Variational Stabilized Linear Forgetting in State-Space Models

Thijs van de Laar*, Marco Cox*, Anouk van Diepen*, and Bert de Vries*,†
*Department of Electrical Engineering, Eindhoven University of Technology
†GN Hearing, Eindhoven, the Netherlands

***Abstract*—State-space modeling of non-stationary natural signals is a notoriously difficult task. As a result of context switches, the memory depth of the model should ideally be adapted online. Stabilized linear forgetting (SLF) has been proposed as an elegant method for state-space tracking in context-switching environments. In practice, SLF leads to state and parameter estimation tasks for which no analytical solutions exist. In the literature, a few approximate solutions have been derived, making use of specific model simplifications. This paper proposes an alternative approach, in which SLF is described as an inference task on a generative probabilistic model. SLF is then executed by a variational message passing algorithm on a factor graph representation of the generative model. This approach enjoys a number of advantages relative to previous work. First, variational message passing (VMP) is an automatable procedure that adapts appropriately under changing model assumptions. This eases the search process for the best model. Secondly, VMP easily extends to estimate model parameters. Thirdly, the modular make-up of the factor graph framework allows SLF to be used as a click-on feature in a large variety of complex models. The functionality of the proposed method is verified by simulating an SLF state-space model in a context-switching data environment.**

I. Introduction

Natural signal modeling is a key issue in various scientific and engineering disciplines, e.g. [1], [2]. Unfortunately, natural signals often exhibit characteristics, such as a switching context, which are notoriously difficult to track by standard methods like the Kalman filter.

As an illustrative example, consider tracking a ball during a soccer game. When a player kicks the ball, the direction and velocity of the ball suddenly changes. At that instance, the state estimator should forget the current trajectory and start tracking a new trajectory. Standard modeling assumptions do not suffice here. For instance, a Kalman filter will only adapt gradually to the new trajectory. A model with switching dynamics [3] is not appropriate here either, because the dynamical system that governs ball movements does not change after a kick. Instead, a model is required that retains the dynamical system model, but resets the tracked state upon sudden changes. However, a complete reset of the state is not required either, because ball direction and speed are constrained by physical boundaries such as player strength and the size of the playing field. Therefore, the state after reset is constrained as well.

Stabilized forgetting (SF) was introduced as a solution for robust state tracking in the presence of "partial state resetting" dynamics [4], such as described above. In the absence of observations, SF also ensures that the uncertainty about the state estimate remains bounded [5]. A general characterization of SF is given by [6]. SF is discussed in the context of control by [7], and SF is justified in the context of recursive estimation by [5], [8]. In general terms, recursive Bayesian estimation distinguishes two stages: a *time step* that predicts a new state belief given previous measurements, and a *data step* that updates the predicted state belief with information from a new measurement, see e.g. [9]. In stabilized forgetting, the state estimate comprises a mixture of two estimates, where the first estimate derives from a standard state transition model and the second estimate relates to a fixed distribution (the alternative model). A time-dependent mixing variable governs the mixture of the default and alternative model. Furthermore, the literature distinguishes between linear mixing of the estimates in the so-called stabilized *linear* forgetting (SLF) model, and exponential mixing in the stabilized *exponential* forgetting model [10].

In practice, the mixing coefficient is generally not known beforehand. Algorithms for estimating the mixing coefficient in the context of recursive estimation have been derived in [11]–[13]. This paper proposes an alternative approach to SLF. Instead of adapting the recursive estimator, this paper describes SLF as an inference task in an augmented probabilistic model. We represent the model by a (Forney-style) factor graph (FFG). In the factor graph framework, joint state-parameter estimation can be executed through message passing algorithms. Relative to previous work, a major advantage is that message passing in a factor graph is an automatable procedure. Hence, if one were to make different modeling assumptions, state and parameter estimation algorithms are updated accordingly. This is important, because signal modeling often involves an iterative search process for the best model. This paper develops a variational message passing (VMP) approach to SLF that is based on the variational Bayes method [14], [15], which is a well-known and principled technique [16] for approximate inference.

