ denotes digamma function.

Now, let us consider the average energy $U[q]$

$$-\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})q(\gamma)}\left[\log\frac{\gamma^{1/2}}{\sqrt{2\pi}} - \frac{\gamma}{2}(y - \boldsymbol{\theta}^{\top}\boldsymbol{x})^2\right]$$

We split the expression under the expectation into two terms:

I: $-\mathbb{E}_{q(\gamma)}\left[\log\frac{\gamma^{1/2}}{\sqrt{2\pi}}\right]$

II: $-\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})q(\gamma)}\left[-\frac{\gamma}{2}(y - \boldsymbol{\theta}^{\top}\boldsymbol{x})^2\right]$

Term I:

$$-\mathbb{E}_{q(\gamma)}\left[\log\frac{\gamma^{1/2}}{\sqrt{2\pi}}\right] = -\mathbb{E}_{q(\gamma)}\left[\frac{1}{2}\log\gamma - \frac{1}{2}\log 2\pi\right] = -\frac{1}{2}[\psi(\alpha) - \log\beta] + \frac{1}{2}\log 2\pi$$

Term II:

$$\begin{aligned}
&-\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})q(\gamma)}\left[-\frac{\gamma}{2}(y - \boldsymbol{\theta}^{\top}\boldsymbol{x})^2\right] = \frac{m_{\gamma}}{2}\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})}\left[(y - \boldsymbol{\theta}^{\top}\boldsymbol{x})^2\right] \\
&\quad = \frac{m_{\gamma}}{2}\mathbb{E}_{q(\boldsymbol{x},y)q(\boldsymbol{\theta})}\left[y^2 - 2y\boldsymbol{\theta}^{\top}\boldsymbol{x} + \boldsymbol{\theta}^{\top}\boldsymbol{x}\boldsymbol{x}^{\top}\boldsymbol{\theta}\right] \\
&\quad = \frac{m_{\gamma}}{2}\left[\sigma_y^2 + m_y^2 - 2\left[V_{x_t\boldsymbol{x}^{\top}} + m_y m_{\boldsymbol{x}}^{\top}\right]m_{\boldsymbol{\theta}} + \mathrm{tr}\left[(V_{\boldsymbol{\theta}} + m_{\boldsymbol{\theta}}m_{\boldsymbol{\theta}}^{\top})V_{\boldsymbol{x}}\right]\right] \\
&\quad + \frac{m_{\gamma}}{2}\left[m_{\boldsymbol{\theta}}^{\top}(V_{\boldsymbol{x}} + m_{\boldsymbol{x}}m_{\boldsymbol{x}}^{\top})m_{\boldsymbol{\theta}}\right]
\end{aligned}$$

hence

$$U[q] = -\frac{1}{2}[\psi(\alpha) - \log\beta] + \frac{1}{2}\log 2\pi + \frac{m_\gamma}{2}d$$

where

$$d = \sigma_y^2 + m_y^2 - 2\left[V_{y\boldsymbol{x}^\top} + m_y m_{\boldsymbol{x}}^\top\right]m_{\boldsymbol{\theta}} + \mathrm{tr}\left[(V_{\boldsymbol{\theta}+m_{\boldsymbol{\theta}}m_{\boldsymbol{\theta}}^\top)V_{\boldsymbol{x}}}\right] + m_{\boldsymbol{\theta}}^\top(V_{\boldsymbol{x}} + m_{\boldsymbol{x}}m_{\boldsymbol{x}}^\top)m_{\boldsymbol{\theta}}$$

