

Message Passing Algorithms for Hierarchical Dynamical Models

Citation for published version (APA):

enöz, I. (2022). *Message Passing Algorithms for Hierarchical Dynamical Models*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Eindhoven University of Technology.

Document status and date:

Published: 24/06/2022

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Message Passing Algorithms for Hierarchical Dynamical Models

İsmail Şenöz

Copyright © 2022 by İsmail Şenöz. All rights reserved.

No parts of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written permission of the author.

A catalogue record is available from the TU/e Library.
ISBN: 978-90-386-5532-1

TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY



The research presented in this dissertation was conducted in the Bayesian Autonomous Systems Lab (BIASlab) of the Signal Processing Systems (SPS) group at the Department of Electrical Engineering, Eindhoven University of Technology (TU/e).

Message Passing Algorithms for Hierarchical Dynamical Models

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 24 juni 2022 om 13:30 uur

door

İsmail Şenöz

geboren te İzmir, Turkije

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr.ir. Elena Lomonova
1e promotor: prof.dr.ir. A. de Vries
copromotoren: assoc.prof.dr. C.D. Mathys (Aarhus University, Interacting Minds Centre)
dr. T. W. van de Laar
leden: prof.dr.ir. F.M.J. Willems
prof.dr.ir. P.M.J. Van den Hof
prof.dr. T.M. Heskes (Radboud University, Data Science)
assoc.prof.dr.ir. J.H.G. Dauwels (Technische Universiteit Delft, Microelectronic)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Summary

Building models to understand patterns is one of the fundamental pursuits of science. Among such models, dynamical models are ubiquitous in most scientific fields as these models describe how processes evolve. These models can be found in finance, navigation, control engineering, audio signal processing, and telecommunications. Applications range from tracking the position of an aircraft to estimating the variance of returns on assets. Inference corresponds to computing posterior distributions over the variables of a given model.

This dissertation describes a theoretical framework for deriving customized message passing-based inference algorithms in factor graphs and illustrates the framework's application to hierarchical dynamical models. Factor graphs are visual representations of the dependency structures among the variables of a model. Inference tasks on a given model can be realized using message passing algorithms on the corresponding factor graph, where propagated messages are computed by integration (summation). Often, dynamical models of natural processes are constructed hierarchically. Because the hierarchies in the models may grow the complexity of dependence structures, exact inference by message passing in these models becomes infeasible and computationally impossible in a real-time setting. To employ hierarchical dynamical models in applications that require real-time processing, inference by message passing needs to be approximated.

This dissertation proposes a constraint manipulation strategy to derive message passing algorithms on Forney-style factor graphs that pave the way for an efficient implementation of automated approximate inference. By changing the constraints on the local sub-graphs, we derive various local message update rules as the stationary solutions of a constrained Bethe free energy from first principles. By combining these local updates, one can perform hybrid message passing. Constraint manipulation is a modular way of generating message-passing algorithms by combining local updates such that factorized computations of local updates allow efficient implementation.

This thesis then demonstrates how message passing by constraint manipulation applies to hierarchical dynamical systems. The focus is on the hierarchical Gaussian filter, a time-series

model for volatile processes where non-linear transforms couple the states in this process. A composite factor node (named GCV), representing the state transition distribution of an HGF, is constructed and subsequently can be used as a plug-in module for any factor graph. Various message update rules for the GCV node under multiple constraints are derived. Combining derived update rules, it is possible to implement automated hybrid message passing for the variants of the HGF-like models in software packages ForneyLab.jl and ReactiveMP.jl.

Natural processes are often non-stationary. Therefore, the realizations of natural processes have statistical properties that change with time. A source of non-stationarity is due to regime changes in the parameter values. A parameterized transition distribution may govern changes in the statistical properties. If the parameters of this transition distribution are subject to regime switches, then the statistical properties of the transition distribution will depend on the regimes. To account for context switches, this dissertation provides a switching extension to the HGF model with a hidden Markov model that governs a selection mechanism for the parameters of the ordinary HGF model. A composite factor node (named GCSV) is constructed as a successor of GCV, and closed-form message update rules are derived. The derived message update rules allow automated real-time message passing in graphs containing state transitions with switching volatile dynamics.

Moreover, this dissertation illustrates how the graphical formalism of factor graphs allows us to build complex models from primitive node structures. To that end, this dissertation focuses on auto-regressive(AR) processes that are ubiquitous for time-series modeling. AR processes are often constructed under the assumption that the precision of the innovation noise and AR coefficients are constant to ensure stationarity. The dissertation shows how the GCV node could be used as a plug-in module within the graphs of auto-regressive models to extend the auto-regressive models such that the deriving noise processes are time-varying. Message passing in the corresponding model leads to online state and parameter estimation in auto-regressive models with time-varying process noise.

In summary, this dissertation formulates a guideline for automated approximate inference and performance evaluation for discrete-time hierarchical dynamical models. This thesis advocates that inference and performance evaluation can be automated and efficiently implemented by message passing on factor graphs for hierarchical dynamical models via locally approximated message update rules. A product of the dissertation is a message-passing framework equipped with modular factor nodes with available message update rules to create hierarchical dynamical systems.

Hasan Taşdemir'in anısına

Contents

Summary	v
List of Symbols	1
1 General Introduction	5
1.1 Motivation	5
1.2 Analysis of Dynamical Models	7
1.2.1 Hierarchical Dynamical Models	7
1.2.2 Approximate Inference	8
1.2.3 Scoring Models and Constraint Specifications	9
1.3 Research Questions	10
1.4 Summary of Contributions	12
1.5 Outline	12
2 Preliminaries	15
2.1 Background on Graphs	15
2.1.1 Directed Graphs	15
2.1.2 Undirected Graphs	16
2.1.3 Factor Graphs	16
2.1.4 Contingency Tables	17
2.2 Statistical Inference	18
2.2.1 Sum-product Algorithm for Trees	18
2.3 Hierarchical Models	19
2.4 Time Series	20
3 Message Passing and Local Constraint Manipulation in Factor Graphs	23
3.1 Introduction	23

3.2	Factor Graphs and Bethe Free Energy	25
3.2.1	Forney-style Factor Graphs	25
3.2.2	Variational Free Energy	27
3.2.3	Approximations to Variational Free Energy	27
3.2.4	Problem Statement for Approximate Inference	28
3.2.5	Sketch of Solution Approach	29
3.3	Bethe Lagrangian Optimization by Message Passing	30
3.3.1	Stationary Points of the Bethe Lagrangian	30
3.3.2	Minimizing the Bethe Free Energy by Belief Propagation	32
3.4	Message Passing Variations through Constraint Manipulation	34
3.4.1	Factorization Constraints	34
3.4.1.1	Structured Mean-field Variational Message Passing	36
3.4.1.2	Naive Mean-field Variational Message Passing	39
3.4.2	Form Constraints	40
3.4.2.1	Data Constraints	41
3.4.2.2	Laplace Propagation	43
3.4.2.3	Expectation Propagation	45
3.4.3	Hybrid Constraints	48
3.4.3.1	Mean-field Variational Laplace	48
3.4.3.2	Expectation Maximization	49
3.4.4	Overview of Message Passing Algorithms	52
3.5	Scoring Models by Minimized Variational Free Energy	52
3.5.1	Evaluation of the Entropy of Dirac-delta Constrained Beliefs	53
3.5.2	Evaluation of Node-Local Free Energy for Deterministic Nodes	54
3.5.3	Evaluating the Bethe Free Energy	55
3.6	Implementation of Algorithms and Simulations	57
3.7	Related Work	57
3.8	Discussion and Conclusions	58
4	The Hierarchical Gaussian Filter	61
4.1	Introduction	61
4.2	Model Definition	63
4.2.1	State Transition Dynamics	63
4.2.2	Likelihood Specifications	64
4.2.3	Prior Specifications	64
4.2.4	FFG Representation	65
4.3	Local Constraints and Lagrangians	66
4.3.1	Factorization Constraints	66
4.3.2	Form Constraints	67
4.3.3	Lagrangian formulations	68
4.4	Problem Definition	70

4.5	Gaussian with Controlled Variance	71
4.6	Message Computations	72
4.6.1	Structured Factorization Computations	72
4.6.2	Naive Mean-Field Factorization Computations	78
4.6.3	Laplace Approximated Messages	81
4.6.4	Laplace Approximated Marginals	83
4.6.5	Moment-matching based Marginal and Message Approximations	85
4.6.6	Expectation Propagation for Discrete Likelihoods	88
4.7	Simulations	91
4.7.1	Verification on Synthetic Data	92
4.7.1.1	Experimental Setup	92
4.7.1.2	Experimental Results	94
4.7.2	Validation on Stock Prices	96
4.7.2.1	Choices of Priors	96
4.7.2.2	Experimental Results	96
4.7.3	Validation on Currency Exchange	99
4.7.3.1	Choices of Priors	100
4.7.3.2	Experimental Results	100
4.8	Discussion and Conclusions	101
5	The Switching Hierarchical Gaussian Filter	103
5.1	Introduction	103
5.2	Model Definition	104
5.3	Problem Definition and Message Passing	107
5.4	Message Computations for the SHGF	110
5.5	Simulations	119
5.5.1	Verification on Synthetic Data	119
5.5.2	Validation on Stock Prices	120
5.6	Discussion and Conclusions	121
6	Auto-regressive Models with Time-varying Noise Processes	123
6.1	Introduction	123
6.2	Model Specification and FFG Representation	124
6.3	Problem Definition	127
6.4	Variational Inference	127
6.5	Experimental Verification	128
6.6	Discussion and Conclusions	131
7	Discussion and Conclusions	133
7.1	Contributions	133
7.2	Strengths and Limitations	134
7.3	Outlook	138

A Appendix	139
A.1 Free Energy Minimization by Variational Inference	139
A.2 Lagrangian optimization and the dual problem	142
A.3 Local free energy example for a deterministic node	143
A.4 Proofs	145
A.4.1 Proof of Lemma 3.1	145
A.4.2 Proof of Lemma 3.2	145
A.4.3 Proof of Theorem 3.1	146
A.4.4 Proof of Lemma 3.3	147
A.4.5 Proof of Theorem 3.2	148
A.4.6 Proof of Corollary 3.1	149
A.4.7 Proof of Lemma 3.4	149
A.4.8 Proof of Theorem 3.3	149
A.4.9 Proof of Lemma 3.5	149
A.4.10 Proof of Theorem 3.4	149
A.4.11 Proof of Lemma 3.6	150
A.4.12 Proof of Lemma 3.7	150
A.4.13 Proof of Theorem 3.5	150
A.4.14 Proof of Theorem 3.6	151
A.4.15 Proof of Theorem 3.7	151
A.4.16 Proof of Theorem 3.8	152
A.4.17 Proof of Theorem 3.9	153
References	154
List of Publications	165
Acknowledgements	169
Biography	173

List of Symbols

Abbreviations	Description
AR	Auto-regressive
AMSE	Average Mean-Squared Error
BFE	Bethe Free Energy
BP	Belief Propagation
DC	Data Constraint
EM	Expectation Maximization
EP	Expectation Propagation
FFG	Forney-style Factor Graph
GBP	Generalized Belief Propagation
GCV	Gaussian with Controlled Variance
GCSV	Gaussian with Controlled Switching Variance
HA	Hearing Aid
HGF	Hierarchical Gaussian Filter
HMC	Hamiltonian Monte Carlo
KLD	Kullback Leibler Divergence
LP	Laplace Propagation
MCMC	Markov Chain Monte Carlo
MFVLP	Mean-Field Variational Laplace
MFVMP	Mean-Field Variational Message Passing
NLE	Negative Log-Evidence
TFFG	Terminated Forney-style Factor Graph
VFE	Variational Free Energy
VMP	Variational Message Passing
PPP	Probabilistic Programming Package

SHGF	Switching Hierarchical Gaussian Filter
SVMP	Structured Variational Message Passing
SP	Sum Product

Mathematical notation	Description
x	A random variable (usually left implicit)
\hat{x}	An observed value for random variable x
\mathbf{x}	A collection of random variables
$x^{(i)}$	A random variable representing i -th level of hierarchy
x_k	The k -th element of a collection \mathbf{x}
$\mathbf{x}_{1:k}$	Random variable collection indexed from 1 to k
\mathbf{A}	A matrix
\mathbf{A}^\top	The transpose of matrix \mathbf{A}
\mathbf{A}^{-1}	The inverse of matrix \mathbf{A}
A_{jk}	The j, k -th element of matrix \mathbf{A}
f	A factor function
p	A probability distribution
q	A recognition (posterior) distributions
tr	The matrix trace operator
\log	Natural logarithm
H	Differential entropy
U	Average Energy
\mathbb{E}	Expectation operator
μ_{jb}	Message on the j edge toward factor b
\vec{v}	Forward message
\bar{v}	Backward message

Probability Distribution Description

$\delta(x - \hat{x})$	Dirac (Kronecker) delta with location \hat{x}
$\mathcal{N}(\mathbf{x} \mathbf{m}, \Sigma)$	Gaussian distribution with mean \mathbf{m} and covariance Σ
$\Gamma(\gamma \alpha, \beta)$	Gamma distribution with shape α and rate β parameters
$\mathcal{W}(\mathbf{\Lambda} \mathbf{V}, \nu)$	Wishart distribution with scale matrix \mathbf{V} and degrees of freedom ν
$\mathcal{C}at(s \mathbf{p})$	Categorical distribution with event probability vector \mathbf{p}
$\mathcal{D}ir(\boldsymbol{\pi} \mathbf{a})$	Dirichlet distribution with parameter vector \mathbf{a}
$\mathcal{T}(x m, v, a, b)$	Truncated Gaussian distribution whose support is limited to $[a, b]$

CHAPTER 1

General Introduction

"You can't always get what you want, but if you try, sometimes, you might find you get what you need."

–The Rolling Stones

1.1 Motivation

Machine learning-enhanced electronic devices are now the norm in the world. Smartphones, smartwatches, wearable health trackers, headphones, and hearing aids are a few examples of these gadgets. The market for wearable electronic devices is increasing [1], as these devices are becoming an indispensable part of our daily lives. Equipped with machine learning algorithms, such electronic devices are often expected to perform challenging tasks that require online processing of sensory and user preferences information.

For example, consider a scenario where a hearing aid (HA) client is having trouble understanding a conversation partner at a cocktail party. Perhaps she needs more amplification or maybe the active noise reduction module actually has an inactive babble noise mode that would help in this setting. In an ideal scenario, a hearing professional (audiologist) would be stand-by and use her expertise to try various HA algorithm alternatives while the conversation is ongoing. Unfortunately, this is not a realistic scenario. Instead, we might consider that HA alternatives are proposed in-situ by a software-based HA design agent that runs on a smartwatch. In this scenario, whenever the client indicates by a simple gesture (e.g., head or wrist shake) that she is unhappy with the current performance of the HA algorithm, it is the task of the agent to respond on-the-spot with a *good* algorithm alternative. We will discuss below that this HA design agent should internally build a model for its sensory inputs, namely the environmental acoustics and user appraisals.

As a second example, consider a trading agent equipped with machine learning algorithms that tries to automate financial decisions. This agent is expected to analyze economic factors for the management of investments and provide automated buy/sell actions for algorithmic trading. The theory of active management postulates that the key to generating profit is having accurate return forecasts combined with acting on these forecasts [2]. The competition of sophisticated investors in financial markets implies that making precise predictions to generate profit requires superior information, either through access to better data, a superior ability to process it, or both [3, Chapter 1]. Applications for trading typically aim to produce better and more actionable forecasts, thus improving the quality of investment decisions and results. The quality of profits is represented with the Sharpe ratio of the return difference between the portfolio and a benchmark to the volatility of those returns [4]. A suitable trading agent needs to learn about the financial factors and take actions based on these factors in a volatile market environment. Moreover, an agent needs to incorporate streaming financial derivatives data to make predictions and take actions in real-time while competing with other investors and trade agents.

Even though the preceding two examples are from two very different fields, the agents in these examples share a common ground as both agents can be interpreted as a controller (*regulator*) of their environment. According to the Good Regulator Theorem [5], a proper controller should contain a generative model for its sensory inputs that are generated by the environment. The HA design agent's sensory inputs comprise acoustic signals from the hearing aid and appraisals from the end-user. In other words, *good* HA design agents need to contain generative models for both acoustic signals and user appraisals. A financial trading agent needs models for various financial factors such as returns, volatility of returns, etc. Moreover, Conant and Ashby also show that if an agent is a good regulator, then it succeeded in internalizing a model either directly or implicitly [5]. This dissertation advocates designing intelligent agents that take the good regulator theorem as a guide.

Often the agent needs to perform in real-time. For example, a financial agent can not afford to wait minutes in high-frequency trading tasks. If the agent must perform in real-time, then all decisions that are associated with these tasks must be automated. Here, the application of probability theory provides an exciting prospect. If a probabilistic description of a model is available, then the agent's task can often be described as a probabilistic inference task, which in principle is automatable. A *probabilistic generative model* describes how observations (variables) are generated (related) with a certain degree of uncertainty such that it is possible to produce a version of the underlying phenomena artificially. For example, a generative model of bird sounds should produce waveforms that sound similar to bird sounds when listened to. The observations in a probabilistic generative model might correspond to sound waveforms, stock prices, or any other measurement done in a particular field. Once a generative model is proposed, the relationship between variables can be discovered (inferred) using probability theory. Together with inference for latent variables, probability theory provides a metric (Bayesian evidence) by which to measure the performance of the proposed model. Depending on the model's performance, the model can be revised, or if satisfactory

performance is achieved, it can be applied in practice.

Non-rigorously, a generative probabilistic model is a joint probability distribution [6] that specifies the relationships among both latent and observed variables with a degree of uncertainty about the quantities involved in the model. For the agent to reason about the variables of interest, it needs a rigorous way to manipulate probability distributions. Probability theory offers a comprehensive framework such that the agent can perform inference once it starts receiving observations [7]. As a matter of convention, the literature distinguishes between *time-varying* latent variables called states and fixed latent variables that we call parameters. The latent states refer to the hidden causes that generate observations via a likelihood function. The signal processing task amounts to estimating these hidden states of the model. From a Bayesian modeling viewpoint, both states and parameters are latent (unobserved) variables. In principle, we are interested in three tasks: signal processing (technically: state estimation), parameter estimation, and model performance evaluation. Nevertheless, the distinction between these tasks is excessive, and this thesis advocates a unified view for these tasks in Chapter 3.

Bayesian inference provides a framework for computing posterior distributions over the variables of interest and critiquing the proposed models. Computing the posteriors is equivalent to executing Bayes rule [6]. In contrast, model critique is equivalent to computing the ratio of model evidence [6] as a measure of relative performance among the models. Even though proposing models is an enterprise of human creativity, all the remaining steps of Bayesian inference can be automated. This is why we favor designing the agent as a fully generative probabilistic model, where all design tasks are implemented as probabilistic inference tasks. In short, real-time design automation implies the need to design agents as probabilistic models capable of executing all design steps through automated inference. Though automated inference can be achieved by application of the Bayes rule, the challenge lies at the computational level as Bayesian inference is known to be an NP-hard (non-polynomial time) problem [8]. Once automation is achieved, the designer can propose and iterate models instead of working out tedious operations for inference.

1.2 Analysis of Dynamical Models

1.2.1 Hierarchical Dynamical Models

Dynamical models are at the core of engineering applications as they model the underlying physical phenomena that change over time. These models can be found in finance, navigation, control engineering, audio signal processing, and telecommunications. Applications range from tracking the position of an aircraft to estimating the variance of returns on assets. For instance, time-varying autoregressive models are commonly used in speech detection [9], prediction for interest rates [10] and analysis of non-stationary EEG signals in biomedical engineering [11]. Gaussian state-space models and its variants are popular in position (velocity) tracking [12], while hidden Markov models are used for speech enhancement tasks in audio

signal processing [13]. Stochastic volatility models are used in finance to track time-varying variances to adjust agents account for volatility in decision-making [14].

Often, models of natural processes are constructed hierarchically. For example, natural auditory scenes tend to be collected by hierarchically layered processes that produce acoustic primitives, such as animal vocalizations in a forest, words in speech or amplitude-modulated sinusoids [15] over time. The auditory system can separate different source signals by recognizing processes that go by the name Auditory Scene Analysis [16]. To process sounds within this hierarchy, the auditory system needs to adapt to the statistical structure of the acoustic scene [17]. Simoncelli and McDermott investigate perceptual modeling of natural sounds in [18] and demonstrate the statistics of these sounds govern perception.

A hierarchical generative model specifies the relationships among latent and observed variables such that the dependency structures are convoluted with layers of functional relations. These dependent structures allow a designer to model complex phenomena, and inference enables the discovery of hidden ties between hierarchically coupled variables [19]. Because the hierarchies in the models may grow the complexity of dependence structures, inference in these models becomes challenging and prohibit using these models in practical applications [20]. When this is the case, a designer might simplify the model to make inference tractable. Nevertheless, the simplified model is no longer an accurate representation of the underlying phenomena, and inference results in solutions that are biased [7]. As the generative model represents a problem formulation, altering the model in favor of a convenient inference procedure changes the problem formulation. In this case, a designer obtains solutions to a different problem than the initially intended one. This dissertation defends that getting approximate solutions to an accurately described problem should be preferred over exact solutions to a modified problem statement [21]. We advocate obtaining approximate algorithms through manipulating the constraint specification without altering the model (see Chapter 3).

1.2.2 Approximate Inference

Obtaining posterior distributions via the Bayes rule involves integration with respect to latent variables in the model [6]. Unfortunately, for profound model dimensions, all inference-related tasks include high dimensional integrals, which are often computationally not tractable. This poses a significant challenge to the automation of inference, the primary concern for real-time automated design. The computational challenge for inference worsens when the model is hierarchical due to complex dependence structures. Moreover, if the underlying dynamical model characterizes a time series, the number of latent variables and observations grows significantly over time, rendering brute force approaches to inference infeasible.

Fortunately, the literature on approximate solutions to inference in probabilistic generative models is vast. In the literature, approximate solutions are divided into two *stochastic* and *deterministic* approaches. The stochastic approach to approximate inference relies on sampling methods such that a set of samples represents a desired intractable posterior distri-

bution [22,23]. Sampling methods are quite accurate as they ensure certain guarantees on the posterior distribution. Nevertheless, a major drawback of sampling methods is the computational complexity, as they often require a significant number of samples to describe desired posterior distributions accurately. Although computational complexity is a drawback, it is not justified to disregard sampling methods. For the real-time automated design of signal processing algorithms, sampling methods are generally too slow and hinder the realization of the models in practical applications. Alternatively, deterministic approaches to inference attempt to obtain analytic approximations [24,25]. The main advantage of deterministic approaches is the speed of computations carried out to obtain analytic approximations to posterior distributions. The speed of deterministic methods puts them in a favorable position for applications that require real-time inference and analysis of large data sets. Originated and popularized by Euler and Lagrange [26], variational methods are the forerunner of deterministic approximations to inference, which we will discuss in Chapter 3.

Deterministic and stochastic methods provide different perspectives to the same problem and can be combined to make hybrid inference algorithms to improve inference. Graphical models prove to be a comprehensive framework to unify different classes of inference methods [27]. Graphical representations of generative probabilistic models allow a message-passing interpretation for inference as a unifying principle. Intuitively, a graphical model for a global multivariate function is a visual representation. In this thesis, we will be particularly interested in a class of graphical models called factor graphs [28] and argue that factor graphs provide a valuable framework to perform approximate inference in hierarchical dynamical systems.

In recent years, Bayesian inference communities have been extremely fruitful in implementing inference algorithms in software packages called probabilistic programming packages (PPPs). Probabilistic programming aims to automate and facilitate probabilistic inference for end-users with or without expertise in Bayesian inference. A considerable amount of PPPs, such as Turing.jl [29], Pyro [30] and TensorFlow Probability [31], attain this goal by using stochastic methods. Message passing-based PPPs aim to execute automated Bayesian inference by employing predefined, deterministic rules in a separate probabilistic programming thread. Infer.NET [32], ForneyLab.jl [33] and ReactiveMP.jl [34] are three examples of that thread. The algorithms discussed in this thesis are implemented with ForneyLab.jl and ReactiveMP.jl.

1.2.3 Scoring Models and Constraint Specifications

Accurate evaluation of Bayesian model evidence is a fundamental problem in model development as it forms the basis of the model critique step. Since evidence evaluations are usually intractable, in practice, variational free energy (VFE) minimization provides an attractive alternative, as the VFE is an upper bound on negative model log-evidence (NLE). To improve the tractability of the VFE, it is common to manipulate the constraints in the search space for the posterior distribution of the latent variables. Unfortunately, constraint manipulation may

also lead to a less accurate estimate of the NLE. Thus, constraint manipulation implies an engineering trade-off between tractability and accuracy of model evidence estimation. Interestingly, constraint manipulation has consequences for inference algorithms. This is mainly because the constraints convert the optimization procedure into a constrained optimization procedure such that the steps to find a solution change compared to an unconstrained specification.

This thesis argues that constraint specification adds another dimension to model development in Chapter 3. A model might be dismissed due to a complicated (approximate) inference procedure under a particular set of constraints. For example, assuming independence among variables is a popular constraint choice. Due to independence inference becomes computationally cheaper at the cost of decreased accuracy. A designer might choose to sacrifice accuracy while another designer might not afford to lose accuracy. Then, the designer needs to look for alternative constraints. In Chapter 3 we discuss alternative constraint specifications that are popular in the literature. Criticizing a model without considering the effects of constraints might lead researchers to resort to computationally expensive methods. Nevertheless, manipulating the constraints can allow more straightforward or/and improved inference, effectively making the model more useful in practice.

1.3 Research Questions

The primary purpose of this thesis is to establish a theoretical framework for approximate inference and demonstrate the application of the theory to hierarchical dynamical models. The theoretical framework will explore approximate inference rigorously and provide the required machinery for applying to specific models. The application framework will illustrate how to approximate inference and estimate posterior distributions for the variables in hierarchical dynamical models. The entire space of hierarchical dynamical models is exponentially large, and addressing inference for each model is not practically possible. That is why we will consider some exemplary hierarchical models, extend these models and illustrate potential applications of these models. In this thesis, we will attempt to explore possible answers to the following question:

Q *How can approximate inference and performance evaluation for hierarchical dynamical models be automated and efficiently implemented?*

Investigation of the possible answers to **Q** has two dimensions. The first dimension is automation. In Section 1.1, we motivated our interest in hierarchical models and the need for automated inference. The second dimension is efficient implementation. In Section 1.2, we initiated our argument for searching approximate solutions and briefly mentioned that the

probabilistic programming paradigm offers an implementation level framework. This means that a possible answer to this question will allow a framework that can efficiently implement automated approximate inference. To explore different dimensions of the deriving research question, we will divide it into sub-parts and formulate more minor research questions.

To explore approximate solutions, we will need to present a rigorous definition of inference and how it can be approximated. Our primary tool for this investigation will be *message passing on factor graphs*. Analysis of approximate inference will be closely related to the specification of constraints and will allow us to understand how to measure performance. To that end, the first research question will serve to unify approximate inference and performance evaluation under the message-passing paradigm.

Q1 How can approximate inference algorithms for probabilistic generative models be derived from first principles?

We will present an answer to **Q1** by adhering to a generic high-level recipe as given in [35]. Considering this recipe from a constraint specification view, we will advocate a constraint manipulation strategy to arrive at different message-passing algorithms on factor graphs to automate approximate inference.

To design an agent that can make decisions in a volatile environment, we need to build models that account for time-varying variances. Stochastic volatility (SV) models are examples of such models. Though SV models have been used extensively, these models' inference procedures heavily rely on sampling methods. We look for more generic hierarchical dynamical models than SV models and try to apply message-passing-based inference for modeling in *hierarchically* volatile environments. In the neuroscience literature, the Hierarchical Gaussian Filter (HGF) [36] has been developed as a hierarchical dynamical system in the context of decision making under volatile environments. Although variational inference in the HGF has been investigated, the framework it relies on is not easy to extend upon to use the HGF as a primitive block for more complex hierarchical models.

Q2 How can message passing algorithms for models of volatile systems be efficiently implemented?

Natural processes are often non-stationary, generating signals whose statistical properties change with time. A parameterized transition distribution may govern changes in the statistical properties. If the parameters of this transition distribution are subject to regime switches, then the statistical properties of the transition distribution will depend on the regimes. The next set of research questions will revolve around context switching behavior.

Q3 How can models of volatile environments be equipped with context switching dynamics?

Finally, we illustrate how graphical formalism allows us to build complex models from primitive node structures. Automated hybrid message passing achieves approximate inference to reason about variables in the models. To that end, we formulate our final research question, focused on auto-regressive modeling.

Q4 How can inference on autoregressive models with time-varying noise processes be efficiently implemented?

Answers to these questions illustrate a solution framework to achieving automated approximate inference for hierarchical dynamical models.

1.4 Summary of Contributions

The following list is an overview of the contributions of this thesis:

- A constraint manipulation framework to derive message passing algorithms for factor graphs (Chapter 3);
- Development and improvement of a hierarchical dynamical model capable of modelling volatile environments (Chapter 4);
- Extending the model in Chapter 4 to model non-stationary context switches (Chapter 5);
- Applying the machinery developed in Chapter 3 to the analysis of auto-regressive models with time-varying noise processes by utilizing the node level construction of Chapter 4.

1.5 Outline

Chapter 2 gives background information on graphs, hierarchical models and inference. It is aimed to familiarize the reader with the concepts that will be discussed in future chapters. The contents of the chapter are presented briefly, and the interested reader is referred

to more detailed references. This chapter includes a toy example of message passing on a hidden Markov model to give a sense of the models and inference mechanisms that will be investigated dissertation.

Chapter 3 addresses **Q1** by formulating development of approximate inference algorithms for generative models as a constrained optimization process. The process of optimization is achieved by obtaining stationary solutions to a Lagrangian. Undetermined multipliers of a Lagrangian, which enforce the constraints on the optimization process, are then interpreted as messages flowing on a factor graph. We show how to derive approximate message passing algorithms that operate on local sub-graphs by constraint manipulation. Combining sub-graphs makes it possible to obtain complex graphs where inference in the corresponding model computes local messages.

Chapter 4 addresses questions **Q2**. We take a well-known example of a hierarchical dynamical model from the neuroscience literature capable of modeling hierarchically coupled volatile signals and transferring this model to a factor graph framework. The way we translate into the factor graph formalism starts by isolating individual factors and analyzing inference in these factors on their own. Then we explore different constraint specifications to obtain local message updates for the individual factors. When the isolated factors are combined in sub-graphs, local message update computations blend in and generate hybrid message passing algorithms. We then show the effectiveness of message passing on simulations. This chapter breaks down the HGF as a plug-in for hierarchical modeling purposes.

Chapter 5 addresses question **Q3** by augmenting the HGF with Markovian switching dynamics such that the HGF becomes capable of modeling non-stationarities due to regime switches. We derive closed-form update equations and illustrate how the model, including switching dynamics, improves upon the non-switching counterpart on a financial modeling task.

Chapter 6 addresses question **Q4**. We show that it is possible to perform inference by message-passing in auto-regressive models with time-varying noise processes. We first augment FFGs of auto-regressive models with the composite structure that is defined in Chapter 4.

Finally, in Chapter 7 we conclude the dissertation with a summary and discuss possible improvements to the presented methods for future research.

CHAPTER 2

Preliminaries

"There's no sense in being precise when you don't even know what you're talking about."

–John von Neumann

2.1 Background on Graphs

This chapter will summarize information on graph theoretic concepts and inference in models represented by graphs. The content of this chapter is a distilled mixture of [37, Chapter 2] and [38, Chapter 1] with adjusted notation. The interested reader should refer to these works for a more comprehensive treatment.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is formed by a collection of vertices \mathcal{V} and a collection of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each edge consists of a pair of vertices $(s, t) \in \mathcal{E}$. The edges may be directed ($s \rightarrow t$) or undirected (s, t) . Graphs can represent probability distributions if each vertex $s \in \mathcal{V}$ is associated with a random variable X_s taking values in some space $x_s \in \mathcal{X}_s$. For example, $\mathcal{X}_s = \{0, \dots, r - 1\}$ corresponds to a discrete space and $\mathcal{X}_s = \mathbb{R}$ corresponds to a continuous space. For any subset $A \subseteq \mathcal{V}$, the notation $X_A \triangleq \{X_s | s \in A\}$ corresponds to the subvector indexed by the subset A . Representation of a probability distribution by a graph is not unique and depends on the type of the graph.

2.1.1 Directed Graphs

Given a directed graph with edges $(s \rightarrow t) \in \mathcal{E}$ we say that t is a child of s , or conversely, s is a parent of t . We denote the parents of a vertex s by $\pi(s)$. A sequence (s_1, s_2, \dots, s_k) is a cycle if $(s_i \rightarrow s_{i+1}) \in \mathcal{E}$ for all i and $(s_k \rightarrow s_1)$. We say that a node s is an ancestor of u

if there is a directed path $(s \rightarrow t_1 \rightarrow t_2 \cdots \rightarrow u)$. Suppose that \mathcal{G} is a directed acyclic graph (DAG) without any cycles. For each vertex $s \in \mathcal{V}$ and its parent set $\pi(s)$ let $p_s(x_s|x_{\pi(s)})$ be a nonnegative local function such that $\int p_s(x_s|x_{\pi(s)})dx_s = 1$. These local functions define a probability distribution that can be represented as a *directed graphical model* via:

$$p(\mathbf{x}) = \prod_{s \in \mathcal{V}} p_s(x_s|x_{\pi(s)}), \quad (2.1)$$

2.1.2 Undirected Graphs

If the underlying graph is undirected the probability distribution can be defined over the *cliques* of the graph. A clique C is a fully connected subset of \mathcal{V} such that $(s, t) \in \mathcal{E}$ for all $s, t \in C$. With each clique, a compatibility function can be associated $\psi_C : \otimes_{s \in C} \mathcal{X}_s \rightarrow \mathbb{R}_{>0}$. With these definitions, an *undirected graphical model* can be defined by the normalized product of the compatibility functions:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C). \quad (2.2)$$

where \mathcal{C} is a collection of cliques. An example of a directed graphical models is given Figure 2.1. In the directed graph formalism, factors represent conditional or marginal probability distributions. In contrast, in undirected formalism, the compatibility functions do not directly equate to the cliques' conditional or marginal probability distributions.

2.1.3 Factor Graphs

Factor graph formalism provides an alternative graphical representation, where the emphasis is on the factors. Let \mathcal{F} represent an index set for the factors defining a probability distribution represented by a graphical model. Consider the bipartite graph $\mathcal{G}' = (\mathcal{V}, \mathcal{F}, \mathcal{E}')$, where \mathcal{V} is the original vertex set, and \mathcal{E}' is a new edge set connecting only vertices $s \in \mathcal{V}$ to factors $a \in \mathcal{F}$. This means, an edge $(s, a) \in \mathcal{E}'$ if and only if x_s is an argument to factor indexed by

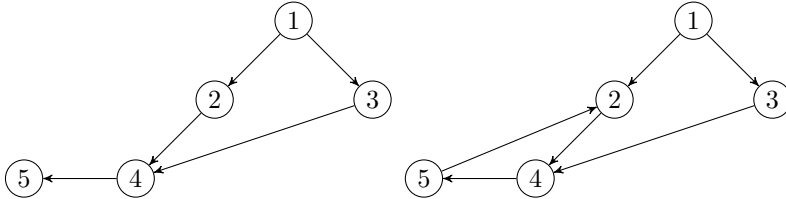


Figure 2.1: (Left) A directed acyclic graph defined with five variables X_1, X_2, X_3, X_4, X_5 . Vertex 5 is a child of vertex 4, and vertex 1 is an ancestor of vertex 5. (Right) A directed graph containing a cycle $(2 \rightarrow 4 \rightarrow 5 \rightarrow 2)$.

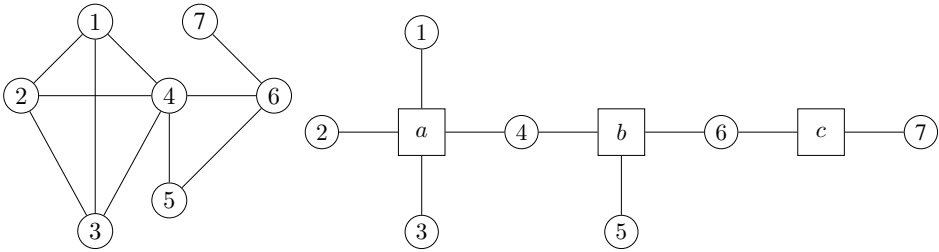


Figure 2.2: (Left) An undirected graph on seven vertices, with maximal cliques $\{1, 2, 3, 4\}$, $\{4, 5, 6\}$ and $\{6, 7\}$. (Right) Equivalent representation of the graph on the left as a factor graph, where the compatibility functions are defined on the maximal cliques. The graph is a bipartite graph with vertex set $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7\}$ and factor index set $\mathcal{F} = \{a, b, c\}$.

$a \in \mathcal{F}$. Illustration of an undirected graphical model and its bipartite counterpart is given in Figure 2.2.

2.1.4 Contingency Tables

Given m random variables X_1, \dots, X_m , each of which having r possible values, an m -dimensional contingency table defines the probability distribution over the random vector (X_1, \dots, X_m) . The contingency table has r^m values and sums to 1. Contingency tables are used in the analysis of categorical data. For example, an important question in data analysis is to identify whether a given set of variables are independent or not. Suppose that we are given $m = 3$ variables. We want to test whether the random variables exhibit independence, pair-wise or three-way interactions. A graphical illustration of this scenario is given in Figure 2.3. This illustration reveals that an undirected formalism can not differentiate between the pairwise and three-way interactions, whereas the factor graph formalism can differentiate between these types of interactions. This simple example illustrates that factor graphs have more representative power than undirected graphical models. That is why we will be working with a specific type of factor graph called a Forney-style factor graph in the remaining of this dissertation. We will introduce Forney-style factor graphs in Chapter 3.

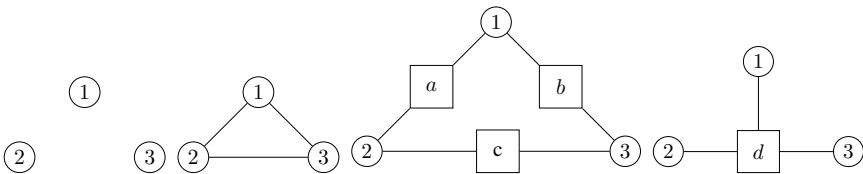


Figure 2.3: (Left) A fully disconnected undirected graph representing independence between variables. (Middle-left) A general dependent model. (Middle-right) Pairwise interactions given by a factor graph. (Right) Triplet interactions.

2.2 Statistical Inference

Given a probability distribution $p(\mathbf{x})$ defined by a graphical model, we will be concerned with the following inference problems:

- Computing marginal distribution $p(x_A)$ over a subset of nodes $A \subset \mathcal{V}$.
- Computing the conditional distributions of the form $p(x_A|x_B)$ for disjoint $A \subset \mathcal{V}$ and $B \subset \mathcal{V}$.
- Computing a mode of the density by solving $\arg \max_{\mathbf{x} \in \mathcal{X}^m} p(\mathbf{x})$.

The first two problems are similar since both of them require marginalization explicitly. The third problem is different as it is a maximization problem and does not require integration.

We can illustrate the computational challenges with inference in a simple example. Let us consider a categorical variable with $\mathcal{X} = \{0, 1, \dots, r-1\}$. Suppose we are interested in obtaining the marginal

$$p_s(x_s) = \sum_{\mathbf{x} \setminus x_s} p(\mathbf{x}). \quad (2.3)$$

Because we sum over the set $\{\mathbf{x}' \in \mathcal{X}^m | x'_s = x_s\}$ with r^{m-1} elements, it is evident that a brute force approach is not feasible. Even when the number of nodes is relatively small $m \approx 100$, the computation is impossible within reasonable time limits. Here we solve a problem over a discrete space, and the situation worsens in the continuous case. When the underlying graph is a tree, the sum-product algorithm recovers the marginals with linear computational complexity in the number of nodes. The junction tree algorithm generalizes the sum-product algorithm when the underlying graphical model has cycles. However, the computational complexity of the junction tree algorithm is exponential in the treewidth of the graph.

2.2.1 Sum-product Algorithm for Trees

We suppose that the underlying graph is a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}(\mathcal{T}))$. For an arbitrary vertex $s \in \mathcal{V}$ we consider the set of neighboring nodes defined by

$$N(s) \triangleq \{u \in \mathcal{V} | (s, u) \in \mathcal{E}(\mathcal{T})\}. \quad (2.4)$$

For each $u \in N(s)$, we denote the subgraph formed by the set of nodes that can be reached from u by paths that *do not* pass through s with $\mathcal{T}_u = (\mathcal{V}_u, \mathcal{E}_u)$. Because the underlying graph is tree, the subgraphs \mathcal{T}_u are also trees. With a subtree \mathcal{T}_s we can associate a subvector $x_{\mathcal{V}_s} \triangleq \{x_u | u \in \mathcal{V}_s\}$ formed by the vertex set of the subtree. Because the cliques of a tree graph are the individual nodes and edges, we can write the probability distribution associated with the subvector by using (2.2) as

$$p(x_{\mathcal{V}_s}; \mathcal{T}_s) \propto \prod_{u \in \mathcal{V}_s} \psi_u(x_u) \prod_{(u,v) \in \mathcal{E}_s} \psi_{uv}(x_u, x_v). \quad (2.5)$$

Without loss of generality suppose that we are interested in obtaining marginal at node s in a tree \mathcal{T} . Since the underlying graph is a tree, we can partition the graph into subtrees \mathcal{T}_u for each neighboring node $u \in N(s)$ being at the root of \mathcal{T}_u . All the subtrees are disconnected from each other if the node s is removed from \mathcal{T} . Hence, given the node s the vectors $x_{\mathcal{V}_w}$ and $x_{\mathcal{V}_u}$ are conditionally independent for every $u \neq w \in N(s)$. This property of trees allow computation of the marginal in the following way

$$p_s(x_s) = \frac{1}{Z_s} \psi_s(x_s) \prod_{w \in N(s)} \mu_{ws}^*(x_s) \quad (2.6a)$$

$$\mu_{ws}^*(x_s) \triangleq \sum_{x'_{\mathcal{V}_w}} \psi_{sw}(x_s, x'_w) p(x'_{\mathcal{V}_w}; \mathcal{T}_w) \quad (2.6b)$$

where $\mu_{ws}^*(x_s)$ is a function of the possible states $x_s \in \mathcal{X}_s$ and Z_s is a normalization constant to ensure that the marginal $p_s(x_s)$ is normalized to 1. Conditional independence breaks down the marginal computation into summations (2.6b) over smaller trees \mathcal{T}_w than the original tree \mathcal{T} . The advantage of breaking down into smaller trees is that the subproblem (2.6b) is guaranteed to be the fixed point of the following recursion

$$\mu_{ws}(x_s) \propto \sum_{x'_w} \psi_{sw}(x_s, x'_w) \prod_{u \in N(w) \setminus s} \mu_{uw}(x'_w). \quad (2.7)$$

The sum-product algorithm can compute the marginals simultaneously. At each iteration, each node passes a *message* (2.7) to each of its neighbours. Each edge has two associated messages for each direction and consequently, there are $2|\mathcal{E}(\mathcal{T})|$ messages for the entire graph \mathcal{T} . For tree graphs these $2|\mathcal{E}(\mathcal{T})|$ messages will converge to $\mu^* \triangleq \{\mu_{su}^*, \mu_{us}^* | \text{for every } (s, u) \in \mathcal{E}(\mathcal{T})\}$. Collectively, μ^* will be the solution to all of the sub-problems defined by (2.6b). We will revisit the sum-product algorithm in Chapter 3 and derive it from first principles. Moreover, we will build the foundations for approximate message passing methods and show how we can derive customized message passing algorithms.