The main contributions can be summarized as follows:

1) We formulate SLF as an inference task in an augmented generative model in the context of a state-space model (SSM) (Section II). Interestingly, SLF is obtained through augmenting a generative probabilistic model rather than by an algorithmic adaptation.
2) We provide a factor graph representation for the augmented SLF state space model (SLF-SSM) (Section

ISBN 978-0-9928626-7-1

II-B). Joint estimation of the hidden state sequence and parameters is executed by VMP (Section III). The FFG framework allows for modular model extensions to which VMP can automatically adapt.
3) We present simulations to verify proper functioning of the augmented SLF-SSM model (Section IV).

II. Generative model description

A generative model describes the random process by which observed data are generated as a probability distribution over all model variables [17]. Sampling from the generative model generates artificial data, whose statistics ideally match the observed data. In this section we describe the generative model for an SSM with SLF. An SSM is generally described by a state transition model and an observation model [18]. The state transition model describes how hidden states evolve over time. The observation model describes how an observation relates to the state at a given instant.

Consider the following simple discrete-time SSM for an observed sequence of data $y_1, \dots, y_K$:

$$\boldsymbol{x}_k | \boldsymbol{x}_{k-1} \sim \mathcal{N}\big(\mathbf{A}\boldsymbol{x}_{k-1}, \mathbf{W}^{-1}\big) \; ; \tag{1a}$$

$$y_k | \boldsymbol{x}_k \sim \mathcal{N}\big(\mathbf{C}\boldsymbol{x}_k, \sigma^2\big) \; . \tag{1b}$$

The hidden states $\boldsymbol{x}_0, \dots, \boldsymbol{x}_K \in \mathbb{R}^d$ are linked through a transition model (1a) with transition matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and stationary Gaussian process noise with inverse covariance matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$. The observation model (1b) relates the hidden state to observations through $\mathbf{C}^T \in \mathbb{R}^d$, and i.i.d. Gaussian observation noise. Note the choice for a simple linear Gaussian observation model and the omission of a control model for the hidden states. These assumptions are however for instructive reasons; the described approach allows for straightforward extensions to more complex observation models, and the inclusion of control.

SLF is incorporated in the generative model by augmenting the standard state transition model from (1a) with a time-dependent binary switch, $z_k \in \{0, 1\}$. This switch determines whether the state transition at time k is governed by the standard transition model, or if the next state is drawn from an alternative fixed distribution (also known as the *stable distribution*):

$$\boldsymbol{x}_k | \boldsymbol{x}_{k-1} \sim \begin{cases} \mathcal{N}\big(\mathbf{A}\boldsymbol{x}_{k-1}, \mathbf{W}^{-1}\big) & \text{if} \quad z_k = 1\,; \\ \mathcal{N}\big(\boldsymbol{m}, \boldsymbol{\Lambda}^{-1}\big) & \text{if} \quad z_k = 0\,. \end{cases} \tag{2}$$

In case $z_k = 0$, this augmented state transition model "forgets" the information from previous observations about $\boldsymbol{x}_k$, which allows the model to cope with abrupt state transitions that are very unlikely under the standard transition model.

To complete the augmented SSM, a model for z_k must be defined. One could assume the switch variables to be governed by a hidden Markov model. For simplicity however, but without loss of generality, this paper assumes the switches to be i.i.d. according to a Bernoulli distribution:

$$z_k \sim \mathcal{B}er(\pi) \; . \tag{3}$$

If $0 < \pi < 1$, the value of the latent switch variables is uncertain. The augmented state transition model is then formulated as a linear Gaussian mixture of the standard transition model and the stable distribution.

The generative model specified in (1b)–(3) implies the generative model factorization of (4):

$$\begin{aligned} &p(\boldsymbol{x}_{0:K}, y_{1:K}, z_{1:K}, \mathbf{W}, \pi, \boldsymbol{m}, \boldsymbol{\Lambda}, \sigma^2) = \\ &\quad p(\mathbf{W})\, p(\pi)\, p(\boldsymbol{m})\, p(\boldsymbol{\Lambda})\, p(\sigma^2)\, p(\boldsymbol{x}_0) \prod_{k=1}^{K} p(y_k|\boldsymbol{x}_k, \sigma^2) \\ &\quad p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{W}, z_k, \boldsymbol{m}, \boldsymbol{\Lambda})\, p(z_k|\pi)\,. \end{aligned} \tag{4}$$

From this probabilistic model factorization and observed data, the goal is to estimate the hidden state sequence and tuning parameters.