References

1. Akaike, H. Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.* **1969**, *21*, 243–247. [CrossRef]
2. Charbonnier, R.; Barlaud, M.; Alengrin, G.; Menez, J. Results on AR-modelling of nonstationary signals. *Signal Process.* **1987**, *12*, 143–151. [CrossRef]
3. Tahir, S.M.; Shaameri, A.Z.; Salleh, S.H.S. Time-varying autoregressive modeling approach for speech segmentation. In Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat.No.01EX467), Kuala Lumpur, Malaysia, 13–16 August 2001; Volume 2, pp. 715–718. [CrossRef]
4. Rudoy, D.; Quatieri, T.F.; Wolfe, P.J. Time-Varying Autoregressions in Speech: Detection Theory and Applications. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 977–989. [CrossRef]
5. Chu, Y.J.; Chan, S.C.; Zhang, Z.G.; Tsui, K.M. A new regularized TVAR-based algorithm for recursive detection of nonstationarity and its application to speech signals. In Proceedings of the 2012 IEEE Statistical Signal Processing Workshop (SSP), Ann Arbor, MI, USA, 5–8 August 2012; pp. 361–364. [CrossRef]
6. Paulik, M.J.; Mohankrishnan, N.; Nikiforuk, M. A time varying vector autoregressive model for signature verification. In Proceedings of the 1994 37th Midwest Symposium on Circuits and Systems, Lafayette, LA, USA, 3–5 August 1994; Volume 2, pp. 1395–1398. [CrossRef]
7. Kostoglou, K.; Robertson, A.D.; MacIntosh, B.J.; Mitsis, G.D. A Novel Framework for Estimating Time-Varying Multivariate Autoregressive Models and Application to Cardiovascular Responses to Acute Exercise. *IEEE Trans. Biomed. Eng.* **2019**, *66*, 3257–3266. [CrossRef]
8. Eom, K.B. Analysis of Acoustic Signatures from Moving Vehicles Using Time-Varying Autoregressive Models. *Multidimens. Syst. Signal Process.* **1999**, *10*, 357–378. [CrossRef]
9. Abramovich, Y.I.; Spencer, N.K.; Turley, M.D.E. Time-Varying Autoregressive (TVAR) Models for Multiple Radar Observations. *IEEE Trans. Signal Process.* **2007**, *55*, 1298–1311. [CrossRef]
10. Zhang, Z.G.; Hung, Y.S.; Chan, S.C. Local Polynomial Modeling of Time-Varying Autoregressive Models With Application to Time–Frequency Analysis of Event-Related EEG. *IEEE Trans. Biomed. Eng.* **2011**, *58*, 557–566. [CrossRef]
11. Wang, H.; Bai, L.; Xu, J.; Fei, W. EEG recognition through Time-varying Vector Autoregressive Model. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 292–296. [CrossRef]
12. Sharman, K.; Friedlander, B. Time-varying autoregressive modeling of a class of nonstationary signals. In Proceedings of the ICASSP'84—IEEE International Conference on Acoustics, Speech, and Signal Processing, San Diego, CA, USA, 19–21 March 1984; Volume 9, pp. 227–230. [CrossRef]
13. Reddy, G.R.S.; Rao, R. Non stationary signal prediction using TVAR model. In Proceedings of the 2014 International Conference on Communication and Signal Processing, Bangkok, Thailand, 10–12 October 2014; pp. 1692–1697. [CrossRef]
14. Souza, D.B.d.; Kuhn, E.V.; Seara, R. A Time-Varying Autoregressive Model for Characterizing Nonstationary Processes. *IEEE Signal Process. Lett.* **2019**, *26*, 134–138. [CrossRef]
15. Zheng, Y.; Lin, Z. Time-varying autoregressive system identification using wavelets. In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (Cat. No.00CH37100), Istanbul, Turkey, 5–9 June 2000; Volume 1, pp. 572–575. [CrossRef]
16. Moon, T.K.; Gunther, J.H. Estimation of Autoregressive Parameters from Noisy Observations Using Iterated Covariance Updates. *Entropy* **2020**, *22*, 572. [CrossRef]
17. Rajan, J.J.; Rayner, P.J.W.; Godsill, S.J. Bayesian approach to parameter estimation and interpolation of time-varying autoregressive processes using the Gibbs sampler. *IEE Proc. Vis. Image Signal Process.* **1997**, *144*, 249–256. [CrossRef]
18. Prado, R.; Huerta, G.; West, M. *Bayesian tIme-Varying Autoregressions: Theory, Methods and Applications*; University of Sao Paolo: Sao Paulo, Brazil, 2000; p. 2000.
19. Nakajima, J.; Kasuya, M.; Watanabe, T. Bayesian analysis of time-varying parameter vector autoregressive model for the Japanese economy and monetary policy. *J. Jpn. Int. Econ.* **2011**, *25*, 225–245. [CrossRef]
20. Barber, D. *Bayesian Reasoning and Machine Learning*; Cambridge University Press: Cambridge, UK, 2012.
21. Zhong, X.; Song, S.; Pei, C. Time-varying Parameters Estimation based on Kalman Particle Filter with Forgetting Factors. In Proceedings of the EUROCON 2005—The International Conference on "Computer as a Tool", Belgrade, Serbia, 21–24 November 2005; Volume 2, pp. 1558–1561. [CrossRef]