2.3 Hierarchical Models

The Bayesian approach to modeling treats all model variables such as observed data, states, and parameters as random variables. Often, prior specification involves extra parameters called hyperparameters. A hierarchical model then expresses conditional probabilities linking hyperparameters, parameters, and data. Graphical models give convenient representations for hierarchical models, and the benefits are as follows. Hierarchical models usually include conditionally independent structures. Graphic treatment of hierarchical models provides a systematic method to check all these independent relationships. Another advantage is that the visualization of a model can help understand the proposed model better and offer possible

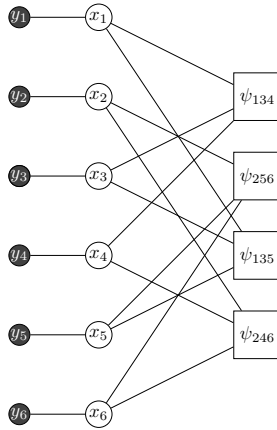


Figure 2.4: A factor graph example of a parity check code of length 6. The dark circles indicate observed bits y_t . Light circles denote the unobserved bits. Finally, rectangles correspond to parity factors. In this example each parity bit is connected to 2 factors and each parity factor involves 3 parity bits.

extensions. Next, we give an example of a hierarchical model often used in communication theory for decoding purposes.

Example 1. In communication theory, a central problem is defining good codewords to decrease decoding errors caused by noisy channels. A heavily utilized strategy is to add redundancy to the transmitted bit sequence. The graphical formalism allows a convenient framework to build codes. Parity check codes are examples of codes that are represented by factor graphs. The parity relation is given by addition in modulo two as follows

$$\psi_{swu}(x_s, x_w, x_u) \triangleq \begin{cases} 1 & \text{if } x_s \oplus x_w \oplus x_u = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

Figure 2.4 illustrates a very small parity check code. The white circles represent the bits that make up the codeword, whereas the shaded circles represent observed noisy bits. The rectangles represent the compatibility (parity) functions. In this example the parity checks are given over the triplets $\{1, 3, 4\}, \{1, 3, 5\}, \{2, 4, 6\}$, and $\{2, 5, 6\}$. The decoding problem corresponds to estimating the transmitted codeword based on the noisy observations.

2.4 Time Series

A *time series* is defined to be a collection of data that can be represented as a sequence indexed by time. An example is financial data, where the index indicates time. On the other hand index of a sequence of genetic data has no temporal meaning. This dissertation will focus on discrete-time probabilistic models for time series. To define a probabilistic model

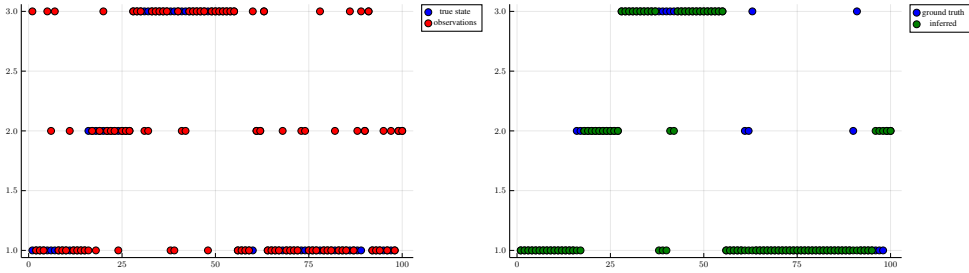


Figure 2.5: (Left) An example of states and noisy observations generated synthetically according to an order one HMM. Blue dots represent ground truth states, and red dots represent the noisy observations. (Right) Result of sum-product message passing on HMM with noisy observations given in the left plot. Blue dots represent the ground truth, and green dots represent the mode of the posterior obtained by sum-product message passing.

for a time series $\mathbf{y} \triangleq \{y_1, y_2, \dots, y_T\}$, specification of a joint distribution $p(\mathbf{y})$ is necessary. To define a joint distribution, one might be tempted to specify all the independent entries of $p(\mathbf{y})$. Nevertheless, this attempt is infeasible. Consider binary data $y_t \in \{0, 1\}$. Then, the joint distribution contains $2^T - 1$ independent entries maximally. Unless T is fairly small, this approach is practically not plausible. Often simplifying assumptions are made to ensure tractability for the joint distribution. Statistical independence is one of the strongest forms of assumptions paving the way for a tractable joint description. A popular way to achieve statistical independence is through conditioning. Using the Bayes rule, we can write

$$p(y_T | y_{1:T-1}) = \frac{p(\mathbf{y})}{p(y_{1:T-1})}. \quad (2.9)$$

Similarly we can carry out the same steps for $p(y_{1:T-1})$ and by induction we can decompose the joint distribution $p(\mathbf{y})$ as

$$p(\mathbf{y}) = \prod_{t=1}^T p(y_t | y_{1:t-1}). \quad (2.10)$$

Equation (2.10) specifies the joint distribution in terms of factors representing a generative model of a variable conditioned on the past variables. Conditioning on the past is consistent with the causal nature of time. Further simplifications can be made by introducing extra conditional dependencies. A popular constraining simplification can be achieved by setting $p(y_t | y_{1:t-1}) = p(y_t | y_{t-M:t-1})$. This assumption is known as an M^{th} order Markov assumption. The following example introduces a *hierarchical dynamical model* that is known as a hidden Markov Model.

Example 2. Consider an M^{th} order Markov chain specified by the following factorization corresponding to a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$p(\mathbf{x}) = p(x_{0:M-1}) \prod_{t=M}^T p(x_t | x_{t-M:t-1}). \quad (2.11)$$

The factors in (2.11) represent transition densities conditioned on the parent set of each variable over time. Cliques of the underlying undirected graphical model are given by the sets of the form $\{t - M, \dots, t\}$ and the separators of the cliques are of the sets $\{t - M, \dots, t - 1\}$ for every $t = M, M + 1, \dots, T$. The probability distribution $p(\mathbf{x})$ can not be written as a product of clique marginals only. However, if we include the separator marginals, we can write the factorization of $p(\mathbf{x})$ in terms of clique marginals and separator marginals.

A hidden Markov model (HMM) is a particular realization of a Markov chain (2.11) when the states are discrete, and the state transitions are governed by a stochastic matrix. HMMs are popular in speech and language processing. To make matters concrete, let us assume that we have an $M = 1$ order HMM with the following state transition and likelihood specifications

$$p(x_t|x_{t-1}) = \text{Cat}(x_t|Ax_{t-1}) \quad (2.12a)$$

$$p(y_t|x_t) = \text{Cat}(y_t|Bx_t), \quad (2.12b)$$

where y_t represents noisy observations (data) of the state x_t having values in $\mathcal{X}_t = \{1, 2, 3\}$. A and B are stochastic matrices assumed to be known as

$$A = \begin{bmatrix} 0.9 & 0.0 & 0.1 \\ 0.1 & 0.9 & 0.0 \\ 0.0 & 0.1 & 0.9 \end{bmatrix} \quad B = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.05 & 0.8 & 0.15 \\ 0.15 & 0.05 & 0.85 \end{bmatrix}. \quad (2.13)$$

Moreover, we assume that the prior is $p(x_0) = \text{Cat}(x_0|[1/3, 1/3, 1/3])$. An example of noisy observations y_t generated from x_t is given in Figure 2.5. Given a set of noisy observations (data), we are interested in obtaining the posterior for the underlying hidden states $p(x_t|\mathbf{y})$ given a set of observations $\mathbf{y} = \{y_1, \dots, y_T\}$. Since the underlying graph is a tree, we can utilize sum-product message passing to obtain the desired marginals at every node. That is, we are interested in computing (2.7) for every node. Right plot of Figure 2.5 displays the result of sum-product message passing for the observations given in left plot of Figure 2.5.

CHAPTER 3

Message Passing and Local Constraint Manipulation in Factor Graphs

"I have approximate answers, and possible beliefs, and different degrees of certainty about different things, but I am not absolutely sure of anything."

–Richard Phillips Feynmann

3.1 Introduction

Building models from data is at the core of both science and engineering applications. The search for suitable models requires a performance measure that scores how well a particular model m captures the hidden patterns in a data set D . In a Bayesian framework, that measure is the *Bayesian evidence* $p(D|m)$, i.e., the probability that model m would generate D if we were to draw data from m . The art of modeling is then the iterative process of proposing new model specifications, evaluating the evidence for each model, and retaining the model with the highest evidence [39].

Unfortunately, Bayesian evidence is intractable for most exciting models. A popular solution to evidence evaluation is provided by *variational* inference, which describes the process of Bayesian evidence evaluation as a (free energy) minimization process since the variational free energy (VFE) is a tractable upper bound on Bayesian (negative log-)evidence [6]. In practice, the model development process then consists of proposing various candidate models, minimizing VFE for each model, and selecting the model with the lowest minimized

VFE.

The difference between VFE and negative log-evidence (NLE) is equal to the Kullback-Leibler divergence (KLD) [40] from the (perfect) Bayesian posterior distribution to the variational distribution for the latent variables in the model. The KLD can be interpreted as the cost of making variational rather than Bayesian inference. Perfect (Bayesian) inference would lead to zero inference costs (KLD= 0), and the KLD increases as the variational posterior diverge further from the Bayesian posterior. As a result, model development in a variational inference context is a balancing act. We search for models with considerable evidence for the data and small inference costs (small KLD). In other words, in a variational inference context, the researcher has two knobs to tune models. The first knob alters the model specification, which affects model evidence. The second knob constrains the search space for the variational posterior, which may affect the inference costs.

In this chapter, we are concerned with developing algorithms for tuning the second knob. How do we constrain the range of variational posteriors to make variational inference both tractable and accurate (i.e., resulting in low KLD)? We present our framework in the context of a (Forney-style) factor graph representation of the model [28, 41]. In that context, variational inference can be understood as an automatable and efficient message passing-based inference procedure [33, 42, 43].

Traditional constraints include mean-field [42] and Bethe approximations [44]. However, more recently it has become clear how alternative local constraints, such as posterior factorization [45], expectation and chance constraints [46, 47], and local Laplace approximation [48] may impact both tractability and inference accuracy, and thereby potentially lead to lower VFE. The main contribution of the current work lies in unifying the various ideas on local posterior constraints into a principled method for deriving variational message passing-based inference algorithms. The proposed method derives existing message-passing algorithms and supports the development of new message-passing variants.

Sec. 3.2 reviews Forney-style Factor Graphs (FFGs) and variational inference by minimizing the Bethe Free Energy (BFE). This review is continued in Sec. 3.3, where we discuss BFE optimization from a Lagrangian optimization viewpoint. In Appendix A.1 we include an example to illustrate that the Bayes rule can be derived from Lagrangian optimization with data constraints. Our main contribution lies in Sec. 3.4, which provides a rigorous treatment of the effects of imposing local constraints on the BFE and the resulting message update rules. We build upon several previous works that describe how manipulation of (local) constraints and variational objectives can be employed to improve variational approximations in the context of message passing. For example, [46] shows how inference algorithms can be unified in terms of hybrid message passing by Lagrangian constraint manipulation. We extend this view by bringing form (Sec. 3.4.2) and factorization constraints (Sec. 3.4.1) into a constrained optimization framework. In [35], a high-level recipe for generating message-passing algorithms from divergence measures is described. We apply their general recipe in the current work, where we adhere to the view on local stationary points for region-based approximations on general graphs [49]. In Appendix A.2, we also show that locally stationary

solutions are also the global stationary solutions. In Sec. 3.5, we develop an algorithm for VFE evaluation in an FFG. In previous work, [50] describes a factor softening approach to evaluate the VFE for models with deterministic factors. We extend this work in Sec. 3.5 and show how to avoid factor softening for free energy evaluation and inference of posteriors. We show an example of how to compute VFE for a deterministic node in Appendix A.3. A more detailed comparison to related work is deferred to Sec. 3.7.

In the literature, proofs and descriptions of message passing-based inference algorithms are scattered across multiple papers and varying graphical representations, including Bayesian networks [42, 51], Markov random fields [49], bi-partite (Tanner) factor graphs [46, 50, 52] and Forney-style factor graphs (FFGs) [41, 45]. In Appendix A.4, we provide first-principle proofs for a large collection of familiar message passing algorithms in the context of Forney-style factor graphs, which is the preferred framework in the information and communication theory communities [28, 53].

3.2 Factor Graphs and Bethe Free Energy

3.2.1 Forney-style Factor Graphs

A Forney-style factor graph (FFG) is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes \mathcal{V} and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote the neighboring edges of a node $a \in \mathcal{V}$ by $\mathcal{E}(a)$. Vice versa, for an edge $i \in \mathcal{E}$, the notation $\mathcal{V}(i)$ collects all neighboring nodes. As a notational convention, we index nodes by a, b, c and edges by i, j, k , unless stated otherwise. We will mainly use a and i as summation indices, and use the other indices to refer to a node or edge of interest.

In this chapter, we will frequently refer to the notion of a subgraph. We define an edge-induced subgraph by $\mathcal{G}(i) = (\mathcal{V}(i), i)$, and a node-induced subgraph by $\mathcal{G}(a) = (a, \mathcal{E}(a))$. Furthermore, we denote a local subgraph by $\mathcal{G}(a, i) = (\mathcal{V}(i), \mathcal{E}(a))$, which collects all local nodes and edges around i and a respectively.

An FFG can be used to represent a factorized function,

$$f(\mathbf{s}) = \prod_{a \in \mathcal{V}} f_a(\mathbf{s}_a), \quad (3.1)$$

where \mathbf{s}_a collects the argument variables of factor f_a . We assume that all the factors are positive valued. In an FFG, a node $a \in \mathcal{V}$ corresponds to a factor f_a , and the neighboring edges $\mathcal{E}(a)$ correspond to the variables \mathbf{s}_a that are the arguments of f_a .

As an example model, the following factorization (3.2), for which the corresponding FFG is shown in Fig. 3.1.

$$f(s_1, \dots, s_5) = f_a(s_1) f_b(s_1, s_2, s_3) f_c(s_2) f_d(s_3, s_4, s_5) f_e(s_5). \quad (3.2)$$

The FFG of Fig. 3.1 consists of five nodes $\mathcal{V} = \{a, \dots, e\}$ as annotated by their corresponding factor functions, and five edges $\mathcal{E} = \{(a, b), \dots, (d, e)\}$ as annotated by their correspond-

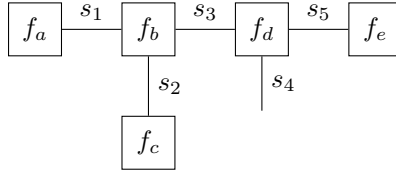


Figure 3.1: Example Forney-style factor graph for the model of (3.2).

ing variables. An edge that connects to only one node (e.g., the edge for s_4) is called a half-edge. In this example, the neighborhood $\mathcal{E}(b) = \{(a, b), (b, c), (b, d)\}$, and $\mathcal{V}((b, c)) = \{b, c\}$.

In the FFG representation, a node can be connected to an arbitrary number of edges, while an edge can only be connected to at most two nodes. Therefore, FFGs often contain “equality nodes” that constrain connected edges to carry identical beliefs, with the implication that these beliefs can be made available to more than two factors. An equality node has the factor function

$$f_a(s_i, s_j, s_k) = \delta(s_j - s_i) \delta(s_j - s_k), \quad (3.3)$$

for which the node-induced subgraph $\mathcal{G}(a)$ is drawn in Fig. 3.2.

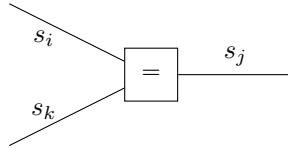


Figure 3.2: Visualization of the node-induced subgraph for an equality node. If the node function f_a is known, a symbol representing the node function is often substituted within the node (“=” in this case).

If every edge in the FFG has exactly two connected nodes (including equality nodes), then we designate the graph as a terminated FFG (TFFG). Since multiplication of a function $f(s)$ by 1 does not alter the function, any FFG can be terminated by connecting any half-edge i to a node a that represents the unity factor $f_a(s_i) = 1$.

In Sec. 3.4.2 we discuss form constraints on posterior distributions. If such a constraint takes on a Dirac-delta functional form, we visualize the constraint on the FFG by a small circle in the middle of the edge. For example, the small shaded circle in Fig. 3.11 indicates that the variable has been observed. In Sec. 3.4.3.2 we consider form constraints in the context of optimization, in which case the circle annotation will be left open (see, e.g., Fig. 3.14).

3.2.2 Variational Free Energy

Given a model $f(\mathbf{s})$ and a (normalized) probability distribution $q(\mathbf{s})$, we can define a Variational Free Energy (VFE) functional as

$$F[q, f] \triangleq \int q(\mathbf{s}) \log \frac{q(\mathbf{s})}{f(\mathbf{s})} d\mathbf{s}. \quad (3.4)$$

Variational inference is concerned with finding solutions to the minimization problem

$$q^*(\mathbf{s}) = \arg \min_{q \in \mathcal{Q}} F[q, f], \quad (3.5)$$

where \mathcal{Q} imposes some constraints on q .

If q is unconstrained, then the optimal solution is obtained for $q^*(\mathbf{s}) = p(\mathbf{s})$, with $p(\mathbf{s}) = \frac{1}{Z} f(\mathbf{s})$ the exact posterior, and $Z = \int f(\mathbf{s}) d\mathbf{s}$ a normalizing constant that is commonly referred to as the evidence. The minimum value of the free energy then follows as the negative log-evidence (NLE),

$$F[q^*, f] = -\log Z,$$

which is also known as the surprisal. The NLE can be interpreted as a measure of model performance, where low NLE is preferred.

Because an unconstrained search space for q grows exponentially with the number of variables, the optimization of (3.5) quickly becomes intractable beyond the most basic models. Therefore, constraints and approximations to the variational free energy (3.4) are often utilized. As a result, the *constrained* variational free energy with $q^* \in \mathcal{Q}$ bounds the NLE by

$$F[q^*, f] = -\log Z + \int q^*(\mathbf{s}) \log \frac{q^*(\mathbf{s})}{p(\mathbf{s})} d\mathbf{s}, \quad (3.6)$$

where the latter term expresses the divergence from the (intractable) exact solution to the optimal variational belief.

In practice, the functional form of $q(\mathbf{s}) = q(\mathbf{s}; \boldsymbol{\theta})$ is often parameterized, such that gradients of F can be derived w.r.t. the parameters $\boldsymbol{\theta}$. This effectively converts the variational optimization of $F[q, f]$ to a parametric optimization of $F(\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$. This problem can then be solved by a (stochastic) gradient descent procedure [54, 55].

In the context of variational calculus, while form constraints may lead to interesting properties (see Sec. 3.4.2), they are generally not required. Interestingly, in a variational optimization context, the functional form of q is often not an *assumption*, but rather a *result* of optimization (see Sec. 3.4.3.1). An example of variational inference is provided in Appendix A.1.

3.2.3 Approximations to Variational Free Energy

The Bethe approximation enjoys a unique place in the landscape of \mathcal{Q} because the Bethe free energy (BFE) defines the fundamental objective of the celebrated belief propagation (BP) algorithm [50, 56]. The origin of the Bethe approximation is rooted in tree-like approximations

to subgraphs (possibly containing cycles) by enforcing local consistency conditions on the beliefs associated with edges and nodes [37].

Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for a factorized function $f(\mathbf{s}) = \prod_{a \in \mathcal{V}} f_a(\mathbf{s}_a)$ (3.1), the Bethe free energy (BFE) is defined as [57]:

$$F[q, f] \triangleq \sum_{a \in \mathcal{V}} \underbrace{\int q_a(\mathbf{s}_a) \log \frac{q_a(\mathbf{s}_a)}{f_a(\mathbf{s}_a)} d\mathbf{s}_a}_{F[q_a, f_a]} + \sum_{i \in \mathcal{E}} \underbrace{\int q_i(s_i) \log \frac{1}{q_i(s_i)} ds_i}_{H[q_i]} \quad (3.7)$$

such that the factorized beliefs

$$q(\mathbf{s}) = \prod_{a \in \mathcal{V}} q_a(\mathbf{s}_a) \prod_{i \in \mathcal{E}} q_i(s_i)^{-1} \quad (3.8)$$

satisfy the following constraints:

$$\int q_a(\mathbf{s}_a) d\mathbf{s}_a = 1, \quad \text{for all } a \in \mathcal{V} \quad (3.9a)$$

$$\int q_a(\mathbf{s}_a) d\mathbf{s}_{a \setminus i} = q_i(s_i), \quad \text{for all } a \in \mathcal{V} \text{ and all } i \in \mathcal{E}(a). \quad (3.9b)$$

Together, the normalization constraints (3.9a) and marginalization constraints (3.9b) imply that the edge marginals are also normalized:

$$\int q_i(s_i) ds_i = 1, \quad \text{for all } i \in \mathcal{E}. \quad (3.10)$$

The Bethe free energy (3.7) includes a local free energy term $F[q_a, f_a]$ for each node $a \in \mathcal{V}$, and an entropy term $H[q_i]$ for each edge $i \in \mathcal{E}$. Note that the local free energy also depends on the node function f_a as specified in the factorization of f (3.1), whereas the entropy only depends on the local belief q_i .

The Bethe factorization (3.8) and constraints are summarized by the local polytope [58]

$$\mathcal{L}(\mathcal{G}) = \{q_a \text{ for all } a \in \mathcal{V} \text{ s.t. (3.9a), and } q_i \text{ for all } i \in \mathcal{E}(a) \text{ s.t. (3.9b)}\}, \quad (3.11)$$

which defines the constrained search space for the factorized variational distribution (3.8).

3.2.4 Problem Statement for Approximate Inference

In this chapter, the problem is to find the beliefs in the local polytope that minimize the Bethe free energy

$$q^*(\mathbf{s}) = \arg \min_{q \in \mathcal{L}(\mathcal{G})} F[q, f], \quad (3.12)$$

where q is defined by (3.8), and where $q \in \mathcal{L}(\mathcal{G})$ offers a shorthand notation for optimizing over the individual beliefs in the local polytope. In the following sections we will follow the Lagrangian optimization approach to derive various message passing-based inference algorithms.

3.2.5 Sketch of Solution Approach

The problem statement of Sec. 3.2.4 defines a global minimization of the beliefs in the Bethe factorization. Instead of solving the global optimization problem directly, we employ the factorization of the variational posterior and local polytope to subdivide the global problem statement in multiple *interdependent* local objectives.

From the BFE objective (3.12) and local polytope of (3.11), we can construct the Lagrangian

$$\begin{aligned}
 L[q, f] = & \sum_{a \in \mathcal{V}} F[q_a, f_a] + \sum_{a \in \mathcal{V}} \psi_a \left[\int q_a(\mathbf{s}_a) d\mathbf{s}_a - 1 \right] \\
 & + \sum_{a \in \mathcal{V}} \sum_{i \in \mathcal{E}(a)} \int \lambda_{ia}(s_i) \left[q_i(s_i) - \int q_a(\mathbf{s}_a) ds_{a \setminus i} \right] ds_i \\
 & + \sum_{i \in \mathcal{E}} H[q_i] + \sum_{i \in \mathcal{E}} \psi_i \left[\int q_i(s_i) ds_i - 1 \right], \tag{3.13}
 \end{aligned}$$

where the Lagrange multipliers ψ_a , ψ_i and λ_{ia} enforce the normalization and marginalization constraints of (3.9). It can be seen that this Lagrangian contains local beliefs q_a and q_i , which are coupled through the λ_{ia} Lagrange multipliers. The Lagrange multipliers λ_{ia} are doubly indexed, because there is a multiplier associated with each marginalization constraint. The Lagrangian method then converts a constrained optimization problem of $F[q, f]$ to an unconstrained optimization problem of $L[q, f]$. The total variation of the Lagrangian (3.13) can then be approached from the perspective of variations of the individual (coupled) local beliefs.

More specifically, given a locally connected pair $b \in \mathcal{V}, j \in \mathcal{E}(b)$, we can rewrite the optimization of (3.12) in terms of the local beliefs q_b, q_j , and the constraints in the local polytope

$$\mathcal{L}(\mathcal{G}(b, j)) = \{q_b \text{ s.t. (3.9a), and } q_j \text{ s.t. (3.9b)}\}, \tag{3.14}$$

that pertains to these beliefs. The problem then becomes to find local stationary solutions

$$\{q_b^*, q_j^*\} = \arg \min_{\mathcal{L}(\mathcal{G}(b, j))} F[q, f]. \tag{3.15}$$

Using (3.13), the optimization of (3.15) can then be written in the Lagrangian form

$$q_b^* = \arg \min_{q_b} L_b[q_b, f_b], \tag{3.16a}$$

$$q_j^* = \arg \min_{q_j} L_j[q_j], \tag{3.16b}$$

where the Lagrangians L_b and L_j include the local polytope of (3.14) to rewrite (3.13) as an explicit functional of beliefs q_b and q_j (see e.g. Lemma 3.1 and 3.2). The combined

stationary solutions to the local objectives then also comprise a stationary solution to the global objective (App. A.2).

The current chapter shows how to identify stationary solutions to local objectives of the form (3.15), with the use of variational calculus, under varying constraints as imposed by the local polytope (3.14). Interestingly, the resulting fixed-point equations can be interpreted as message passing updates on the underlying TFFG representation of the model. In the following sections 3.3 and 3.4 we derive the local stationary solutions under a selection of constraints, and show how these relate to known message-passing update rules (Table 3.1). It becomes possible to derive novel message updates and algorithms by simply altering the local polytope.

3.3 Bethe Lagrangian Optimization by Message Passing

3.3.1 Stationary Points of the Bethe Lagrangian

We wish to minimize the Bethe free energy under variations of the variational density. Because the Bethe free energy factorizes over factors and variables (3.7), we first consider variations on a separate node- and edge-induced subgraphs.

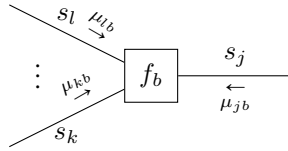


Figure 3.3: The subgraph around node b with indicated messages. Ellipses indicate an arbitrary (possibly zero) amount of edges.

Lemma 3.1. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the node-induced subgraph $\mathcal{G}(b)$ (Fig. 3.3). The stationary points of the Lagrangian (3.16a) as a functional of q_b ,

$$L_b[q_b, f_b] = F[q_b, f_b] + \psi_b \left[\int q_b(\mathbf{s}_b) d\mathbf{s}_b - 1 \right] + \sum_{i \in \mathcal{E}(b)} \int \lambda_{ib}(s_i) \left[q_i(s_i) - \int q_b(\mathbf{s}_b) ds_{b \setminus i} \right] ds_i + C_b, \quad (3.17)$$

where C_b collects all terms that are independent of q_b , are of the form

$$q_b(\mathbf{s}_b) = \frac{f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i)}{\int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b}. \quad (3.18)$$

Proof. See Appendix A.4.1. □

The $\mu_{ib}(s_i)$ are any set of positive functions that makes (3.18) satisfy (3.9b), and will be identified in Theorem 3.1.

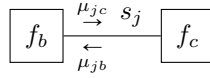


Figure 3.4: An edge-induced subgraph $\mathcal{G}(j)$ with indicated messages.

Lemma 3.2. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider an edge-induced subgraph $\mathcal{G}(j)$ (Fig. 3.4). The stationary points of the Lagrangian (3.16b) as a functional of q_j ,

$$L_j[q_j] = H[q_j] + \psi_j \left[\int q_j(s_j) ds_j - 1 \right] + \sum_{a \in \mathcal{V}(j)} \int \lambda_{ja}(s_j) \left[q_j(s_j) - \int q_a(s_a) ds_{a \setminus j} \right] ds_j + C_j, \quad (3.19)$$

where C_j collects all terms that are independent of q_j , are of the form

$$q_j(s_j) = \frac{\mu_{jb}(s_j)\mu_{jc}(s_j)}{\int \mu_{jb}(s_j)\mu_{jc}(s_j) ds_j}. \quad (3.20)$$

Proof. See Appendix A.4.2. □

Example 3. (Linear Dynamical System) Consider a linear Gaussian state space model specified by the following factors:

$$g_0(x_0) = \mathcal{N}(x_0 | m_{x_0}, V_{x_0}) \quad (3.21a)$$

$$g_t(x_{t-1}, z_t, A_t) = \delta(z_t - A_t x_{t-1}) \quad (3.21b)$$

$$h_t(x'_t, z_t, Q_t) = \mathcal{N}(x'_t | z_t, Q_t^{-1}) \quad (3.21c)$$

$$n_t(x_t, x'_t, x''_t) = \delta(x_t - x'_t) \delta(x_t - x''_t) \quad (3.21d)$$

$$m_t(o_t, x''_t, B_t) = \delta(o_t - B_t x''_t) \quad (3.21e)$$

$$r_t(y_t, o_t, R_t) = \mathcal{N}(y_t | o_t, R_t^{-1}). \quad (3.21f)$$

The FFG corresponding to the one-time segment of the state space model is given in Figure 3.5. We assume that we know the following matrices that are used to generate the data:

$$\hat{A}_t = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \hat{Q}_t^{-1} = \begin{bmatrix} 3 & 0.1 \\ 0.1 & 2 \end{bmatrix}, \hat{B}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \hat{R}_t^{-1} = \begin{bmatrix} 10 & 2 \\ 2 & 20 \end{bmatrix} \quad (3.22)$$

with $\theta = \pi/8$. Given a collection of observations $\hat{\mathbf{y}} = \{\hat{y}_1, \dots, \hat{y}_T\}$, we constrain the latent states $\mathbf{x} = \{x_0, \dots, x_T\}$ by local marginalization and normalization constraints (for brevity we omit writing the normalization constraints explicitly) in accordance with Theorem 3.1, i.e.,

$$\int q(x_{t-1}, z_t, A_t) dx_{t-1} dz_t = q(A_t), \quad \int q(x_{t-1}, z_t, A_t) dA_t = q(z_t | x_{t-1}) q(x_{t-1}) \quad (3.23a)$$

$$\int q(x'_t, z_t, Q_t) dx'_t dz_t = q(Q_t), \quad \int q(x'_t, z_t, Q_t) dz_t dQ_t = q(x'_t)$$

$$\int q(x'_t, z_t, Q_t) dx'_t dQ_t = q(z_t) \quad (3.23b)$$

$$q(x_t, x'_t, x''_t) = q(x_t) \delta(x_t - x'_t) \delta(x_t - x''_t) \quad (3.23c)$$

$$\int q(o_t, x''_t, B_t) do_t, dx''_t = q(B_t), \quad \int q(o_t, x''_t, B_t) dB_t = q(o_t | x''_t) q(x''_t) \quad (3.23d)$$

$$\int q(o_t, y_t, R_t) do_t dy_t = q(R_t), \quad \int q(o_t, y_t, R_t) dR_t do_t = q(y_t)$$

$$\int q(o_t, y_t, R_t) dR_t dy_t = q(o_t) \quad (3.23e)$$

Moreover, we use data-constraints in accordance with Theorem 3.3 (explained in Section 3.4.2.1) for the observations, state transition matrices and precision matrices, i.e.,

$$q(y_t) = \delta(y_t - \hat{y}_t), \quad q(A_t) = \delta(A_t - \hat{A}_t), \quad q(B_t) = \delta(B_t - \hat{B}_t)$$

$$q(Q_t) = \delta(Q_t - \hat{Q}_t), \quad q(R_t) = \delta(R_t - \hat{R}_t).$$

Computation of sum-product messages by (3.25) is analytically tractable and detailed algebraic manipulation can be found in [59]. If the backward messages are not passed, the resulting sum-product message passing algorithm is equivalent to Kalman filtering. If both forward and backward messages are propagated, then the Rauch-Tung-Striebel smoother is obtained [60, Ch. 8].

We generated $T = 100$ observations $\hat{\mathbf{y}}$ using the matrices specified in (3.22) and the initial condition $\hat{x}_0 = [5, -5]^\top$. Due to (3.21a) we have $\mu_{x_0, g_1} = \mathcal{N}(m_{x_0}, V_{x_0})$. We choose $V_{x_0} = 100 \cdot \mathbf{I}$ and $m_{x_0} = \hat{x}_0$. Under these constraints, the results of sum-product message passing and Bethe free energy evaluation is given in Figure 3.5. Because the underlying graph is a tree, sum-product message passing results are exact and the evaluated BFE corresponds to negative log-evidence. In the follow-up example 4, we will modify the constraints and give a comparative free energy plot for the examples in Figure 3.10 and 3.16.

3.3.2 Minimizing the Bethe Free Energy by Belief Propagation

We now combine Lemmas 3.1 and 3.2 to derive the sum-product message update.

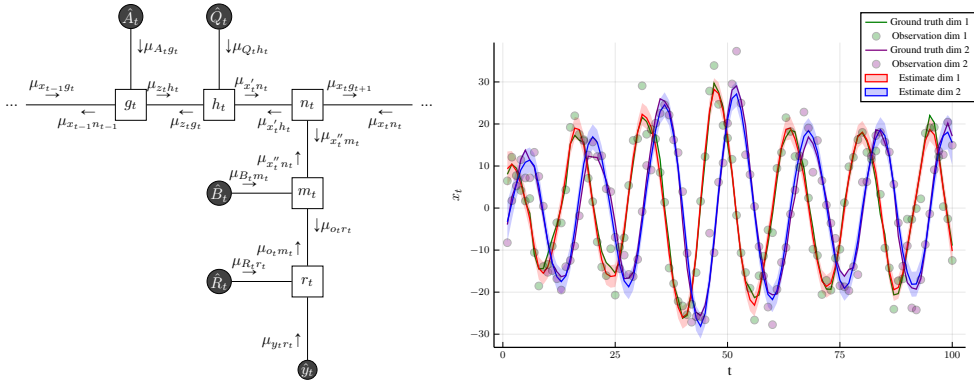


Figure 3.5: (Left) One time-segment of the FFG corresponding to the linear Gaussian state space model specified in Example 3 with the sum-product messages computed according to (3.25). The three small dots at both sides of the graph indicate identical continuation of the graph over time. (Right) The small dots indicate the noisy observations that are synthetically generated by the linear state space model of (3.21) using parameter matrices as specified in (3.22). The posterior distribution for the hidden states are inferred by sum-product message passing and are drawn with shaded regions indicating plus and minus the variance. The Bethe free energy evaluates to $F[q, f] = 580.698$.

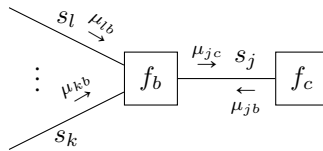


Figure 3.6: Visualization of a subgraph with indicated sum-product messages.

Theorem 3.1. (Sum-Product Message Update). Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.6). Given the local polytope $\mathcal{L}(\mathcal{G}(b, j))$ of (3.14), then the local stationary solutions to (3.15) are given by

$$q_b^*(s_b) = \frac{f_b(s_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i)}{\int f_b(s_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i) ds_b} \quad (3.24a)$$

$$q_j^*(s_j) = \frac{\mu_{jb}^*(s_j) \mu_{jc}^*(s_j)}{\int \mu_{jb}^*(s_j) \mu_{jc}^*(s_j) ds_j}, \quad (3.24b)$$

with messages $\mu_{jc}^*(s_j)$ corresponding to the fixed points of

$$\mu_{jc}^{(k+1)}(s_j) = \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j}, \quad (3.25)$$

with k an iteration index.

Proof. See Appendix A.4.3. \square

The sum-product algorithm has proven useful in many engineering applications and disciplines. For example, it is widely used for decoding in communication systems [28, 53, 61]. Furthermore, for a linear Gaussian state-space model, Kalman filtering and smoothing can be expressed in terms of sum-product message passing for state inference on a factor graph [62, 63]. This equivalence has inspired applications ranging from localization [64] to estimation [59].

The sum-product algorithm with updates (3.25) obtains the exact Bayesian posterior when the underlying graph is a tree [19, 37, 57]. Applying the sum-product algorithm to cyclic graphs is not guaranteed to converge and might lead to oscillations in the BFE over iterations. Theorems 3.1 and 3.2 in [65] show that the BFE of a graph with a single cycle is convex, which implies that the sum-product algorithm will converge in this case. Moreover, [52] shows that it is possible to obtain a double-loop message passing algorithm if the graph has a cycle such that the stable fixed-points will correspond to local minima of the BFE.

3.4 Message Passing Variations through Constraint Manipulation

There is no guarantee that the sum-product updates can be solved analytically for generic node functions with arbitrary connectivity. When analytic solutions are not possible, there are two ways to proceed. One way is to solve the sum-product update equations numerically, e.g., by Monte Carlo methods. Alternatively, we can add additional constraints to the BFE that lead to simpler update equations at the cost of inference accuracy. In the remainder of the chapter, we explore a variety of constraints that have proven to yield practical inference solutions.

3.4.1 Factorization Constraints

Additional factorizations of the variational density $q_a(\mathbf{s}_a)$ are often assumed to ease computation. In particular, we assume a *structured mean-field factorization* such that

$$q_b(\mathbf{s}_b) \triangleq \prod_{n \in l(b)} q_b^n(\mathbf{s}_b^n), \quad (3.26)$$

where n indicates a local cluster as a set of edges. To define a local cluster rigorously, let us first denote by $\mathcal{P}(a)$ the power set of an edge set $\mathcal{E}(a)$, where the power set is the set of all subsets of $\mathcal{E}(a)$. Then, a mean-field factorization $l(a) \subseteq \mathcal{P}(a)$ can be chosen such that all elements in $\mathcal{E}(a)$ are included in $l(a)$ exactly once. Therefore, $l(a)$ is defined as a set of one or multiple sets of edges. For example, if $\mathcal{E}(a) = \{i, j, k\}$, then $l(a) = \{\{i\}, \{j, k\}\}$ is allowed, as is $l(a) = \{\{i, j, k\}\}$ itself, but $l(a) = \{\{i, j\}, \{j, k\}\}$ is not allowed, since the element j occurs twice. More formally, in (3.26) the intersection of the super- and sub-script collects the required variables, see Fig. 3.7 for an example. The special case of a fully factorized $l(b)$ for all edges $i \in \mathcal{E}(b)$ is known as the *naive mean-field factorization* [37, 45].

We will analyze the effect of a structured mean-field factorization (3.26) on the Bethe free energy (3.7), for a specific factor-node $b \in \mathcal{V}$. Substituting (3.26) in the local free energy for factor b , yields

$$\begin{aligned} F[q_b, f_b] &= F[\{q_b^n\}, f_b] \\ &= \sum_{n \in l(b)} \int q_b^n(\mathbf{s}_b^n) \log q_b^n(\mathbf{s}_b^n) d\mathbf{s}_b^n - \int \left\{ \prod_{n \in l(b)} q_b^n(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b. \end{aligned} \quad (3.27)$$

We are then interested in

$$q_b^{m,*} = \arg \min_{q_b^m} L_b^m[q_b^m, f_b], \quad (3.28)$$

where the Lagrangian L_b^m (Lemma 3.3) enforces the normalization and marginalization constraints

$$\int q_b^m(\mathbf{s}_b^m) d\mathbf{s}_b^m = 1, \quad (3.29a)$$

$$\int q_b^m(\mathbf{s}_b^m) d\mathbf{s}_{b \setminus i} = q_i(s_i), \text{ for all } i \in m, m \in l(b). \quad (3.29b)$$

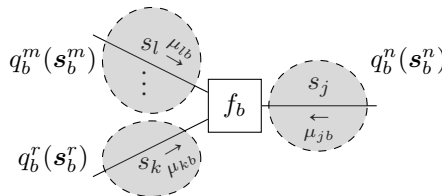


Figure 3.7: A node-induced subgraph $\mathcal{G}(b)$ with shaded sections that enclose the edges of an exemplary structured mean-field factorization $l(b) = \{m, n, r\}$. Note that in this example the cluster n only encompasses the single edge j , such that $q_b^n(\mathbf{s}_b^n) = q_j(s_j)$. In general, the assignment and number of edges in a cluster can be arbitrary.

Lemma 3.3. Given a terminated FFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider a node-induced subgraph $\mathcal{G}(b)$ with a structured mean-field factorization $l(b)$ (e.g. Fig. 3.7). Then local stationary solutions to the Lagrangian

$$\begin{aligned} L_b^m[q_b^m] &= \int q_b^m(\mathbf{s}_b^m) \log q_b^m(\mathbf{s}_b^m) d\mathbf{s}_b^m - \int \left\{ \prod_{n \in l(b)} q_b^n(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b + \\ &\quad \sum_{i \in m} \int \lambda_{ib}(s_i) \left[q_i(s_i) - \int q_b^m(\mathbf{s}_b^m) d\mathbf{s}_{m \setminus i} \right] ds_i + \\ &\quad \psi_b^m \left[\int q_b^m(\mathbf{s}_b^m) d\mathbf{s}_b^m - 1 \right] + C_b^m, \end{aligned} \quad (3.30)$$

where C_b^m collects all terms independent of q_b^m , are of the form

$$q_b^m(\mathbf{s}_b^m) = \frac{\tilde{f}_b^m(\mathbf{s}_b^m) \prod_{i \in m} \mu_{ib}(s_i)}{\int \tilde{f}_b^m(\mathbf{s}_b^m) \prod_{i \in m} \mu_{ib}(s_i) d\mathbf{s}_b^m}, \quad (3.31)$$

where

$$\tilde{f}_b^m(\mathbf{s}_b^m) = \exp \left(\int \left\{ \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^n(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b^{\setminus m} \right). \quad (3.32)$$

Proof. See Appendix A.4.4. □

3.4.1.1 Structured Mean-field Variational Message Passing

We now combine Lemma 3.3 and 3.2 to derive the structured variational message passing algorithm.

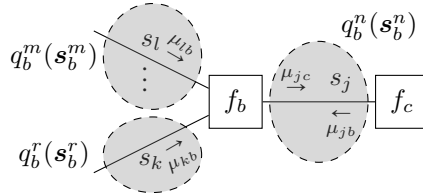


Figure 3.8: An example subgraph corresponding to $\mathcal{G}(b, j)$. Dashed ellipses enclose the edges of an exemplary exact cover $l(b) = \{m, n, r\}$. In general, the assignment and number of edges in a cluster can be arbitrary.