A. Forney-style factor graph example

A Forney-style factor graph (FFG) offers both a visually intuitive representation of factorized probabilistic models as well as a principled method for solving inference problems by message passing. An FFG represents variables as edges, and relations between variables (the factors) as nodes. Edges in an FFG are undirected. An excellent introduction to FFGs and message passing is available in [19].

As a practical example, consider a model with the factorization

$$f(x_1, \dots, x_5) = f_a(x_1, x_2) \times f_b(x_2, x_3, x_4) \times f_c(x_4, x_5)\,. \tag{5}$$

Now suppose x_1, x_3 and x_5 are observed, and the marginal distribution of x_2 is requested. Then, after rearranging the integrals through the distributive law, the following nested integration evaluates to the exact marginal:

$$\begin{aligned} f(x_2) &= \int \cdots \int f(x_1, \dots, x_5)\, \mathrm{d}x_1\, \mathrm{d}x_3\, \mathrm{d}x_4\, \mathrm{d}x_5 \\ &= \underbrace{\int f_a(x_1, x_2)\, \mathrm{d}x_1}_{①} \times \\ &\underbrace{\int \Big(\int f_b(x_2, x_3, x_4)\, \mathrm{d}x_3 \times \overbrace{\int f_c(x_4, x_5)\, \mathrm{d}x_5}^{②} \Big)\, \mathrm{d}x_4}_{③}\,. \end{aligned} \tag{6}$$

The outcome of each integral can be visualized effectively as a messages that is passed over an edge of the corresponding graph, see Fig. 1. The resulting marginal $f(x_2)$ is simply the product of the colliding messages ① and ③.

In order to work efficiently with FFGs, it helps to store a lookup table of message update rules for commonly used factors. In that case, if we design probabilistic models with factors from the lookup table, complex inference procedures reduce to consulting the message update rules in the lookup table. This makes the FFG an attractive framework for proposing and testing alternative model hypotheses.

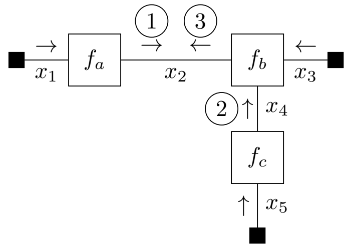


Fig. 1. Example message passing schedule on a Forney-style factor graph, for computing the marginal over x_2. Solid terminal nodes indicate observed variables.

B. The SLF Generative Model as an FFG

The factorization of the stabilized linear forgetting model of (4) is represented by the Forney-style factor graph in Fig. 2. In order to avoid a cluttered graph, dotted half-edges represent the tuning parameters of the model. It is possible to apply priors on these parameters by terminating the half-edges by a prior distribution node.

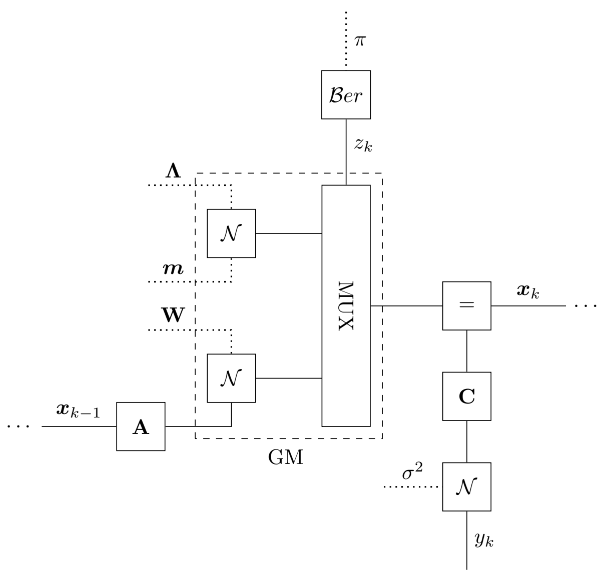


Fig. 2. A Forney style factor graph representation of a single time section of the augmented state-space model (with stabilized linear forgetting). Ellipses denote a continuation of the displayed section over time. The complete factor graph across K time steps is a concatenation of K identical sections. The network inside the dashed box denoted by GM specifies SLF as a Gaussian mixture model. Dotted edges denote time-independent parameters.