22. Winn, J.; Bishop, C.M.; Jaakkola, T. Variational Message Passing. *J. Mach. Learn. Res.* **2005**, *6*, 661–694.
23. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
24. Korl, S. A Factor Graph Approach to Signal Modelling, System Identification and Filtering. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 2005.
25. Penny, W.D.; Roberts, S.J. Bayesian multivariate autoregressive models with structured priors. *IEE Proc. Vis. Image Signal Process.* **2002**, *149*, 33–41. [CrossRef]
26. Loeliger, H.A.; Dauwels, J.; Hu, J.; Korl, S.; Ping, L.; Kschischang, F.R. The Factor Graph Approach to Model-Based Signal Processing. *Proc. IEEE* **2007**, *95*, 1295–1322. [CrossRef]
27. Dauwels, J.; Korl, S.; Loeliger, H.A. Expectation maximization as message passing. *Int. Symp. Inf. Theory* **2005**, 583–586. [CrossRef]
28. Cox, M.; van de Laar, T.; de Vries, B. A factor graph approach to automated design of Bayesian signal processing algorithms. *Int. J. Approx. Reason.* **2019**, *104*, 185–204. [CrossRef]
29. De Vries, B.; Friston, K.J. A Factor Graph Description of Deep Temporal Active Inference. *Front. Comput. Neurosci.* **2017**, *11*. [CrossRef] [PubMed]
30. Beck, J. Bayesian system identification based on probability logic. *Struct. Control. Health Monit.* **2010**. [CrossRef]
31. Zhang, D.; Song, X.; Wang, W.; Fettweis, G.; Gao, X. Unifying Message Passing Algorithms Under the Framework of Constrained Bethe Free Energy Minimization. *arXiv* **2019**, arXiv:1703.10932.
32. Dauwels, J. On Variational Message Passing on Factor Graphs. In Proceedings of the IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007; pp. 2546–2550. [CrossRef]
33. Zhang, D.; Wang, W.; Fettweis, G.; Gao, X. Unifying Message Passing Algorithms Under the Framework of Constrained Bethe Free Energy Minimization. *arXiv* **2017**, arXiv:1703.10932v3.
34. Cui, G.; Yu, X.; Iommelli, S.; Kong, L. Exact Distribution for the Product of Two Correlated Gaussian Random Variables. *IEEE Signal Process. Lett.* **2016**, *23*, 1662–1666. [CrossRef]
35. Wu, W.-R.; Chen, P.-C. Subband Kalman filtering for speech enhancement. *IEEE Trans. Circuits Syst. Analog. Digit. Process.* **1998**, *45*, 1072–1083. [CrossRef]
36. So, S.; Paliwal, K.K. Modulation-domain Kalman filtering for single-channel speech enhancement. *Speech Commun.* **2011**, *53*, 818–829. [CrossRef]
37. Nossier, S.A.; Wall, J.; Moniri, M.; Glackin, C.; Cannings, N. An Experimental Analysis of Deep Learning Architectures for Supervised Speech Enhancement. *Electronics* **2021**, *10*, 17. [CrossRef]
38. Hu, Y.; Loizou, P.C. Subjective comparison and evaluation of speech enhancement algorithms. *Speech Commun.* **2007**, *49*, 588–601. [CrossRef]
39. Paliwal, K.; Basu, A. A speech enhancement method based on Kalman filtering. In Proceedings of the ICASSP'87—IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, USA, 6–9 April 1987; Volume 12, pp. 177–180. [CrossRef]
40. You, C.H.; Rahardja, S.; Koh, S.N. Autoregressive Parameter Estimation for Kalman Filtering Speech Enhancement. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP'07, Honolulu, HI, USA, 15–20 April 2007; Volume 4, pp. IV-913–IV-916. [CrossRef]
41. Grenier, Y. Time-dependent ARMA modeling of nonstationary signals. *IEEE Trans. Acoust. Speech Signal Process.* **1983**, *31*, 899–911. [CrossRef]
42. Kamary, K.; Mengersen, K.; Robert, C.P.; Rousseau, J. Testing hypotheses via a mixture estimation model. *arXiv* **2014**, arXiv: 1412.2044.
43. Friston, K.; Parr, T.; Zeidman, P. Bayesian model reduction. *arXiv* **2019**, arXiv:1805.07092.
44. Podusenko, A.; Kouw, W.M.; de Vries, B. Online variational message passing in hierarchical autoregressive models. In Proceedings of the 2020 IEEE International Symposium on Information Theory, Los Angeles, CA, USA, 21–26 June 2020; pp. 1337–1342.