Theorem 3.2. (Structured Variational Message Passing) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ with a structured mean-field factorization $l(b) \subseteq \mathcal{P}(b)$, with local clusters $n \in l(b)$. Let $m \in l(b)$ be the cluster where $j \in m$ (see, e.g., Fig. 3.8). Given the local polytope

$$\mathcal{L}(\mathcal{G}(b, j)) = \{q_b^n \text{ for all } n \in l(b) \text{ s.t. (3.29a), and } q_j \text{ s.t. (3.29b)}\}, \quad (3.33)$$

then local stationary solutions to

$$\{q_b^{m,*}, q_j^*\} = \arg \min_{\mathcal{L}(\mathcal{G}(b, j))} F[q, f], \quad (3.34)$$

are given by

$$q_b^{m,*}(\mathbf{s}_b^m) = \frac{\tilde{f}_b^{m,*}(\mathbf{s}_b^m) \prod_{i \in m} \mu_{ib}^*(s_i)}{\int \tilde{f}_b^{m,*}(\mathbf{s}_b^m) \prod_{i \in m} \mu_{ib}^*(s_i) d\mathbf{s}_b^m} \quad (3.35a)$$

$$q_j^*(s_j) = \frac{\mu_{jb}^*(s_j) \mu_{jc}^*(s_j)}{\int \mu_{jb}^*(s_j) \mu_{jc}^*(s_j) ds_j}, \quad (3.35b)$$

with messages $\mu_{jc}^*(s_j)$ corresponding to the fixed points of

$$\mu_{jc}^{(k+1)}(s_j) = \int \tilde{f}_b^{m,(k)}(\mathbf{s}_b^m) \prod_{\substack{i \in m \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_b^m, \quad (3.36)$$

with iteration index k , and where

$$\tilde{f}_b^{m,(k)} = \exp \left(\int \left\{ \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^{n,(k)}(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b^m \right). \quad (3.37)$$

Proof. See Appendix A.4.5. □

The structured mean-field factorization applies the marginalization constraint only to the local cluster beliefs instead of the joint node belief. As a result, computation for the local cluster beliefs might become tractable [37, Ch.5]. The practical appeal of Variational Message Passing (VMP) based inference becomes evident when the underlying model is composed of conjugate factor pairs from the exponential family. When the underlying factors are conjugate exponential family distributions, the message passing updates (3.36) amounts to adding natural parameters [66] of the underlying exponential family distributions. Structured variational message passing is popular in acoustic signal modeling, e.g., [67] as it allows to keep track of correlations over time. In [68], a stochastic variant of structured variational

inference is utilized for Latent Dirichlet Allocation. Structured approximations are also used to improve inference in auto-encoders. In [69], inference involving non-parametric Beta-Bernoulli process priors is improved by developing a structured approximation to variational auto-encoders. When the data being modeled is time series, structured approximations reflect the transition structure over time. In [70], an efficient, structured black-box variational inference algorithm for fitting Gaussian variational models to latent time series is proposed.

Example 4. Consider the linear Gaussian state space model of Example 3. Let us assume that the precision matrix for latent-state transitions Q_t is not known and can not be constrained by data. Then, we can augment state space model by including a prior for Q_t and try to infer a posterior over Q_t from the observations. Since Q_t is the precision of a Normal factor we choose a conjugate Wishart prior and assume that Q_t is time-invariant by adding the following factors

$$w_0(Q_0, V, \nu) = \mathcal{W}(Q_0|V, \nu) \quad (3.38a)$$

$$w_t(Q_{t-1}, Q_t, Q_{t+1}) = \delta(Q_{t-1} - Q_t)\delta(Q_t - Q_{t+1}), \text{ for every } t = 1, \dots, T. \quad (3.38b)$$

It is undoubtedly possible to assume a time-varying structure for Q_t . However, our purpose is to illustrate a change in constraints rather than analyzing time-varying properties. This is why we presume time-invariance.

In this setting, the sum-product equations around the factor h_t are not analytically tractable. Therefore we change the constraints associated with h_t (3.23b) to those given in Theorem 3.2 as follows

$$\int q(x'_t, z_t, Q_t) dx'_t dz_t = q(Q_t), \quad \int q(x'_t, z_t, Q_t) dQ_t = q(x'_t, z_t) \quad (3.39a)$$

$$\int q(Q_t) dQ_t = 1, \quad \int q(x'_t, z_t) dx'_t dz_t = 1. \quad (3.39b)$$

We remove the data constraint on $q(Q_t)$ and instead include data constraints on the hyper-parameters

$$q(V) = \delta(V - \hat{V}), \quad q(\nu) = \delta(\nu - \hat{\nu}). \quad (3.40)$$

With the new set of constraints (3.39), we obtain a hybrid of sum-product and structured VMP algorithm, where structured messages around the factor h_t are computed by (3.36) and the rest of the messages are computed by sum-product (3.25). One time segment of the modified FFG along with the messages is given Figure 3.9. We use the same observations $\hat{\mathbf{y}}$ that was generated in Example 3 and the same initialization for the hidden states. For the hyper-parameters of the Wishart prior we choose $\hat{V} = 0.1 \cdot \mathbf{I}$ and $\hat{\nu} = 2$. Under these constraints, the result of structured variational message passing results along with the Bethe free energy evaluation is given in Figure 3.9.

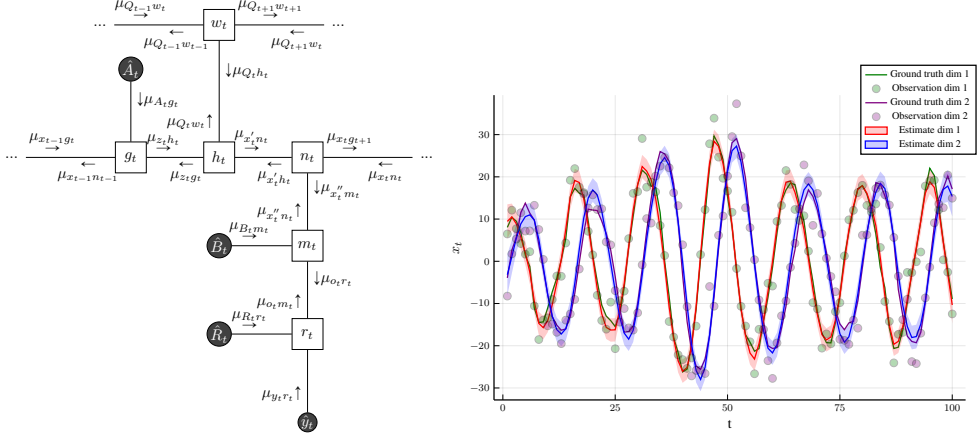


Figure 3.9: (Left) One time-segment of the FFG corresponding to the linear Gaussian state space model specified in Example 4 with the sum-product messages computed according to (3.36). (Right) The small dots indicate the noisy observations that are synthetically generated by the linear state space model of (3.21) using matrices specified in (3.22). The posterior distribution of the hidden states inferred by structured variational message passing is depicted with shaded regions representing plus and minus one variance. The minimum of the evaluated Bethe free energy over all iterations is $F[q, f] = 586.178$ (compared to $F[q, f] = 580.698$ in Example 3). The posterior distribution for the precision matrix is given by $Q \sim \mathcal{W} \left(\begin{bmatrix} 0.00266 & 0.000334 \\ 0.00034 & 0.00670 \end{bmatrix}, 102.0 \right)$.

3.4.1.2 Naive Mean-field Variational Message Passing

As a corollary of Theorem 3.2, we can consider the special case of a naive mean-field factorization, which is defined for node b as

$$q_b(\mathbf{s}_b) = \prod_{i \in \mathcal{E}(b)} q_i(s_i). \quad (3.41)$$

The naive mean-field constraint (3.41) transforms the local free energy into

$$\begin{aligned} F[q_b, f_b] &= F[\{q_i\}, f_b] \\ &= \sum_{i \in \mathcal{E}(b)} \int q_i(s_i) \log q_i(s_i) ds_i - \int \left\{ \prod_{i \in \mathcal{E}(b)} q_i(s_i) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b. \end{aligned} \quad (3.42a)$$

Corollary 3.1. (Naive Variational Message Passing) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ with a naive mean-field factorization $l(b) = \{i\}$ for all $i \in \mathcal{E}(b)$. Let $m \in l(b)$ be the cluster where $j = m$. Given the local polytope of (3.33), the local

stationary solutions to (3.34) are given by

$$q_b^{m,*}(\mathbf{s}_b^m) = q_j^*(s_j) = \frac{\mu_{jb}^*(s_j)\mu_{jc}^*(s_j)}{\int \mu_{jb}^*(s_j)\mu_{jc}^*(s_j) ds_j},$$

where the messages $\mu_{jc}^*(s_j)$ are the fixed points of the following iterations

$$\mu_{jc}^{(k+1)}(s_j) = \exp\left(\int \left\{ \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} q_i^{(k)}(s_i) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus j}\right), \quad (3.43)$$

where k is an iteration index.

Proof. See Appendix A.4.6. □

The naive mean-field factorization limits the search space of beliefs by imposing strict constraints on the variational posterior. As a result, the variational posterior also lose flexibility. To improve inference performance for sparse Bayesian learning, [71] proposes a hybrid mechanism by augmenting naive mean-field VMP with sum-product updates. This hybrid scheme reduces the complexity of the sum-product algorithm while improving the accuracy of the naive VMP approach. In [72], naive VMP is applied to semi-parametric regression and allows for scaling regression models to large data sets.

Example 5. As a follow up on Example 4 we relax the constraints in (3.39) to the following constraints presented in Corollary 3.1 as

$$\int q(x'_t, z_t, Q_t) dx'_t dz_t = q(Q_t), \quad \int q(x'_t, z_t, Q_t) dQ_t = q(x'_t, z_t) = q(x'_t)q(z_t) \quad (3.44a)$$

$$\int q(Q_t) dQ_t = 1, \quad \int q(x'_t) dx'_t = 1, \quad \int q(z_t) dz_t = 1. \quad (3.44b)$$

The FFG remains the same, and we use identical data constraints as in Example 4. Together with constraints (3.44), we obtain a hybrid of naive variational message passing and sum-product message passing algorithm where the messages around the factor h_t are computed by (3.43) and the rest of the messages by sum-product (3.25). Using the same data as in Example 3, the results for naive VMP are given in Figure 3.10 along with the evaluated Bethe free energy.

3.4.2 Form Constraints

Form constraints limit the functional form of the variational factors $q_a(\mathbf{s}_a)$ and $q_i(s_i)$. One of the most widely used form constraints, the data constraint, is also illustrated in Appendix A.1.

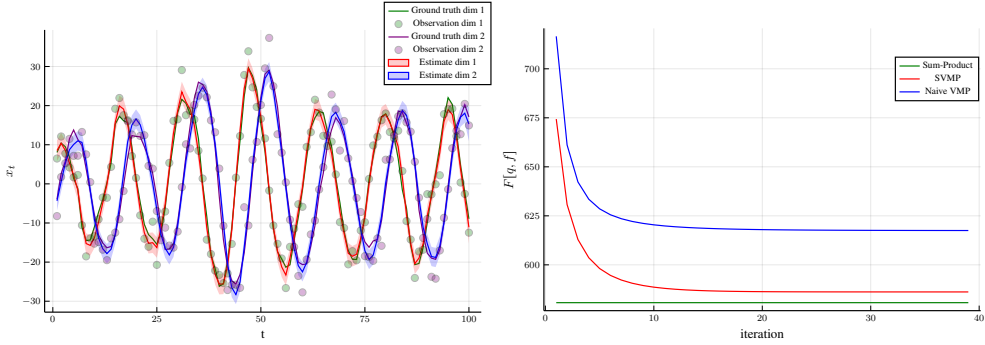


Figure 3.10: (Left) The small dots indicate the noisy observations that were synthetically generated by the linear state space model of (3.21) using matrices specified in (3.22). The posterior distribution for the hidden states inferred by naive variational message passing is depicted with shaded regions representing plus and minus one variance. The minimum of the evaluated Bethe free energy over all iterations is $F[q, f] = 617.468$, which is more than for the less-constrained Example 4 (with $F[q, f] = 586.178$) and example 3 (with $F[q, f] = 580.698$). The posterior for the precision matrix is given by $Q \sim \mathcal{W} \left(\begin{bmatrix} 0.00141 & -6.00549e^{-5} \\ -6.00549e^{-5} & 0.00187 \end{bmatrix}, 102.0 \right)$ (Right) A comparison of the Bethe free energies for sum-product, structured and naive variational message passing algorithms for the data generated in Example 3.

3.4.2.1 Data Constraints

A data constraint can be viewed as a special case of (3.9b), where the belief q_j is constrained to be a Dirac-delta function [73], such that

$$\int q_a(\mathbf{s}_a) d\mathbf{s}_{a \setminus j} = q_j(s_j) = \delta(s_j - \hat{s}_j), \quad (3.45)$$

where \hat{s}_j is a known value, e.g., an observation.

Lemma 3.4. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the node-induced subgraph $\mathcal{G}(b)$ (Fig. 3.3). Then local stationary solutions to the Lagrangian

$$\begin{aligned} L_b[q_b, f_b] = & F[q_b, f_b] + \psi_b \left[\int q_b(\mathbf{s}_b) d\mathbf{s}_b - 1 \right] + \\ & \sum_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \int \lambda_{ib}(s_i) \left[q_i(s_i) - \int q_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus i} \right] ds_i + \\ & \int \lambda_{jb}(s_j) \left[\delta(s_j - \hat{s}_j) - \int q_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus j} \right] ds_j + C_b. \end{aligned} \quad (3.46)$$

where C_b collects all terms that are independent of q_b , are of the form

$$q_b(\mathbf{s}_b) = \frac{f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i)}{\int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b}. \quad (3.47)$$

Proof. See Appendix A.4.7. □

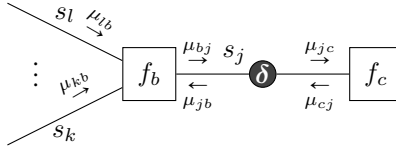


Figure 3.11: Visualization of a subgraph $\mathcal{G}(b, j)$ with indicated messages, where the dark circled delta indicates a data constraint; i.e., the variable s_j is constrained to have a distribution of the form $\delta(s_j - \hat{s}_j)$.

Theorem 3.3. (Data-Constrained Sum-Product) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.11). Given the local polytope

$$\mathcal{L}(\mathcal{G}(b, j)) = \{q_b \text{ s.t. (3.45)}\}, \quad (3.48)$$

the local stationary solutions to

$$q_b^* = \arg \min_{\mathcal{L}(\mathcal{G}(b, j))} F[q, f],$$

are of the form

$$q_b^*(\mathbf{s}_b) = \frac{f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i)}{\int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i) d\mathbf{s}_b}, \quad (3.49)$$

with message

$$\mu_{jb}^*(s_j) = \delta(s_j - \hat{s}_j). \quad (3.50)$$

Proof. See Appendix A.4.8. □

Note that the resulting message $\mu_{jb}^*(s_j)$ to node b does not depend on messages from node c , as would be the case for a sum-product update. By symmetry of Theorem 3.3 for the subgraph $\mathcal{L}\{\mathcal{G}(c, j)\}$, (A.32) identifies

$$\mu_{cj}(s_j) = \int f_c(\mathbf{s}_c) \prod_{\substack{i \in \mathcal{E}(c) \\ i \neq j}} \mu_{ic}(s_i) d\mathbf{s}_{c \setminus j} \neq \delta(s_j - \hat{s}_j).$$

This implies that messages incoming to a data constraint (such as μ_{cj}) are not further propagated through the data constraint. The data constraint thus effectively introduces conditional independence between the variables of neighboring factors (conditioned on the shared constrained variable). Interestingly, this is similar to the notion of an intervention [74], where a decision variable is externally forced to a realization.

Data constraints allow information from data sets to be absorbed into the model. Essentially, (variational) Bayesian machine learning applies inference in a graph with data constraints. In our framework, data is a constraint, and machine learning via the Bayes rule follows naturally from the minimization of the Bethe free energy (see also Appendix A.1).

3.4.2.2 Laplace Propagation

A second type of form constraint we consider is the Laplace constraint, see also [48]. Consider a second-order Taylor approximation on the local log-node function

$$\mathcal{L}_a(\mathbf{s}_a) = \log f_a(\mathbf{s}_a), \quad (3.51)$$

around an approximation point $\hat{\mathbf{s}}_a$, as

$$\tilde{\mathcal{L}}_a(\mathbf{s}_a; \hat{\mathbf{s}}_a) = \mathcal{L}_a(\hat{\mathbf{s}}_a) + \nabla^\top \mathcal{L}_a(\hat{\mathbf{s}}_a) (\mathbf{s}_a - \hat{\mathbf{s}}_a) + \frac{1}{2} (\mathbf{s}_a - \hat{\mathbf{s}}_a)^\top \nabla^2 \mathcal{L}_a(\hat{\mathbf{s}}_a) (\mathbf{s}_a - \hat{\mathbf{s}}_a). \quad (3.52)$$

From this approximation, we define the Laplace-approximated node-function as

$$\tilde{f}_a(\mathbf{s}_a; \hat{\mathbf{s}}_a) \triangleq \exp\left(\tilde{\mathcal{L}}_a(\mathbf{s}_a; \hat{\mathbf{s}}_a)\right), \quad (3.53)$$

which is substituted in the local free energy to obtain the Laplace-encoded local free energy as

$$F[q_a, \tilde{f}_a; \hat{\mathbf{s}}_a] = \int q_a(\mathbf{s}_a) \log \frac{q_a(\mathbf{s}_a)}{\tilde{f}_a(\mathbf{s}_a; \hat{\mathbf{s}}_a)} d\mathbf{s}_a. \quad (3.54)$$

It follows that the Laplace-encoded optimization of the local free energy becomes

$$q_a^* = \arg \min_{q_a} L_a[q_a, \tilde{f}_a; \hat{\mathbf{s}}_a], \quad (3.55)$$

where the Lagrangian L_a imposes the marginalization and normalization constraints of (3.9) on (3.54).

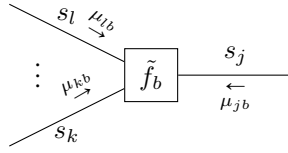


Figure 3.12: The subgraph around a Laplace-approximated node b with indicated messages.

Lemma 3.5. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the node-induced subgraph $\mathcal{G}(b)$ (Fig. 3.12). The stationary points of the Laplace-approximated Lagrangian (3.55) as a functional of q_b ,

$$L_b[q_b, \tilde{f}_b; \hat{\mathbf{s}}_b] = F[q_b, \tilde{f}_b; \hat{\mathbf{s}}_b] + \psi_b \left[\int q_b(\mathbf{s}_b) d\mathbf{s}_b - 1 \right] + \sum_{i \in \mathcal{E}(b)} \int \lambda_{ib}(s_i) \left[q_i(s_i) - \int q_b(\mathbf{s}_b) ds_{b \setminus i} \right] ds_i + C_b, \quad (3.56)$$

where C_b collects all terms that are independent of q_b , are of the form

$$q_b(\mathbf{s}_b) = \frac{\tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i)}{\int \tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b}. \quad (3.57)$$

Proof. See Appendix A.4.9. □

We can now formulate Laplace propagation as an iterative procedure, where the approximation point $\hat{\mathbf{s}}_b$ is chosen as the mode of the belief $q_b(\mathbf{s}_b)$.

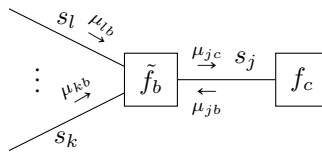


Figure 3.13: Visualization of a subgraph with indicated Laplace propagation messages. The node function f_b is denoted by \tilde{f}_b according to (3.53).

Theorem 3.4. (Laplace Propagation) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.13) with the Laplace-encoded factor \tilde{f}_b as per (3.53). We write the model (3.1) with substituted Laplace-encoded factor \tilde{f}_b for f_b , as f . Given the local polytope $\mathcal{L}(\mathcal{G}(b, j))$ of (3.14), then the local stationary solutions to

$$\{q_b^*, q_j^*\} = \arg \min_{\mathcal{L}(\mathcal{G}(b, j))} F[q, \tilde{f}; \hat{\mathbf{s}}_b], \quad (3.58)$$

are given by

$$q_b^*(\mathbf{s}_b) = \frac{\tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^*) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i)}{\int \tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^*) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i) d\mathbf{s}_b}$$

$$q_j^*(s_j) = \frac{\mu_{jb}^*(s_j) \mu_{jc}^*(s_j)}{\int \mu_{jb}^*(s_j) \mu_{jc}^*(s_j) ds_j},$$

with $\hat{\mathbf{s}}_b^*$ and the messages $\mu_{jc}^*(s_j)$ the fixed points of

$$\hat{\mathbf{s}}_b^{(k)} = \arg \max_{\mathbf{s}_b} \log q_b^{(k)}(\mathbf{s}_b)$$

$$q_b^{(k)}(\mathbf{s}_b) = \frac{\tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^{(k)}) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^{(k)}(s_i)}{\int \tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^{(k)}) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_b}$$

$$\mu_{jc}^{(k+1)}(s_j) = \int \tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^{(k)}) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j}.$$

Laplace propagation is introduced in [48] as an algorithm that propagates mean and variance information when exact updates are expensive to compute. Laplace propagation has found applications in the context of Gaussian processes and support vector machines [48]. In the jointly normal case, Laplace propagation coincides with sum-product and expectation propagation [48, 51].

3.4.2.3 Expectation Propagation

Expectation propagation can be derived in terms of constraint manipulation by relaxing the marginalization constraints to expectation constraints. Expectation constraints are of the form

$$\int q_a(\mathbf{s}_a) T_i(s_i) d\mathbf{s}_a = \int q_i(s_i) T_i(s_i) ds_i, \quad (3.59)$$

for a given function (statistic) $T_i(s_i)$. Technically, the statistic $T_i(s_i)$ can be chosen arbitrarily. Nevertheless, they are often chosen as sufficient statistics of an exponential family distribution. An exponential family distribution is defined by

$$q_i(s_i) = h(s_i) \exp(\eta_i^\top T_i(s_i) - \log Z(\eta_i)), \quad (3.60)$$

where η_i is the natural parameter, $Z(\eta_i)$ is the partition function, $T_i(s_i)$ is the sufficient statistics and $h(s_i)$ is a base measure [37]. The reason $T_i(s_i)$ is a sufficient statistic is because if there are observed values of the random variable s_i , then the parameter η_i can be estimated by using only the statistics $T_i(s_i)$. This means that the estimator of η_i will depend only on the statistics.

The idea in expectation propagation [51] is to relax the marginalization constraints with moment-matching constraints by choosing sufficient statistics from exponential family distributions [46]. Relaxation allows approximating the marginals of the sum-product algorithm with exponential family distributions. By keeping the marginals within the exponential family, the complexity of the resulting computations is reduced.

Lemma 3.6. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the node-induced subgraph $\mathcal{G}(b)$ (Fig. 3.3). The stationary points of the Lagrangian

$$\begin{aligned} L_b[q_b, f_b] = & F[q_b, f_b] + \psi_b \left[\int q_b(\mathbf{s}_b) d\mathbf{s}_b - 1 \right] + \\ & \sum_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \int \lambda_{ib}(s_i) \left[q_i(s_i) - \int q_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus i} \right] ds_i + \\ & \eta_{jb}^\top \left[\int q_j(s_j) T_j(s_j) ds_j - \int q_b(\mathbf{s}_b) T_j(s_j) d\mathbf{s}_b \right] + C_b, \end{aligned} \quad (3.61)$$

with sufficient statistics T_j , and where C_b collects all terms that are independent of q_b , are of the form

$$q_b(\mathbf{s}_b) = \frac{f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i)}{\int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b}, \quad (3.62)$$

with incoming exponential family message

$$\mu_{jb}(s_j) = \exp(\eta_{jb}^\top T_j(s_j)). \quad (3.63)$$

Proof. See Appendix A.4.11. □

Lemma 3.7. Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider an edge-induced subgraph $\mathcal{G}(j)$ (Fig. 3.4). The stationary solutions of the Lagrangian

$$\begin{aligned} L_j[q_j] = & H[q_j] + \psi_j \left[\int q_j(s_j) ds_j - 1 \right] + \\ & \sum_{a \in \mathcal{V}(j)} \eta_{ja}^\top \left[\int q_j(s_j) T_j(s_j) ds_j - \int q_a(\mathbf{s}_a) T_j(s_j) d\mathbf{s}_a \right] + C_j, \end{aligned}$$

with sufficient statistics $T_j(s_j)$, and where C_j collects all terms that are independent of q_j , are of the form

$$q_j(s_j) = \frac{\exp([\eta_{jb} + \eta_{jc}]^\top T_j(s_j))}{\int \exp([\eta_{jb} + \eta_{jc}]^\top T_j(s_j)) ds_j}. \quad (3.64)$$

Proof. See Appendix A.4.12. \square

Theorem 3.5. (Expectation Propagation) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.6). Given the local polytope

$$\mathcal{L}(\mathcal{G}(b, j)) = \{q_b \text{ s.t. (3.9a), and } q_j \text{ s.t. (3.59) and (3.10)}\}, \quad (3.65)$$

and $\mu_{jb}(s_j) = \exp(\eta_{jb}^\top T_j(s_j))$ an exponential family message (from Lemma 3.6). Then the local stationary solutions to (3.15) are given by

$$q_b^*(\mathbf{s}_b) = \frac{f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i)}{\int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}^*(s_i) d\mathbf{s}_b} \quad (3.66a)$$

$$q_j^*(s_j) = \frac{\exp([\eta_{jb}^* + \eta_{jc}^*]^\top T_j(s_j))}{\int \exp([\eta_{jb}^* + \eta_{jc}^*]^\top T_j(s_j)) ds_j}, \quad (3.66b)$$

with η_{jb}^* , η_{jc}^* and $\mu_{jc}^*(s_j)$ the fixed points of the iterations

$$\begin{aligned} \tilde{\mu}_{jc}^{(k)}(s_j) &= \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j} \\ \tilde{q}_j^{(k)}(s_j) &= \frac{\mu_{jb}^{(k)}(s_j) \tilde{\mu}_{jc}^{(k)}(s_j)}{\int \mu_{jb}^{(k)}(s_j) \tilde{\mu}_{jc}^{(k)}(s_j) ds_j}. \end{aligned}$$

By moment matching on $\tilde{q}_j^{(k)}(s_j)$, we obtain the natural parameter $\tilde{\eta}_j^{(k)}$. The message update then follows from

$$\begin{aligned} \eta_{jc}^{(k)} &= \tilde{\eta}_j^{(k)} - \eta_{jb}^{(k)} \\ \mu_{jc}^{(k+1)}(s_j) &= \exp(T_j(s_j)^\top \eta_{jc}^{(k)}). \end{aligned}$$

Proof. See Appendix A.4.13. \square

Moment matching can be performed by solving [37, Proposition 3.1]

$$\nabla_{\eta_j} \log Z_j(\eta_j) = \int \tilde{q}_j(s_j) T_j(s_j) ds_j$$

for η_j , where

$$Z_j(\eta_j) = \int \exp(\eta_j^\top T_j(s_j)) ds_j.$$

In practice, for a Gaussian approximation, the natural parameters can be obtained by converting the matched mean, and variance of $\tilde{q}_j(s_j)$ to the canonical form [51]. Computing the moments of $\tilde{q}_j(s_j)$ is often challenging due to the lack of closed-form solutions of the normalization constant. To address the computation of moments in EP, [75] proposes to evaluate challenging moments by quadrature methods. For multivariate random variables, moment matching by spherical radial cubature would be advantageous as it will reduce the computational complexity [76]. Another popular way of evaluating the moments is through importance sampling [77, Ch. 7] [78].

Expectation propagation has been utilized in various applications ranging from time-series estimation with Gaussian processes [79] to Bayesian learning with stochastic natural gradients [80]. When the likelihood functions for Gaussian process classification are not Gaussian, EP is often utilized [81, Chapter 3]. In [82], a message passing-based expectation propagation algorithm is developed for models that involve both continuous and discrete random variables. Perhaps the most practical applications of EP are in the context of probabilistic programming [32], where it is heavily used in real-world applications.

3.4.3 Hybrid Constraints

This section considers hybrid methods that combine factorization and form constraints and formalize some well-known algorithms in terms of message passing.

3.4.3.1 Mean-field Variational Laplace

Mean-field variational Laplace applies the mean-field factorization to the Laplace-approximated factor function. The appeal of this method is that Gaussians can represent all messages out-bound from the Laplace-approximated factor.

Theorem 3.6. (Mean-Field Variational Laplace) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.13) with the Laplace-encoded factor \tilde{f}_b as per (3.53). We write the model (3.1) with substituted Laplace-encoded factor \tilde{f}_b for f_b , as \tilde{f} . Furthermore, assume a naive mean-field factorization $l(b) = \{\{i\} \text{ for all } i \in \mathcal{E}(b)\}$. Let $m \in l(b)$ be the cluster where $j = m$. Given the local polytope of (3.33), the local stationary solutions to

$$\{q_b^{m,*}, q_j^*\} = \arg \min_{\mathcal{L}(\mathcal{G}(b,j))} F[q, \tilde{f}; \hat{s}_b], \quad (3.67)$$

are given by

$$q_b^{m,*}(s_b^m) = q_j^*(s_j) = \frac{\mu_{jb}^*(s_j)\mu_{jc}^*(s_j)}{\int \mu_{jb}^*(s_j)\mu_{jc}^*(s_j) ds_j},$$

where μ_{jc}^* are the fixed points of the following iterations

$$\mu_{jc}^{(k+1)}(s_j) = \exp\left(\int \left(\prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} q_i^{(k)}(s_i)\right) \log \tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^{(k)}) d\mathbf{s}_{b \setminus j}\right), \quad (3.68)$$

with

$$\hat{\mathbf{s}}_b^{(k)} = \arg \max_{\mathbf{s}_b} \log q_b^{(k)}(\mathbf{s}_b).$$

Proof. See Appendix A.4.14. □

Conveniently, under these constraints every outbound message from node b will be proportional to a Gaussian. Substituting the Laplace-approximated factor function, we obtain:

$$\log \mu_{jc}^{(k)}(s_j) = \int \left(\prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} q_i^{(k)}(s_i)\right) \tilde{\mathcal{L}}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b^{(k)}) d\mathbf{s}_{b \setminus j} + C.$$

Resolving this expectation yields a quadratic form in s_j , which after completing the square leads to a proportionally Gaussian message $\mu_{jc}(s_j)$. This argument holds for any edge adjacent to b , therefore, for all outbound messages from node b . Moreover, suppose the incoming messages are represented by Gaussians (e.g., because these are also computed under the mean-field variational Laplace constraint). In that case, all beliefs on the adjacent edges to b will also be Gaussian. This significantly simplifies computing the expectations, which illustrates the computational appeal of mean-field variational Laplace.

Mean-field variational Laplace is widely used in dynamic causal modeling [83] and more generally in cognitive neuroscience, partly because the resulting computations are deemed neurologically plausible [84–86].

3.4.3.2 Expectation Maximization

Expectation Maximization (EM) can be viewed as a hybrid algorithm that combines a structured variational factorization with a Dirac-delta constraint, where the constrained value itself is optimized. Given a structured mean-field factorization $l(a) \subseteq \mathcal{P}(a)$, with a single-edge cluster $m = j$, then Expectation maximization considers local factorizations of the form

$$q_a(\mathbf{s}_a) = \delta(s_j - \theta_j) \prod_{\substack{n \in l(a) \\ n \neq m}} q_a^n(\mathbf{s}_a^n), \quad (3.69)$$

where the belief for s_j is constrained by a Dirac-delta distribution, similar to Sec. 3.4.2.1. In (3.69) however, the variable s_j represents a random variable with (unknown) value $\theta_j \in \mathbb{R}^d$, where d is the dimension of the random variable s_j . We explicitly use the notation θ_j (as opposed to \hat{s}_j for the data constraint in Sec. 3.4.2.1) to clarify that this value is a parameter for the constrained belief over s_j that will be optimized. That is, θ_j does not represent a model parameter in itself. To make this distinction even more explicit, in the context of optimization, we will refer to Dirac-delta constraints as point-mass constraints.

The factor-local free energy $F[q_a, f_a; \theta_j]$ then becomes a function of the θ_j parameter.

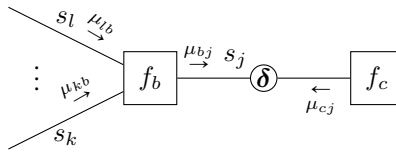


Figure 3.14: Visualization of a subgraph $\mathcal{G}(b, j)$ with indicated messages. The open circle indicates a point-mass constraint of the form $\delta(s_j - \theta_j)$, where the value θ_j is optimized.

Theorem 3.7. (Expectation Maximization) Given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the induced subgraph $\mathcal{G}(b, j)$ (Fig. 3.14) with a structured mean-field factorization $l(b) \subseteq \mathcal{P}(b)$, with local clusters $n \in l(b)$. Let $m \in l(b)$ be the cluster where $j = m$. Given the local polytope

$$\mathcal{L}(\mathcal{G}(b, j)) = \{q_b^n \text{ for all } n \in l(b) \text{ s.t. (3.29a)}\}, \quad (3.70)$$

the local stationary solutions to

$$\theta_j^* = \arg \min_{\mathcal{L}(\mathcal{G}(b, j))} F[q, f; \theta_j],$$

are given by the fixed points of

$$\mu_{bj}^{(k+1)}(s_j) = \exp \left(\int \left\{ \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^{n, (k)}(s_b^n) \right\} \log f_b(s_b) ds_{b \setminus j} \right) \quad (3.71a)$$

$$\theta_j^{(k+1)} = \arg \max_{s_j} \left(\log \mu_{bj}^{(k+1)}(s_j) + \log \mu_{cj}^{(k+1)}(s_j) \right). \quad (3.71b)$$

Proof. See Appendix A.4.15. □

Expectation Maximization was formulated in [87] as an iterative method that optimizes log-expectations of likelihood functions, where each EM iteration is guaranteed to increase the expected log-likelihood. Moreover, under some differentiability conditions, the EM algorithm is guaranteed to converge [87, Theorem 3]. A detailed overview of EM for exponential

families is available in [37, Ch. 6]. A formulation of EM in terms of message passing is given by [88], where message passing for EM is applied in a filtering and system identification context. In [88], derivations are based on [87, Theorem 1], whereas our derivations directly follow from variational principles.

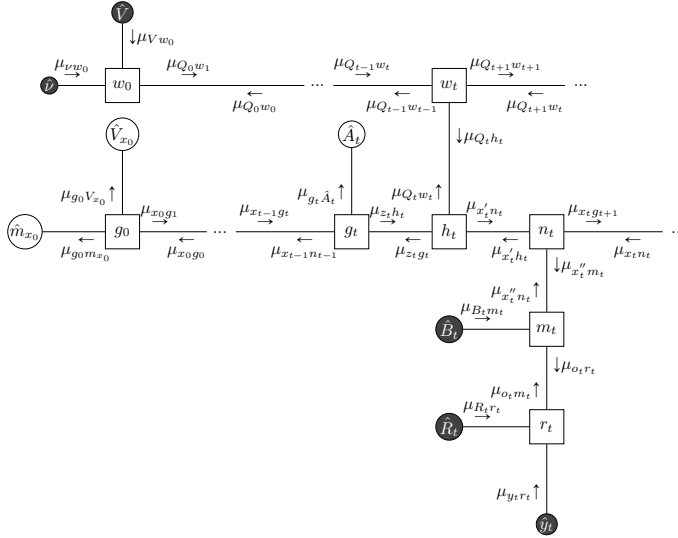


Figure 3.15: The FFG of the linear Gaussian state space model augmented with the EM constraints in Example 6.

Example 6. Now suppose we do not know the angle θ for the state transition matrix A_t in Example 4 and would like to estimate the value of θ . Moreover, further suppose that we are interested in estimating the hyper-parameters for the prior m_{x_0} and V_{x_0} , as well as the precision matrix for the state transitions Q_t . For this purpose, we change the constraints of (3.23a) into EM constraints by Theorem 3.7:

$$q(x_{t-1}, z_t, A_t(\theta)) = \delta(A_t(\theta) - A_t(\hat{\theta}))q(z_t|x_{t-1}, A_t(\theta))q(x_{t-1}) \quad (3.72a)$$

$$q(x_0, m_{x_0}, V_{x_0}) = q(x_0)\delta(m_{x_0} - \hat{m}_{x_0})\delta(V_{x_0} - \hat{V}_{x_0}), \quad (3.72b)$$

where we optimize $\hat{\theta}$, \hat{V}_{x_0} and \hat{m}_{x_0} with EM. With the addition of the new EM constraints, the resulting FFG is given in Figure 3.15. The hybrid message passing algorithm consists of structured variational messages around the factor h_t , sum-product messages around w_t , n_t , m_t and r_t , and EM messages around g_0 and g_t . We used identical observations as in the previous examples. The results for the hybrid SVMP-EM-SP algorithm is given in Figure 3.16 along with the evaluated Bethe free energy over all iterations.

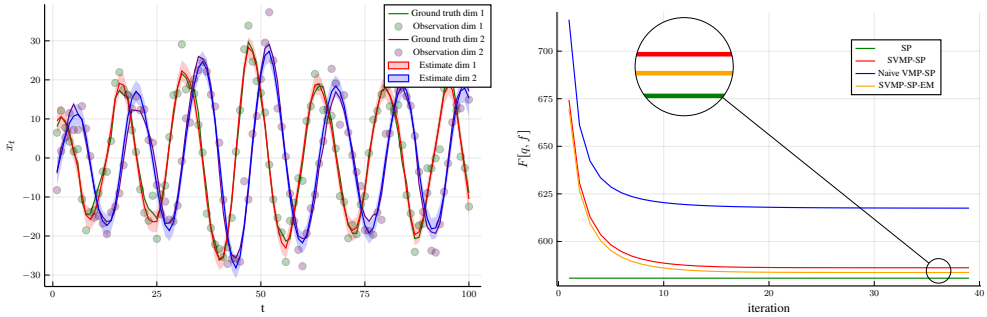


Figure 3.16: (Left) The small dots indicate the noisy observations that are synthetically generated by the linear state space model of (3.21) using matrices specified in (3.22). The posterior distribution of the hidden states inferred by structured variational message passing is depicted with shaded regions representing plus and minus one variance. The minimum of the evaluated Bethe free energy over iterations is $F[q, f] = 583.683$. Moreover, the posterior distribution for the precision matrix is given by $Q \sim \mathcal{W}\left(\begin{bmatrix} 0.00286 & 0.00038 \\ 0.00038 & 0.00691 \end{bmatrix}, 102.0\right)$. The EM estimates are $\theta = \pi/7.821$, $\hat{m}_{x_0} = [7.23, -7.016]$ and $\hat{V}_{x_0} = \begin{bmatrix} 11.028 & -1.926 \\ -1.926 & 10.918 \end{bmatrix}$. (Right) Free energy plots of the 4 algorithms discussed in Examples 1 through 4 on the same data set.

3.4.4 Overview of Message Passing Algorithms

In Secs. 3.4.1 through 3.4.3, following a high-level recipe pioneered by [35], we presented first-principle derivations of some of the popular message passing-based inference algorithms by manipulating the local constraints of the Bethe free energy. The results are summarized in Table 3.1.

Crucially, the method of constrained BFE minimization goes beyond the reviewed algorithms. By creating a new set of local constraints and following similar derivations based on variational calculus, one can obtain new message passing-based inference algorithms that better match the specifics of the generative model or application. It is also possible to extend the formulation to a more generic constrained divergence minimization problem. Changing the divergence will have consequences on the properties of the stationary solutions obtained by a message passing algorithm. For a review of the properties of posteriors obtained by α -divergence minimization, we refer the interested reader to [35]. For certain choices of divergence measures, local minimization will not guarantee global minimization and convergence is not guaranteed [35].

3.5 Scoring Models by Minimized Variational Free Energy

As discussed in Sec. 3.2.2, the variational free energy is an important measure of model performance. In Sec. 3.5.1 and 3.5.2 we discuss some problems that occur when evaluating

Local Constraint	SP	SVMP	MFVMP	DC	LP	MFVLP	EM	EP
Normalization	✓	✓	✓	✓	✓	✓	✓	✓
Marginalization	✓	✓	✓	✓	✓	✓	✓	✓
Moment Matching								✓
Structured Mean-Field		✓					✓	
Naive Mean-Field			✓			✓		
Laplace Approximation					✓	✓		
Dirac-delta Estimation				✓			✓	✓

Table 3.1: Relation between local constraints and derived message updates. The rows refer to different constraints related to factor-variable combinations, factors, and variables. Note that each message passing algorithm combines a set of constraints. Abbreviations: Sum-Product (SP), Structured Variational Message Passing (SVMP), Mean-Field Variational Message Passing (MFVMP), Data Constraint (DC), Laplace Propagation (LP), Mean-Field Variational Laplace (MFVLP), Expectation Maximization (EM), and Expectation Propagation (EP).

the BFE on a TFFG. In Sec. 3.5.3 we propose an algorithm that evaluates the constrained BFE as a summation of local contributions on the TFFG.

3.5.1 Evaluation of the Entropy of Dirac-delta Constrained Beliefs

For continuous variables, data and point-mass constraints as discussed in Sec. 3.4.2.1, 3.4.3.2 and Appendix A.1, collapse the information density to infinity, which leads to singularities in entropy evaluation [89]. More specifically, for a continuous variable s_j , the entropies for beliefs of the form $q_j(s_j) = \delta(s_j - \hat{s}_j)$ and $q_a(\mathbf{s}_a) = q_{a \setminus j}(\mathbf{s}_{a \setminus j} | s_j) \delta(s_j - \hat{s}_j)$ both evaluate to $-\infty$.

In variational inference, it is common to define the VFE only with respect to the latent (unobserved) variables [6, Sec. 10.1]. In contrast, in this paper we explicitly define the BFE in terms of an iteration over all nodes and edges (3.7), which also includes non-latent beliefs in the BFE definition. Therefore, we define

$$q_j(s_j) = \delta(s_j - \hat{s}_j) \Rightarrow H[q_j] \triangleq 0,$$

$$q_a(\mathbf{s}_a) = q_{a \setminus j}(\mathbf{s}_{a \setminus j} | s_j) \delta(s_j - \hat{s}_j) \Rightarrow H[q_a] \triangleq H[q_{a \setminus j}],$$

where $q_{a \setminus j}(\mathbf{s}_{a \setminus j} | s_j)$ indicates the conditional belief, and $q_{a \setminus j}(\mathbf{s}_{a \setminus j})$ the joint belief. These definitions effectively remove the entropies for observed variables from the BFE evaluation. Note that although $q_{a \setminus j}(\mathbf{s}_{a \setminus j})$ is technically not a part of our belief set (3.7), it can be obtained by marginalization of $q_a(\mathbf{s}_a)$ (3.9b).

3.5.2 Evaluation of Node-Local Free Energy for Deterministic Nodes

Another difficulty arises with the evaluation of the node-local free energy $F[q_a]$ for factors of the form

$$f_a(\mathbf{s}_a) = \delta(h_a(\mathbf{s}_a)). \quad (3.73)$$

This type of node function reflects deterministic operations, e.g., $h(x, y, z) = z - x - y$ corresponds to the summation $z = x + y$. In this case, directly evaluating $F[q_a]$ again leads to singularities.