III. Inference by variational message passing

Given the generative model and observations, probabilistic inference can estimate the hidden state sequence. This is achieved by conditioning the joint distribution on the observed signal, and integrating over all variables except the state sequence. Observation noise parameter σ^2 is assumed to be known, which yields:

$$p(\boldsymbol{x}_{0:K}|y_{1:K}) = \iint p(\boldsymbol{x}_{0:K}, \boldsymbol{z}_{1:K}, \theta|y_{1:K})\, \mathrm{d}\theta\, \mathrm{d}\boldsymbol{z}_{1:K}, \quad (7)$$

where $\theta = \{\mathbf{W}, \pi, \boldsymbol{m}, \mathbf{\Lambda}\}$ denotes the set of tuning parameters. Other than for a standard linear SSM, this distribution cannot be computed analytically for the augmented SLF-SSM. However, variational Bayesian inference approximates the exact posterior by a simpler distribution $q(\boldsymbol{x}_{0:K}, \boldsymbol{z}_{1:K}, \theta)$. Variational inference searches for an approximate distribution by minimizing an approximation error that is measured as the Kullback-Leibler divergence from the exact posterior to the approximating distribution. This approach replaces computation of the intractable exact posterior with a well-defined optimization problem. With the *mean-field approximation*, all variables are governed by independent approximate distributions:

$$q(\boldsymbol{x}_{0:K}, \boldsymbol{z}_{1:K}, \theta) = q(\mathbf{W})q(\pi)q(\boldsymbol{m})q(\mathbf{\Lambda})q(\boldsymbol{x}_0)\prod_{k=1}^{K} q(\boldsymbol{x}_k)q(z_k)\,. \quad (8)$$

The factorization of the approximate distribution implicitly defines the distribution types [17] and free parameters for each q-factor. The goal is to find the free parameter values that minimize the approximation error. After optimization, the approximate posterior assumes the role of the exact posterior in any further computations. An approximate distribution for the state sequence is then obtained by substituting the approximate posterior in (7), which yields:

$$p(\boldsymbol{x}_{0:K}|y_{1:K}) \approx q(\boldsymbol{x}_{0:K})\,. \quad (9)$$

The variational inference procedure can be implemented as a message passing algorithm on the factor graph. This paper follows the variational message passing (VMP) formulation from [15], which defines a recipe for iterative message updates, such that the variational approximation is guaranteed to converge to a local minimum of the KL divergence.

Fig. 3 illustrates the message passing schedule that achieves approximate posterior inference in the augmented SLF-SSM. The circled numbers represent the messages and their update order. Crucially, the message update equations only depend on local factors (i.e. the Gaussian mixture factor, Bernoulli factor, etc.), and are not influenced by distant nodes or global modeling context. As a result, once-derived message update rules for a factor can be re-applied across many different models.

IV. Simulations

As an illustrative example, we tracked the one-dimensional hidden state from a model for a non-stationary data sequence. The performance of a standard SSM was compared with the augmented SLF-SSM. As a model performance measure, the (natural) logarithm of the state sequence posterior density was evaluated at the observed state sequence:

$$Q_q = \log q(X = \boldsymbol{x}_{0:K})\,. \quad (10)$$

The largest value for this metric identifies the best performing model in the sense that under this model the observed state sequence is more likely than for alternative models.

ISBN 978-0-9928626-7-1

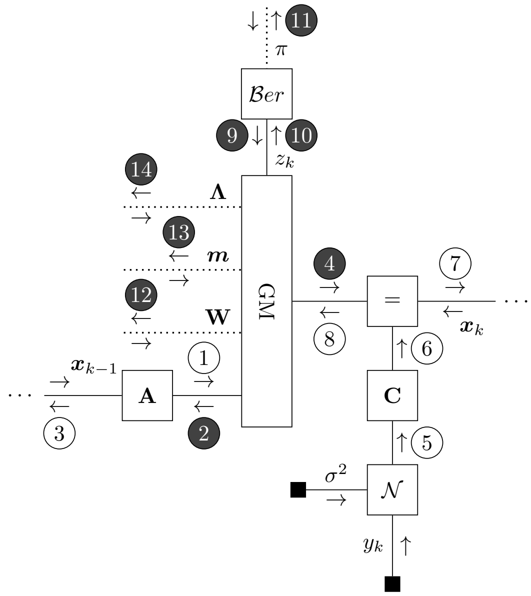


Fig. 3. Variational message passing schedule for the augmented SLF state-space model. Messages denoted by black circles are computed according to the variational message update rule (see Appendix). White circles represent sum-product messages as derived in [20]. Solid black nodes indicate observed variables, and dotted edges denote time-independent parameters.