There are (at least) two strategies available in the literature that resolve this issue. The first strategy “softens” the Dirac-delta by re-defining:

$$f_a(\mathbf{s}_a) \triangleq \frac{1}{\sqrt{2\pi\epsilon}} \exp\left(-\frac{1}{2\epsilon} h_a(\mathbf{s}_a)^2\right),$$

with $0 < \epsilon \ll 1$ [50]. A drawback of this approach is that it may alter the model definition in a numerically unstable way, leading to a different inference solution and variational free energy than originally intended.

The second strategy combines the deterministic factor f_a with a neighboring stochastic factor f_b into a new *composite* factor f_c , by marginalizing over a shared variable s_j , leading to [90]

$$f_c(\mathbf{s}_c) \triangleq \int \delta(h_a(\mathbf{s}_a)) f_b(\mathbf{s}_b) ds_j,$$

where $\mathbf{s}_c = \{\mathbf{s}_a \cup \mathbf{s}_b\} \setminus s_j$. This procedure has drawbacks for models that involve many deterministic factors. Namely, the convenient model modularity and resulting distributed compatibility are lost when large groups of factors are compacted in model-specific composite factors. We propose here a third strategy.

Theorem 3.8. Let $f_a(\mathbf{s}_a) = \delta(h_a(\mathbf{s}_a))$, with $h_a(\mathbf{s}_a) = s_j - g_a(\mathbf{s}_{a \setminus j})$, and node-local belief $q_a(\mathbf{s}_a) = q_{j|a}(s_j | \mathbf{s}_{a \setminus j}) q_{a \setminus j}(\mathbf{s}_{a \setminus j})$. Then, the node-local free energy evaluates to

$$F[q_a, f_a] = \begin{cases} -H[q_{a \setminus j}] & \text{if } q_{j|a}(s_j | \mathbf{s}_{a \setminus j}) = \delta(s_j - g_a(\mathbf{s}_{a \setminus j})) \\ \infty & \text{otherwise.} \end{cases}$$

Proof. See Appendix A.4.16. □

An example that evaluates the node-local free energy for a non-trivial deterministic node can be found in Appendix A.3.

The equality node is a particular case of deterministic node, with a node function of the form (3.3). The argument of (Theorem 3.8) does not directly apply to this node. Because the equality node function comprises of two Dirac-delta functions, it can not be written in the form of Theorem 3.8. However, we can still reduce the node-local free energy contribution.

Theorem 3.9. Let $f_a(\mathbf{s}_a) = \delta(s_j - s_i) \delta(s_j - s_k)$, with node-local belief written as $q_a(\mathbf{s}_a) = q_{ik|j}(s_i, s_k | s_j) q_j(s_j)$. Then, the node-local free energy evaluates to

$$F[q_a, f_a] = \begin{cases} -H[q_j] & \text{if } q_{ik|j}(s_i, s_k | s_j) = \delta(s_j - s_i) \delta(s_j - s_k) \\ \infty & \text{otherwise.} \end{cases}$$

Proof. See Appendix A.4.17. □

3.5.3 Evaluating the Bethe Free Energy

We propose an algorithm that evaluates the BFE on a TFFG representation of a factorized model. The algorithm is based on the following results:

- The definitions for the computation of data-constrained entropies ensure that only variables with associated stochastic beliefs count towards the Bethe entropy. This makes the BFE evaluation consistent with Theorems 3.3 and 3.7, where the single-variable beliefs for observed variables are excluded from the BFE definition;
- We assume that a local mean-field factorization $l(a)$ is available for each $a \in \mathcal{V}$ (Sec. 3.4.1). If the mean-field factorization is not explicitly defined, we assume $l(a) = \{a\}$ is the unfactored set;
- Deterministic nodes are accounted for by Theorem 3.8, which reduces the joint entropy to entropy over the ‘inbound’ edges. Although the belief over the ‘inbounds’ $q_{a \setminus j}(\mathbf{s}_{a \setminus j})$ is not a term in the Bethe factorization (3.8), it can simply be obtained by marginalization of $q_a(\mathbf{s}_a)$;
- The equality node is a special case, where we let the node-entropy discount the degree of the associated variable in the original model definition. While the BFE definition on a TFFG (3.7) does not explicitly account for edge degrees, this mechanism implicitly corrects for ‘double-counting’ [50]. In this case edge selection for counting is arbitrary, because all associated edges are (by definition) constrained to share the same belief (Sec. 3.2.1, Theorem 3.9).

The decomposition of (3.7) shows that an iteration can compute the BFE over the nodes and edges of the graph. Because some contributions to the BFE might cancel each other, the algorithm first tracks counting numbers u_a for the average energies

$$U_a[q_a] = - \int q_a(\mathbf{s}_a) \log f_a(\mathbf{s}_a) d\mathbf{s}_a,$$

and counting numbers h_k for the (joint) entropies

$$H[q_k] = - \int q_k(\mathbf{s}_k) \log q_k(\mathbf{s}_k) d\mathbf{s}_k,$$

which are ultimately combined and evaluated. We use an index k to indicate that the entropy computation may include not only the edges but a generic set of variables. We will give the definition of the set that k belongs to in Algorithm 1.

Algorithm 1 Evaluation of the Bethe free energy on a Terminated Forney-style factor graph.

given a TFFG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
given a local mean-field factorization $l(a)$ for all $a \in \mathcal{V}$
define $q_j(s_j) = \delta(s_j - \hat{s}_j) \Rightarrow H[q_j] \stackrel{\Delta}{=} 0$ \triangleright Ignore entropy of Dirac-delta constrained beliefs
define $q_a(\mathbf{s}_a) = q_{a|j}(\mathbf{s}_{a \setminus j} | s_j) \delta(s_j - \hat{s}_j) \Rightarrow H[q_a] \stackrel{\Delta}{=} H[q_{a \setminus j}]$ \triangleright Reduce entropy of Dirac-delta constrained joint beliefs
define $\mathcal{K} = \{a, a \setminus i, n, \text{ for all } a \in \mathcal{V}, i \in \mathcal{E}(a), n \in l(a)\}$ the set of (joint) belief indices
initialize counting numbers $u_a = 0$ for all $a \in \mathcal{V}$, $h_k = 0$ for all $k \in \mathcal{K}$

for all nodes $a \in \mathcal{V}$ **do**
 if a is a stochastic node **then**
 $u_a += 1$ \triangleright Count the average energy
 for all clusters $n \in l(a)$ **do**
 $h_n += 1$ \triangleright Count the (joint) cluster entropy
 end for
 else if a is an equality node **then**
 Select an edge $j \in \mathcal{E}(a)$
 $h_j += 1$ \triangleright Count the variable entropy
 else \triangleright Deterministic node a
 Obtain the node function $f_a(\mathbf{s}_a) = \delta(s_j - g_a(\mathbf{s}_{a \setminus j}))$
 $h_{a \setminus j} += 1$ \triangleright Count the (joint) entropy of the inbounds
 end if
end for

for all edges $i \in \mathcal{E}$ **do**
 $h_i -= 1$ \triangleright Discount the variable entropy
end for

$U = \sum_{a \in \mathcal{V}} u_a U_a[q_a]$
 $H = \sum_{k \in \mathcal{K}} h_k H[q_k]$
return $F = U - H$

3.6 Implementation of Algorithms and Simulations

We have developed probabilistic programming toolboxes *ForneyLab.jl* and *ReactiveMP.jl* in the Julia language [34,91,92]. The majority of algorithms that are reviewed in Table 3.1 have been implemented in ForneyLab and ReactiveMP along with a variety of demos.¹² Both toolboxes are extendable and supports postulating new local constraints of the BFE for the creation of custom message passing-based inference algorithms.

To limit the length of this chapter, we refer the reader to the demonstration folder of ForneyLab and several of our previous papers with code. For instance, our previous work in [93] implemented Mean-Field Variational Laplace propagation for the hierarchical Gaussian filter (HGF) [94]. In the follow-up work [95], inference results improved by changing to structured factorization and moment-matching local constraints. In that case, modification of local constraints created a hybrid EP-VMP algorithm that better suited the model. Moreover, in [47] we formulated the idea of *chance constraints* in the form of violation probabilities leading to a new message-passing algorithm that supports goal-directed behavior within the context of active inference. A similar line of reasoning led to improved inference procedures for auto-regressive models [96].

3.7 Related Work

The seminal work [50] inspires our work in this chapter, which discusses the equivalence between the fixed points of the belief propagation algorithm [19] and the stationary points of the Bethe free energy. This equivalence is established through a Lagrangian formalism, which allows for the derivation of Generalized Belief Propagation (GBP) algorithms by introducing region-based graphs and the region-based (Kikuchi) free energy [49].

Region graph-based methods allow for overlapping clusters (Sec. 3.4.1) and thus offer a more generic message-passing approach. However, the selection of appropriate regions (clusters) proves to be difficult, and the resulting algorithms may grow prohibitively complex. In this context, [97] addresses how to manipulate regions and manage the complexity of GBP algorithms. Furthermore, [98] also establishes a connection between GBP and expectation propagation (EP) by introducing structured region graphs.

The inspirational work of [35] derives message-passing algorithms by minimization of α -divergences. A fixed point projection scheme obtains the stationary points of α -divergences. This projection scheme is reminiscent of the minimization scheme of the expectation propagation (EP) algorithm [51]. Compared to [35], our work focuses on a single divergence objective (namely the VFE). The work of [46] derives the EP algorithm by manipulating the marginalization and factorization constraints of the Bethe free energy objective (see also Sec. 3.4.2.3). However, the EP algorithm is not guaranteed to converge to a minimum of the associated divergence metric.

¹<https://github.com/biaslab/ForneyLab.jl/tree/master/demo>

²<https://github.com/biaslab/ReactiveMP.jl/tree/master/demo>

To address the convergence properties of the algorithms that are obtained by region graph methods, the outstanding work of [65] derives conditions on the region counting numbers that guarantee the convexity of the underlying objective. In general, however, the constrained Bethe free energy is not guaranteed to be convex, and therefore the derived message passing updates are not guaranteed to converge.

3.8 Discussion and Conclusions

The key message in this chapter is that a (variational) Bayesian model designer may tune the tractability-accuracy trade-off for evidence and posterior evaluation through constraint manipulation. It is interesting to note that the technique to derive message-passing algorithms is always the same. We followed the recipe pioneered in [35] to derive a large variety of message passing algorithms solely through minimizing constrained Bethe free energy. This minimization leads to local fixed-point equations, which we can interpret as message passing updates on a (terminated) FFG. The presented lemmas showed how the constraints affect the Lagrangians locally. The presented theorems determined the stationary solutions of the Lagrangians and obtained the message passing equations. Thus, if a designer proposes a new set of constraints, the first place to start is to analyze the effect on the Lagrangian. Once the impact of the constraint on the Lagrangian is known, then variational optimization may result in stationary solutions that a fixed-point iteration scheme can obtain.

This chapter selected the Forney-style factor graph framework to illustrate our ideas. FFGs are mathematically comparable to the more common bi-partite factor graphs that associate round nodes with variables and square nodes with factors [53]. Bi-partite factor graphs require two distinct types of message updates (one leaving variable nodes and one leaving factor nodes), while message passing on a (T)FFG requires only a single type of message updates [99]. The (T)FFG paradigm thus substantially simplifies the derivations and resulting message passing update equations.

The message passing update rules in this chapter are presented without guarantees on the convergence of the (local) minimization process except for the EM and VMP algorithms. In practice, however, algorithm convergence can be easily checked by evaluating the BFE (Algorithm 1) after each belief update.

In this chapter, we formulated a message-passing approach to probabilistic inference by identifying local stationary solutions of a constrained Bethe free energy objective (Sec. 3.3, 3.4). The proposed framework constructs a graph for the generative model and specifies local constraints for variational optimization in a local polytope. The constraints are then imposed on the variational objective by a Lagrangian construct. Unconstrained optimization of the Lagrangian then leads to local expressions of stationary points, which can be obtained by iterative execution of the resulting fixed point equations, which we identify with message passing updates.

Furthermore, we presented an approach to evaluate the BFE on a (terminated) Forney-style factor graph (Sec. 3.5). This procedure allows an algorithm designer to assess algo-

rithms and models' performance readily.

We have included detailed derivations of message passing updates (Appendix A), and hope that the presented formulation inspires the discovery of novel and customized message passing algorithms.

CHAPTER 4

The Hierarchical Gaussian Filter

"At the beginning of every problem in probability theory, there arises a need to assign some initial probability distribution; or what is the same thing, to set up an ensemble. This is a problem which cannot be evaded, and for which the laws of physics give us no help."

–Edwin Thompson Jaynes

4.1 Introduction

Online updating of non-linear dynamic models for possibly non-stationary time series remains much-studied in various disciplines. In this chapter, we focus on these issues for the Hierarchical Gaussian Filter (HGF), which is a successful model in the computational neuroscience community, e.g., [36] [100]. The HGF is positioned as a generative probabilistic non-linear hierarchical model for sensory observations in this community. In this view, perceptual processes are modeled as a Bayesian inference (state estimation) task, and learning corresponds to Bayesian inference of the model parameters. The HGF is a continuation of observing the observer framework presented in [101], which gives a generic approach on how subjects decide in the presence of uncertainty. The HGF is a generative probabilistic non-linear hierarchical model for sensory observations in the computational neuroscience community. In this view, perceptual processes are modeled as a Bayesian inference (state estimation) task, and learning corresponds to Bayesian inference of the model parameters (perceptual parameters) [84], [102].

In inspiring work by [36], analytic equations for online state estimation in the HGF are derived. [36] recommends offline ("batch") variational Bayesian learning for the HGF model parameters. Unfortunately, offline parameter estimation is not well-suited to track

non-stationarities that are part of real-world sensoria. Moreover, while an open-source toolbox for HGF-based modeling is available [103], the toolbox does not automatically update the inference equations if the HGF model specifications are slightly modified. These are somewhat limiting factors to a wide application range of HGF-inspired models for non-stationary processes.

Derivation of update equations for state estimation in [36] is based on naive mean-field factorization constraint of Section 3.4.1.2. Even though the derivation relies on variational principles, it lacks an explicit Lagrangian formalism. Due to the lack of direct involvement of constraints, it is impossible to explore the effect of varying constraints on the model performance. For example, assuming an independent structure for the approximating distribution (naive mean-field) over temporal dimension is not realistic for data with inherent correlation. It will decrease the performance of the HGF.

This chapter will focus on message passing in the HGF as a framework for analyzing non-stationary time series. Message passing variants that were discussed in Chapter 3 will find their direct applications to the HGF. The appeal of message passing will become evident once we show its modularity and computational efficiency of analytic update equations. The modularity of message passing due to the inherent divide and conquer approach will allow us to extend the HGF beyond its current state of applicability. In contrast, analytic updates will allow us to perform fast inference.

This chapter aims to illustrate how the message passing framework can be utilized for time-series modeling with the HGF as a generic method of inference that unifies state and parameter estimation together with performance evaluation. After introducing the model definition of the HGF along with a graphical representation in Section 4.2, we will establish a Lagrangian framework for the HGF in Section 4.3. In the spirit of Chapter 3, we will formulate the problem definition as a Lagrangian minimization in Section 4.4 where changing constraints will lead to a variety of message passing algorithms for the HGF. In Section 4.5 and 4.6, we will demonstrate the divide and conquer approach of the graphical formalism by abstracting challenging parts (where the computation of messages is not straightforward) of the HGF into composite structures and then computing the required messages for the composite structures, respectively. We will show simulations of time-series modeling with the HGF in Section 4.7 and conclude this chapter in Section 4.8.

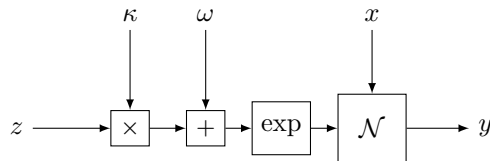


Figure 4.1: Internal structure of the GCV. The functional form of GCV is $\mathcal{N}(y|x, \exp(\kappa z + \omega))$. GCV do not include time or layer information explicitly. Once used in the context of hierarchical time-series modelling, the variables might have time or layers explicitly included.

4.2 Model Definition

The hierarchical Gaussian filter is a Gaussian random walk model for a sequence of observations, where the variance of the random walk is itself modeled as a Gaussian random walk and so on [36]. Specifically, for an observation sequence $\mathbf{y} \triangleq [y_1 \ y_2 \ \dots \ y_T]^\top$, an HGF model is specified as a joint probability distribution

$$p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) = \underbrace{p(\boldsymbol{\theta})p(\mathbf{x}_0)}_{\text{prior}} \underbrace{\prod_{t=1}^T p(y_t | x_t^{(1)})}_{\text{likelihood}} \underbrace{\prod_{i=1}^N p(x_t^{(i)} | x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)})}_{\text{state transitions}}. \quad (4.1)$$

4.2.1 State Transition Dynamics

The state transition model is given by

$$p(x_t^{(i)} | x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)}) = \begin{cases} \mathcal{N}(x_t^{(i)} | x_{t-1}^{(i)}, g_t^{(i)}) & i < N \\ \mathcal{N}(x_t^{(i)} | x_{t-1}^{(i)}, \xi^{-1}) & i = N, \end{cases} \quad (4.2)$$

$g_t^{(i)} = \exp(\kappa^{(i)} x_t^{(i+1)} + \omega^{(i)})$ and ξ is precision. Here, $x_t^{(i)}$ and $\boldsymbol{\theta}^{(i)} = [\kappa^{(i)} \ \omega^{(i)}]^\top$ respectively denote the hidden state and parameters of layer $i = 1, \dots, N-1$ at time t . The only parameter of the highest level is the transition precision, i.e., $\boldsymbol{\theta}^{(N)} = \xi$. The HGF state transition model couples the state of a random walk layer to the variance parameter of the layer below through a positive non-linearity g_t . The model parameters $\kappa^{(i)}$ and $\omega^{(i)}$ determine the scale and bias of the random walks. The state vector at time t is denoted by $\mathbf{x}_t = [x_t^{(1)} \ x_t^{(2)} \ \dots \ x_t^{(N)}]^\top$ and the collection of all states and parameters are written as $\mathbf{x} \triangleq [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_T]^\top$ and $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}^{(1)} \ \boldsymbol{\theta}^{(2)} \ \dots \ \boldsymbol{\theta}^{(N)}]^\top$.

Equation 4.2 factorizes the model into layers with Markovian dynamics at each layer and specifies that the state transition model is a Gaussian random walk with *time-varying* variance that is determined by the state of the superior layer. The non-linearity $g_t^{(i)}$ describes how the variance in the random walk step depends on the state of the superior layer. The exponent in the non-linearity enforces a non-negative variance that contains a phasic (time-varying) component $\exp(\kappa^{(i)} x_t^{(i+1)})$ and a tonic (time-invariant) component $\exp(\omega^{(i)})$. The parameters $\kappa^{(i)}$ and $\omega^{(i)}$ affinely transform the higher layer random walk before exponential non-linearity.

The state transition model is a primitive building block of the HGF model. It repeats over time and layers. This structure is of prime importance for the rest of the chapter. We call the state transition function $f_t^{(i)} \triangleq p(x_t^{(i)} | x_{t-1}^{(i)}, x_t^{(i+1)}, \boldsymbol{\theta}^{(i)})$ a Gaussian-with-Controlled-Variance (GCV), which can be represented as a composite node in an FFG. The internal structure of GCV is given in Figure 4.1.

4.2.2 Likelihood Specifications

Technically any likelihood model can be combined with the state transitions. For continuous valued observations $y_t \in \mathbb{R}$, we choose a Gaussian likelihood

$$p\left(y_t | x_t^{(1)}\right) = \mathcal{N}\left(y_t | x_t^{(1)}, \psi^{-1}\right) \quad (4.3)$$

where ψ is the precision.

For binary outcomes $y_t \in \{0, 1\}$, firstly we use the cumulative density function of a standard Normal [82] as a link factor such that \mathbb{R} is mapped into $[0, 1]$ and then use these values as the parameter of the Bernoulli distribution

$$p\left(y_t | x_t^{(1)}\right) = \Phi\left(x_t^{(1)}\right)^{y_t} \left(1 - \Phi\left(x_t^{(1)}\right)\right)^{1-y_t} \quad (4.4)$$

$$\Phi\left(x_t^{(1)}\right) = \int_{-\infty}^{x_t^{(1)}} \mathcal{N}(z | 0, 1) dz. \quad (4.5)$$

4.2.3 Prior Specifications

Theoretically any kind of distribution, whose support agrees with the support of the random variables, can be chosen to reflect prior knowledge. Nevertheless, choosing an arbitrary prior has computational consequences. We choose the following priors

$$p\left(x_0^{(i)}\right) = \mathcal{N}\left(x_0^{(i)} | m_{x_0}^{(i)}, v_{x_0}^{(i)}\right), \quad i = 1, \dots, N \quad (4.6)$$

$$p\left(\kappa^{(i)}\right) = \mathcal{N}\left(\kappa^{(i)} | m_{\kappa}^{(i)}, v_{\kappa}^{(i)}\right), \quad i = 1, \dots, N - 1 \quad (4.7)$$

$$p\left(\omega^{(i)}\right) = \mathcal{N}\left(\omega^{(i)} | m_{\omega}^{(i)}, v_{\omega}^{(i)}\right), \quad i = 1, \dots, N - 1 \quad (4.8)$$

$$p(\xi) = \Gamma(\xi | a_{\xi}, b_{\xi}) \quad (4.9)$$

$$p(\psi) = \Gamma(\psi | a_{\psi}, b_{\psi}). \quad (4.10)$$

The computational motivation behind the prior choice is that these priors are conjugate to factors in the generative model. Since all of the factors are from the exponential family, choosing conjugate priors will ease the computation of variational messages (see Section 3.4.1.1). Another motivation is due to the maximum entropy principle, which tells us to choose priors that reflect our ignorance in agreement with the given constraints [7, Chapter 12]. For a continuous random variable whose support is \mathbb{R} , the maximum entropy distribution with fixed mean and variance is Gaussian. For a continuous random variable whose support is $\mathbb{R}_{>0}$, the maximum entropy distribution with a fixed positive mean and a fixed expectation of logarithm is Gamma distribution.

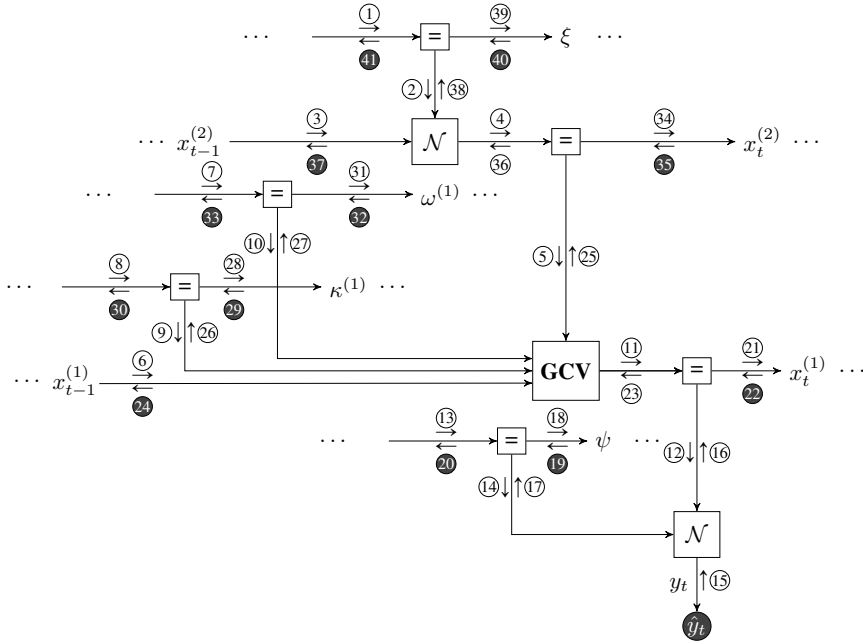


Figure 4.2: A Forney-style factor graph (FFG) for one time-segment of a 2 layer HGF model in Eq. 4.1 with a continuous likelihood model.

4.2.4 FFG Representation

Having specified functional forms of the factors for the HGF, we will now translate the mathematical notation into a graphical formalism. Figure 4.2 represents one time-segment of a 2-layer HGF model. The arrowheads indicate the generative directionthe associated variables name edges. The triple dots indicate a graph continuation (replication) in both temporal directions. Dark small nodes indicate data constraints. Equality nodes resolve the constraint that each variable can be connected to only two nodes. Details of the composite GCV node are provided in Figure 4.1. Messages propagated during inference are denoted on the edges with circles. Numbers on the messages indicate a computation order which results in a scheduled updating mechanism. Dark circles represent messages received from either future or messages to be passed to the past. If dark circled messages are fixed as proportional to 1 (making them effectively un-informative), the resulting message update scheme will correspond to filtering. The resulting message passing algorithm will correspond to smoothing if dark circled messages are propagated.

In essence, Figure 4.2 is a sub-graph of the terminated FFG for the 2 layer HGF model specified by (4.1). Terminated FFG corresponding to an N layered HGF model can be visualized as a chain expanding through the temporal dimension. The section at $t = 0$ will contain the prior, and the section at $t = T$ will not contain the equality factors and hierarchical di-

mension coupled via the GCV factor node. Due to hierarchical couplings and replication throughout the temporal dimension, TFFGs of the HGF models are doomed to contain loops.

If the number of layers in the HGF increases, the corresponding FFG will contain more layers. Consequently, this will imply more complex connective structures. Even though connectivity will get more complicated, underlying primitive forms such as Gaussian, GCV, and equality nodes will remain unchanged. Because the involved nodes are limited, we need only concern ourselves with the availability of message update rules for these limited cases. By isolating the complicated primitive structures and tabulating the required computation rules, we will be equipped with the machinery to automatically generate message-passing algorithms for arbitrarily connected HGFs. Instead of attacking inference from a global perspective, we will divide inference into sub-tasks composed of locally computing messages for a limited number of nodes.

The idea of the divide and conquer approach comes in handy when designing automated message passing algorithm generator frameworks. While automatically generating message-passing algorithms is straightforward, the applicability of the developed algorithm depends on the availability of a pre-computed set of message update rules. Tabulating message update rules for all the involved nodes and variables is a prerequisite for inference. In this chapter, the focus is on establishing the availability of message update rules for the HGF such that it can be added to a library of nodes in a message-passing framework.

4.3 Local Constraints and Lagrangians

Having specified the generative model for the HGF, we will now discuss constraints on the approximate distributions. The presentation of the corresponding Lagrangians will then follow the specification of constraints.

4.3.1 Factorization Constraints

Throughout the chapter we will assume that approximate distributions for observations, parameters and states for an HGF model are always decoupled, i.e.,

$$q(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y}) = q(\mathbf{y})q(\mathbf{x})q(\boldsymbol{\theta}). \quad (4.11)$$

Over the parameters and observations we assume a naive-mean field factorization such that

$$q(\boldsymbol{\theta}) = q(\xi)q(\psi) \prod_{i=1}^{N-1} q(\kappa^{(i)}) q(\omega^{(i)}) \quad (4.12)$$

$$q(\mathbf{y}) = \prod_{t=1}^T q(y_t). \quad (4.13)$$

For the states we will either assume the same structure as the model definition or assume a naive-mean field. Firstly, we assume the following Bethe factorization on the states

$$q(\mathbf{x}) = \prod_{i=1}^N \frac{\prod_{t=0}^T q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)}{\prod_{t=1}^{T-1} q\left(x_t^{(i)}\right)}. \quad (4.14)$$

Equation (4.14) implies that states are assumed to be decoupled over hierarchical layers whereas they are assumed to contain correlated structure over temporal dimension. If we assume (4.14) together with (4.12), we will obtain a structured factorization that is discussed in Section 3.4.1.1. Another alternative is to further assume independence over temporal dimension as well. Independence assumption over temporal dimension will yield a factorization of the form

$$q(\mathbf{x}) = \prod_{i=1}^N \prod_{t=0}^T q\left(x_t^{(i)}\right). \quad (4.15)$$

If we assume the factorization given by (4.15) and (4.12), we will obtain a naive mean-field factorization of Section 3.4.1.2 for the entire HGF model.

4.3.2 Form Constraints

In this chapter we will assume that observations are always data constrained which means

$$q(y_t) = \delta(y_t - \hat{y}_t), \quad (4.16)$$

where \hat{y}_t are observed values.

Often times functional forms of messages will hinder conjugate operations for message passing in the HGF. In order to retain fixed-form parametric approximations to non-conjugate message operations we will assume moment-matching constraint. Moment-matching will allow us to approximate non-exponential family marginals with exponential family marginals such that pre-computed rules that depend on an exponential family representation can be applied. For instance, we will assume that

$$\mathbb{E}_{q(\kappa^{(i)})} \left[T \left(\kappa^{(i)} \right) \right] = \mathbb{E}_{\tilde{q}(\kappa^{(i)})} \left[T \left(\kappa^{(i)} \right) \right], \quad (4.17)$$

where $T = \left[\kappa^{(i)} \left(\kappa^{(i)} \right)^2 \right]$. Further constraining $\tilde{q}(\kappa^{(i)})$ to have a fixed functional form such as a Gaussian, we can obtain a Gaussian approximation to $q(\kappa^{(i)})$. This approximation relies on estimation of the expected value of the sufficient statistics, where the underlying distribution $q(\kappa^{(i)})$ is constraint to be normalized. First a normalization constant is needed. Once normalization constant is determined, the moments of the normalized distribution is

needed such that we can match the moments of $\tilde{q}(\kappa^{(i)})$ to those obtained by an approximation method for $q(\kappa^{(i)})$ such as particle-filters [104] and importance sampling [105]. In Section 4.6.3 we will detail our methodology for moment-matching.

4.3.3 Lagrangian formulations

In this section, we will present useful Lagrangians for the HGF under constraints that are previously discussed. We will start with the simplest of constraint specifications and will build the complexity of constraints gradually. First combination of constraints have naive mean-field factorization for the entire variables, normalization and data-constraint for the observations. Under these specifications the first Lagrangian we consider is as follows

$$\begin{aligned}
L^1[q] = & \sum_{t=1}^T \left(\sum_{i=1}^N U[f_t^{(i)}] + U[p(y_t|x_t^{(1)})] \right) + \sum_{i=1}^{N-1} \left(U[p(\omega^{(i)})] + U[p(\kappa^{(i)})] \right) + \\
& \sum_{i=1}^N \left(U[p(x_0^{(i)})] - \sum_{t=1}^{T-1} H[q(x_t^{(i)})] \right) + U[p(\xi)] + U[p(\psi)] + \\
& \sum_{i=1}^{N-1} \left(H[q(\kappa^{(i)})] - H[q(\omega^{(i)})] \right) + \sum_{i=1}^N \sum_{t=1}^T \alpha_t^i \left(\int q(x_t^{(i)}) dx_t^{(i)} - 1 \right) + \\
& \sum_{i=1}^{N-1} \left(\beta^i \left(\int q(\kappa^{(i)}) d\kappa^{(i)} - 1 \right) + \gamma^i \left(\int q(\omega^{(i)}) d\omega^{(i)} - 1 \right) \right) + \\
& \beta^N \left(\int q(\xi) d\xi - 1 \right) + \gamma^N \left(\int q(\psi) d\psi - 1 \right) - H[q(\xi)]
\end{aligned} \tag{4.18}$$

The Lagrangian given by (4.18) is decomposed over temporal and hierarchical dimensions. Its summation indices are a bit different than those considered in Chapter 3 in the sense they do not correspond to indices for the actual node and edge sets. There will be replica variables to ensure the degree 2 constraints of the underlying FFG. Nevertheless, the Lagrangians represent the same quantity and are equivalent, as discussed in Chapter 3. We will not consider the replicas variables in this chapter for time-series models, as we find it more convenient to work with Lagrangians indexed by temporal and hierarchical dimensions. The only type of Lagrange multipliers is for the normalization constraints. For parameters, undetermined multipliers are not indexed by temporal index because we assume that these parameters do not vary over time. On the other hand, undetermined multipliers for the states are indexed by t . This is because distributions of states change over time, and their normalization needs to be ensured at every time step. L^1 given by (4.18) will lead to messages that will be computed by the result of Corollary 3.1.

The second set of constraints extends the first set of constraints by assuming a structured factorization for states on top of the marginalization constraints. Mean-field factorization

over parameters is not altered. The corresponding Lagrangian is equal to

$$\begin{aligned}
L^2[q] = & \sum_{i=1}^N \sum_{t=1}^T U \left[f_t^{(i)} \right] + \sum_{t=1}^T U \left[p \left(y_t | x_t^{(1)} \right) \right] + \sum_{i=1}^N U \left[p \left(x_0^{(i)} \right) \right] + \\
& \sum_{i=1}^{N-1} U \left[p \left(\omega^{(i)} \right) \right] + U \left[p(\xi) \right] + U \left[p(\psi) \right] - \sum_{i=1}^N \sum_{t=1}^T H \left[q \left(x_{t-1}^{(i)}, x_t^{(i)} \right) \right] - H \left[q(\xi) \right] + \\
& \sum_{i=1}^N \sum_{t=1}^{T-1} H \left[q \left(x_t^{(i)} \right) \right] - \sum_{i=1}^{N-1} H \left[q \left(\kappa^{(i)} \right) \right] - \sum_{i=1}^{N-1} H \left[q \left(\omega^{(i)} \right) \right] - H \left[q(\psi) \right] + \\
& \sum_{i=1}^N \sum_{t=1}^T \int \lambda_t^i \left(x_t^{(i)} \right) \left(\int q \left(x_{t-1}^{(i)}, x_t^{(i)} \right) dx_{t-1}^{(i)} - q \left(x_t^{(i)} \right) \right) dx_t^{(i)} + \sum_{i=1}^{N-1} U \left[p \left(\kappa^{(i)} \right) \right] + \\
& \sum_{i=1}^N \sum_{t=1}^T \alpha_t^i \left(\int q \left(x_t^{(i)} \right) dx_t^{(i)} - 1 \right) + \sum_{i=1}^{N-1} \beta^i \left(\int q \left(\kappa^{(i)} \right) d\kappa^{(i)} - 1 \right) + \\
& \sum_{i=1}^{N-1} \gamma^i \left(\int q \left(\omega^{(i)} \right) d\omega^{(i)} - 1 \right) + \beta^N \left(\int q(\xi) d\xi - 1 \right) + \gamma^N \left(\int q(\psi) d\psi - 1 \right) \quad (4.19)
\end{aligned}$$

Difference between (4.19) and (4.18) is that (4.19) includes an additional marginalization constraint in accordance with Theorem 3.2. Moreover, due to Bethe assumption (4.19) includes an extra negative joint entropy term which consequently necessitates addition of edge entropies for the states. This is the reason that $H \left[q \left(x_t^{(i)} \right) \right]$ terms have opposite signs in (4.19) and (4.18).

We will now consider Lagrangians that include moment matching constraints. Due to exponential non-linearity in the state transition variances, variational updates are not guaranteed to be in the exponential family. Indeed, we will see that the resulting distributions will not be in the exponential family for certain factorization choices. We will approximate the non-exponential family distribution with an exponential family distribution when we know that this is the case. Suppose that we constrain the parameters of the HGF by moment matching such that L^2 is augmented by

$$\begin{aligned}
L^3[q] = & L^2[q] + \sum_{i=1}^{N-1} \eta_i^\top \left(\mathbb{E}_{q(\kappa^{(i)})} \left[T \left(\kappa^{(i)} \right) \right] - \mathbb{E}_{\tilde{q}(\kappa^{(i)})} \left[T \left(\kappa^{(i)} \right) \right] \right) + \\
& \sum_{i=1}^{N-1} \zeta_i^\top \left(\mathbb{E}_{q(\omega^{(i)})} \left[T \left(\omega^{(i)} \right) \right] - \mathbb{E}_{\tilde{q}(\omega^{(i)})} \left[T \left(\omega^{(i)} \right) \right] \right). \quad (4.20)
\end{aligned}$$

Lagrangian given by (4.20) includes entropy terms for the approximating distributions \tilde{q} such that if minimization is carried out with respect to \tilde{q} then we recover the maximum-entropy exponential family distributions under the moment-matching constraints.

Lastly, we consider the case where we also constraint the states via moment matching. The resulting Lagrangian will then correspond to

$$L^4[q] = L^3[q] + \sum_{t=1}^T \sum_{i=1}^N v_i^\top \left(\mathbb{E}_{q(x_t^{(i)})} [T(x_t^{(i)})] - \mathbb{E}_{\bar{q}(x_t^{(i)})} [T(x_t^{(i)})] \right) \quad (4.21)$$

where we again choose maximum-entropy exponential family approximating distribution for the state marginals.

4.4 Problem Definition

From a signal processing perspective, we are interested in joint tracking of states and parameters for the HGF model (4.1). This can be achieved by sequential Bayesian updating that is equivalent to solving Chapman-Kolmogorov integral [60, Ch.4]

$$p(\mathbf{x}_t, \boldsymbol{\theta} | \mathbf{y}_{1:t}) = \frac{p(y_t | \mathbf{x}_t)}{p(y_t | \mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \boldsymbol{\theta}) p(\mathbf{x}_{t-1}, \boldsymbol{\theta} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}, \quad (4.22)$$

where $\mathbf{x}_t = [x_t^{(1)} \ x_t^{(2)} \ \dots \ x_t^{(N)}]^\top$ is the collection of hierarchical states at a particular time t , and the denominator $p(y_t | \mathbf{y}_{1:t-1})$ is a running Bayesian evidence score, which can be evaluated as

$$p(y_t | \mathbf{y}_{1:t-1}) = \int p(y_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \boldsymbol{\theta}) p(\mathbf{x}_{t-1}, \boldsymbol{\theta} | \mathbf{y}_{1:t-1}) d\boldsymbol{\theta} d\mathbf{x}_{t-1} d\mathbf{x}_t. \quad (4.23)$$

Inside the integral, in (4.22) the first term corresponds to state transition whose parameters are $\boldsymbol{\theta}$ and the second term is the posterior from the previous time step. While (4.22) and (4.23) represent the exact solutions to joint tracking and evidence updating, due to the integration over states (and parameters) and non-conjugate prior-posterior pairing, the computation of these integrals is intractable in the HGF model. Indeed, the intractable updating mechanism for the joint posterior via (4.22) is a consequence of an underlying marginalization constraint specification on the Bethe free energy corresponding to the HGF model.

Due to the intractability of exact Bayesian inference (sum-product), we are interested in obtaining approximate distributions that minimize the Lagrangians of the previous section. As we have argued in Chapter 3, the variational view of the problem offers a principled approach that includes exact Bayesian inference as a particular case. We will try to obtain approximate solutions employing variational message passing by changing the constraint specifications.

We are generally interested in computing variational distributions for even a larger class of Lagrangians, where constraint specifications include hybrid forms or constraints specified by a designer. Even though the complexity of the Lagrangians will grow with the arbitrary specification of constraints, the problem definition will remain the same. Suppose that we

have formed a Lagrangian L by a constraint set $\mathcal{L}(\mathcal{G})$ corresponding to the graph of an N -layered HGF $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Then, in the spirit of Chapter 3, we are interested in obtaining the solutions to the following minimization problem

$$q^* = \arg \min_q L[q]. \quad (4.24)$$

Solutions to the minimization problem can be obtained locally by computing messages around each factor (node) of the graph \mathcal{G} for all the involved random variables (edges). In the HGF model, we do not have access to tabulated messages only around a GCV node. All the required variational and sum-product messages for Gamma and Gaussian factors are tabulated and can be found in [106]. This means we only need to determine the functional form of messages for a GCV node for all the involved variables and combine them with the already tabulated messages.

4.5 Gaussian with Controlled Variance

Before we dwell into computation of the message update equations for the GCV node, we will discuss the mechanics of the GCV node. Technically, GCV is a composite node where a Gaussian factor is augmented with a non-linear transform for the variance (precision) term. We will show the reasoning behind choosing an affine transform followed and exponential non-linearity as the source of non-linearity in the HGF. Suppose we have a node function defined by

$$f(y, x, z) = \mathcal{N}(y|x, g(z)), \quad (4.25)$$

where g is a positive valued function. Since g is a positive valued function then there must be a function h such that $g(z) = \exp(h(z))$. Let us expand $h(z)$ around a point a such that

$$\begin{aligned} h(z) &= h(a) + \left. \frac{d}{dz} h(z) \right|_a (z - a) + \mathcal{O}(2) \\ &= \log g(a) + \frac{g'(a)}{g(a)} (z - a) + \mathcal{O}(2) \\ &= \frac{g'(a)}{g(a)} z + \log g(a) - a \frac{g'(a)}{g(a)} + \mathcal{O}(2). \end{aligned} \quad (4.26)$$

If we make the following identifications

$$\kappa \triangleq \frac{g'(a)}{g(a)} \text{ and } \omega \triangleq \log g(a) - a \frac{g'(a)}{g(a)}, \quad (4.27)$$

we will obtain a first order approximation to g as

$$h(z) = \kappa z + \omega + \mathcal{O}(2) \quad (4.28)$$

$$g(z) \approx \tilde{g}(z) = \exp(\kappa z + \omega). \quad (4.29)$$

In Equation (4.29), \tilde{g} is an approximation which is only exact when g itself is an exponential of a first-order polynomial. Ignoring higher-order terms means that the HGF is restricted to first-order coupling functions. If the effect of higher terms is stronger, then the approximation will severely degrade as we move away from the expansion point. Moreover, κ and ω are unique given the expansion point a . Without knowing the underlying non-linearity g and an expansion point a , infinitely many combinations of κ and ω yield the same transformation result.

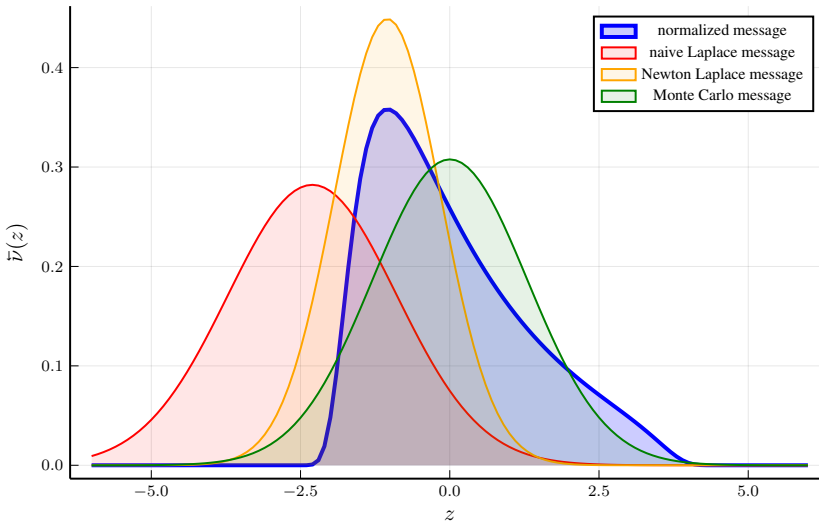


Figure 4.3: Illustration of Laplace approximations to the message $\tilde{v}(z)$ with $m_\kappa = 1, v_\kappa = 1$ and $\gamma_3\gamma_8 = 0.1$. The normalization constant for the message $\tilde{v}(z)$ is obtained through Monte-Carlo summation and the normalized message is plotted in blue. We can see that this message is unimodal and has skewness. A Gaussian approximation by Monte-Carlo summation is given in color green. Naive Laplace approximation under the assumption that $v_\kappa \rightarrow 0$ is plotted with color red. Laplace approximation via Householder's method is plotted in yellow.

4.6 Message Computations

4.6.1 Structured Factorization Computations

We will now compute the messages for the GCV node under structured factorization assumptions. We will denote the factor function of the GCV by

$$f(y, x, z, \kappa, \omega) = \mathcal{N}(y|x, \exp(\kappa z + \omega)) . \quad (4.30)$$

Table 4.1: Message passing update rules for the GCV Node under structured factorization.

GCV Node	Auxiliary						
<p style="text-align: center;">$\mathcal{N}(y x, \exp(\kappa z + \omega))$</p>	γ_1	$m_z^2 v_\kappa + m_\kappa^2 v_z + v_z v_\kappa$					
	γ_2	$\exp(-m_\kappa m_z + 0.5 \gamma_1)$					
	γ_3	$\exp(-m_\omega + 0.5 v_\omega)$					
	γ_4	$(m_1 - m_2)^2 + \Sigma_{11} + \Sigma_{22} - \Sigma_{12} - \Sigma_{21}$					
	$\gamma_5(z)$	$\gamma_4 \gamma_3 \exp(-m_\kappa z + 0.5 z^2 v_\kappa)$					
	$\gamma_6(\kappa)$	$\gamma_4 \gamma_3 \exp(-m_z \kappa + 0.5 \kappa^2 v_z)$					
	$\gamma_7(\omega)$	$\gamma_4 \gamma_2 \exp(-\omega)$					
	m	Σ	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>\vec{m}_x / \vec{v}_x</td> <td></td> </tr> <tr> <td>\vec{m}_y / \vec{v}_y</td> <td></td> </tr> </table>	\vec{m}_x / \vec{v}_x		\vec{m}_y / \vec{v}_y	
	\vec{m}_x / \vec{v}_x						
	\vec{m}_y / \vec{v}_y						
	Σ^{-1}		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>$1/\vec{v}_x + \gamma_2 \gamma_3$</td> <td>$-\gamma_2 \gamma_3$</td> </tr> <tr> <td>$-\gamma_2 \gamma_3$</td> <td>$1/\vec{v}_y + \gamma_2 \gamma_3$</td> </tr> </table>	$1/\vec{v}_x + \gamma_2 \gamma_3$	$-\gamma_2 \gamma_3$	$-\gamma_2 \gamma_3$	$1/\vec{v}_y + \gamma_2 \gamma_3$
	$1/\vec{v}_x + \gamma_2 \gamma_3$	$-\gamma_2 \gamma_3$					
	$-\gamma_2 \gamma_3$	$1/\vec{v}_y + \gamma_2 \gamma_3$					
	Messages						
$\vec{v}(y)$		$\mathcal{N}(y \vec{m}_y, \vec{v}_y)$					
$\vec{v}(y)$		$\mathcal{N}\left(y \vec{m}_x, \vec{v}_x + \frac{1}{\gamma_2 \gamma_3}\right)$					
$\vec{v}(x)$		$\mathcal{N}(x \vec{m}_x, \vec{v}_x)$					
$\vec{v}(x)$		$\mathcal{N}\left(x \vec{m}_y, \vec{v}_y + \frac{1}{\gamma_2 \gamma_3}\right)$					
$\vec{v}(z)$		$\mathcal{N}(z \vec{m}_z, \vec{v}_z)$					
Marginals							
$\vec{v}(z)$		$\exp(-0.5(m_\kappa z + \gamma_5(z)))$					
$q(x, y) = \mathcal{N}\left(\begin{array}{c c} y & \\ \hline x & \end{array} \middle m, \Sigma\right)$	$\vec{v}(\kappa)$	$\mathcal{N}(\kappa \vec{m}_\kappa, \vec{v}_\kappa)$					
$q(z) = \mathcal{N}(z m_z, v_z)$	$\vec{v}(\kappa)$	$\exp(-0.5(m_z \kappa + \gamma_6(\kappa)))$					
$q(\kappa) = \mathcal{N}(\kappa m_\kappa, v_\kappa)$	$\vec{v}(\omega)$	$\mathcal{N}(\omega \vec{m}_\omega, \vec{v}_\omega)$					
$q(\omega) = \mathcal{N}(m_\omega, v_\omega)$	$\vec{v}(\omega)$	$\exp(-0.5(\omega + \gamma_7(\omega)))$					
Entropy		Average Energy					
$0.5 \log\left((2\pi e)^5 \Sigma v_z v_\kappa v_\omega\right)$		$0.5(\log 2\pi + m_\kappa m_z + m_\omega + \gamma_4 \gamma_3 \gamma_2)$					

In Table 4.1 we have included the result of messages that we intend to compute. For a structured factorization for the variables associated with the GCV y, x, z, κ, ω , is we assume

$$q(y, x, z, \kappa, \omega) = q(y, x)q(z)q(\kappa)q(\omega). \quad (4.31)$$

Before we proceed with the message update rules, we compute intermediate expectations $\mathbb{E}_{q_{xy}}[(x - y)^2]$, $\mathbb{E}_{q_z q_\kappa}[\exp(\kappa z)]$ and $\mathbb{E}_{q_\omega}[\exp(\omega)]$ that are needed for computation of messages and marginals. Assuming $q(x, y) \propto \mathcal{N}(m, \Sigma)$, where $m \in \mathbb{R}^2$ is the mean and $\Sigma \in \mathbb{R}^{2 \times 2}$ is the covariance of the joint distribution, we can write

$$\mathbb{E}_{q_{xy}}[(x - y)^2] = \underbrace{(m_1 - m_2)^2 + \Sigma_{11} + \Sigma_{22} - \Sigma_{21} - \Sigma_{12}}_{\gamma_4}. \quad (4.32)$$

Following [107] we can approximate the multiplication of two Gaussian random variables with a Gaussian random variable. Assuming independence between κ and z we write the approximate distribution corresponding to the multiplication as

$$q(z\kappa) \stackrel{approx.}{\propto} \mathcal{N}(z\kappa | m_z m_\kappa, m_z^2 v_\kappa + m_\kappa^2 v_z + v_z v_\kappa), \quad (4.33)$$

where we define

$$\gamma_1 \triangleq m_z^2 v_\kappa + m_\kappa^2 v_z + v_z v_\kappa. \quad (4.34)$$

This means $\exp(z\kappa)$ is log-normally distributed with mean and variance as in Eq. 4.35a. We approximate the joint expectation of this log-normally distributed product of two independent normal random variables as per Eq. 4.35b. To arrive at Eq. 4.35b we use the fact that for a log-normally distributed z , for all $n \in \mathbb{R}$ all moments are well defined and analytically given by $\mathbb{E}_{q_z}[z^n] = \exp\left(nm_z + \frac{n^2 v_z}{2}\right)$. For $n = -1$ we obtain Eq. 4.35b.

$$\exp(z\kappa) \stackrel{approx.}{\sim} \log \mathcal{N}(m_z m_\kappa, \gamma_1) \quad (4.35a)$$

$$\mathbb{E}_{q_z q_\kappa}[\exp(\kappa z)] \approx \exp\left(m_z m_\kappa + \frac{\gamma_1}{2}\right) \quad (4.35b)$$

$$\mathbb{E}_{q_z q_\kappa}\left[\frac{1}{\exp(\kappa z)}\right] \approx \exp\left(-m_z m_\kappa + \frac{\gamma_1}{2}\right) \triangleq \gamma_2. \quad (4.35c)$$

Similarly we can write the expectation with respect to $q(\omega)$ as

$$\mathbb{E}_{q_\omega}[\exp(\omega)] \approx \exp\left(m_\omega + \frac{v_\omega}{2}\right) \quad (4.36a)$$

$$\mathbb{E}_{q_\omega}\left[\frac{1}{\exp(\omega)}\right] \approx \exp\left(-m_\omega + \frac{v_\omega}{2}\right) \triangleq \gamma_3. \quad (4.36b)$$

We are now equipped with the needed expectations to obtain expressions of Table 4.1. We start our derivation with $\vec{\nu}(y)$. Using results of Theorem 3.2 we can write

$$\vec{\nu}(y) \propto \int \vec{\nu}(x) \tilde{f}(x, y) dx \quad (4.37a)$$

$$\propto \int \vec{\nu}(x) \exp\left(-0.5 \mathbb{E}_{q_{xy}} \left[\frac{(y-x)^2}{\exp(\kappa z + \omega)} \right]\right) dx. \quad (4.37b)$$

Let us assume that incoming message is a Gaussian i.e. $\vec{\nu}(x) \propto \mathcal{N}(x | \vec{m}_x, \vec{v}_x)$. Using Eq. (4.36b), (4.35c) and recognizing that the integral in Eq. 4.37b is a convolution of two Gaussian forms we can write

$$\vec{\nu}(y) \propto \int \vec{\nu}(x) \exp(-0.5 \gamma_2 \gamma_3 (y-x)^2) dx \quad (4.38a)$$

$$\propto \mathcal{N}\left(y | \vec{m}_x, \vec{v}_x + \frac{1}{\gamma_2 \gamma_3}\right). \quad (4.38b)$$

Assuming $\vec{\nu}(y) \propto \mathcal{N}(y | \vec{m}_y, \vec{v}_y)$, we can compute the message $\vec{\nu}(x)$ in a similar manner

$$\vec{\nu}(x) \propto \mathcal{N}\left(x | \vec{m}_y, \vec{v}_y + \frac{1}{\gamma_2 \gamma_3}\right). \quad (4.39)$$

Using $\vec{\nu}(y)$, $\vec{\nu}(x)$ and $\tilde{f}(x, y)$ we can compute the joint by Theorem 3.2

$$q(x, y) \propto \vec{\nu}(x) \vec{\nu}(y) \tilde{f}(x, y) \quad (4.40a)$$

$$\propto \mathcal{N}(x | \vec{m}_x, \vec{v}_x) \mathcal{N}(y | \vec{m}_y, \vec{v}_y) \mathcal{N}\left(y | x, \frac{1}{\gamma_2 \gamma_3}\right) \quad (4.40b)$$

$$\propto \mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix} \middle| \begin{bmatrix} \vec{m}_x \\ \vec{m}_y \end{bmatrix}, \begin{bmatrix} \vec{v}_x & 0 \\ 0 & \vec{v}_y \end{bmatrix}\right) \mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix} \middle| \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \frac{1}{\gamma_2 \gamma_3} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\right) \quad (4.40c)$$

$$\propto \mathcal{N}_\Lambda\left(\begin{bmatrix} x \\ y \end{bmatrix} \middle| \Sigma \begin{bmatrix} \vec{m}_x / \vec{v}_x \\ \vec{m}_y / \vec{v}_y \end{bmatrix}, \begin{bmatrix} 1/\vec{v}_x + \gamma_2 \gamma_3 & -\gamma_2 \gamma_3 \\ -\gamma_2 \gamma_3 & 1/\vec{v}_y + \gamma_2 \gamma_3 \end{bmatrix}\right), \quad (4.40d)$$

where we identify the precision matrix as

$$\Sigma^{-1} = \begin{bmatrix} 1/\vec{v}_x + \gamma_2 \gamma_3 & -\gamma_2 \gamma_3 \\ -\gamma_2 \gamma_3 & 1/\vec{v}_y + \gamma_2 \gamma_3 \end{bmatrix}. \quad (4.41)$$

Eq.(4.40c) follows by grouping the first two terms of (4.40b) into a multivariate Gaussian distribution and the last term can be obtained by arranging the quadratic terms of the Gaussian distribution. Note however that the second term of (4.40c) is not a proper distribution as the determinant of covariance matrix is 0. We can however write (4.40d) by noting that

Table 4.2: Message passing update rules for the GCV Node under naive mean-field factorization. Definitions for γ_1 to γ_7 follow from Table 4.1.

GCV Node	Auxiliary	
<p style="text-align: center;">$\mathcal{N}(y x, \exp(\kappa z + \omega))$</p>	γ_8	$(m_y - m_x)^2 + v_x + v_y$
	$\gamma_9(z)$	$\gamma_3 \gamma_8 \exp(-m_\kappa z + 0.5 z^2 v_\kappa)$
	$\gamma_{10}(\kappa)$	$\gamma_3 \gamma_8 \exp(-\kappa m_z + 0.5 \kappa^2 v_z)$
	$\gamma_{11}(\omega)$	$\gamma_2 \gamma_8 \exp(-\omega)$
	Messages	
	$\vec{v}(y)$	$\mathcal{N}(y \vec{m}_y, \vec{v}_y)$
	$\vec{v}(y)$	$\mathcal{N}\left(y m_x, \frac{1}{\gamma_2 \gamma_3}\right)$
	$\vec{v}(x)$	$\mathcal{N}(x \vec{m}_x, \vec{v}_x)$
	$\vec{v}(x)$	$\mathcal{N}\left(x m_y, \frac{1}{\gamma_2 \gamma_3}\right)$
	Marginals	
	$q(y) = \mathcal{N}(y m_y, v_y)$	$\vec{v}(z)$
$q(x) = \mathcal{N}(x m_x, v_x)$	$\vec{v}(z)$	$\exp(-0.5(m_\kappa z + \gamma_9(z)))$
$q(z) = \mathcal{N}(z m_z, v_z)$	$\vec{v}(\kappa)$	$\mathcal{N}(\kappa \vec{m}_\kappa, \vec{v}_\kappa)$
$q(\kappa) = \mathcal{N}(\kappa m_\kappa, v_\kappa)$	$\vec{v}(\kappa)$	$\exp(-0.5(m_z \kappa + \gamma_{10}(\kappa)))$
$q(\omega) = \mathcal{N}(\omega m_\omega, v_\omega)$	$\vec{v}(\omega)$	$\mathcal{N}(\omega \vec{m}_\omega, \vec{v}_\omega)$
$\vec{v}(\omega)$	$\vec{v}(\omega)$	$\exp(-0.5(\omega + \gamma_{11}(\omega)))$
Entropy		Average Energy
$0.5 \log\left((2\pi e)^5 v_y v_x v_z v_\kappa v_\omega\right)$	$0.5 (\log 2\pi + m_\kappa m_z + m_\omega + \gamma_4 \gamma_3 \gamma_8)$	

summation of two quadratic terms in the exponent of (4.40c) results in a quadratic term which is proportional to a multivariate Gaussian with a precision parameterization (that is the reason we put Λ as a subscript). Note that (4.40d) is a proper distribution, since the covariance is a proper positive definite matrix.

We now compute the message from the GCV node towards the control state z as follows

$$\bar{\nu}(z) \propto \exp(\mathbb{E}_{\setminus q_z}[\log f(y, x, z, \kappa, \omega)]) \quad (4.42a)$$

$$\propto \exp\left(\mathbb{E}_{\setminus q_z}\left[-\frac{\kappa z + \omega}{2}\right] + \mathbb{E}_{\setminus q_z}\left[-\frac{(y-x)^2}{2 \exp(\kappa z + \omega)}\right]\right) \quad (4.42b)$$

$$\propto \exp\left(-0.5\left(zm_\kappa + \frac{\gamma_4 \gamma_3}{\exp\left(zm_\kappa - \frac{z^2 v_\kappa}{2}\right)}\right)\right) \quad (4.42c)$$

$$\propto \exp(-0.5(zm_\kappa + \gamma_5(z))), \quad (4.42d)$$

where we define

$$\gamma_5(z) \triangleq \frac{\gamma_4 \gamma_3}{\exp\left(zm_\kappa - \frac{z^2 v_\kappa}{2}\right)}. \quad (4.43)$$

Expectations of (4.42b) are given by (4.36b) and (4.32). Similarly for $\bar{\nu}(\kappa)$ we can write

$$\bar{\nu}(\kappa) \propto \exp(\mathbb{E}_{\setminus q_\kappa}[\log f(y, x, z, \kappa, \omega)]) \quad (4.44a)$$

$$\propto \exp\left(\mathbb{E}_{\setminus q_\kappa}\left[-\frac{\kappa z + \omega}{2}\right] + \mathbb{E}_{\setminus q_\kappa}\left[-\frac{(y-x)^2}{2 \exp(\kappa z + \omega)}\right]\right) \quad (4.44b)$$

$$\propto \exp\left(-0.5\left(\kappa m_z + \frac{\gamma_4 \gamma_3}{\exp\left(\kappa m_z - \frac{\kappa^2 v_z}{2}\right)}\right)\right) \quad (4.44c)$$

$$\propto \exp(-0.5(\kappa m_z + \gamma_6(\kappa))), \quad (4.44d)$$

where we define

$$\gamma_6(\kappa) \triangleq \frac{\gamma_4 \gamma_3}{\exp\left(\kappa m_z - \frac{\kappa^2 v_z}{2}\right)}. \quad (4.45)$$

Last remaining message is towards ω edge. We compute the message as

$$\bar{\nu}(\omega) \propto \exp(\mathbb{E}_{\setminus q_\omega}[\log f(y, x, z, \kappa, \omega)]) \quad (4.46a)$$

$$\propto \exp\left(\mathbb{E}_{\setminus q_\omega}\left[-\frac{\kappa z + \omega}{2}\right] + \mathbb{E}_{\setminus q_\omega}\left[-\frac{(y-x)^2}{2 \exp(\kappa z + \omega)}\right]\right) \quad (4.46b)$$

$$\propto \exp\left(-0.5\left(\omega + \frac{\gamma_4 \gamma_2}{\exp(\omega)}\right)\right) \quad (4.46c)$$

$$\propto \exp(-0.5(\omega + \gamma_7(\omega))), \quad (4.46d)$$

where we substitute

$$\gamma_7(\omega) \triangleq \frac{\gamma_4 \gamma_2}{\exp(\omega)}. \quad (4.47)$$

4.6.2 Naive Mean-Field Factorization Computations

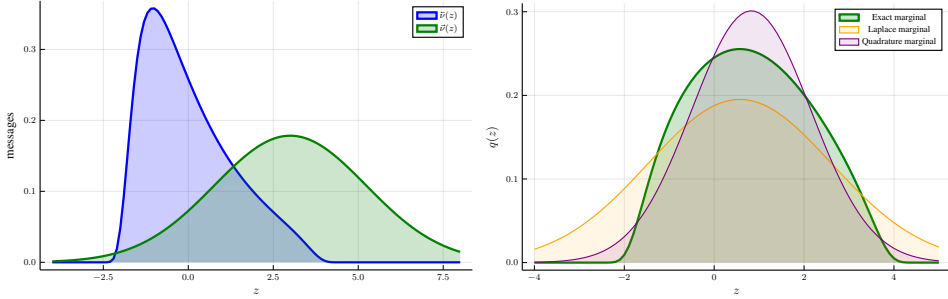


Figure 4.4: (Left) Blue curve represents the message $\vec{v}(z)$ with $m_\kappa = 1$, $v_\kappa = 1$ and $\gamma_3\gamma_8 = 0.1$ same as in Figure 4.3. Green curve represents the message $\vec{v}(z)$ with $\tilde{m}_z = 3.0$ and $\tilde{v}_z = 5.0$. (Right) Green curve plots the marginal, that corresponds to the multiplication of the messages on the left of the figure, computed by Monte-Carlo sampling. Orange curve represents the marginal approximated by Laplace's method corresponding to (4.82). Purple curve displays the marginal obtained by moment-matching through quadrature (4.94).

We will now compute message updates for the GCV node under naive mean-field factorization, i.e.

$$q(y, x, z, \kappa, \omega) = q(y)q(x)q(z)q(\kappa)q(\omega). \quad (4.48)$$

A quick reference for the message computation rules can be found in Table 4.2. We start our derivation with $\vec{v}(y)$. Using the result of Corollary 3.1 we compute

$$\vec{v}(y) \propto \exp(\mathbb{E}_{q_y} [\log f(y, x, z, \kappa, \omega)]) \quad (4.49a)$$

$$\propto \exp \left(\mathbb{E}_{q_y} \left[-0.5 \frac{(y-x)^2}{\exp(\kappa z + \omega)} \right] \right) \quad (4.49b)$$

Combining Eq. 4.36b and 4.35c we see that

$$\vec{v}(y) \propto \exp \left(-0.5\gamma_2\gamma_3 (y - m_x)^2 \right) \quad (4.50a)$$

$$\propto \mathcal{N} \left(y \mid m_x, \frac{1}{\gamma_2\gamma_3} \right) \quad (4.50b)$$

where m_x is the mean of q_x . The difference between the naive-mean field and the structured update for the y edge is that: naive-mean field uses only the mean of the recognition distribution q_x , whereas structured update utilizes the mean and variance of the incoming message

$\tilde{\nu}(x)$. Finding the message $\tilde{\nu}(x)$ is straightforward as it is identical with $\tilde{\nu}(y)$.

$$\tilde{\nu}(x) \propto \exp(\mathbb{E}_{q_x}[\log f(y, x, z, \kappa, \omega)]) \quad (4.51a)$$

$$\propto \exp\left(\mathbb{E}_{q_x}\left[-0.5 \frac{(y-x)^2}{\exp(\kappa z + \omega)}\right]\right) \quad (4.51b)$$

Combining Eq. 4.36b and 4.35c we see that

$$\tilde{\nu}(x) \propto \exp\left(-0.5\gamma_2\gamma_3(x - m_y)^2\right) \quad (4.52a)$$

$$\propto \mathcal{N}\left(x \mid m_y, \frac{1}{\gamma_2\gamma_3}\right) \quad (4.52b)$$

We now proceed with computation of the message $\tilde{\nu}(z)$. Again using the result of Corollary 3.1, we compute the message as

$$\tilde{\nu}(z) \propto \exp(\mathbb{E}_{q_z}[\log f(y, x, z, \kappa, \omega)]) \quad (4.53a)$$

$$\propto \exp\left(\mathbb{E}_{q_z}\left[-\frac{\kappa z + \omega}{2}\right] + \mathbb{E}_{q_z}\left[-0.5 \frac{(y-x)^2}{\exp(\kappa z + \omega)}\right]\right) \quad (4.53b)$$

$$\propto \exp\left(-0.5\left(zm_\kappa + m_\omega + \frac{(m_y - m_x)^2 + v_x + v_y}{\exp(zm_\kappa + m_\omega - \frac{z^2 v_\kappa + v_\omega}{2})}\right)\right) \quad (4.53c)$$

$$\propto \exp\left(-0.5\left(zm_\kappa + \gamma_3 \exp(-m_\kappa z + 0.5z^2 v_\kappa) \left((m_y - m_x)^2 + v_x + v_y\right)\right)\right) \quad (4.53d)$$

If we make the following definitions

$$\gamma_8 \triangleq (m_y - m_x)^2 + v_x + v_y \quad (4.54a)$$

$$\gamma_9(z) \triangleq \gamma_3 \gamma_8 \exp(-m_\kappa z + 0.5z^2 v_\kappa) \quad (4.54b)$$

we can write the message towards z edge as

$$\tilde{\nu}(z) \propto \exp(-0.5(zm_\kappa + \gamma_9(z))) \quad (4.55)$$

Message towards the κ edge can be computed by

$$\bar{v}(\kappa) \propto \exp \left(\mathbb{E}_{\backslash q_\kappa} [\log f(y, x, z, \kappa, \omega)] \right) \quad (4.56a)$$

$$\propto \exp \left(\mathbb{E}_{\backslash q_\kappa} \left[-\frac{\kappa z + \omega}{2} \right] + \mathbb{E}_{\backslash q_\kappa} \left[-0.5 \frac{(y-x)^2}{\exp(\kappa z + \omega)} \right] \right) \quad (4.56b)$$

$$\propto \exp \left(-0.5 \left(\kappa m_z + m_\omega + \frac{(m_y - m_x)^2 + v_x + v_y}{\exp(\kappa m_z + m_\omega - \frac{\kappa^2 v_z + v_\omega}{2})} \right) \right) \quad (4.56c)$$

$$\propto \exp \left(-0.5 \left(\kappa m_z + \gamma_3 \exp(-\kappa m_z + 0.5 \kappa^2 v_z) \left((m_y - m_x)^2 + v_x + v_y \right) \right) \right) \quad (4.56d)$$

If we make the following definition

$$\gamma_{10}(\kappa) \triangleq \gamma_3 \gamma_8 \exp(-\kappa m_z + 0.5 \kappa^2 v_z) \quad (4.57)$$

we can write the message towards κ edge as

$$\bar{v}(\kappa) \propto \exp(-0.5(\kappa m_z + \gamma_{10}(\kappa))) . \quad (4.58)$$

Lastly we compute the message towards ω edge.

$$\bar{v}(\omega) \propto \exp \left(\mathbb{E}_{\backslash q_\omega} [\log f(y, x, z, \kappa, \omega)] \right) \quad (4.59a)$$

$$\propto \exp \left(\mathbb{E}_{\backslash q_\omega} \left[-\frac{\kappa z + \omega}{2} \right] + \mathbb{E}_{\backslash q_\omega} \left[-0.5 \frac{(y-x)^2}{\exp(\kappa z + \omega)} \right] \right) \quad (4.59b)$$

$$\propto \exp \left(-0.5 \left(\omega + \gamma_2 \exp(-\omega) \left((m_y - m_x)^2 + v_x + v_y \right) \right) \right) \quad (4.59c)$$

Finally, making the following definition

$$\gamma_{11}(\omega) \triangleq \gamma_2 \gamma_8 \exp(-\omega) \quad (4.60)$$

we can write the message towards ω edge as

$$\bar{v}(\omega) \propto \exp(-0.5(\omega + \gamma_{11}(\omega))) \quad (4.61)$$

Messages $\bar{v}(z)$, $\bar{v}(\omega)$ and $\bar{v}(\kappa)$ as derived in this section are not in the exponential family. Since these messages are not in the exponential family, computation of convolution integrals and marginal distributions by means of multiplication become problematic. In order to avoid these problems, we will introduce a set of possible approximations in the next sections.

4.6.3 Laplace Approximated Messages

Let us consider the message $\tilde{\nu}(z)$ as derived under naive mean-field constraint (4.55). There is no loss of generality with this consideration as the methods described in this section can be applied to messages that are derived under structured factorization as well as to the messages $\tilde{\nu}(\omega)$ and $\tilde{\nu}(\kappa)$. We can write the log message as proportional to

$$\log \tilde{\nu}(z) \propto -0.5 \underbrace{(zm_\kappa + \gamma_9(z))}_{l(z)}. \quad (4.62)$$

We can make a second order Taylor approximation to $l(z)$ around z_0 by

$$l(z) \approx l(z_0) + l'(z_0)(z - z_0) + \frac{l''(z_0)(z - z_0)^2}{2}. \quad (4.63)$$

If we can determine a point z_0 where the first order derivative vanishes $l'(z_0) = 0$, then $\log \tilde{\nu}(z)$ will be proportional to only a quadratic term. Hence the message $\tilde{\nu}(z)$ will be proportional to a Gaussian with mean z_0 and variance $\frac{1}{l''(z_0)}$ [6, Chapter 4.4]. This means that we are looking for a solution for the equation

$$l'(z) = 0 \quad (4.64)$$

$$-0.5 (m_\kappa + \gamma_9'(z)) = 0 \quad (4.65)$$

$$-m_\kappa = \gamma_9'(z) \quad (4.66)$$

$$-m_\kappa = \gamma_3 \gamma_8 \exp(-m_\kappa z + 0.5z^2 v_\kappa) (-m_\kappa + zv_\kappa) \quad (4.67)$$

Equation 4.67 has no analytic solution for a root without further approximations. There are two ways to proceed here. First one is to assume that $v_\kappa \rightarrow 0$. This assumption allows us to find a root analytically as

$$-m_\kappa = -\gamma_3 \gamma_8 \exp(-m_\kappa z) m_\kappa \quad (4.68)$$

$$1 = \gamma_3 \gamma_8 \exp(-m_\kappa z) \quad (4.69)$$

$$z_0 = \frac{\log(\gamma_3 \gamma_8)}{m_\kappa} \quad (4.70)$$

Choosing z_0 as given by (4.70) allows us to approximate the message $\tilde{\nu}(z)$ as

$$\tilde{\nu}(z) \stackrel{\text{approx.}}{\propto} \mathcal{N} \left(z \left| \frac{\log(\gamma_3 \gamma_8)}{m_\kappa}, \frac{2}{m_\kappa^2} \right. \right). \quad (4.71)$$

The derivation above can only be exact if a Dirac delta belief constrains κ as to the condition $v_\kappa \rightarrow 0$ is valid only for this case. Moreover, the result of the message in (4.71)

has a variance term that depends only on m_κ , which will potentially decrease the information content that is going to be propagated to the higher layers by the GCV node. To avoid these problems, we can drop the assumption that $v_\kappa \rightarrow 0$ and use a numerical method to find an approximate numerical solution for (4.67). We will describe this second method to approximate a non-exponential family message with a Gaussian form.

Numerical solutions to obtain the roots of a function is of fundamental importance since Babylonians and the literature on these methods are vast [108]. We choose to work with Householder's method for finding the roots. Householder's method looks for the solution of $f(z) = 0$ by an iterative procedure [109]

$$z_{n+1} = z_n + (p+1) \frac{\left(\frac{1}{f(z_n)}\right)^{(p)}}{\left(\frac{1}{f(z_n)}\right)^{(p+1)}} \quad (4.72)$$

where $\left(\frac{1}{f(z)}\right)^{(p)}$ is the p^{th} order derivative of the inverse function. Iterations given by (4.72) has a convergence rate of $p+2$. Choosing $p=0$ yields Newton's method with quadratic convergence while choosing $p=1$ yields Halley's method with cubic convergence [108] [109]. We choose to work with $p=1$ and apply the iteration to find the solution of $l'(z) = 0$ such that we obtain

$$z_{n+1} = z_n - \frac{l'(z_n)}{l''(z_n)} \left(1 - \frac{l'(z_n) l'''(z_n)}{l''(z_n) 2l''(z_n)}\right)^{-1} \quad (4.73)$$

$$= z_n - \frac{m_\kappa + \gamma_9'(z_n)}{\gamma_9''(z_n)} \left(1 - \frac{m_\kappa + \gamma_9'(z_n)}{\gamma_9''(z_n)} \frac{\gamma_9'''(z_n)}{2\gamma_9''(z_n)}\right)^{-1} \quad (4.74)$$

where the derivatives can be found analytically as

$$\gamma_9'(z) = \gamma_3 \gamma_8 \exp(-m_\kappa z_n + 0.5 z_n^2 v_\kappa) (z_n v_\kappa - m_\kappa) \quad (4.75)$$

$$\gamma_9''(z) = \gamma_3 \gamma_8 \exp(-m_\kappa z_n + 0.5 z_n^2 v_\kappa) (v_\kappa + (z_n v_\kappa - m_\kappa)^2) \quad (4.76)$$

$$\gamma_9'''(z) = \gamma_3 \gamma_8 \exp(-m_\kappa z_n + 0.5 z_n^2 v_\kappa) (v_\kappa - 3m_\kappa + 4z_n v_\kappa + (z_n v_\kappa - m_\kappa)^2). \quad (4.77)$$

We choose $p=1$ as opposed to $p=0$ due to numerical stability. Newton's method encounters problems when $\gamma_3 \gamma_8$ is close to 0 because the update step blows up. The higher-order Householder methods prevent this problem. Numerical stability can be ensured by introducing a step size that meets Robbins-Monro condition [110] nevertheless introducing a step size will necessitate properly choosing the step size. We want to avoid choosing a step size for this relatively simple problem. We illustrate approximation methods visually in Figure 4.3. The figure shows that when the assumption $v_\kappa \rightarrow 0$ does not hold, the naive method fails

to describe the mode. In contrast, the Laplace approximation with the Householder method describes the mode fairly well. To quantify the quality of approximations, we compute the KL divergence between the approximating messages and $\tilde{\nu}(z)$. KL divergence between naive Laplace approximated message and $\tilde{\nu}(z)$ is 1217.04, between Householder Laplace approximate message and $\tilde{\nu}(z)$ is 167.78, between Monte-Carlo estimate and $\tilde{\nu}(z)$ is 73.18. Even though the Laplace estimate with the Householder method captures the mode, it fails to capture the variance information accurately and, as a result, has higher KL divergence compared with the Monte-Carlo estimate. The main problem of the Laplace approximation is that it fails to capture the mass of the probability distribution as it can not accurately describe the variance and higher-order moment information. This is because the message $\tilde{\nu}(z)$ does not resemble a Gaussian at all. A better alternative could be formulated by taking into account the incoming message $\vec{\nu}(z)$. Instead of approximating the $\tilde{\nu}(z)$ directly, we can try to approximate marginal distribution $q(z) \propto \tilde{\nu}(z)\vec{\nu}(z)$. In the next section, we will discuss this procedure.

4.6.4 Laplace Approximated Marginals

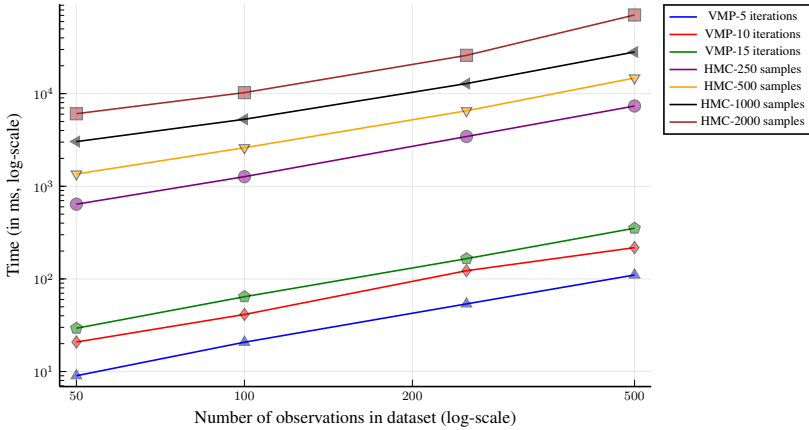
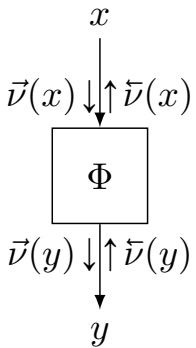


Figure 4.5: Plot of filtering run-time duration vs. the number of observations for a 2-layered HGF model. Both axes are plotted on a logarithmic scale. Lines corresponding to the VMP algorithm have the associated number of iterations in the legend. Lines corresponding to the HMC algorithm have the number of samples in the chains in the legend. The HMC algorithm leapfrog integrator has 10 leapfrog steps and a step size of 0.1.

As we have mentioned at the end of Section 4.6.5, the message $\tilde{\nu}(z)$ does not resemble a Gaussian, and approximating it with a Gaussian by computing the mean and variance via Laplace approximation does not yield reasonable estimates. We can, however, use the fact that the incoming message $\vec{\nu}(z)$ is a Gaussian, and its multiplication with $\tilde{\nu}(z)$ might resem-

Table 4.3: EP updates for the probit node. Calculation of average energy is done through quadrature as described in Section 4.6.5 since there is no analytic solution. W_i denotes the weights and ω_i denotes the sigma points of the Gauss-Hermite quadrature.

Probit Node	Auxiliary	
	ζ	$\frac{\vec{m}_x}{\sqrt{1 + \vec{v}_x}}$
	α	$\frac{\beta \Phi(\zeta)}{\beta \Phi(\zeta) + (1 - \beta)(1 - \Phi(\zeta))}$
	$h(y)$	$-\alpha \log \Phi(y) - (1 - \alpha) \log \Phi(-y)$
	C	$1 - \beta + (2\beta - 1) \Phi(\zeta)$
	μ_1	$\Phi(\zeta) \vec{m}_x + \frac{\vec{v}_x \mathcal{N}(\zeta 0, 1)}{\sqrt{1 + \vec{v}_x}}$
	μ_2	$2\vec{m}_x \mu_1 + (\vec{v}_x - \vec{m}_x^2) \Phi(\zeta) + \frac{\vec{v}_x^2 \zeta \mathcal{N}(\zeta 0, 1)}{1 + \vec{v}_x}$
	m_x	$\frac{(1 - \beta) \vec{m}_x + (2\beta - 1) \mu_1}{C}$
	v_x	$\frac{(1 - \beta) (\vec{m}_x^2 + \vec{v}_x) + (2\beta - 1) \mu_2}{C} - m_x^2$
	\tilde{v}_x	$(v_x^{-1} - \vec{v}_x^{-1})^{-1}$
	\tilde{m}_x	$\vec{v}_x (v_x^{-1} m_x^{-1} - \vec{v}_x^{-1} \vec{m}_x)$
Marginals	Messages	
$q(x) = \mathcal{N}(x m_x, v_x)$	$\tilde{v}(y)$	$\beta^y (1 - \beta)^{1-y}$
$q(y) = \alpha^y (1 - \alpha)^{1-y}$	$\vec{v}(y)$	$\Phi(\zeta)^y (1 - \Phi(\zeta))^{1-y}$
Average Energy	$\vec{v}(x)$	$\mathcal{N}(x \vec{m}_x, \vec{v}_x)$
$\frac{1}{\sqrt{\pi}} \sum_i W_i h(\sqrt{2v_x} \omega_i + m_x)$	$\tilde{v}(x)$	$\mathcal{N}(x \tilde{m}_x, \tilde{v}_x)$

ble a Gaussian better than the message $\vec{v}(z)$ itself. Consequently, this will mean that we will not alter the functional form of the message $\vec{v}(z)$ such that it will be propagated through the graph and instead approximate the marginal of $q(z)$.

Consider the log of the marginal distribution

$$\log q(z) \propto \log \vec{v}(z) + \log \tilde{v}(z) \quad (4.78)$$

$$\propto -0.5 \underbrace{\left(\frac{(z - \vec{m}_z)^2}{\vec{v}_z} + m_\kappa z + \gamma_9(z) \right)}_{l(z)}. \quad (4.79)$$

In order to approximate the marginal $q(z)$ with a Gaussian we again look for the solution of $l'(z) = 0$. We utilize Householder's method with $p = 1$ to find the root by the following update scheme

$$z_{n+1} = z_n - \frac{l'(z_n)}{l''(z_n)} \left(1 - \frac{l'(z_n) l'''(z_n)}{l''(z_n) 2l''(z_n)} \right)^{-1} \quad (4.80)$$

$$= z_n - \frac{2 \frac{z_n - \vec{m}_z}{\vec{v}_z} + m_\kappa + \gamma_9'(z_n)}{\frac{2}{\vec{v}_z} + \gamma_9''(z_n)} \left(1 - \frac{2 \frac{z_n - \vec{m}_z}{\vec{v}_z} + m_\kappa + \gamma_9'(z_n)}{\frac{2}{\vec{v}_z} + \gamma_9''(z_n)} \frac{\gamma_9'''(z_n)}{\frac{4}{\vec{v}_z} + 2\gamma_9''(z_n)} \right)^{-1} \quad (4.81)$$

where derivatives of $\gamma_9(z)$ are given in (4.75). When the Householder iteration terminates at z^* we approximate the marginal distribution by

$$q(z) \stackrel{\text{approx.}}{\propto} \mathcal{N} \left(z \mid z^*, \frac{1}{l''(z^*)} \right). \quad (4.82)$$

Often in practice, the Laplace approximation is known to suffer from an inaccurate estimate of variance information by taking into account the second derivative information [111]. The following section will present another method to approximate the non-conjugate message multiplication that remedies propagating inaccurate variance information.

4.6.5 Moment-matching based Marginal and Message Approximations

We are again interested in approximating the non-conjugate multiplication with a tractable distribution such as a Gaussian. This time we will approach the problem from a different perspective. Consider the non-conjugate multiplication

$$q(z) = \frac{\vec{v}(z) \tilde{v}(z)}{\int \vec{v}(z) \tilde{v}(z) dz}. \quad (4.83)$$

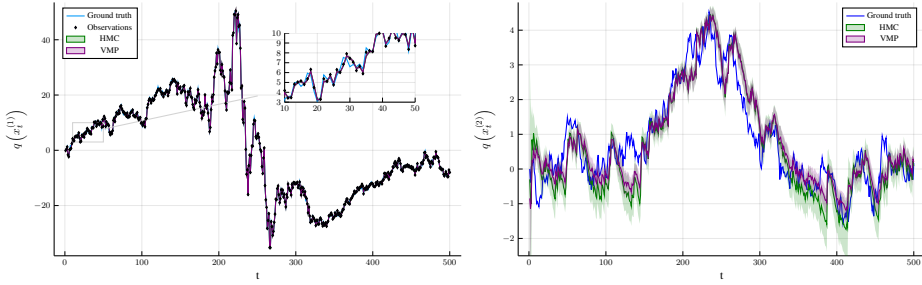


Figure 4.6: (Left) Estimates for the $q(x_t^{(1)})$. Both algorithms return almost identical estimates. Observations are generated with high precision. seen from the inset plot. (Right) Estimates for $q(x_t^{(2)})$ obtained by the VMP algorithm is shown in purple while the estimate obtained by the HMC algorithm is shown in green. Both algorithms return estimates that are close to each other. This closeness results in similar error measures for the algorithms.

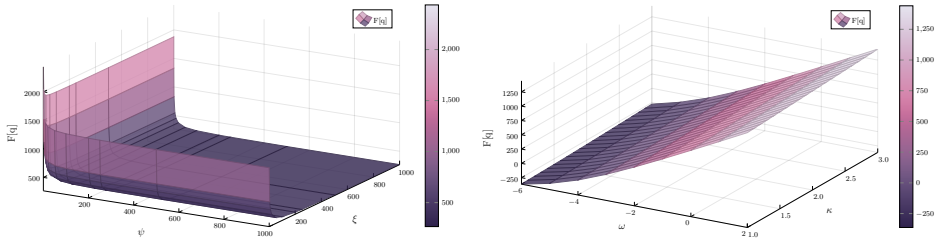


Figure 4.7: (Left) The plot of the free energy versus the precision values for the state transition ξ and likelihood ψ nodes. Free energy values correspond to the converged value of the free energy over iterations. (Right) The plot of the free energy versus κ and ω values for the state transition and likelihood nodes. Free energy values correspond to the converged value of the free energy over iterations.

A fundamental problem with the multiplication of (4.83) is that its normalization constant is not known analytically. After plugging in the functional form of the messages to the integral computation in the denominator of (4.83), we see that the normalization constant can be written as

$$Z \triangleq \int_{-\infty}^{\infty} \bar{v}(z) \bar{v}(z) dz = \int_{-\infty}^{\infty} \exp(-0.5(zm_{\kappa} + \gamma_9(z))) \mathcal{N}(z | \bar{m}_z, \bar{v}_z) dz. \quad (4.84)$$

Integral to compute Z is known as Gauss-Hermite integral, and its solution can be obtained by Gauss-Hermite quadrature [60, Ch. 6]. Gauss-Hermite integration approximates the integral by a summation of the following form

$$Z \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^p W_i \bar{v}(\sqrt{2\bar{v}_z} \zeta_i + \bar{m}_z) \quad (4.85)$$

$$\approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^p W_i \exp\left(-0.5\left(m_{\kappa}\left(\sqrt{2\bar{v}_z} \zeta_i + \bar{m}_z\right) + \gamma_9\left(\sqrt{2\bar{v}_z} \zeta_i + \bar{m}_z\right)\right)\right) \quad (4.86)$$

where p is the order of Hermite polynomials that are used to obtain the sigma point ζ_i and the weights W_i . The i^{th} order physicist's Hermite polynomial is given by

$$H_i(\zeta) = (-1)^i \exp(\zeta^2) \frac{d^i}{d\zeta^i} \exp(-\zeta^2). \quad (4.87)$$

Roots of the Hermite polynomial correspond to the sigma points ζ_i for $i = 1, \dots, p$ and the weights are determined by

$$W_i = \frac{p! 2^{p-1} \sqrt{\pi}}{p^2 [H_{p-1}(\zeta_i)]^2}. \quad (4.88)$$

In practice we use Golub-Welsch algorithm to find the roots and weights of the Hermite polynomials [112].