A. Training set

The data y in Fig. 4 were generated from the generative process as defined in (1b)–(3), with parameters: $x_0 = 0$; $\mathrm{W} = 10$; $\mathrm{C} = \mathrm{A} = 1$; $\pi = 0.95$; $m = 0$; $\Lambda = 0.01$, and $\sigma^2 = 3$.

B. State estimation

State estimation was performed with the standard SSM and augmented SLF-SSM. For the standard model, vague priors were chosen as: $x_0 \sim \mathcal{N}(0, 100)$, $\mathrm{W} \sim \mathcal{G}am(0.01, 0.01)$ (in shape-rate parametrization). For the SLF-SSM, additional vague priors were chosen as: $\pi \sim \mathcal{B}eta(1, 1)$, $m \sim \mathcal{N}(0, 100)$, and $\Lambda \sim \mathcal{G}am(0.01, 0.01)$. Estimation results and model performance after 500 iterations are shown in Fig. 4.

Comparing model performance (10), it is evident that the SLF-SSM outperforms the standard SSM. For the standard model, the transition noise precision is estimated lower than for the augmented model, as a result of the large jumps in state that are required to track the data. The effects of a low transition noise precision can be seen in the top diagram in Fig. 4, where there is almost no smoothing effect. In contrast, the SLF-SSM model explicitly models jumps through the augmented state transition model, which is active when $z_k = 0$. The augmented state transition thus avoids the need to model large jumps with a low transition noise precision. Small jumps however, e.g. relating to the third switch activation, can also occur under high transition noise precision and therefore remain difficult to identify.

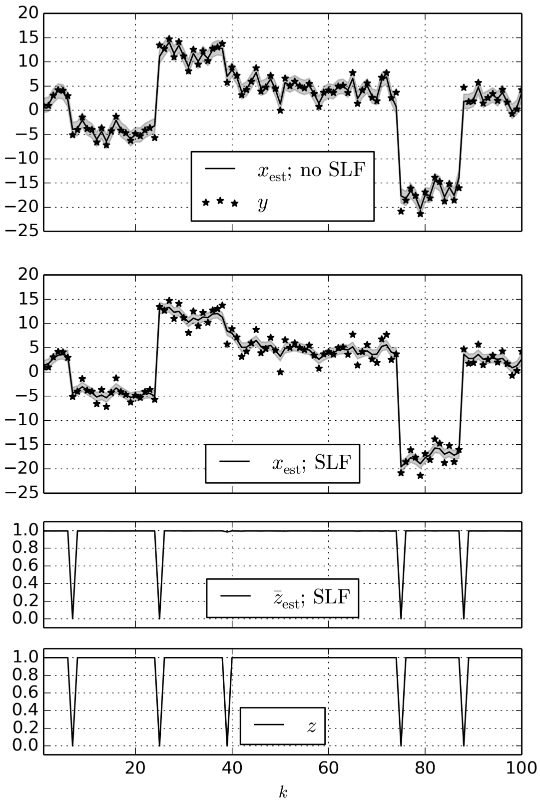


Fig. 4. Top diagram: estimated state x_{est} from data $y_{1:K}$ with the standard SSM; $\mathrm{W}_{\mathrm{est}} = 0.060 \pm 0.009$; performance $Q_q = -184$. Grey bands indicate the estimated standard deviation. Second diagram: estimated state with augmented SSM; $\mathrm{W}_{\mathrm{est}} = 0.39 \pm 0.06$; $\pi_{\mathrm{est}} = 0.96 \pm 0.02$; $m_{\mathrm{est}} = -0.85 \pm 6.57$; $\Lambda_{\mathrm{est}} = 0.0030 \pm 0.0027$; performance $Q_q = -146$. Third diagram: mean of estimated switch state. Bottom diagram: true switch state.

V. Conclusions

This paper described a probabilistic modeling approach to stabilized linear forgetting (SLF), and an automatable variational message passing (VMP) algorithm to infer hidden states and parameters. A Forney-style factor graph (FFG) was employed to represent the factorized generative SLF-SSM. The FFG provides an easily adaptable model description, from which tabulated message updates were locally derived.

In contrast to previous work, the current approach allows for convenient adaptation of model proposals with automated updating of the state (and parameter) inference procedures. SLF was described as a click-on model feature and can easily be incorporated across a wide range of models. However, more experiments are required to test the robustness of the VMP approach to SLF in different modeling contexts.