Once the normalization constant Z is determined by (4.86) we can further apply Gauss-Hermite quadrature to approximate the moments of the non-conjugate marginal by the following rule

$$\mathbb{E}[z^n] \approx \frac{1}{Z \sqrt{\pi}} \sum_{i=1}^p W_i \bar{v}(\sqrt{2\bar{v}_z} \zeta_i + \bar{m}_z) \left(\sqrt{2\bar{v}_z} \zeta_i + \bar{m}_z\right)^n. \quad (4.89)$$

Obtaining the first two moments will then allow us to approximate the non-conjugate marginal with a Gaussian whose moments are matched with the moments of the non-conjugate marginal.

We set the mean and the variance as

$$m_z = \mathbb{E}[z] \quad (4.90)$$

$$\approx \frac{1}{Z\sqrt{\pi}} \sum_{i=1}^p W_i \tilde{\nu} \left(\sqrt{2\tilde{\nu}_z} \zeta_i + \tilde{m}_z \right) \left(\sqrt{2\tilde{\nu}_z} \zeta_i + \tilde{m}_z \right) \quad (4.91)$$

$$v_z = \mathbb{E} \left[(z - m_z)^2 \right] \quad (4.92)$$

$$\approx \frac{1}{Z\sqrt{\pi}} \sum_{i=1}^p W_i \tilde{\nu} \left(\sqrt{2\tilde{\nu}_z} \zeta_i + \tilde{m}_z \right) \left(\sqrt{2\tilde{\nu}_z} \zeta_i + \tilde{m}_z - m_z \right)^2. \quad (4.93)$$

Then the moment matched approximation to the non-conjugate marginal is obtained as

$$q(z) \stackrel{approx.}{\propto} \mathcal{N}(z|m_z, v_z). \quad (4.94)$$

We illustrate marginal approximations of Sections 4.6.4 and 4.6.5 in Figure 4.4. The figure shows that the Laplace approximation puts probability mass on a region of the Monte-Carlo marginal that has little mass. In contrast, moment-matched marginal does not have a drastic mismatch. We compute the KL divergence between approximations and the Monte-Carlo marginal to quantify approximation quality. KL divergence between the Laplace marginal and the Monte-Carlo marginal is 228.17, and KL divergence between the quadrature marginal and Monte-Carlo marginal is 23.20.

A different version of the approach we discussed in this section has already been used in the quadrature EP algorithm of [75] where an expectation of natural statistics of an exponential family is determined by quadrature. Then the parameters are found by inverting the link function if an analytic relation is known for the inverse link function. Once the marginal is obtained, the message $\tilde{\nu}(z)$ can be obtained by a division procedure as described in [75] corresponding to the EP algorithm [51]. As opposed to EP, we are propagating the non-conjugate message $\tilde{\nu}(z)$ as it is without any approximation, and we approximate the marginal $q(z)$. Therefore, a division to obtain the message is unnecessary. We have already obtained a marginal $q(z)$ that will evaluate expectations to obtain messages for other edges in the graph.

This section concluded the message and marginal computations for the GCV node and proposed various approximation strategies. In the next section, we will review the EP algorithm for discrete likelihoods presented by [82] such that the HGF can model discrete observations.

4.6.6 Expectation Propagation for Discrete Likelihoods

In this section we will describe how to connect HGF models to binary observations. In Section 4.2.2 we have specified a binary likelihood in (4.4) where a continuous valued latent signal is mapped into binary observations by a Bernoulli likelihood whose parameter is governed by a probit link function [81, Chp. 3]. Again removing the time indices from (4.4) we

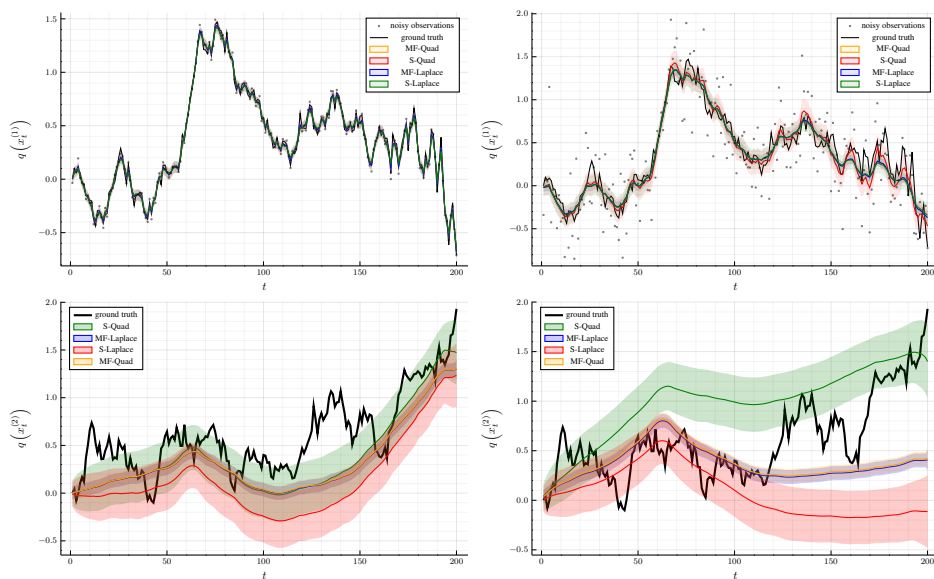


Figure 4.8: The figure displays the estimated trajectories of the hierarchical states of a 2-layered HGF model under two different observation noise regimes obtained by four different algorithms. These algorithms are SVMP and naive mean-field VMP with quadrature and Laplace approximations. The left column corresponds to estimates obtained in a high-precision regime whereas the right column corresponds to estimates obtained in a low-precision regime.

obtain the following node function

$$p(y|x) = \Phi(x)^y (1 - \Phi(x))^{1-y} \tag{4.95}$$

which is called a probit node in [82]. In Table 4.3 we present the probit factor node and the messages associated with it. We want to compute these messages adhering to Theorem 3.5. For the derivation of the messages we refer to [82].

EP for the probit node solves the problem that $\vec{\nu}(x)$ can not be computed as a (normalized) message causing issues in the inference if it is to be passed as a message along the graph [82]. EP first approximates the exact marginal, corresponding to the multiplication $\vec{\nu}(x)\vec{\nu}(x)$, with a Gaussian distribution by moment matching as described in Section 4.6.5. Once a Gaussian approximation to the marginal is obtained, EP algorithm [51] computes the message $\vec{\nu}(x)$ by dividing the marginal with $\vec{\nu}(x)$ resulting in the updates of Table 4.3 as derived by [82].

4

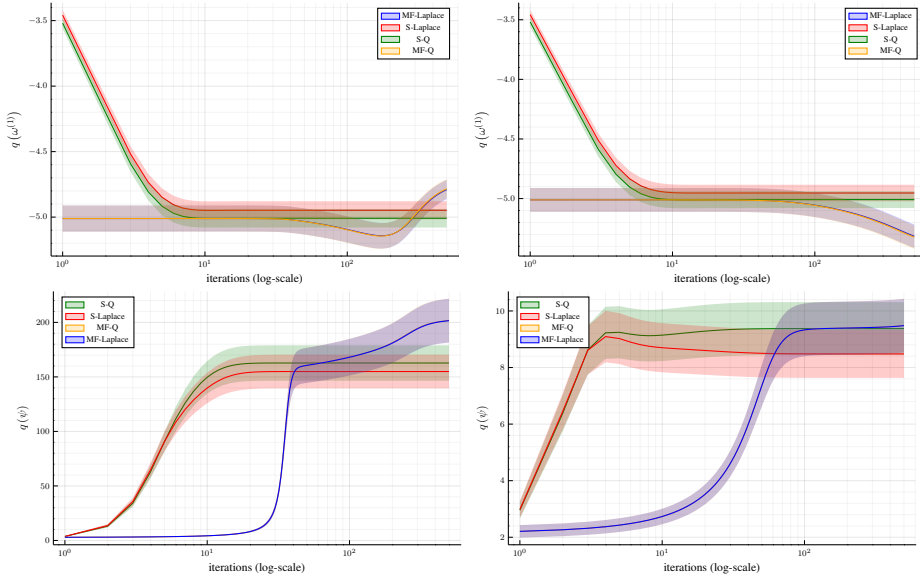


Figure 4.9: The figure displays some trajectories of the estimated parameters $\omega^{(1)}$ and ψ over iterations. The left column estimates are recovered in a high precision observation case. The right column estimates are recovered in a low precision observation case

This section concludes message computations. We are equipped with the required machinery to perform inference in the HGF model with continuous and binary observations. We will experimentally verify that the message update equations can compute posterior distributions given in the following sections. We will compare these solutions with sampling algorithms to establish their quality. Moreover, we will validate the HGF update equations on real datasets.

	Number of observations					
	1 th layer error			2 nd layer error		
Algorithm	50	100	250	50	100	250
VMP	0.89	0.79	0.75	0.36	0.35	0.35
HMC (500)	1.41	1.29	1.51	0.45	0.62	4.74
HMC (1000)	1.30	1.14	1.22	0.41	0.49	2.56

Table 4.4: Accuracy comparison of filtering results in terms of (4.96) metric for 2-layered HGF model across HMC and VMP on the synthetic data-set. Lower is better. Number of VMP iterations is equal to 15 around converging point. HMC method is run with 500 and 1000 number of samples per chain.

4.7 Simulations

This section will present message passing simulations for the HGF model that use message update rules given in Section 4.6. We will carry out simulations in the filtering and smoothing schemes. Before we move on to verification simulations and compare the performance of the methods described in Section 4.6, we will compare the VMP algorithm with the sampling-based Hamiltonian Monte Carlo (HMC) method. We will compare the inference accuracy and the performance (in terms of run-time) of VMP and HMC given a simple 2-layered HGF model. Specifically, we fix $\kappa^{(1)} = 1$, $\omega^{(1)} = 0$, $\psi = 5$, $\xi = 20$, $x_0^{(1)} = 0$ and $x_0^{(2)} = 0$ in the model definition (4.1) and generate observations by varying the number of observations $T = 50, 100, 250, 500$. We denote this data-set with \mathcal{D} . Then on this data-set \mathcal{D} we ran HMC [23] and VMP algorithms to perform filtering to obtain $q(x_t^{(1)})$ and $q(x_t^{(2)})$. We use the following number of iterations for the VMP algorithm: 5, 10, 20, 25. For the HMC algorithm, we vary the number of samples taken as follows 250, 500, 1000, 2000. Figure 4.5 plots the run-time for the varying number of observations. The VMP algorithm reduces the time required to perform filtering compared to HMC by orders of magnitude.

Since the data-set is generated synthetically we have access to the generated signals and we can compare the accuracy based on a metric. For the VMP algorithm the objective metric is Bethe Free Energy as it is derived by minimization of the Bethe Free Energy. On the other hand HMC optimizes Hamiltonian energy. These two objectives are not necessarily the same and comparing them is not appropriate. Therefore we resort to average mean-squared error for the posteriors obtained by HMC and VMP algorithms by computing

$$\text{err}(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \left(\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q(x_t^{(i)})} \left[\left(x_t^{(i)} - \hat{x}_t^{(i)} \right)^2 \right] \right), \quad (4.96)$$

where \mathcal{D} is a data-set containing $|\mathcal{D}|$ number of realizations of ground truth latent states $\hat{\mathbf{x}}^{(i)}$ for an N -layered HGF model. In Table 4.4 we present the results of evaluating the Bayesian error metric on the estimates returned by HMC and VMP algorithms for the generated data-set \mathcal{D} . Table 4.4 shows that VMP algorithm consistently results in lower error compared to HMC.

An empirical comparison of VMP and HMC on a simplistic version of the HGF model

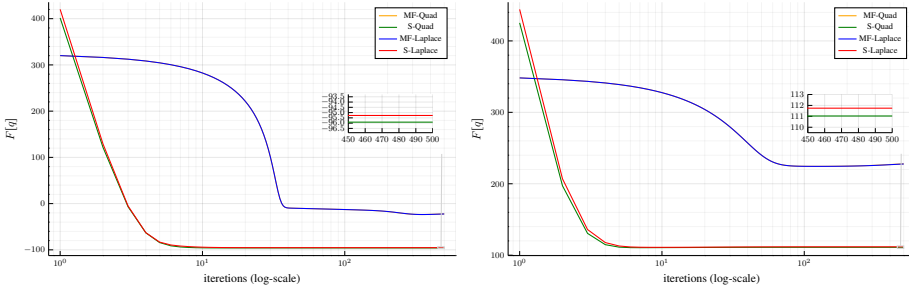


Figure 4.10: The figure plots the evolution of Bethe free energy over iteration in high observation precision (left) and low observation precision regimes (right).

shows that the VMP algorithm, which uses the message update equations derived in Section 4.6, outperforms HMC both in terms of accuracy and run-time. Inference time constraints will not scale well if the HMC algorithm is used on multi-layered HGFs with a significant number of observations. For time series analysis, it is essential to have many observations to infer the parameters and latent states simultaneously. This empirical study demonstrates that the VMP algorithm scales better with the number of observations, resulting in higher estimation accuracy. From now on, we will analyze the inference solutions returned by the message-passing algorithms and do not compare them to the sampling-based solutions further.

4.7.1 Verification on Synthetic Data

This section will show simulations carried out on synthetic data using the HGF model. We carry out simulations on synthetic data to have an empiric understanding of how parameters affect the free energy minimization procedure and to understand the performance of the message passing algorithms derived from various constraints.

4.7.1.1 Experimental Setup

The data-set \mathcal{D}_i for an i^{th} -layered HGF is generated by sweeping through the parameter values of the HGF models (4.1) as given by Table 4.5. Given a dataset D_i , we first construct an FFG and determine a message-passing schedule as shown in Figure 4.2 for a 2-layered HGF. Then we iterate the messages until free-energy converges.

To understand the relationship between free energy minimization and the parameter val-

κ	0.01 0.1 1 2 4
ω	-4 -3 -2 -1 0 1 2 3 4
ψ	0.1 1 5 10 20 50 100 150
ξ	1 5 10 20 50 100 150
T	500

Table 4.5: Parameter values to be swept over to generate data-set \mathcal{D}_i .

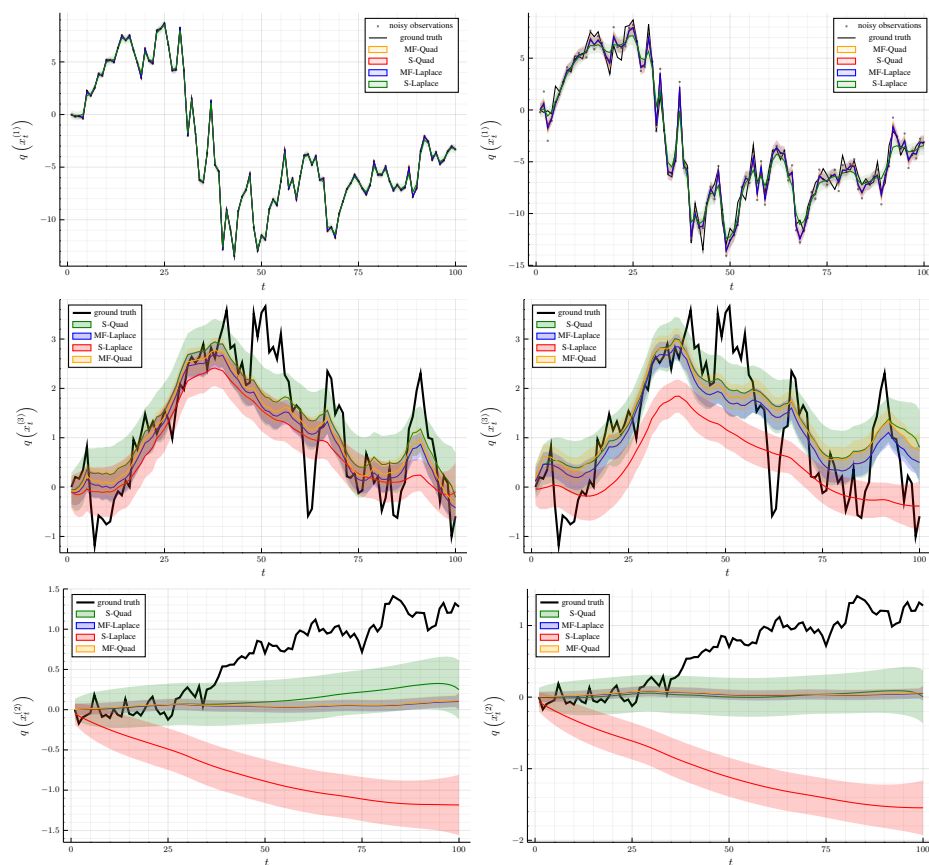


Figure 4.11: The figure displays the estimated trajectories of the hierarchical states of a 3-layered HGF model under two different observation noise regimes obtained by four different algorithms. These algorithms are SVMP and naive mean-field VMP with quadrature and Laplace approximations. The left column corresponds to estimates obtained in a high-precision regime whereas the right column corresponds to estimates obtained in a low-precision regime.

ues, we will analyze the convergent values of the free energy metric with different parameter values. To simplify things, we will consider a 2-layered HGF. For a 2-layered HGF, there are four parameters in total $\kappa^{(1)}, \omega^{(1)}$ and ψ, ξ where the former are specific to the HGF. We assume that we know the parameters for this experiment and only try to infer the hidden states. We will perform inference using SVMP with moment-matching constraint and obtain a free energy minimization curve for each data in D_i . The following section will present two surface plots indicating the relationship between the free energy minimization and parameter values. This setup will be referred to as setup 1.

In the following experimental setup, we create 2 and 3 layered HGF datasets and perform inference in these datasets using the approximation methods of Section 4.6. We do not assume that the parameters are fixed for this setup but rather to be learned along with the latent states. We use priors defined in Section 4.2.3 and put a point-mass constraint on the hyper-parameters to optimize over the values of the hyper-parameters, leading to an EM procedure for the estimation of hyper-parameters. We compare four hybrid VMP algorithms for the HGF model and perform smoothing and learning on datasets D_i . We record the free energies corresponding to each algorithm and compare the free energy values to compare the performance of each algorithm. This setup will be referred to as setup 2.

In experimental setup 3, we generate synthetic binary observations with a two-layered-HGF using probit link-function as in (4.4) and show that the HGF model can handle discrete data by including local EP updates of Table 4.3.

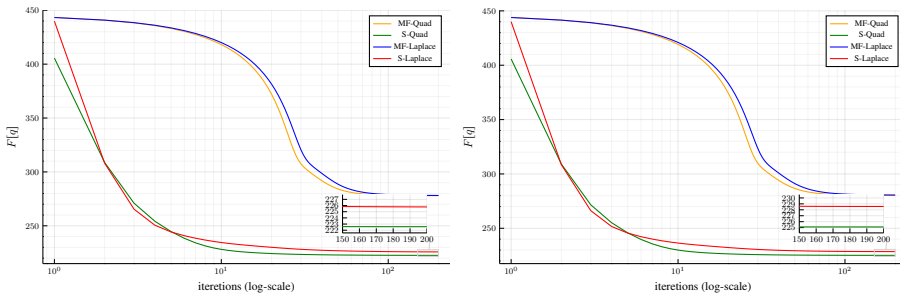


Figure 4.12: The figure plots the evolution of Bethe free energy over iteration in high observation precision (left) and low observation precision regimes (right) for a 3 layered HGF model.

4.7.1.2 Experimental Results

The results of experimental setup 1 are summarized in Figure 4.7. The plots display free energies over a ranging set of parameters. Free energy values correspond to minima obtained over the iterations during message passing. The left plot in Figure 4.7 reveals that free energy decreases monotonically with increasing precision. The right plot in Figure 4.7 reveals that free energy increases monotonically as ω and κ increase. Although the relationship between

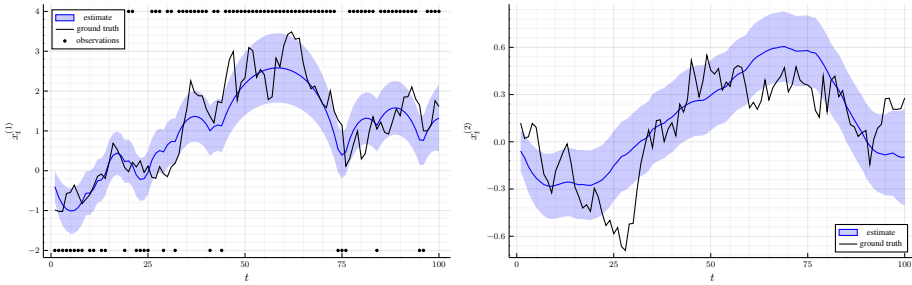


Figure 4.13: The figure plots the trajectories of the estimated states of a 2 layered HGF model with discrete observations. (Left) Binary observations are modelled by the probit node and estimates for $x_t^{(1)}$ are obtained by EP. (Right) Estimates for $x_t^{(2)}$ are obtained by SVMP updates.

κ and the free energy is monotonic, the effect of changing κ (for fixed ω) for a two-layered HGF does cause a relatively small change in the free energy compared to changing ω (for fixed κ).

The results of experimental setup 2 are visually summarized in Figures 4.8, 4.9, 4.10, 4.11 and 4.12. We refer to estimates of variables returned by four variants of VMP algorithms in these figures. We denote the mean-field VMP algorithm with quadrature approximated marginals as MF-Quad, the structured VMP algorithm with quadrature approximated marginals as S-Quad, and the mean-field VMP algorithm with Laplace approximated marginals as MF-Laplace. The structured VMP algorithm with Laplace approximated marginals as S-Laplace. Performance comparison of these algorithms immediately follows from comparing the free energy minimization plots. The algorithm that minimizes Bethe's free energy lower gives better performance. Figures 4.10 and 4.12 indicates that the structured VMP algorithms consistently outperform naive mean-field VMP algorithms. Better performance of structured factorization can be explained because naive mean-field factorization ignores the inherent correlations present in the time series. Ignoring the correlations causes underestimation of variances of hierarchical states (see the Figures 4.8, 4.9), which is penalized by the Bethe free energy for mean-field approximation. Quadrature and Laplace approximation yield reasonably close estimates for both factorizations. Nevertheless, as the number of hierarchical layers is increased to three, the difference between quadrature and Laplace approximation opens up in favor of quadrature. Moreover, the Laplace approximation requires a numerical update on the covariance matrix, which is constrained to be positive definite. Due to the numerical nature of the optimization, positivity constraint adds another level of complexity to message passing with Laplace approximation. We can conclude that the S-Quad algorithm outperforms all the remaining based on free energy comparison.

In Figures 4.8 and 4.11, we illustrate the estimated states of 2 and 3 layered HGF models under two different noise regimes. When the precision of the likelihoods is high, the first layer estimates are almost visually indistinguishable. In the lower precision cases, all the algorithms except S-Quad underestimate the intrinsic volatility and attribute more weight to the observation noise, resulting in overly smoothed estimates. Figures 4.8 and 4.11 indicate

that as the likelihood precision increase, hierarchical states can be identified finer on the higher layers. Figure 4.9 displays the evolution of some of the parameters over the iterations for a 2-layered HGF. Plots in Figure 4.9 indicate that the parameter estimates converge for all the algorithms.

The results of experimental setup three are summarized in Figure 4.13. We performed a hybrid message passing algorithm where EP, SVMP, and BP messages are utilized in this setup. We use the quadrature approximation as our method of choice for approximating non-conjugate marginal computations. The recovery of states at the first layer is cruder compared to estimates with a Gaussian likelihood as in Figure 4.8 and 4.11. The crudeness of recovery can be attributed to the nature of messages obtained with EP for the link between continuous-valued variables and discrete-valued observations. EP messages around the prohibit node display a distribution covering property such that the marginals computed by EP messages tend to cover as much of the exact posterior as possible [35, 51]. This behavior can be seen in the left plot of Figure 4.13 for $t \in [50, 75]$, where the variance of the state estimates increases as the approximate marginal tends to cover a larger area. This setup illustrates that discrete likelihoods can augment HGF and that the modularity of update equations allows for a hybrid algorithm.

4.7.2 Validation on Stock Prices

We validate the message update equations for the HGF by measuring the predictive performance of Bitcoin prices between 25/10/2010 and 29/11/2011. We measure the predictive performance by the free energy. The top row in Fig. 4.14 shows the recorded prices. We employed a 3-layer HGF model to predict this sequence and discuss the estimation and evaluation results. We look at one-day prediction and assume that the data arrives sequentially such that future prices are not known to the algorithm.

4.7.2.1 Choices of Priors

For the parameters $\omega^{(i)}$ and $\kappa^{(i)}$ we choose uninformative priors $\omega^{(i)} \sim \mathcal{N}(0.0, 10.0)$ and for the “scale” parameters $\kappa^{(i)}$ we choose $\kappa^{(i)} \sim \mathcal{N}(1, 0.01)$, where $i = 1, 2$. In [36], the $\kappa^{(i)}$ values are fixed to 1 and the justification behind this choice is to ensure parameter identifiability. However, we suspect that allowing κ to vary relatively slowly relative to states $x_t^{(i)}$ might have benefits in terms of adapting to changing market dynamics. Hence we set the mean of the κ priors to 1 and add a small variance. Finally, we choose $\xi \sim \Gamma(0.001, 0.001)$ as the state transition precision in the third layer and $\psi \sim \Gamma(0.0001, 0.0001)$ for the precision of the observation model.

4.7.2.2 Experimental Results

We will examine the state estimates first; see the second and the third subplots in Fig. 4.14. At the second layer, both algorithms return estimates that share similar patterns. The tracks

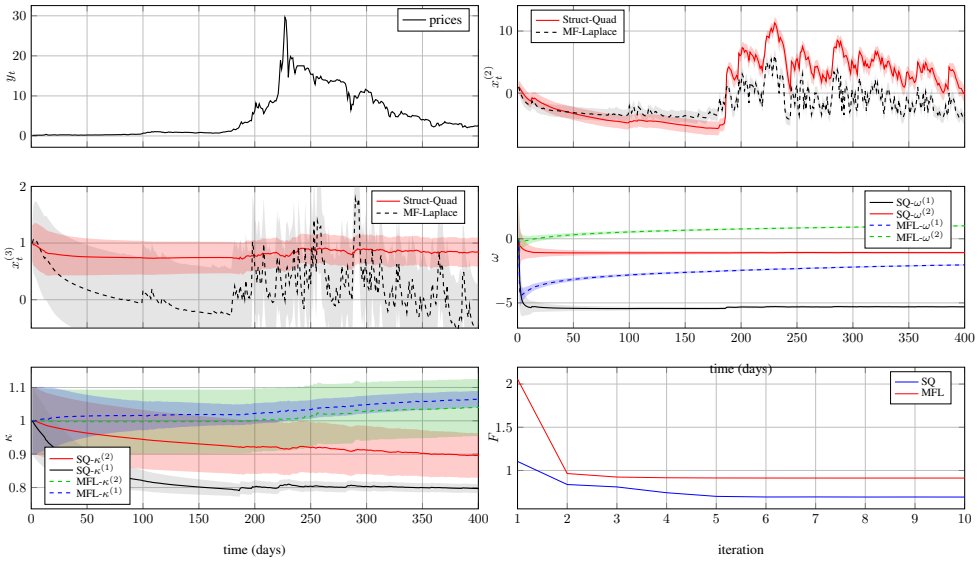


Figure 4.14: The top left shows scaled Bitcoin prices, i.e., the observations y_t . The top right and middle left subplots show state estimates $x_t^{(2)}$ (the “volatility”) and $x_t^{(3)}$ respectively. The solid and dashed lines represent the mean estimates for SQ and MFL, respectively, and shaded regions represent one standard deviation (mean \pm standard deviation). The middle right and bottom left subplots depict estimates for the ω and κ parameters over time, respectively. The plot in the bottom right shows time-averaged free-energy (in *nats*) over iterations. Both algorithms converge, but the proposed S-Quad algorithm converges faster.

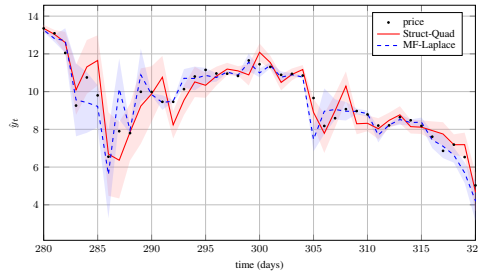


Figure 4.15: Predictions for Bitcoin prices. For clarity, we only plot 40 days. The dotted line shows the actual prices, and the solid and dashed lines represent the mean of Struct-Quad and MF-Laplace predictions. The width of the shaded areas indicates two standard deviations.

of $x_t^{(2)}$ capture the trends of increase and decrease in volatility. Nevertheless, the Struct-Quad algorithm tracks more smoothly compared to MF-Laplace. While there are still some salient events, the third-layer state of Struct-Quad evolves smoothly. On the other hand, the third-layer state for MF-Laplace is quite active. The smoothness of the Struct-Quad estimates is due to a structured assumption that keeps track of temporal correlations.

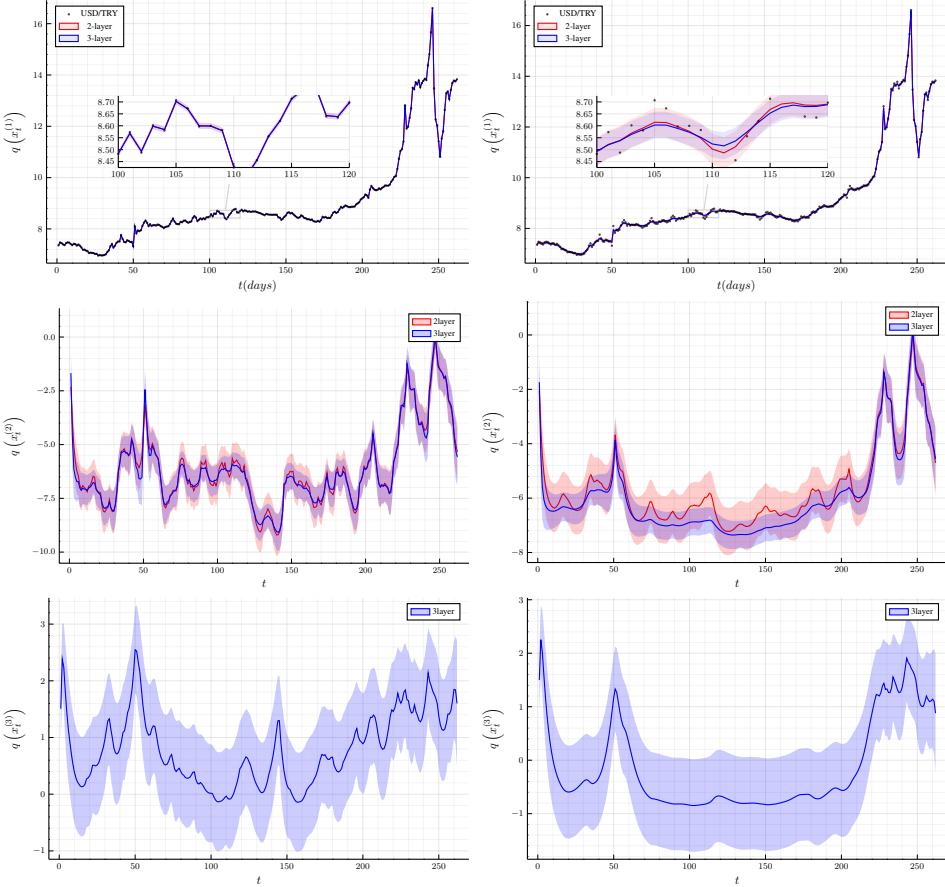


Figure 4.16: The figure plots smoothing results on the currency exchange data. Left column represents estimates obtained with the prior $\psi \sim \Gamma(0.001, 0.001)$ and the right column represents estimates obtained with the prior $\psi \sim \Gamma(0.001, 0.001)$. Top row plots show the currency values and the estimates of the states on the first layer. Second and third row corresponds to estimates of states (log-volatility) and (log-log-volatility) at the second and third layer of the HGF respectively.

The fourth and fifth subplots show the estimation tracks for tonic and phasic parameters

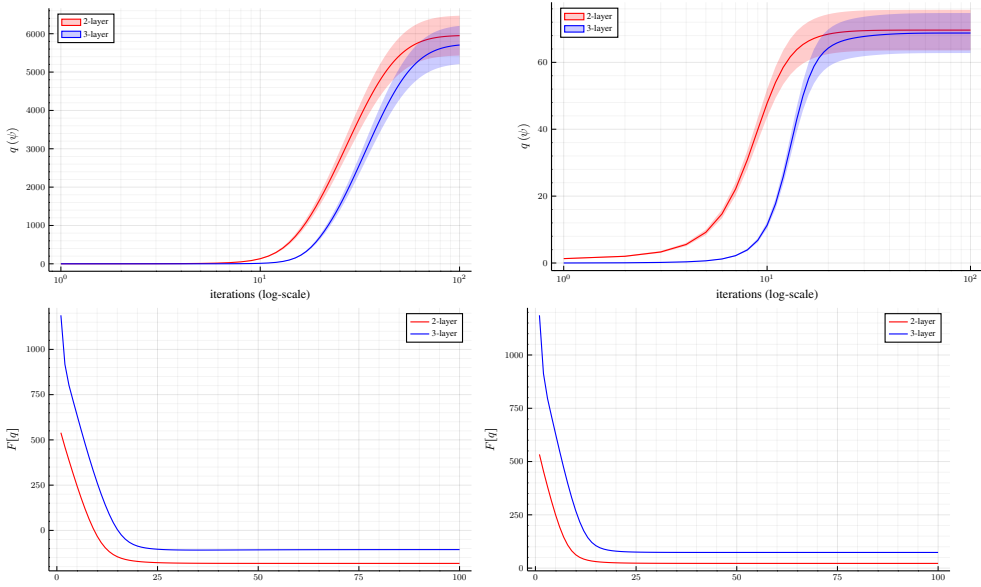


Figure 4.17: The first row are the estimates of the precision ψ of the Gaussian likelihood over the iterations. The second row plots the free-energy minimization curves over the iterations.

$\omega^{(i)}$ and $\kappa^{(i)}$ respectively. Estimates for $\omega^{(i)}$ vary at a slower time scale than the states, but they exhibit specific variation and adaptation. The κ tracks in the last subplot show a decreasing trend for Struct-Quad, leading to a reduced impact of superior layer states on the inferior layer parameters.

We measure the performance of the two algorithms by the free energy function, which can be interpreted as accuracy plus model complexity cost function. The free energy plot in the bottom right subplot of Fig. 4.14 shows the free energy averaged over iterations per time step. Smaller free energy values represent a better fit. While both algorithms converge, the proposed S-Quad algorithm converges to a lower free energy value than the MF-Laplace algorithm. We plot the prediction results in Fig 4.15.

4.7.3 Validation on Currency Exchange

We validate the update equations for the HGF model on modeling the volatility of currency exchange between United States Dollars (USD) and Turkish Lira (TRY) between 11/01/2021 and 11/01/2022. Here we are not interested in predicting, but instead, we would like to analyze the time-varying variance of the currency exchange USD/TRY in retrospect. That means we have access to accurate historical data and can perform smoothing on the graph of the HGF model. For this purpose, we utilize 2-layered and 3-layered HGF models to estimate the volatility.

4.7.3.1 Choices of Priors

For smoothing task the prior choices can be made more informed as opposed to the filtering task where we have no data available to start with. This means we can use the data to extract empirical values for the hyper-parameters of the some of the priors. A parameter where we can use an informed prior is the tonic component of the HGF $\omega^{(i)}$. This tonic component represents the logarithm of a constant variance term which we can use sample variance to start with. For example, in a 2 layered HGF we can compute sample variance and choose the prior as

$$p\left(\omega^{(1)}\right) = \mathcal{N}\left(\omega^{(1)}|\hat{\omega}^{(1)}, 1.0\right) \quad (4.97)$$

$$\hat{\omega}^{(1)} = \log\left(\frac{1}{T^2} \sum_{\substack{t,t' \\ t < t'}} (\hat{y}_t - \hat{y}_{t'})^2\right), \quad (4.98)$$

where \hat{y}_t are the observations. The choice of mean for the higher layer tonic parameters require certain assumptions. We initially assume that $\hat{\omega}^{(i+1)} \leq \hat{\omega}^{(i)}$ to reflect the prior assumption that the variance of the volatility of the currency exchange do not fluctuate as much as the volatility itself. Thus we choose to set the mean of the priors by the following iterative scheme

$$\hat{\omega}^{(i+1)} = \hat{\omega}^{(i)} - c, \quad (4.99)$$

where c is an arbitrary positive constant. We choose to set $\hat{\kappa}^{(i)} = 1$ as to ensure parameter identification.

We will use two different hyper-parameter settings for the precision of the likelihood. The shape and rate parameters of the Gamma likelihood determine the degree of smoothing for the noisy observations at the first layer. We will use a non-informative prior $\psi \sim \Gamma(0.001, 0.001)$ and an informed prior $\psi \sim \Gamma(1, 1)$.

4.7.3.2 Experimental Results

Figure 4.16 displays the estimation results for the modeling of the currency exchange task. When the prior for the precision is uninformative $\Gamma(0.001, 0.001)$, both the 2-layer and 3-layer HGF models attribute a high precision to the observations by returning a large mean estimate for ψ . On the other hand, when the prior is informed $\Gamma(1, 1)$, the posterior estimate for ψ does not move away from the starting point much. As a result of this, the estimates of the first layer change. With informed prior, estimates are smoother compared to the uninformed prior. Since an increase in the observation precision is going to decrease the free energy (see Figure 4.7), the model with uninformed prior results in lower free energy, as seen in the bottom row plots of Figure 4.17. Estimation of the precision of the likelihood influences the state and parameter estimates at the higher layers of the HGF model. With increasing precision, states at the second and third layers become more active, and they start to capture

more delicate details of the volatility of the exchange values. Therefore, state estimates of high precision regimes fluctuate more than the low precision regime estimates.

Based on the free energy metric, the 2-layer HGF model with SVMP inference methodology outperforms the 3-layer HGF model with SVMP inference methodology. We are not solely comparing the models in this comparison of the free energy. The comparison includes the approximate posteriors as well. This is because the free energy functional is a function of both the model and the approximate posterior. Since both the model and the associated recognition factor change, free energy is not appropriate for model comparison here. That being said, the presence of an extra layer has influenced the smoothness of the estimates. In both precision regimes, the first layer and second layer state estimates of a 3-layer HGF model are smoother than the estimates of a 2-layer HGF model. Sudden changes in the currency manifest in the form of peaks for the higher layer states. 2 peaks are very noticeable during the days 220-250 for the second layer estimates. During these days, the TRY devalued steadily, and then the currency made a peak on 17th December 2021. After this day, the Central Bank of Turkey intervenes, and the USD/TRY is forced to drop. Interestingly, the steady increase in the USD/TRY until 17th December 2021 is reflected as a decrease in volatility. After the intervention of the Central Bank, the volatility shoots up again, reflecting the forced reduction of the USD/TRY. In a 3-layered HGF model, this phenomenon is seen in the third layer during the days 200-250. The trajectory of the third layer states increases and starts fluctuating when the second layer state trajectory makes valleys and peaks.

4.8 Discussion and Conclusions

In this chapter, we have cast the Hierarchical Gaussian Filter in a factor graph framework and derived local (variational) message passing update rules for the nonlinear connection factor between the layers in this model. Moreover, we derived formulae for the local free energy contributions of the connection factor. As a result, both online state and parameter estimation and performance tracking of the HGF or any variants thereof can now be automatically simulated in a software toolbox that stores the results lookup tables. The central contribution of this work is to reformulate a complex hierarchical dynamic system in a factor graph framework, thus yielding a divide-and-conquer approach to inference that isolates the nonlinear part of the model from (more straightforward) linear functions. We needed to resort to approximation methods only for the non-linear factor through this isolation. The FFG framework plays a central role in this isolation through the concept of a *composite node* that allows us to group a set of connected factors into one substituting factor.

We made a few simplifying assumptions in the derivation of the variational update rules of Table 4.1. For instance, we assumed no correlation between state x and its corresponding κ parameter to estimate their product κx as a Gaussian distributed random variable. Note, however, that this assumption is not necessary, and [107] gives a closed-form approximation without making the above assumption. The only remaining challenge in the case of correlation between the state and its corresponding κ is to estimate the joint distribution between κ

and z . This factorization is indeed implemented in the `ReactiveMP.jl` implementation of the GCV node. Secondly, while setting up the Laplace approximation, we assumed that the variance of the recognition distribution of the κ parameter is negligibly small. This assumption is valid if we fix κ , which we do in our experiments. However, if one also wishes to estimate κ , further care about the assumption needs to be considered. We then improved upon this simplification by improving the Laplace approximation with numerical optimization to find the desired marginals. Finally, we proposed moment-matching with quadrature to approximate non-conjugate marginal computations and showed that this approximation outperforms the other approximations.

We have tested the proposed update rules on synthetic datasets and compared different alternatives to update schemes. We showed on a real-world Bitcoin time series that online variational tracking with structured factorization of states and slowly-varying parameters in a 3-layer HGF with quadrature resulted in convergence to a lower final free energy value than the Laplace approximation. Furthermore, we showed by an example of modeling an exchange rate data set that the HGF model with the proposed update equations can model time series that are volatile and non-stationary.

CHAPTER 5

The Switching Hierarchical Gaussian Filter

"The most important questions of life... are indeed for the most part only problems of probability."

–Pierre Simon Laplace

5.1 Introduction

Hierarchical Dynamic Models (HDM) have often been used to explain the variation of parameters and states of natural processes [20, 113–116]. The Hierarchical Gaussian filter (HGF) is a specific type of HDM that is popular in the neuroscience community, which is partly due to the availability of an open-source modeling toolbox [36, 103, 117]. As discussed in Chapter 4, the HGF is a multi-layer nonlinear state-space model where the states at a higher layer control the variance of state transitions at a particular layer. In the literature, parameters and hidden states of the HGF can be recovered by closed-form variational message passing updates. These properties make the HGF an interesting model for modeling natural signals.

However, in many practical applications, the observed signal can be subject to Markovian regime-switching behavior [14, 118]. The "classical" HGF model will not accurately describe a time series when parameter regime switches rule the underlying dynamics.

Another example is coming from computational neuroscience. The neurons of the prefrontal cortex change their firing activity when animals are switching between different behavioral strategies [119].

While it is not difficult to describe the forward mechanics of regime-switching behavior in a generative model, inference for states and parameters in these models is problematic. In

[120], a switching state-space model (SSSM) that employs a variational inference technique for tracking the posterior of the hidden states was introduced. This work took a pivotal position in the literature and was followed by further diverse developments on state inference for SSSMs [121–124]. Examples include efficient Gaussian Sum Filtering to track a Gaussian Mixture state posterior [125, Ch. 25] and Rao-Blackwellised particle filters for state tracking by analytical marginalization of continuous variables conditioned on sampled discrete latent variables [126].

In this chapter, we develop a state and parameter inference framework for a *Switching Hierarchical Gaussian Filter* (SHGF) that extends the original HGF by supporting a selector mechanism for the model’s parameters. The SHGF is a complex generative model that features hierarchical regime-switching dynamics and non-linear couplings between the layers. Since our target applications require real-time inference on wearable devices, we are interested in developing closed-form inference updates for states and (both slowly time-varying and regime-switching) parameters and tracking a Bayesian evidence performance measure. Inference by Monte Carlo sampling is computationally too expensive for these applications. We build on Chapter 4 for the HGF by representing the model as a factor graph and executing message passing-based inference via divergence minimization. The contributions of this chapter include:

- In Section 5.2, we present a new switching hierarchical dynamical model, the SHGF. We map the SHGF onto a Forney-style Factor Graph (FFG), which supports a fully modular message passing-based approach to inference.
- In Section 5.3, we identify and isolate a "Gaussian with controlled switching variance (GCSV) node" as the module that causes inference issues.
- In Section 5.4, we derive variational update rules for the GCSV node and combine these rules with moment-matching to show that quadrature-based moment-matching [75] can handle the non-conjugate operations yielding a hybrid algorithm [46].
- We experimentally verify the proposed inference procedure on synthetic data for a 2-layer SHGF in Section 5.5. We also provide a real-world example on a stock market data set to compare the SHGF to a (non-switching) HGF model. Additionally, we show how the free energy (FE) score can determine the number of regimes in the data.

5.2 Model Definition

Let $y_t \in \mathbb{R}$ represent observations. We denote latent continuously valued states at layer i by $x_t^{(i)} \in \mathbb{R}$ and categorical states by $s_t^{(i)} \in \{1, \dots, M_i\}$. State transitions of categorical variables are governed by transition matrices $\mathbf{A}^{(i)} \in \mathbb{R}^{M_i \times M_i}$ and continuous state transitions are parameterized by $\kappa^{(i)} \in \mathbb{R}^{M_i}$ and $\omega^{(i)} \in \mathbb{R}^{M_i}$.

One layer of an N -layer switching hierarchical Gaussian filter is defined by the state transitions

$$p\left(x_t^{(i)} | x_{t-1}^{(i)}, s_t^{(i)}, g_t^{(i)}\right) = \prod_{m=1}^{M_i} \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, g_t^{(i)}\right)^{[s_t^{(i)}=m]} \quad (5.1a)$$

$$p\left(s_t^{(i)} | s_{t-1}^{(i)}, \mathbf{A}^{(i)}\right) = \prod_{k=1}^{M_i} \prod_{m=1}^{M_i} \left(\alpha_{km}^{(i)}\right)^{[s_t^{(i)}=k][s_{t-1}^{(i)}=m]} \quad (5.1b)$$

where we used the following definition and constraint, respectively, for every $i = 1, \dots, N - 1$:

$$g_t^{(i)}\left(x_t^{(i+1)}, \kappa_m^{(i)}, \omega_m^{(i)}\right) \triangleq \exp\left(\kappa_m^{(i)} x_t^{(i+1)} + \omega_m^{(i)}\right) \quad (5.2)$$

$$\sum_{k=1}^{M_i} \alpha_{km}^{(i)} = 1. \quad (5.3)$$

We use Iverson bracket notation in (5.1), which is defined as

$$[s_t = m] = \begin{cases} 1 & \text{if } s_t = m \text{ is true} \\ 0 & \text{otherwise} \end{cases}.$$

A non-switching HGF layer is recovered for $M_i = 1$. At the top layer ($i = N$), we assume non-switching random walk dynamics with transition variance ξ^{-1} :

$$p\left(x_t^{(N)} | x_{t-1}^{(N)}\right) = \mathcal{N}\left(x_t^{(N)} | x_{t-1}^{(N)}, \xi^{-1}\right). \quad (5.4)$$

While other likelihood functions are compatible with the SHGF, for simplicity we will assume that observations are generated by a Gaussian likelihood from the first (bottom) layer hidden states with variance τ^{-1} :

$$p\left(y_t | x_t^{(1)}\right) = \mathcal{N}\left(y_t | x_t^{(1)}, \tau^{-1}\right). \quad (5.5)$$

As (5.2) shows, the essential characteristic of an HGF model is that the variance of state transitions $g_t^{(i)}$ for the continuously valued states at layer i are controlled by a non-linear mapping of the continuously valued state at layer $i + 1$. In the extension to a *switching* HGF, the m^{th} component of the parameters $\kappa^{(i)}$ and $\omega^{(i)}$ of the nonlinear transformation (5.2) are selected by a discrete categorical state $s_t^{(i)} = m$ that evolves according to Markovian dynamics given by (5.1b). After selection of the component of parameters $\kappa^{(i)}$ and $\omega^{(i)}$, the corresponding transition in (5.1a) is selected by the categorical variable. Columns of transition matrices $\mathbf{A}^{(i)}$ define probability distributions that lie in $M_i - 1$ dimensional simplex (5.3).

A Forney-style factor graph (FFG) representation that corresponds to the SHGF model and a description of the graphical notation is given in Figure 5.1.

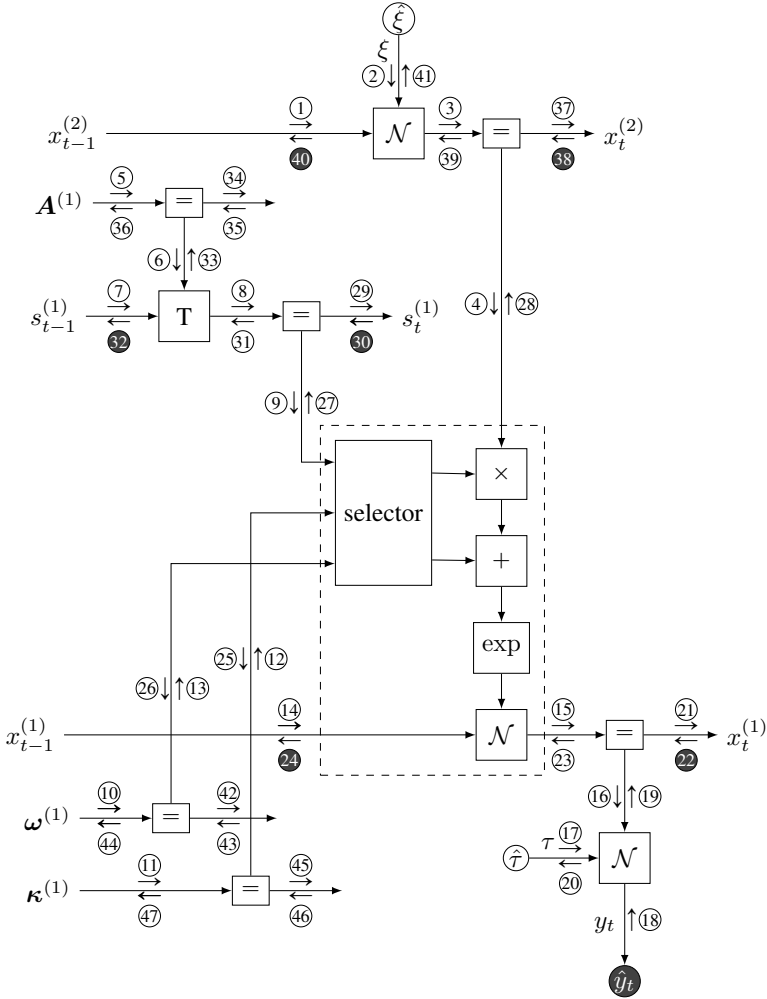


Figure 5.1: One time segment of an FFG corresponding to a 2-layer SHGF model. Nodes represent the factors. An abbreviation of the underlying functional form of the factors are given in the nodes. Small dark squares indicate an observation constraint, sending point mass messages. Arrows represent messages flowing at an edge. Circled numbers indicate a computation schedule that can be chosen differently. A dark circle refers to a backward message (from observations to latent variables). If dark messages are set proportional to 1 (i.e., uninformative), this message passing schedule results in filtering; otherwise, the schedule leads to smoothing. The selector node chooses the components of $\kappa^{(1)}$ and $\omega^{(1)}$, depending on the value of the categorical selector. The output of the selector node is then passed to the non-linearity block (5.2). The dashed box corresponds to the "composite" GCSV node $f_t^{(i)}$ defined by (5.8c).

5.3 Problem Definition and Message Passing

For a given SHGF model m and collection of data $\mathbf{y} \triangleq \mathbf{y}_{1:T} = [y_1 \dots y_T]$, we are interested in obtaining the posterior distributions for every layer i for the states $p(x_t^{(i)}|\mathbf{y})$, $p(s_t^{(i)}|\mathbf{y})$, and parameters $p(\kappa^{(i)}|\mathbf{y})$, $p(\omega^{(i)}|\mathbf{y})$, $p(A^{(i)}|\mathbf{y})$. Furthermore, to score model performance, we are interested in computing Bayesian evidence $p(\mathbf{y}|m)$.

To make matters concrete, suppose that we are interested in obtaining $p(x_t^{(i)}|\mathbf{y})$, then the corresponding Bayesian smoothing equations are given by [60]

$$p(x_t^{(i)}|\mathbf{y}) = p(x_t^{(i)}|\mathbf{y}_{1:t}) \int \frac{p(x_{t+1}^{(i)}|x_t^{(i)}) p(x_{t+1}^{(i)}|\mathbf{y})}{p(x_{t+1}^{(i)}|\mathbf{y}_{1:t})} dx_{t+1}^{(i)} \quad (5.6)$$

where the filtering equation is evaluated as

$$p(x_t^{(i)}|\mathbf{y}_{1:t}) = \frac{p(x_t^{(i)}|\mathbf{y}_{1:t-1}) p(x_t^{(i)}|y_t)}{\int p(x_t^{(i)}|\mathbf{y}_{1:t-1}) p(x_t^{(i)}|y_t) dx_t^{(i)}}. \quad (5.7)$$

The filtering process (5.7) requires evaluating

$$p(x_t^{(i)}|\mathbf{y}_{1:t-1}) = \int p(x_t^{(i)}|x_{t-1}^{(i)}) p(x_{t-1}^{(i)}|\mathbf{y}_{1:t-1}) dx_{t-1}^{(i)} \quad (5.8a)$$

$$p(x_t^{(i)}|y_t) = \mathbb{E}_{\{x_t^{(i)}\}} [f_t^{(i-1)}] \quad (5.8a)$$

$$p(x_t^{(i)}|x_{t-1}^{(i)}) = \mathbb{E}_{\{x_t^{(i)}, x_{t-1}^{(i)}\}} [f_t^{(i)}] \quad (5.8b)$$

$$f_t^{(i)} \triangleq p(x_t^{(i)}|x_{t-1}^{(i)}, s_t^{(i)}, \kappa^{(i)}, \omega^{(i)}, x_t^{(i+1)}) \quad (5.8c)$$

and the factor $f_t^{(i)}$ in (5.8c) is further specified by (5.1a) and (5.2). Note that computing the smoothing posterior (5.6) involves computing the filtering posterior (5.7), which in turn involves an exponentially growing number of summation terms due to the expectations in (5.8a) and (5.8b) with respect to categorical states. For example, if there are M categories, then there will be M indexed Gaussians at time $t = 1$, M^2 Gaussians at $t = 2$, and M^k Gaussians at $t = k$ [125, Ch. 25]. This explosion of terms make exact inference intractable. In addition, the non-linear couplings between the continuous state transitions in (5.8c) cause the complexity of functional dependencies for evaluating (5.8a) and (5.8b) to grow quickly. In short, the smoothing and filtering solutions (5.6) and (5.7) are not analytically tractable. In this chapter, we address approximating the filtering (5.7) and smoothing solutions (5.6) for the SHGF model. Above, we identified the non-linearities inside the factor $f_t^{(i)}$ and expectations

over internal categorical states in this factor as the problematic issues. We call this factor $f_t^{(i)}$ a “Gaussian with Controlled Switching Variance” (GCSV), see the dashed box in Figure 5.1.

Our solution to smoothing and filtering relies on exploiting the factorized structure of the SHGF model and uses variational message passing-based inference on factor graphs. This method supports solving the inference issues inside the GCSV node in isolation and then utilizing the GCSV factor as a plug-in node in any factor graph, including the graph for the SHGF model. Casting the problem into a factor graph will allow us to reduce the inference complexity through localized message computations. The modularity of FFGs exposes opportunities to improve computational performance in hierarchical models such as SHGF.

In the SHGF model, we constrain the approximating distribution to be factorized into normalized terms over hierarchical layers. In order to track correlations of states over time, we utilize a structured factorization that reflects the first-order Markovian structure of the model over the layers:

$$\prod_i q\left(x_t^{(i)}, x_{t-1}^{(i)}\right) q\left(s_t^{(i)}, s_{t-1}^{(i)}\right) q\left(\kappa^{(i)}\right) q\left(\omega^{(i)}\right) q\left(\mathbf{A}^{(i)}\right)$$

such that each factor

$$q\left(x_t^{(i)}\right) = \int q\left(x_t^{(i)}, x_{t-1}^{(i)}\right) dx_{t-1}^{(i)} \approx p\left(x_t^{(i)} | \mathbf{y}\right) \quad (5.9)$$

approximates the desired smoothing marginal (5.6) by imposing a marginalization constraint on the joint marginal of the states. We do not impose marginalization constraint on the parameters. In Theorem 3.2 we have showed that the specified constraints lead to the following distribution

$$q^*\left(x_t^{(i)}, x_{t-1}^{(i)}\right) = \frac{1}{Z_{x_t, t-1}^{(i)}} \exp\left(\mathbb{E}_{\setminus\{x_t^{(i)}, x_{t-1}^{(i)}\}} [\log p(\mathbf{y}, \mathbf{z})]\right) \quad (5.10)$$

where $Z_{x_t, t-1}^{(i)}$ is a normalization constant. Often, we do not have a closed form for the normalization constant and lack of closed form constitutes a challenge to computation of posterior. Due to the factorized model structure, the stationary marginals (5.9) can efficiently be obtained as multiplication of messages on a factor graph corresponding to the SHGF model. Due to the factorized model structure, the computation of (5.10) localizes over the time-segments of the FFG for the SHGF. This means that the model induces a factorization on the approximate posterior q that supports computation of the local marginal approximating the smoothing solution (5.6) by multiplication of

$$q\left(x_t^{(i)}\right) \propto \vec{\nu}\left(x_t^{(i)}\right) \bar{\nu}\left(x_t^{(i)}\right) \nu\uparrow\left(x_t^{(i)}\right) \quad (5.11)$$

where the messages are obtained via

$$\vec{\nu}(x_t^{(i)}) \propto \int \vec{\nu}(x_{t-1}^{(i)}) \tilde{p}(x_t^{(i)}, x_{t-1}^{(i)}) dx_{t-1}^{(i)} \quad (5.12a)$$

$$\tilde{\nu}(x_t^{(i)}) \propto \int \tilde{\nu}(x_{t+1}^{(i)}) \tilde{p}(x_t^{(i)}, x_{t+1}^{(i)}) dx_{t+1}^{(i)} \quad (5.12b)$$

$$\tilde{p}(x_t^{(i)}, x_{t-1}^{(i)}) \triangleq \exp\left(\mathbb{E}_{q(x_t^{(i)}, x_{t-1}^{(i)})} [\log f_t^{(i)}]\right) \quad (5.12c)$$

$$\nu \uparrow(x_t^{(i)}) \propto \exp\left(\mathbb{E}_{q(x_t^{(i)})} [\log f_t^{(i-1)}]\right). \quad (5.12d)$$

See Figure 5.1 for the messages around the GCSV node. Here (5.12a) and (5.12b) correspond to forward and backward messages that are sent by the GCSV node. The upward message to the upper layers is computed by (5.12d), and finally, the marginal is computed at an equality node by (5.11) through multiplying forward, backward, and upwards messages. By iteratively computing equation set (5.12), we obtain a structured variational message passing algorithm. In the FFG corresponding to the SHGF, messages around nodes other than GCSV can already be found in [127]. Around the GCSV node in Figure 5.1, the computation of messages (15), (24), (25), (26), (27) and (28) is the bottleneck to inference in the SHGF model. We give a detailed computation of the required messages and marginals in Section 5.4 and present the results for the GCSV node in Table 5.1 under the assumptions of Table 5.2. Messages (15) and (24) are Gaussian and message (27) is Categorical. The functional forms of the remaining messages do not correspond to known parametric exponential family distributions. We note that the messages associated with the continuously valued states have variances comprised of mixture of terms weighted by the discrete state probabilities. This mixture behaviour is the main difference with the HGF update equations.

Owing to the functional forms of messages (25), (26) and (28), the computation of marginals by multiplication, for example (5.11) is no longer a conjugate multiplication. If a parametric exponential family distribution does not approximate the multiplication, then the complexity of the variational algorithm grows and quickly becomes infeasible. Non-conjugate multiplications are also present in the message passing equations for the HGF, and the SHGF directly inherits these issues. Fortunately, there are various ways to approximate non-conjugate multiplications. For instance, Laplace approximation requires expanding the multiplication into a Taylor series and finding a stationary point where the gradient almost vanishes [6, Ch. 4.4]. Then the multiplication is approximated with a Gaussian distribution whose mean is a point where gradient vanishes and covariance is the inverse Hessian [6, Ch. 4.4]. Another approach is moment matching [60, Ch. 6] which gives rise to notable algorithms such expectation propagation [51] and assumed density filtering [128]. In [75], moment computations in expectation propagation is achieved by quadrature methods. Along the lines of moment matching, in Chapter 4 we have shown that quadrature-based moment matching outperforms Laplace's method. Here, we choose the quadrature-based moment approximation introduced in Chapter 4 to handle non-conjugate message multiplications.

The quadrature-based moment matching approximation starts by determining the normalization constant that corresponds to the marginal computed by (5.11). The computation assumes that the messages $\tilde{\nu}(x_t^{(i)})$ and $\bar{\nu}(x_t^{(i)})$ are Gaussian. This allows us to write the normalization constant in the form of a Gaussian integral with limits at infinity, i.e.,

$$Z_{x_t}^{(i)} = \int_{-\infty}^{\infty} \nu \uparrow (x_t^{(i)}) \mathcal{N} \left(x_t^{(i)} | \tilde{m}_t^{(i)}, \tilde{v}_t^{(i)} \right) dx_t^{(i)} \quad (5.13)$$

where $\tilde{m}_t^{(i)}$ and $\tilde{v}_t^{(i)}$ are the corresponding statistics for the Gaussian resulting from the multiplication of $\tilde{\nu}(x_t^{(i)})$ and $\bar{\nu}(x_t^{(i)})$. Using Hermite polynomials, integration in (5.13) can be obtained by Gaussian quadrature such that

$$Z_{x_t}^{(i)} \approx \frac{1}{\sqrt{\pi}} \sum_k w_k^{(i)} \nu \uparrow \left(\psi_k^{(i)} \sqrt{2\tilde{v}_t^{(i)}} + \tilde{m}_t^{(i)} \right) \quad (5.14)$$

where $\psi_k^{(i)}$ are points that are the roots of Hermite polynomials and $w_k^{(i)}$ are the corresponding weights [60, Ch. 6]. Once the normalization constant (5.14) has been determined, the moments of the distribution corresponding to the non-conjugate multiplication (5.11) can be evaluated by

$$\mathbb{E} \left[\left(x_t^{(i)} \right)^n \right] = \frac{\sum_k \nu \uparrow \left(\psi_k^{(i)} \sqrt{2\tilde{v}_t^{(i)}} + \tilde{m}_t^{(i)} \right) \left(\psi_k^{(i)} \sqrt{2\tilde{v}_t^{(i)}} + \tilde{m}_t^{(i)} \right)^n}{\sqrt{\pi} Z_{x_t}^{(i)}}.$$

Using the first two moments, we can now approximate the non-conjugate multiplication by a Gaussian distribution. Due to one dimensional nature of the problem, Gauss-Hermite quadrature is computationally feasible. The p^{th} order Gauss-Hermite integration is exact for monomials of the form $x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$ and their arbitrary combinations, where each of the order $d_i < 2p - 1$. In our experiments, we fix the order of Gauss-Hermite polynomials to 21 and plan to investigate the effect of the order of the polynomials on the free energy minimization in future work.

5.4 Message Computations for the SHGF

In this section we will compute the required messages $\textcircled{15}$, $\textcircled{24}$, $\textcircled{25}$, $\textcircled{26}$, $\textcircled{27}$ and $\textcircled{28}$ and the associated marginals. The results of this section are tabulated in Table 5.1 under the assumption of the functional forms in Table 5.2.

In order to compute the marginal for the switching state $s_t^{(i)}$ the following computations

Table 5.1: Summary of message computations for the GCSV node. Computations require quantities that are defined in Table 5.2.

Messages	Functional form
⑮	$\mathcal{N}\left(x_t^{(1)} \mid \vec{m}_{t-1}^{(1)}, \vec{v}_{t-1}^{(1)} + \sum_{k=1}^{M_t} \pi_{t,k}^{(1)} \exp\left(\gamma_{t,k}^{(1)} + \beta_{t,k}^{(1)}\right)^{-1}\right)$
⑳	$\mathcal{N}\left(x_t^{(1)} \mid \vec{m}_{t+1}^{(1)}, \vec{v}_{t+1}^{(1)} + \sum_{k=1}^{M_t} \pi_{t,k}^{(1)} \exp\left(\gamma_{t,k}^{(1)} + \beta_{t,k}^{(1)}\right)^{-1}\right)$
㉑	$\exp\left(-0.5 \sum_k \pi_{t,k}^{(1)} \left(\kappa_k^{(1)} m_t^{(2)} + r(\boldsymbol{\kappa}^{(i)})\right)\right)$
㉒	$\exp\left(-0.5 \sum_k \pi_{t,k}^{(1)} \left(\omega_k^{(1)} + \zeta_t^{(1)} \exp\left(-\omega_k^{(1)}\right)\right)\right)$
㉓	$\prod_j \exp\left(-0.5 \left(\eta_{t,j}^{(1)} + \zeta_t^{(1)} \exp\left(\gamma_{t,j}^{(1)} + \beta_{t,j}^{(1)}\right)\right)\right)^{\lfloor s_t^{(1)}=j \rfloor}$
㉔	$\exp\left(-0.5 \sum_j \pi_{t,j}^{(1)} \left(\boldsymbol{\mu}_t^{(1)}\right)_j x_t^{(2)} + h\left(x_t^{(2)}\right)\right)$
Auxiliary	Definition by moment statistics
$\eta_{t,j}^{(i)}$	$\left(\boldsymbol{\mu}_t^{(i)}\right)_j m_t^{(i+1)} + \left(\boldsymbol{\vartheta}_t^{(i)}\right)_j$
$\Psi_t^{(i)}$	$\left(\boldsymbol{\Sigma}_{t,t-1}^{(i)}\right)_{11} + \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)}\right)_{22} - \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)}\right)_{12} - \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)}\right)_{21}$
$\Phi_{t,j}^{(i)}$	$\left(\boldsymbol{\mu}_t^{(i)}\right)_j^2 v_t^{(i+1)} + \left(\boldsymbol{\Omega}_t^{(i)}\right)_{jj} \left(m_t^{(i+1)}\right)_j^2 + v_t^{(i+1)} \left(\boldsymbol{\Omega}_t^{(i)}\right)_{jj}$
$\zeta_t^{(i)}$	$\left(\left(m_{t,t-1}^{(i)}\right)_1 - \left(m_{t,t-1}^{(i)}\right)_2\right)^2 + \Psi_t^{(i)}$
$\gamma_{t,j}^{(i)}$	$-\left(\boldsymbol{\mu}_t^{(i)}\right)_j m_t^{(i+1)} + 0.5 \Phi_{t,j}^{(i)}$
$\beta_{t,j}^{(i)}$	$-\left(\boldsymbol{\vartheta}_t^{(i)}\right)_j + 0.5 \left(\boldsymbol{\Xi}_t^{(i)}\right)_{jj}$
$h\left(x_t^{(i)}\right)$	$\zeta_t^{(i)} \exp\left(-\left(\boldsymbol{\mu}_t^{(i-1)}\right)_j x_t^{(i)} + 0.5 \left(x_t^{(i)}\right)^2 \left(\boldsymbol{\Omega}_t^{(i-1)}\right)_{jj}\right)$
$r\left(\boldsymbol{\kappa}^{(i)}\right)$	$\zeta_t^{(i)} \exp\left(-m_t^{(i+1)} \kappa_k^{(i)} + 0.5 v_t^{(i+1)} \left(\kappa_k^{(i)}\right)^2\right)$

are required.

$$q\left(s_t^{(i)}\right) \propto \vec{v}\left(s_t^{(i)}\right) \nu \uparrow\left(s_t^{(i)}\right) \vec{v}\left(s_t^{(i)}\right) \quad (5.15a)$$

$$\vec{v}\left(s_t^{(i)}\right) \propto \sum_{s_{t-1}^{(i)}} \exp\left(\mathbb{E}_{q\left(s_t^{(i)}, s_{t-1}^{(i)}\right)}\left[\log p\left(s_t^{(i)} \mid s_{t-1}^{(i)}, \mathbf{A}^{(i)}\right)\right]\right) \vec{v}\left(s_{t-1}^{(i)}\right) \quad (5.15b)$$

$$\vec{v}\left(s_t^{(i)}\right) \propto \sum_{s_{t+1}^{(i)}} \exp\left(\mathbb{E}_{q\left(s_{t+1}^{(i)}, s_t^{(i)}\right)}\left[\log p\left(s_{t+1}^{(i)} \mid s_t^{(i)}, \mathbf{A}^{(i)}\right)\right]\right) \vec{v}\left(s_{t+1}^{(i)}\right) \quad (5.15c)$$

$$\nu \uparrow\left(s_t^{(i)}\right) \propto \exp\left(\mathbb{E}_{q\left(s_t^{(i)}\right)}\left[\log f_t^{(i)}\right]\right). \quad (5.15d)$$

Firstly we address the computation of forward message (5.15a). We assume that the messages $\vec{v} \left(s_{t-1}^{(i)} \right) \propto \text{Cat} \left(s_{t-1}^{(i)} | \vec{\pi}_{t-1}^{(i)} \right)$ and $\vec{v} \left(s_{t+1}^{(i)} \right) \propto \text{Cat} \left(s_{t+1}^{(i)} | \vec{\pi}_t^{(i)} \right)$ are Categorical and the marginal of the transition matrix $q_t \left(\mathbf{A}^{(i)} \right) \propto \text{Dir} \left(\mathbf{A}^{(i)} | \alpha_t^{(i)} \right)$ follows Dirichlet distribution. These choices are motivated by conjugacy. To compute the forward message we determine an intermediate expectation form

$$\begin{aligned}
\mathbb{E}_{q \left(s_t^{(i)}, s_{t-1}^{(i)} \right)} \left[\log p \left(s_t^{(i)} | s_{t-1}^{(i)}, \mathbf{A}^{(i)} \right) \right] &= \mathbb{E}_{q_t \left(\mathbf{A}^{(i)} \right)} \left[\log \prod_{m=1}^{M_t} \prod_{k=1}^{M_t} \left(\alpha_{km}^{(i)} \right)^{\left[s_t^{(i)}=k, s_{t-1}^{(i)}=m \right]} \right] \\
&= \sum_m \sum_k \mathbb{E}_{q_t \left(\mathbf{A}^{(i)} \right)} \left[\log \left(\alpha_{km}^{(i)} \right)^{\left[s_t^{(i)}=k, s_{t-1}^{(i)}=m \right]} \right] \\
&= \sum_m \sum_k \mathbb{E}_{q_t \left(\mathbf{A}^{(i)} \right)} \left[\log \left(\alpha_{km}^{(i)} \right) \right]^{\left[s_t^{(i)}=k, s_{t-1}^{(i)}=m \right]} \\
&= \sum_m \sum_k \left(\psi \left(\alpha_{t,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t,jm}^{(i)} \right) \right)^{\left[s_t^{(i)}=k, s_{t-1}^{(i)}=m \right]}. \tag{5.16}
\end{aligned}$$

where ψ denotes digamma function [129].

Substituting (5.16) into (5.15b) we obtain

$$\begin{aligned}
\vec{v} \left(s_t^{(i)} \right) &\propto \sum_{s_{t-1}^{(i)}} \exp \left(\sum_m \sum_k \left(\psi \left(\alpha_{t,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t,jm}^{(i)} \right) \right)^{\left[s_t^{(i)}=k, s_{t-1}^{(i)}=m \right]} \right) \vec{v} \left(s_{t-1}^{(i)} \right) \\
&\propto \exp \left(\sum_k \left(\sum_m \log \vec{\pi}_{t-1,m}^{(i)} \left(\psi \left(\alpha_{t,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t,jm}^{(i)} \right) \right) \right)^{\left[s_t^{(i)}=k \right]} \right) \\
&\propto \prod_k \exp \left(\sum_m \log \vec{\pi}_{t-1,m}^{(i)} \left(\psi \left(\alpha_{t,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t,jm}^{(i)} \right) \right) \right)^{\left[s_t^{(i)}=k \right]} \\
&\propto \prod_k \left(\vec{\pi}_{t,k}^{(i)} \right)^{\left[s_t^{(i)}=k \right]} \tag{5.17a}
\end{aligned}$$

$$\vec{\pi}_{t,k}^{(i)} \triangleq \exp \left(\sum_m \log \vec{\pi}_{t-1,m}^{(i)} \left(\psi \left(\alpha_{t,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t,jm}^{(i)} \right) \right) \right). \tag{5.17b}$$

Equation (5.17a) means that the forward message has a functional form proportional to a Categorical distribution. Following the steps in the derivation of the forward message (5.15c),

we obtain the backward message proportional to a Categorical distribution.

$$\begin{aligned} \tilde{\nu} \left(s_t^{(i)} \right) &\propto \sum_{s_{t+1}^{(i)}} \exp \left(\sum_m \sum_k \left(\psi \left(\alpha_{t+1,km}^{(i)} \right) - \psi \left(\sum_j \alpha_{t+1,jm}^{(i)} \right) \right) \Big|_{s_{t+1}^{(i)}=m, s_t^{(i)}=k} \right) \tilde{\nu} \left(s_{t+1}^{(i)} \right) \\ &\propto \prod_k \left(\tilde{\pi}_{t,k}^{(i)} \right) \Big|_{s_t^{(i)}=k} \end{aligned} \quad (5.18a)$$

$$\tilde{\pi}_{t,k}^{(i)} \triangleq \exp \left(\sum_m \log \tilde{\pi}_{t+1,m}^{(i)} \left(\psi \left(\alpha_{t+1,mk}^{(i)} \right) - \psi \left(\sum_j \alpha_{t+1,kj}^{(i)} \right) \right) \right). \quad (5.18b)$$

Lastly we compute (5.15d). Computation of the message (5.15d) requires the following expectation quantities

$$\eta_{t,j}^{(i)} \triangleq \mathbb{E} \left[\kappa_j^{(i)} x_t^{(i+1)} + \omega_j^{(i)} \right] = \left(\boldsymbol{\mu}_t^{(i)} \right)_j m_t^{(i+1)} + \left(\boldsymbol{\vartheta}_t^{(i)} \right)_j \quad (5.19a)$$

$$\zeta_t^{(i)} \triangleq \mathbb{E} \left[\left(x_t^{(i)} - x_{t-1}^{(i)} \right)^2 \right] \quad (5.19b)$$

$$\begin{aligned} &= \left(\left(\mathbf{m}_{t,t-1}^{(i)} \right)_1 - \left(\mathbf{m}_{t,t-1}^{(i)} \right)_2 \right)^2 + \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)} \right)_{11} + \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)} \right)_{22} - \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)} \right)_{12} - \left(\boldsymbol{\Sigma}_{t,t-1}^{(i)} \right)_{21} \\ \mathbb{E} \left[\exp \left(\kappa_j^{(i)} x_t^{(i+1)} + \omega_j^{(i)} \right)^{-1} \right] &\approx \exp \left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)} \right) \end{aligned} \quad (5.19c)$$

$$\gamma_{t,j}^{(i)} \triangleq - \left(\boldsymbol{\mu}_t^{(i)} \right)_j m_t^{(i+1)} + 0.5 \left(\left(\boldsymbol{\mu}_t^{(i)} \right)_j^2 v_t^{(i+1)} + \left(\boldsymbol{\Omega}_t^{(i)} \right)_{jj} \left(m_t^{(i+1)} \right)^2 + v_t^{(i+1)} \left(\boldsymbol{\Omega}_t^{(i)} \right)_{jj} \right) \quad (5.19d)$$

$$\beta_{t,j}^{(i)} \triangleq - \left(\boldsymbol{\vartheta}_t^{(i)} \right)_j + 0.5 \left(\boldsymbol{\Xi}_t^{(i)} \right)_{jj} \quad (5.19e)$$

Using the results from (5.19), the message (5.15d) is determined as Categorical. Eq. (5.19c) uses results from [130, Section 2] showing that multiplication of two Gaussian random variables can be approximated by a Gaussian.

$$\begin{aligned} \nu \uparrow \left(s_t^{(i)} \right) &\propto \mathbb{E}_{q \left(s_t^{(i)} \right)} \left[\log \prod_{j=1}^{M_i} \mathcal{N} \left(x_t^{(i)} \mid x_{t-1}^{(i)}, g_t^{(i)} \left(x_t^{(i+1)}, \kappa_j^{(i)}, \omega_j^{(i)} \right) \right) \Big|_{s_t^{(i)}=j} \right] \\ &= \prod_j \exp \left(-0.5 \left(\eta_{t,j}^{(i)} + \zeta_t^{(i)} \exp \left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)} \right) \right) \right) \Big|_{s_t^{(i)}=j}. \end{aligned} \quad (5.20a)$$

Due to exponential family being closed under multiplication, we know that the marginal distribution computed by the multiplication of 3 un-normalized Categorical messages (5.15a),

after normalization, will be a Categorical distribution (5.21).

$$q\left(s_t^{(i)}\right) = \frac{\bar{\nu}\left(s_t^{(i)}\right) \nu\left(s_t^{(i)}\right) \bar{\nu}\left(s_t^{(i)}\right)}{\sum_{s_t^{(i)}} \bar{\nu}\left(s_t^{(i)}\right) \nu\left(s_t^{(i)}\right) \bar{\nu}\left(s_t^{(i)}\right)} \quad (5.21a)$$

$$= \frac{\prod_{j=1}^{M_i} \left(\bar{\pi}_{t,j}^{(i)} \bar{\pi}_{t,j}^{(i)} \exp\left(-0.5\left(\eta_{t,j}^{(i)} + \zeta_t^{(i)} \exp\left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)}\right)\right)\right) \right)^{\left[s_t^{(i)}=j\right]}}{\sum_{j=1}^{M_i} \bar{\pi}_{t,j}^{(i)} \bar{\pi}_{t,j}^{(i)} \exp\left(-0.5\left(\eta_{t,j}^{(i)} + \zeta_t^{(i)} \exp\left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)}\right)\right)\right)} \quad (5.21b)$$

$$= \prod_j \left(\pi_{t,j}^{(i)} \right)^{\left[s_t^{(i)}=j\right]} \quad (5.21c)$$

$$\pi_{t,j}^{(i)} \triangleq \frac{\left(\bar{\pi}_{t,j}^{(i)} \bar{\pi}_{t,j}^{(i)} \exp\left(-0.5\left(\eta_{t,j}^{(i)} + \zeta_t^{(i)} \exp\left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)}\right)\right)\right) \right)}{\sum_{j=1}^{M_i} \bar{\pi}_{t,j}^{(i)} \bar{\pi}_{t,j}^{(i)} \exp\left(-0.5\left(\eta_{t,j}^{(i)} + \zeta_t^{(i)} \exp\left(\gamma_{t,j}^{(i)} + \beta_{t,j}^{(i)}\right)\right)\right)}. \quad (5.21d)$$

This concludes the marginal computation for the discrete states $s_t^{(i)}$ such that the resulting marginal for the switch state is Categorical.

Table 5.2: Messages and marginals required in Table 5.1.

Messages	Functional form
$\bar{\nu}\left(x_{t-1}^{(i)}\right)$	$\mathcal{N}\left(x_{t-1}^{(i)} \mid \bar{m}_{t-1}^{(i)}, \bar{v}_{t-1}^{(i)}\right)$
$\bar{\nu}\left(x_t^{(i)}\right)$	$\mathcal{N}\left(x_t^{(i)} \mid \bar{m}_t^{(i)}, \bar{v}_t^{(i)}\right)$.
Marginals	Functional form
$q\left(x_t^{(i-1)}, x_{t-1}^{(i-1)}\right)$	$\mathcal{N}\left(\mathbf{x}_{t,t-1}^{(i+1)} \mid \mathbf{m}_{t,t-1}^{(i+1)}, \boldsymbol{\Sigma}_{t,t-1}^{(i+1)}\right)$
$q\left(x_t^{(i+1)}\right)$	$\mathcal{N}\left(x_t^{(i+1)} \mid m_t^{(i+1)}, v_t^{(i+1)}\right)$
$q_t\left(\boldsymbol{\kappa}^{(i)}\right)$	$\mathcal{N}\left(\boldsymbol{\kappa}^{(i)} \mid \boldsymbol{\mu}_t^{(i)}, \boldsymbol{\Omega}_t^{(i)}\right)$
$q_t\left(\boldsymbol{\omega}^{(i)}\right)$	$\mathcal{N}\left(\boldsymbol{\omega}^{(i)} \mid \boldsymbol{\vartheta}_t^{(i)}, \boldsymbol{\Xi}_t^{(i)}\right)$
$q\left(s_t^{(i)}\right)$	$\prod_{k=1}^{M_i} \left(\pi_{t,k}^{(i)} \right)^{\left[s_t^{(i)}=k\right]}$

Required message computations for the continuous hierarchical states $x_t^{(i)}$ are as follows

$$q(x_t^{(i)}) \propto \bar{\nu}(x_t^{(i)}) \nu \uparrow(x_t^{(i)}) \bar{\nu}(x_t^{(i)}) \quad (5.22a)$$

$$\bar{\nu}(x_t^{(i)}) \propto \int \tilde{p}(x_t^{(i)}, x_{t-1}^{(i)}) \bar{\nu}(x_{t-1}^{(i)}) dx_{t-1}^{(i)} \quad (5.22b)$$

$$\bar{\nu}(x_t^{(i)}) \propto \int \tilde{p}(x_t^{(i)}, x_{t-1}^{(i)}) \bar{\nu}(x_{t-1}^{(i)}) dx_{t-1}^{(i)} \quad (5.22c)$$

$$\nu \uparrow(x_t^{(i)}) \propto \exp\left(\mathbb{E}_{q(x_t^{(i)})} \left[\log f_t^{(i-1)} \right]\right), \quad (5.22d)$$

where \tilde{p} is defined in (5.12c). Firstly, we compute the forward message (5.22b). In order to obtain the message we need determine the following expectation

$$\begin{aligned} & \mathbb{E}_{q(x_t^{(i)}, x_{t-1}^{(i)})} \left[\log \prod_{m=1}^{M_i} \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, g_t^{(i)}\left(x_t^{(i+1)}, \kappa_m^{(i)}, \omega_m^{(i)}\right)\right)^{[s_t^{(i)}=m]} \right] \\ &= \sum_{m=1}^{M_i} \pi_{t,m}^{(i)} \mathbb{E} \left[\log \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, g_t^{(i)}\left(x_t^{(i+1)}, \kappa_m^{(i)}, \omega_m^{(i)}\right)\right) \right] \end{aligned} \quad (5.23a)$$

$$\approx \sum_m \pi_{t,m}^{(i)} \log \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, \exp\left(\gamma_{t,m}^{(i)} + \beta_{t,m}^{(i)}\right)^{-1}\right) \quad (5.23b)$$

Plugging (5.23b) into (5.22b) we obtain

$$\begin{aligned} \bar{\nu}(x_t^{(i)}) &\propto \int \exp\left(\sum_m \pi_{t,m}^{(i)} \log \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, \exp\left(\gamma_{t,m}^{(i)} + \beta_{t,m}^{(i)}\right)^{-1}\right)\right) \bar{\nu}(x_{t-1}^{(i)}) dx_{t-1}^{(i)} \\ &\propto \mathcal{N}\left(x_t^{(i)} | \bar{m}_{t-1}^{(i)}, \bar{v}_{t-1}^{(i)} + \sum_{k=1}^{M_i} \pi_{t,k}^{(i)} \exp\left(\gamma_{t,k}^{(i)} + \beta_{t,k}^{(i)}\right)^{-1}\right). \end{aligned} \quad (5.24a)$$

Backwards message (5.22c) has the identical form with (5.22b) and it is evaluated as

$$\begin{aligned} \bar{\nu}(x_t^{(i)}) &\propto \int \exp\left(\sum_m \pi_{t,m}^{(i)} \log \mathcal{N}\left(x_t^{(i)} | x_{t-1}^{(i)}, \exp\left(\gamma_{t,m}^{(i)} + \beta_{t,m}^{(i)}\right)^{-1}\right)\right) \bar{\nu}(x_{t+1}^{(i)}) dx_{t+1}^{(i)} \\ &\propto \mathcal{N}\left(x_t^{(i)} | \bar{m}_{t+1}^{(i)}, \bar{v}_{t+1}^{(i)} + \sum_{k=1}^{M_i} \pi_{t,k}^{(i)} \exp\left(\gamma_{t,k}^{(i)} + \beta_{t,k}^{(i)}\right)^{-1}\right). \end{aligned} \quad (5.25a)$$

Forward (5.24a) and backward (5.25a) messages have variances that are composed of mixture terms. The categorical probabilities scale these mixture terms. To enforce "pure" switching behavior, we can modify the VMP updates for the backward and forward message by constraining the categorical variable $s_t^{(i)}$ to have a Kronecker delta distribution centered around

the mode of $q\left(s_t^{(i)}\right)$. We first determine the mode of the approximate posterior distribution for the categorical variable.

$$\tilde{s}_t^{(i)} = \arg \max_{s_t^{(i)}} q\left(s_t^{(i)}\right) \quad (5.26a)$$

$$\tilde{q}\left(s_t^{(i)}\right) = \delta\left(s_t^{(i)} - \tilde{s}_t^{(i)}\right). \quad (5.26b)$$

Then we use the point mass distribution to evaluate (5.23b) as

$$\begin{aligned} & \mathbb{E}_{\backslash q\left(x_t^{(i)}, x_{t-1}^{(i)}\right)} \left[\log \prod_{m=1}^{M_i} \mathcal{N}\left(x_t^{(i)} \mid x_{t-1}^{(i)}, g_t^{(i)}\left(x_t^{(i+1)}, \kappa_m^{(i)}, \omega_m^{(i)}\right)\right)^{\left[s_t^{(i)}=m\right]} \right] \\ & \approx \log \mathcal{N}\left(x_t^{(i)} \mid x_{t-1}^{(i)}, \exp\left(\gamma_{t, \tilde{s}_t^{(i)}}^{(i)} + \beta_{t, \tilde{s}_t^{(i)}}^{(i)}\right)^{-1}\right). \end{aligned} \quad (5.27)$$

Using (5.27), we can modify the forward and backward messages such that the variance information corresponds to selecting the parameters that come from the most probable switch. This means that, instead of mixture of components in the variance terms of (5.24a) and (5.25a) there is only one parameter regime influencing the message.

$$\bar{v}\left(x_t^{(i)}\right) \propto \mathcal{N}\left(x_t^{(i)} \mid \bar{m}_{t-1}^{(i)}, \bar{v}_{t-1}^{(i)} + \exp\left(\gamma_{t, \tilde{s}_t^{(i)}}^{(i)} + \beta_{t, \tilde{s}_t^{(i)}}^{(i)}\right)^{-1}\right) \quad (5.28a)$$

$$\bar{v}\left(x_t^{(i)}\right) \propto \mathcal{N}\left(x_t^{(i)} \mid \bar{m}_{t-1}^{(i)}, \bar{v}_{t-1}^{(i)} + \exp\left(\gamma_{t, \tilde{s}_t^{(i)}}^{(i)} + \beta_{t, \tilde{s}_t^{(i)}}^{(i)}\right)^{-1}\right). \quad (5.28b)$$

The desired behavior message updates for the continuous-valued hierarchical states can be changed and used accordingly. Next, we compute the message towards continuous-valued hierarchical states propagated from the lower layers.