Our approach builds on previous literature by its capacity to estimate parameters of the stable distribution. Moreover, the VMP algorithm can be adapted to online inference [21].

In more general terms, deriving algorithms through automated inference in generative probabilistic models is gaining much interest in the machine learning community [22]. VMP is one among many techniques that are currently under development in the effort toward automated inference methods. In this paper we showed by example that the probabilistic modeling method also holds promise to advance algorithm development for adaptive signal processing systems.

ISBN 978-0-9928626-7-1 © EURASIP 2017

References

[1] Y. Karklin and M. S. Lewicki, “A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals,” *Neural computation*, vol. 17, no. 2, pp. 397–423, 2005.

[2] R. E. Turner, “Statistical Models for Natural Sounds,” Ph.D. dissertation, Gatsby Computational Neuroscience Unit, UCL, 2010.

[3] Z. Ghahramani and G. E. Hinton, “Variational learning for switching state-space models,” *Neural computation*, vol. 12, no. 4, pp. 831–864, 2000.

[4] R. Kulhavý and M. Kárný, “Tracking of slowly varying parameters by directional forgetting,” in *Proc. 9th IFAC World Congress, Budapest, Hungary*, 1984, pp. 78–83.

[5] F. J. Kraus, “Stabilized least squares estimators for time-variant processes,” in *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*. IEEE, 1989, pp. 1803–1804.

[6] R. Kulhavý and M. B. Zarrop, “On a general concept of forgetting,” *International Journal of Control*, vol. 58, no. 4, pp. 905–924, Oct. 1993. [Online]. Available: http://dx.doi.org/10.1080/00207179308923034

[7] M. Kárný, J. Böhm, T. V. Guy, and P. Nedoma, “Mixture-based adaptive probabilistic control,” *International Journal of adaptive control and signal processing*, vol. 17, no. 2, pp. 119–132, 2003.

[8] J. T. Milek, *Stabilized adaptive forgetting in recursive parameter estimation*. vdf Hochschulverlag AG, 1995.

[9] S. Azizi and A. Quinn, “Approximate Bayesian filtering using stabilized forgetting,” in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 2711–2715.

[10] R. Kulhavý and F. J. Kraus, “On Duality of Exponential and Linear Forgetting,” *Automatica*, vol. 32, no. 10, pp. 1403–1415, 1996.

[11] V. Smidl and A. Quinn, “Bayesian estimation of non-stationary AR model parameters via an unknown forgetting factor,” in *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th*. IEEE, 2004, pp. 221–225.

[12] V. Smidl and F. Gustafsson, “Bayesian estimation of forgetting factor in adaptive filtering and change detection,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, Aug. 2012, pp. 197–200.

[13] S. Azizi and A. Quinn, “A data-driven forgetting factor for stabilized forgetting in approximate Bayesian filtering,” in *Signals and Systems Conference (ISSC), 2015 26th Irish*, Jun. 2015, pp. 1–6.

[14] H. Attias, “A Variational Baysian Framework for Graphical Models.” in *NIPS*, vol. 12, 1999.

[15] J. Dauwels, “On Variational Message Passing on Factor Graphs,” in *IEEE International Symposium on Information Theory*, Jun. 2007, pp. 2546–2550.

[16] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

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

[18] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

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

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

[21] A. Honkela and H. Valpola, “On-line variational Bayesian learning,” in *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003, pp. 803–808.

[22] D. M. Blei, “Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models,” *Annual Review of Statistics and Its Application*, vol. 1, no. 1, pp. 203–232, 2014. [Online]. Available: http://www.annualreviews.org/doi/abs/10.1146/annurev-statistics-022513-115657

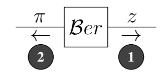
Appendix

This appendix[1] summarizes the non-standard variational messages in Fig. 3. Variational message updates are computed in accordance with [15], with shorthand notations for the mean $\overline{f(\cdot)} = \mathbb{E}_{q(\cdot)}[f(\cdot)]$, covariance $\mathbb{C}\mathrm{ov}[f(\cdot)] = \mathbb{C}\mathrm{ov}_{q(\cdot)}[f(\cdot)]$, and average energy $\mathbb{U}[f(\cdot)] = -\mathbb{E}_{q(\cdot)}[\log f(\cdot)]$.

Bernoulli node

Table I summarizes the variational message updates for the Bernoulli node, with switch $z \in \{0, 1\}$ and parametrized by $\pi \in [0, 1]$ through node function $f(z, \pi) = \pi^z (1 - \pi)^{1-z}$.

TABLE I
MESSAGE UPDATE RULES FOR THE BERNOULLI NODE



1	$\propto \mathcal{B}er\left(z \,\middle|\, \frac{\exp\left(\overline{\log \pi}\right)}{\exp\left(\overline{\log \pi}\right) + \exp\left(\overline{\log(1-\pi)}\right)}\right)$
2	$\propto \mathcal{B}eta(\pi \mid \bar{z} + 1, 2 - \bar{z})$

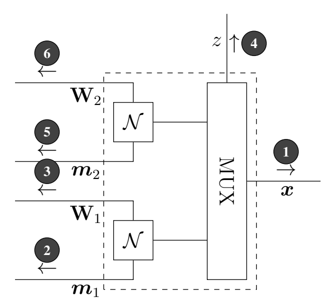
Gaussian mixture node

Table II summarizes the variational message updates for the Gaussian mixture node, which mixes Gaussian models through a switch $z \in \{0, 1\}$ through the node function:

$$f(\boldsymbol{m}_1, \mathbf{W}_1, z, \boldsymbol{m}_2, \mathbf{W}_2, \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{x} | \boldsymbol{m}_1, \mathbf{W}_1^{-1}\big)^z \times \mathcal{N}\big(\boldsymbol{x} | \boldsymbol{m}_2, \mathbf{W}_2^{-1}\big)^{1-z}.$$

For stabilized linear forgetting, the Gaussian mixture has two Gaussian components.

TABLE II
UPDATE RULES FOR THE GAUSSIAN MIXTURE NODE



1	$\propto \mathcal{N}\big(\boldsymbol{x} | \bar{\boldsymbol{m}}_1, (\bar{z}\bar{\mathbf{W}}_1)^{-1}\big) \times \mathcal{N}\big(\boldsymbol{x} | \bar{\boldsymbol{m}}_2, ((1-\bar{z})\bar{\mathbf{W}}_2)^{-1}\big)$
2	$\propto \mathcal{N}\big(\boldsymbol{m}_1 | \bar{\boldsymbol{x}}, (\bar{z}\bar{\mathbf{W}}_1)^{-1}\big)$
3	$\propto \mathcal{W}(\mathbf{W}_1 | \mathbf{V}_1, \bar{z} + d + 1)$, with dimensionality d, and $\mathbf{V}_1 = \left(\bar{z}\left[(\bar{\boldsymbol{m}}_1 - \bar{\boldsymbol{x}})(\bar{\boldsymbol{m}}_1 - \bar{\boldsymbol{x}})^T + \mathbb{C}\mathrm{ov}[\boldsymbol{m}_1] + \mathbb{C}\mathrm{ov}[\boldsymbol{x}]\right]\right)^{-1}$
4	$\propto \mathcal{B}er\left(z \,\middle|\, \frac{\exp\left(-\mathbb{U}\left[\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{m}_1, \mathbf{W}_1^{-1}\right)\right]\right)}{\sum_{i=\{1,2\}} \exp\left(-\mathbb{U}\left[\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{m}_i, \mathbf{W}_i^{-1}\right)\right]\right)}\right)$
5	$\propto \mathcal{N}\Big(\boldsymbol{m}_2 | \bar{\boldsymbol{x}}, \big((1-\bar{z})\bar{\mathbf{W}}_2\big)^{-1}\Big)$
6	$\propto \mathcal{W}(\mathbf{W}_2 | \mathbf{V}_2, 2 - \bar{z} + d)$, with $\mathbf{V}_2 = \left((1-\bar{z})\left[(\bar{\boldsymbol{m}}_2 - \bar{\boldsymbol{x}})(\bar{\boldsymbol{m}}_2 - \bar{\boldsymbol{x}})^T + \mathbb{C}\mathrm{ov}[\boldsymbol{m}_2] + \mathbb{C}\mathrm{ov}[\boldsymbol{x}]\right]\right)^{-1}$

[1]Derivations are available at http://biaslab.github.io/pdf/slf/supplement.pdf

ISBN 978-0-9928626-7-1