Firstly, we evaluate the following expectation

$$\begin{aligned} & \mathbb{E}_{\backslash q\left(x_t^{(i)}\right)} \left[\log \prod_{m=1}^{M_i} \mathcal{N}\left(x_t^{(i-1)} \mid x_{t-1}^{(i-1)}, g_t^{(i-1)}\left(x_t^{(i)}, \kappa_m^{(i-1)}, \omega_m^{(i-1)}\right)\right)^{\left[s_t^{(i-1)}=m\right]} \right] \\ & = \sum_m \pi_{t,m}^{(i)} \mathbb{E} \left[\log \mathcal{N}\left(x_t^{(i-1)} \mid x_{t-1}^{(i-1)}, g_t^{(i-1)}\left(x_t^{(i)}, \kappa_m^{(i-1)}, \omega_m^{(i-1)}\right)\right) \right]. \end{aligned} \quad (5.29a)$$

Evaluation requires the following intermediate result

$$\begin{aligned} & \mathbb{E}_{\backslash q\left(x_t^{(i)}\right)} \left[\exp\left(\kappa_j^{(i-1)} x_t^{(i)} + \omega_j^{(i-1)}\right)^{-1} \right] \\ & \approx \exp\left(-\left(\boldsymbol{\mu}_t^{(i-1)}\right)_j x_t^{(i)} + 0.5 \left(x_t^{(i)}\right)^2 \left(\boldsymbol{\Omega}_t^{(i-1)}\right)_{jj}\right) \end{aligned} \quad (5.30)$$

Using this result (5.30) we can write

$$\nu \uparrow (x_t^{(i)}) \propto \exp \left(\mathbb{E}_{q(x_t^{(i)})} \left[\log f_t^{(i-1)} \right] \right) \quad (5.31a)$$

$$\propto \exp \left(-0.5 \sum_j \pi_{t,j}^{(i-1)} \left(\left(\boldsymbol{\mu}_t^{(i-1)} \right)_j x_t^{(i)} + h \left(x_t^{(i)} \right) \right) \right), \quad (5.31b)$$

where we define

$$h \left(x_t^{(i)} \right) \triangleq \zeta_t^{(i)} \exp \left(- \left(\boldsymbol{\mu}_t^{(i-1)} \right)_j x_t^{(i)} + 0.5 \left(x_t^{(i)} \right)^2 \left(\boldsymbol{\Omega}_t^{(i-1)} \right)_{jj} \right). \quad (5.32)$$

The functional form of (5.31b) suggests that $\nu \uparrow (x_t^{(i)})$ is not in the exponential family and contains a weighted sum term. Again, by constraining the categorical variable to have a point mass distribution centered at the MAP estimate of the variational distribution (5.26a) and (5.26b), it is possible to induce a switching behavior as opposed to a mixture behavior. As (5.31b) is not in the exponential family, the further approximations are needed to obtain the marginal (5.22a). We know that the forward and backward messages are Gaussian (5.25a) and (5.24a). This means that we can utilize these messages to approximate the marginal of $x_t^{(i)}$ with a Gaussian. This is achieved by employing the Gauss-Hermite quadrature to evaluate the non-conjugate multiplication by moment matching. We want to approximate with Gaussian can be motivated from different perspectives. First of all, keeping the functional form and trying to propagate the functional form is computationally challenging as with every functional form, additional rules are needed.

Required messages for computing the marginal of $\boldsymbol{\omega}^{(i)}$ are

$$q_t \left(\boldsymbol{\omega}^{(i)} \right) \propto \bar{\nu} \left(\boldsymbol{\omega}^{(i)} \right) \prod_{t'=1}^t \bar{\nu}_{t'} \left(\boldsymbol{\omega}^{(i)} \right) \quad (5.33a)$$

$$\bar{\nu} \left(\boldsymbol{\omega}^{(i)} \right) \propto p \left(\boldsymbol{\omega}^{(i)} \right) \quad (5.33b)$$

$$\bar{\nu}_t \left(\boldsymbol{\omega}^{(i)} \right) \propto \exp \left(\mathbb{E}_{q(\boldsymbol{\omega}^{(i)})} \left[\log f_t^{(i)} \right] \right). \quad (5.33c)$$

The forward message is the prior hence it does not require computation. We compute the only required message $\bar{\nu}_t \left(\boldsymbol{\omega}^{(i)} \right)$. Using the distributions of Table 5.2 we determine the backward message as

$$\bar{\nu}_t \left(\boldsymbol{\omega}^{(i)} \right) \propto \exp \left(-0.5 \sum_k \pi_{t,k}^{(i)} \left(\omega_k^{(i)} + \zeta_t^{(i)} \exp \left(-\omega_k^{(i)} \right) \right) \right) \quad (5.34)$$

Required messages for $\kappa^{(i)}$ are as follows

$$q_t \left(\kappa^{(i)} \right) \propto \bar{v} \left(\kappa^{(i)} \right) \prod_{t'=1}^t \bar{v}_{t'} \left(\kappa^{(i)} \right) \quad (5.35a)$$

$$\bar{v} \left(\kappa^{(i)} \right) \propto p \left(\kappa^{(i)} \right) \quad (5.35b)$$

$$\bar{v}_t \left(\kappa^{(i)} \right) \propto \exp \left(\mathbb{E}_{q(\kappa^{(i)})} \left[\log f_t^{(i)} \right] \right). \quad (5.35c)$$

We evaluate the backward message towards $\kappa^{(i)}$

$$\bar{v}_t \left(\kappa^{(i)} \right) \propto \exp \left(-0.5 \sum_k \pi_{t,k}^{(i)} \left(\kappa_k^{(i)} m_t^{(i+1)} + r \left(\kappa^{(i)} \right) \right) \right) \quad (5.36)$$

$$r \left(\kappa^{(i)} \right) \triangleq \zeta_t^{(i)} \exp \left(-m_t^{(i+1)} \kappa_k^{(i)} + 0.5 v_t^{(i+1)} \left(\kappa_k^{(i)} \right)^2 \right) \quad (5.37)$$

Required messages for $\mathbf{A}^{(i)}$ are given by

$$q_t \left(\mathbf{A}^{(i)} \right) \propto \bar{v} \left(\mathbf{A}^{(i)} \right) \prod_{t'=1}^t \bar{v}_{t'} \left(\mathbf{A}^{(i)} \right) \quad (5.38a)$$

$$\bar{v} \left(\mathbf{A}^{(i)} \right) \propto p \left(\mathbf{A}^{(i)} \right) \quad (5.38b)$$

$$\bar{v}_t \left(\mathbf{A}^{(i)} \right) \propto \exp \left(\mathbb{E}_{q(\mathbf{A}^{(i)})} \left[\log p \left(s_t^{(i)} | s_{t-1}^{(i)}, \mathbf{A}^{(i)} \right) \right] \right). \quad (5.38c)$$

For a sake of brevity let us rewrite the message $\bar{v}_t \left(\mathbf{A}^{(i)} \right)$ in a log domain.

$$\log \bar{v}_t \left(\mathbf{A}^{(i)} \right) = \mathbb{E}_{q(\mathbf{A}^{(i)})} \left[\log p \left(s_t^{(i)} | s_{t-1}^{(i)}, \mathbf{A}^{(i)} \right) \right] + const \quad (5.39)$$

$$= \mathbb{E}_{q(\mathbf{A}^{(i)})} \left[\log \prod_{k=1}^{M_i} \prod_{m=1}^{M_i} \left(\alpha_{km}^{(i)} \right)^{[s_t^{(i)}=k, s_{t-1}^{(i)}=m]} \right] + const \quad (5.40)$$

$$= \sum_{k=1}^{M_i} \sum_{m=1}^{M_i} \mathbb{E}_{q(\mathbf{A}^{(i)})} \left[\log \left(\alpha_{km}^{(i)} \right)^{[s_t^{(i)}=k, s_{t-1}^{(i)}=m]} \right] + const \quad (5.41)$$

$$= \sum_{k=1}^{M_i} \sum_{m=1}^{M_i} \mathbb{E}_{q(\mathbf{A}^{(i)})} \left[\log \left(\alpha_{km}^{(i)} \right) \right]^{[s_t^{(i)}=k, s_{t-1}^{(i)}=m]} + const \quad (5.42)$$

$$= \sum_{k=1}^{M_i} \sum_{m=1}^{M_i} \mathbb{E}_{q(\mathbf{A}^{(i)})} \left[[s_t^{(i)} = k, s_{t-1}^{(i)} = m] \log \left(\alpha_{km}^{(i)} \right) \right] + const \quad (5.43)$$

$$= \sum_{k=1}^{M_i} \sum_{m=1}^{M_i} \rho_{k,m} \log \left(\alpha_{km}^{(i)} \right) + const \quad (5.44)$$

where $\rho_{k,m}$ are entries of contingency matrix \mathbf{P} of, i.e. $q(s_t^{(i)}, s_{t-1}^{(i)}) \propto \text{Con}(s_t^{(i)}, s_{t-1}^{(i)} | \mathbf{P})$. Hence,

$$\tilde{\nu}_t(\mathbf{A}^{(i)}) = \prod_{m=1}^{M_i} \frac{\Gamma\left(\sum_{k=1}^{M_i} (\rho_{k,m}^{(i)} + 1)\right)}{\prod_{k=1}^{M_i} \Gamma(\rho_{k,m}^{(i)} + 1)} \left(\prod_{k=1}^{M_i} \alpha_{km}^{(i)}\right)^{\rho_{k,m}^{(i)}} \quad (5.45)$$

and defining $\hat{\rho} = \rho + 1$ we can rewrite (5.45) as

$$\tilde{\nu}_t(\mathbf{A}^{(i)}) = \prod_{m=1}^{M_i} \frac{\Gamma\left(\sum_{k=1}^{M_i} (\hat{\rho}_{k,m}^{(i)})\right)}{\prod_{k=1}^{M_i} \Gamma(\hat{\rho}_{k,m}^{(i)})} \left(\prod_{k=1}^{M_i} \alpha_{km}^{(i)}\right)^{\hat{\rho}_{k,m}^{(i)} - 1} \quad (5.46)$$

5.5 Simulations

All experiments have been implemented with the Julia package ForneyLab [33]. The source code for the experiments can be found at <https://github.com/biaslab/SGCV>.

5.5.1 Verification on Synthetic Data

To verify the proposed inference algorithm, we built a 2-layer (2-L) SHGF model (see Fig. 5.1) where $\omega^{(1)}, \kappa^{(1)} \in \mathbb{R}^3$. We generated $N = 100$ data sets with $T = 500$ observation points in each set. We used weakly informative priors for $x_0^{(1)}, x_0^{(2)}, s_0^{(1)}$ and $\mathbf{A}^{(1)}$, but informative for $\omega^{(1)}$, i.e. $\omega^{(1)} \sim \mathcal{N}(\omega^*, \mathbf{I})$ where $\omega^* \sim \mathcal{N}(\omega^{\text{true}}, \mathbf{I})$ (ω^{true} denotes ground-truth parameters). Note that in these experiments we did not learn $\kappa^{(1)}$. As the update equations for $\kappa^{(1)}$ and x_t are symmetrical, one of these random variables should be observed. Otherwise, learning of $\kappa^{(1)}$ together with $\omega^{(1)}$ and x_t would lead to identifiability issues. An approach to overcome identifiability is to constrain $\kappa^{(1)}$ further. For example, constraining the support set of $\kappa^{(1)}$ from \mathbb{R}^{M_1} to $[0, 1]^{M_1}$ and bounding the variance of state transitions that $\kappa^{(1)}$ undergoes, is an approach that can help learning of $\kappa^{(1)}$.

We ran the proposed message-passing algorithm on the full SHGF graph with $T = 500$ time segments and 500×10 nodes in total for the entire data set. The update schedule for a one-time segment of SHGF is shown in Fig. 5.1. Fig. 5.2 reports the results of the verification experiments. The evolution of free energy in Fig. 5.2 (top-left) indicates that the hybrid VMP algorithm consistently decreases free energy averaged over the entire data set and converges to stationary points of the Bethe free energy. Fig. 5.2 (left) reports the convergence of the proposed message-passing algorithm for two-layered SHGF. An example

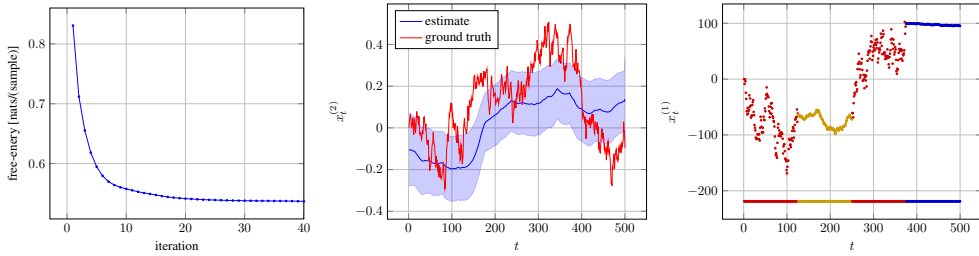


Figure 5.2: Verification results. (left) Evolution of free energy per variational iteration averaged over data sets ($N = 100$) and the number of observations ($T = 500$). The curve indicates that the proposed algorithm consistently minimizes free energy and converges to stationary solutions. (middle) An example of the inference of the upper layer random walk. (right) An example of observations from one of the data sets used to verify the algorithm. Each color represents a particular regime (switch). Observations are color-coded according to the regimes they are generated from.

of successfully recovered states is depicted in Fig. 5.2. The red signal (middle of Fig.5.2) indicates the second layer continuous state $x_t^{(2)}$ that corresponds to the observations at the bottom figure. The blue curve is the estimate of the state obtained by the VMP algorithm. The estimate recovers the trend of the second layer state.

In the right plot of Fig.5.2 plots, the mode of categorical distributions corresponding to the switch variables for the entire time points is marked below the signal. The plot indicates that the recovery of switching regimes matches the ground truth.

5.5.2 Validation on Stock Prices

We applied the SHGF model to a real-world data set to validate our model. The data set corresponds to AAPL stock prices (downloaded from <https://finance.yahoo.com/quote/AAPL/>). We wanted to test if the stock price evolution exhibits regime-switching dynamics over a consecutive period of $T = 252$ days. We used the minimized variational free energy as a model performance score and compared four models: a 2-layer HGF, 2-layer SHGFs with 2 and 3 categories, and a 3-layer SHGF. To keep the comparison fair, we used identical priors for the states and parameters where possible.

Fig. 5.3 highlights the results of validation experiments. The 2-layer SHGF with two categories results in lower free energy than the 3-layer SHGF (too complex) and HGF (too simple). The 2-layer SHGF with three categories assigns vanishing probability to the 3rd category, so the 2-layer SHGF with two regimes is optimal. This indicates that the underlying prices submit to two-category regime-switching dynamics.

All SHGF models outperform the 2-layer HGF, indicating that the prices exhibit switching behavior. Since the 2-layer SHGFs perform better than a 3-layer SHGF, we conclude that the extra model complexity of the 3rd layer outweighs the increased accuracy due to the 3rd layer. The 2-layer SHGF with three categories performs almost as well as the one with two categories. The inference results for the three categories indicate that the model assigns a

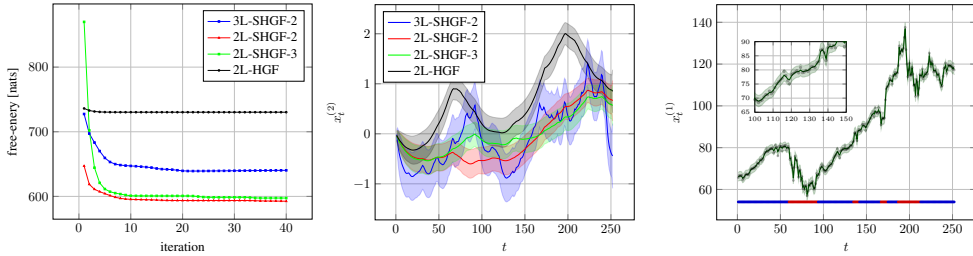


Figure 5.3: Validation results. (left) Free energy plots corresponding to a 3-layer SHGF with 2 categories, a 2-layer SHGF with 3 categories, a 2-layer SHGF with 2 categories and a 2-layer HGF, where the observations are AAPL stock prices. (middle) Second layer state trajectories for the 3-layer SHGF, 2-layer SHGFs and 2-layer HGF models obtained from the AAPL stocks. (right) Black dots correspond to the stock prices. The green curve represents the belief trajectory for $x_t^{(1)}$ obtained by the 2-layer SHGF model. In order to avoid clutter in the plot, we only present the model with the lowest free energy (2L-SHGF-2). We display a zoomed version on the smooth behavior of the belief trajectory. The obtained switches are color coded and displayed beneath the prices. Based on the model, there are 2 underlying regimes governing the prices.

vanishing probability to the third category, meaning that it settles for two categories.

The 3-layer SHGF is quite active due to an extra layer. The 2-layer SHGFs and HGF are smoother. These three trajectories share a similar trend where the volatility makes two peaks around $t = 60$ and $t = 200$. On average, the HGF model attributes higher volatility to the stock prices than the SHGF model.

5.6 Discussion and Conclusions

We introduced a Switching Hierarchical Gaussian Filter (HGF) to model regime-switching non-stationary time series of volatile environments. The proposed model extends the classical HGF by assuming that the parameters in each layer are selected by a discrete state, which evolves according to a hidden Markov model. We presented a closed-form variational message passing framework to track all states and the transition matrix (a matrix of parameters) for the discrete states. The presented message passing framework relies on the hybridization of conjugate and non-conjugate variational message update rules. We verified that the proposed inference algorithm finds the stationary solutions of the minimization problem and consistently minimizes free energy. After verification, we showed that the SHGF provides improved results over the HGF on modeling of stock prices. Crucially, the closed-form update rules for the problematic GCSV factor allow it to be used as a plug-in node in any factor graph, thus enabling message passing-based inference for alternative hierarchical dynamic models.

CHAPTER 6

Auto-regressive Models with Time-varying Noise Processes

"Everything we call real is made of things that cannot be regarded as real."

–Niels Bohr

6

6.1 Introduction

Auto-regressive (AR) models are of fundamental importance to physics, economics, and engineering problems. Although standard AR models have been successfully applied to various practical domains [131, 132], the underlying dynamics are often assumed to be stationary. Still, many applications involve modeling signals where time-varying process statistics would lead to better performance. For example, good models for stock market prices contain time-varying variance parameters [125, Chapter 24]. Therefore, it is essential to be able to track slowly-varying parameters and fast-changing latent states. Unfortunately, online Bayesian tracking leads to intractable equations for many dynamic models.

Classical alternatives for modeling signals that are generated by non-stationary processes include time-varying AR and generalized AR conditional heteroskedasticity models [125, 133]. These models are powerful, but inference methods often lack the Bayesian approach's advantages as these methods are mostly sampling-based. Modern MCMC sampling-based methods generally track these types of dynamics well [134, 135], but are often too slow for online inference. Variational Bayes approaches are both fast and robust to overfitting. [136] model process noise with a Gaussian mixture model, achieving time-varying variance via a dynamic selection of components.

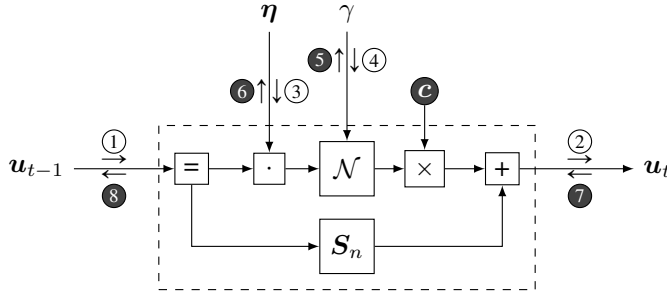


Figure 6.1: Composite node structure of the state transition (6.4) governing an AR process

Time-Varying AR (TVAR) models, where the AR coefficients are allowed to vary over time, support tracking signals produced by non-stationary processes. TVAR models have been successfully applied to a wide range of applications, including speech signal processing [9, 137, 138], signature verification [139], cardiovascular response modeling [140], acoustic signature recognition of vehicles [141], radar signal processing [142], and EEG analysis [143, 144]. In these works, parameter and state estimation in TVAR models have been explored in non-Bayesian settings. Solving modified Yule-Walker equations, [145] and utilizing wavelets for TVAR model identification [146] are examples of popular non-Bayesian approaches. In contrast to the non-Bayesian trend in the literature, TVAR models have been formulated within the factor graph framework along with a low-complexity VMP algorithm for inference in [96]. Our aim in this chapter will be to extend the scope of AR models by allowing non-stationary noise processes as innovations for AR models within a Bayesian setting.

This chapter introduces an AR model with a time-varying noise process. The proposed (AR-HGF) model is flexible in prior assumptions for the parameter dynamics. Our contributions include the following: first, in Sec. 6.2 we introduce our AR-HGF model and present a corresponding Forney-style Factor Graph. In Sec. 6.3 we specify the problem statement and show that the combination of update rules derived in Chapter 4 with the update rules of [96, 147] is sufficient to support online parameter and state estimation in the proposed model. In Sec. 6.5 we verify the performance of an AR-HGF model in a T-step ahead prediction task on a financial data-set.

6.2 Model Specification and FFG Representation

We consider an M -th order auto-regressive representation (AR(M)) for a latent variable $y_t \in \mathbb{R}$, specified by

$$y_t = \sum_{m=1}^M \theta_m y_{t-m} + \epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}(0, \vartheta_t). \quad (6.1)$$

This representation is parameterized by auto-regressive coefficients $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_M]^\top \in \mathbb{R}^M$ and time-varying process noise variance $\vartheta_t \in \mathbb{R}^+$. We assume that observations o_t are obtained from the latent state y_t after corrupted by Gaussian noise as $p(o_t|y_t, \zeta) = \mathcal{N}(o_t|y_t, \zeta^{-1})$, where ζ is the precision of the likelihood. For the process noise variance ϑ_t , we impose an additional AR prior that is mapped to the positive domain to serve as a variance parameter, which is specified by

$$z_t = \sum_{m=1}^N \eta_m z_{t-m} + \psi_t \quad \text{where } \psi_t \sim \mathcal{N}(0, \gamma^{-1}), \quad (6.2a)$$

$$\vartheta_t = \exp(\kappa z_t + \omega), \quad (6.2b)$$

with initial states $z_0 \sim \mathcal{N}(m_{z_0}, v_{z_0})$ and $y_0 \sim \mathcal{N}(m_{y_0}, v_{y_0})$, which we usually choose as rather uninformative (e.g., zero mean and large variance). We will refer to $z_t \in \mathbb{R}$ as a *control* state since it controls the variance of ϵ_t . The control state is also an AR process with $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_N]^\top \in \mathbb{R}^N$ corresponding to AR coefficients. Noise processes of the control state is perturbed by Gaussian transition noise whose precision is γ . We do not assume that γ is known but we assume that γ does not change over time. To complete the model specification, we use the following priors on the parameters:

$$\begin{aligned} \boldsymbol{\theta} &\sim \mathcal{N}(\mathbf{m}_\theta, \mathbf{V}_\theta), \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{m}_\eta, \mathbf{V}_\eta), \\ \gamma &\sim \Gamma(\alpha_\gamma, \beta_\gamma), \quad \zeta \sim \Gamma(\alpha_\zeta, \beta_\zeta), \quad \kappa \sim \mathcal{N}(m_\kappa, v_\kappa), \quad \omega \sim \mathcal{N}(m_\omega, v_\omega). \end{aligned} \quad (6.3)$$

Note that we can rewrite Eq. 6.1 in the state space representation

$$y_t = \boldsymbol{\theta}^\top \mathbf{x}_{t-1} + \epsilon_t, \quad z_t = \boldsymbol{\eta}^\top \mathbf{u}_{t-1} + \psi_t \quad (6.4a)$$

$$\mathbf{x}_t = \mathbf{S}_M \mathbf{x}_{t-1} + \mathbf{c} y_t, \quad \mathbf{u}_t = \mathbf{S}_N \mathbf{u}_{t-1} + \mathbf{c} z_t \quad (6.4b)$$

with the following definitions

$$\mathbf{x}_t \triangleq [y_t, y_{t-1}, \dots, y_{t-M+1}]^\top, \quad \mathbf{u}_t \triangleq [z_t, z_{t-1}, \dots, z_{t-N+1}]^\top \quad (6.5)$$

$$\mathbf{S}_n \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{n-1} & \mathbf{0} \end{bmatrix}, \quad \mathbf{c} \triangleq [1, 0, \dots, 0]^\top, \quad (6.6)$$

where \mathbf{I}_{n-1} is the identity matrix of size $n-1$ by $n-1$. We refer to \mathbf{x}_t and \mathbf{u}_t as data buffers, since they retain M previous latent states. A composite factor node corresponding to the state transitions (6.4) is given in Figure 6.1 The factor graph can now be constructed by rewriting the full model as the following factorized probability distribution:

$$p(\mathbf{o}, \mathbf{y}, \mathbf{z}, \mathbf{x}, \mathbf{u}, \boldsymbol{\Psi}) = \quad (6.7)$$

$$\underbrace{p(z_0, y_0, \boldsymbol{\Psi})}_{\text{priors}} \prod_{t=1}^T \underbrace{p(o_t|y_t, \boldsymbol{\Psi})}_{\text{observation}} \underbrace{p(y_t|z_t, \mathbf{x}_{t-1}, \boldsymbol{\Psi}) p(z_t|\mathbf{u}_{t-1}, \boldsymbol{\Psi})}_{\text{state transition}} \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1}, y_t) p(\mathbf{u}_t|\mathbf{u}_{t-1}, z_t)}_{\text{data buffer}}$$

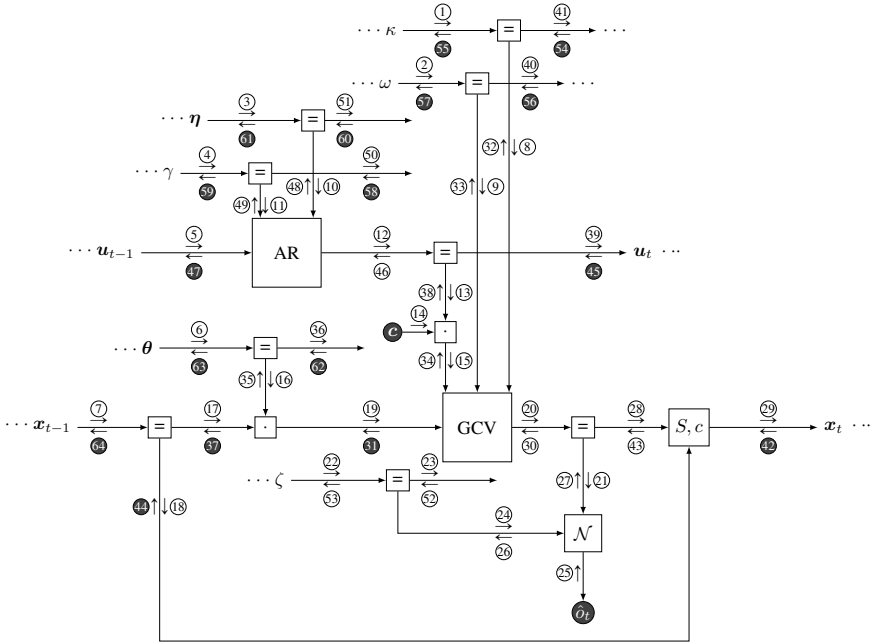


Figure 6.2: An FFG corresponding to the one time segment of the model specified in (6.7). The graph contains loops and non-conjugate factor pairs. AR node represents the state transition $p(\mathbf{u}_t | \mathbf{u}_{t-1}, \boldsymbol{\eta}, \gamma)$ in accordance with [96] as a composite factor. The node S, c represents the data buffer.

where the priors are given by Eq. 6.3. State transitions are given by $p(y_t | z_t, \mathbf{x}_{t-1}, \boldsymbol{\Psi}) = \mathcal{N}(y_t | \boldsymbol{\theta}^\top \mathbf{x}_{t-1}, z_t)$, $p(z_t | \mathbf{u}_{t-1}, \boldsymbol{\Psi}) = \mathcal{N}(z_t | \boldsymbol{\eta}^\top \mathbf{u}_{t-1}, \gamma^{-1})$ and data buffers are represented with $p(\mathbf{x}_t | \mathbf{x}_{t-1}, y_t) = \delta(\mathbf{x}_t - (\mathbf{S}\mathbf{x}_{t-1} + \mathbf{c}y_t))$ and $p(\mathbf{u}_t | \mathbf{u}_{t-1}, z_t) = \delta(\mathbf{u}_t - (\mathbf{S}\mathbf{u}_{t-1} + \mathbf{c}z_t))$. With these specifications we complete the model proposal. In summary, the presented model augments an AR model with a time-varying noise-process that itself is an AR model. Mapping of an AR process to positive domain by an exponential non-linearity is the same transformation that allows for construction of an HGF model. This means that the composite factor GCV can again be used as a building block for the proposed model with already tabulated message update rules. Having specified the model we will now translate it to an FFG representation.

Fig. 6.2 is an FFG corresponding to one time-segment of the model in Eq. 6.7. The FFG contains Gaussian nodes \mathcal{N} , dot product nodes \cdot , a data buffer update node S, c , equality nodes $=$, a GCV node an AR node. Details of the AR node are illustrated in Figure 6.1. The GCV node allows for modeling a time-varying process noise for the AR processes modeling the observations. By adjusting the location of the GCV node in the graph, it is possible to obtain different noise regimes for the underlying AR process. For instance, if the Gaussian likelihood is replaced with another GCV node, the likelihood noise process would be time-varying. Another alternative would be introducing a switching behavior to model context

switches for the non-stationary noise behavior. This can be easily achieved by the GSCV node that is described in Chapter 5.

An interesting feature of the FFG of the model is its loopy structure. The loops are caused due to the necessity of keeping past states. The AR node itself contains loops. By treating the AR node as a composite structure, we break the loops within, and the only remaining loops are around the GCV nodes. The loops around the GCV node As the order of the underlying AR process increase, the number of latent states involved in the loops will increase. The presence of loops involving many states can drastically influence the convergence of free energy.

6.3 Problem Definition

We are interested in joint tracking of states y_t, z_t and parameters Ψ in model Eq. 6.7. On-line inference can be achieved by sequential Bayesian updating which leads to a Chapman-Kolmogorov integral:

$$\underbrace{p(y_t, z_t, \Psi | \mathbf{o}_{1:t})}_{\text{posterior}} \propto \int \underbrace{p(o_t | y_t, \Psi)}_{\text{likelihood}} \underbrace{p(y_t | z_t, \mathbf{x}_{t-1}, \Psi)}_{\text{state transition}} p(z_t | \mathbf{u}_{t-1}, \Psi) \underbrace{p(\mathbf{u}_t | \mathbf{u}_{t-1}, z_t)}_{\text{data buffer}} \underbrace{p(\mathbf{x}_t | \mathbf{x}_{t-1}, y_t)}_{\text{prior}} p(y_{t-1}, z_{t-1}, \Psi | \mathbf{o}_{1:t-1}) d\mathbf{u}_{t-1} d\mathbf{x}_{t-1}. \quad (6.8)$$

Due to non-conjugate couplings, the integral (6.8) is not tractable analytically. Purely sampling-based methods will suffer to approximate the integral (6.8) if the AR order is high. Due to the presence of loops, the naive implementation of BP is not straightforward. Loopy BP is a viable option; however, to solve the BP updates, one needs to resort to sampling, and hence BP will inherit the computational complexity of sampling methods. Again, we resort to computing approximate posterior distributions using variational message passing to avoid computationally expensive algorithms that hinder online inference.

6.4 Variational Inference

We will resort to VMP to approximate the intractable Chapman-Kolmogorov integral (6.8). To perform VMP we assume the following factorization

$$q(\mathbf{x}, \mathbf{u}, \Psi) = q(\mathbf{x})q(\mathbf{u})q(\kappa)q(\omega)q(\boldsymbol{\theta})q(\gamma)q(\zeta) \quad (6.9)$$

We will not further assume that the recognition distributions for the states factorizes over temporal dimension. However, the model induces a factorization over temporal dimension due to factorized model definition. This means that for every time slice of the FFG the states

are factorized according to Bethe assumption, implying that the states will be constrained by the following marginalization constraints

$$\int q(\mathbf{x}_t, \mathbf{x}_{t-1}) d\mathbf{x}_{t-1} = q(\mathbf{x}_t) \quad (6.10)$$

$$\int q(\mathbf{z}_t, \mathbf{z}_{t-1}) d\mathbf{z}_{t-1} = q(\mathbf{z}_t). \quad (6.11)$$

Additionally, we impose moment-matching constraints on the parameters of the GCV as well as the control state as in Theorem 3.5 such that we can propagate exponential family messages in the graph. Minimization of the Bethe free energy given the specified constraints yield a hybrid SVMP algorithm with moment matching to approximate marginals that emerge from the product of non-conjugate messages computed by Theorem 3.5.

Fortunately, the messages and marginals required for inference in the proposed model are readily available for the GCV module from Table 4.1 of Chapter 4 and for the AR module from [96]. We tabulate the messages and marginals associated with the AR node of Figure 6.1 in Table 6.2 and refer the author to [96, Appendix] This is one of the most significant benefits of FFG formalism. Having derived the messages associated with the sub-modules, we are no longer obliged to derive additional messages. Plug and play nature of the FFG approach shows its benefits when composing complex models. Modularity allows us to build a complex hierarchical dynamical system by putting the smaller modules together. The availability of messages paves the way for an automated real-time inference. Because the messages are computed locally, the approximation of the intractable posterior integral (6.8) is performed distributively with local approximations.

Table 6.1: Marginals required in Table 6.2. For the derivation of marginals rules see [96, Appendix 7].

Marginals	Functional form
$q(\mathbf{u}_t, \mathbf{u}_{t-1})$	$\mathcal{N} \left(\begin{array}{c c c} \mathbf{u}_t & & \mathbf{m}_{u_t} \\ \hline \mathbf{u}_{t-1} & & \mathbf{m}_{u_{t-1}} \end{array}, \begin{array}{cc} \Sigma_{u_t} & \Sigma_{u_t u_{t-1}} \\ \Sigma_{u_{t-1} u_t} & \Sigma_{u_{t-1}} \end{array} \right)$
$q(\boldsymbol{\theta})$	$\mathcal{N}(\boldsymbol{\theta} \mathbf{m}_\theta, \Sigma_\theta)$
$q(\gamma)$	$\Gamma(\gamma \alpha_\gamma, \beta_\gamma)$

6.5 Experimental Verification

We will utilize the proposed model (6.7) to predict the future stock prices of Apple Inc. We use the prices between 10.11.2020 to 22.01.2021 to estimate the prices from 23.01.2021 to 17.06.2021. That means we create an FFG that has in total 150 time-slices. The first 100 time-slices of the FFG will have access to past prices, however the last 50 time-slices will not

Table 6.2: Summary of message computations for the AR node of Figure 6.1. The table makes the following definitions $\mathbf{A}_\theta \triangleq \begin{bmatrix} \mathbf{m}_\theta^\top & \\ \mathbf{I}_{n-1} & \mathbf{0} \end{bmatrix}$, $\mathbf{A}_\gamma \triangleq \begin{bmatrix} \frac{\alpha_\gamma}{\beta_\gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\mathbf{B}_t \triangleq \Sigma_{u_t} + \mathbf{m}_{u_t} \mathbf{m}_{u_t}^\top$ and $\bar{\alpha}_\gamma = 3/2$.

Messages	Functional form
①	$\mathcal{N}(u_{t-1} \bar{\mathbf{m}}_{u_{t-1}}, \bar{\Sigma}_{u_{t-1}})$
②	$\mathcal{N}(\theta \bar{\mathbf{m}}_\theta, \bar{\Sigma}_\theta)$
③	$\Gamma(\gamma \bar{\alpha}_\gamma, \bar{\beta}_\gamma)$
④	$\mathcal{N}(u_t \bar{\mathbf{m}}_{u_t}, \bar{\Sigma}_{u_t})$
⑤	$\mathcal{N}(u_t \bar{\mathbf{m}}_{u_t}, \bar{\Sigma}_{u_t})$
⑥	$\Gamma(\gamma \bar{\alpha}_\gamma, \bar{\beta}_\gamma)$
⑦	$\mathcal{N}(\theta \bar{\mathbf{m}}_\theta, \bar{\Sigma}_\theta)$
⑧	$\mathcal{N}(u_{t-1} \bar{\mathbf{m}}_{u_{t-1}}, \bar{\Sigma}_{u_{t-1}})$
Auxiliary	Definition by moment statistics
$\bar{\Sigma}_{u_t}$	$\mathbf{A}_\theta \left(\bar{\Sigma}_{u_{t-1}}^{-1} + \frac{\alpha_\gamma}{\beta_\gamma} \Sigma_\theta \right)^{-1} \mathbf{A}_\theta^\top + \mathbf{A}_\gamma$
$\bar{\mathbf{m}}_{u_t}$	$\mathbf{A}_\theta \left(\bar{\Sigma}_{u_{t-1}}^{-1} + \frac{\alpha_\gamma}{\beta_\gamma} \Sigma_\theta \right)^{-1} \bar{\Sigma}_{u_{t-1}}^{-1} \bar{\mathbf{m}}_{u_{t-1}}$
$\bar{\beta}_\gamma$	$\mathbf{c} \left(\mathbf{B}_t - 2\mathbf{A}_\theta \left(\Sigma_{u_{t-1}u_t} + \mathbf{m}_{u_t} \mathbf{m}_{u_t}^\top \right) + \mathbf{A}_\theta \mathbf{B}_{t-1} \mathbf{A}_\theta^\top + \text{tr}(\Sigma_\theta \mathbf{B}_{t-1}) \right) \mathbf{c}^\top$
$\bar{\Sigma}_\theta^{-1}$	$\frac{\alpha_\gamma}{\beta_\gamma} \left(\Sigma_{u_{t-1}} + \mathbf{m}_{u_{t-1}} \mathbf{m}_{u_{t-1}}^\top \right)$
$\bar{\mathbf{m}}_\theta$	$\bar{\Sigma}_\theta^{-1} \left(\Sigma_{u_t u_{t-1}} + \mathbf{m}_{u_{t-1}} \mathbf{m}_{u_t}^\top \right) \mathbf{c} \frac{\alpha_\gamma}{\beta_\gamma}$
$\bar{\Sigma}_{u_{t-1}}^{-1}$	$\mathbf{A}_\theta^\top \left(\bar{\Sigma}_{u_t} + \mathbf{A}_\gamma \right)^{-1} \mathbf{A}_\theta + \frac{\alpha_\gamma}{\beta_\gamma} \Sigma_\theta$
$\bar{\mathbf{m}}_{u_{t-1}}$	$\mathbf{A}_\theta^\top \left(\bar{\Sigma}_{u_t} + \mathbf{A}_\gamma \right)^{-1} \bar{\mathbf{m}}_{u_t}$

be terminated by observations. The FFG of the corresponding model will not be terminated. Since the FFG is not terminated for the last 50 days, we can not compute free energy in this section as the theorems we have proved in Chapter 3 are valid only if the FFG is terminated. We can however still use the message updates and obtain prediction for the not terminated edges. We utilize the model proposed in (6.7) and assign the following hyper-parameter values to the prior specifications in (6.3)

$$\begin{aligned} \mathbf{m}_\theta = \mathbf{0}, \mathbf{V}_\theta = \mathbf{I}, \mathbf{m}_\eta = \mathbf{0}, \mathbf{V}_\eta = \mathbf{I}, \alpha_\gamma = 1, \beta_\gamma = 1, \\ \alpha_\zeta = 1, \beta_\zeta = 1, m_\kappa = 1, v_\kappa = 0.01, m_\omega = 0.0, v_\omega = 10.0. \end{aligned}$$

We set AR orders of $M = N \in \{25, 35, 40, 45\}$.

For each time step, we iterate the full message-passing schedule (messages ① through ⑥4) until a maximum number of iterations is reached or the free energy corresponding to the subgraph of the FFG covering the observed prices converges. We emphasize that Bethe free energy corresponding to the FFG of the proposed model is not convex, and the message-passing iterations are not guaranteed to converge to local minima. Moreover, some characteristic polynomial roots, constructed with the estimated AR coefficients, may lie inside the unit circle due to high order AR specifications. If this occurs, the underlying AR process is not wide-sense stationary. We do not impose stationarity constraints as we do not know a priori whether the underlying dynamics of the stock prices are stationary or not. To assess the performance of the model on the prediction task, we use the average mean-squared error measure defined in (4.96) and compare the performance of the model with the varying number of AR orders.

Results of the prediction task are presented in Figure 6.3. Based on the average mean-squared error measure, the best performance is obtained with order 40, giving lower error estimates than the other orders. AR-HGF order 25 results in a much smoother estimation of past prices, while AR-HGF order 45 results in a much more rough estimation. For future prices, AR-HGF order 25 results in predictions that are not fluctuating nor capturing the trend, while AR-HGF order 45 overestimates the movement of prices. AR-HGF order 35 overly smoothes the prices before 22.01.2021, causing underestimation of future price patterns. It appears that AR-HGF order 40 finds a balance between overfitting past prices and underfitting future price trends. For comparison, we also present the prediction result with a standard AR process of order 40 with a Gaussian likelihood. We see that AR order 40 produces higher error estimates than the AR-HGF model with the same order. AR order 40 tends to follow the past prices more closely and therefore returns rougher estimates in comparison. AR order 40 predictions are better than the AR-HGF models that are not of the same order; however, compared to AR-HGF order 40, the variance of predictions grows faster. The presence of a GCV node, controlled with another AR process, ensures the time-varying variance of the prices is affected by the past values of the variance. Hence, the variance of an AR-HGF model's future predictions grows slower than standard AR. Moreover, the AR-HGF model is more robust to outliers because the AR process has a time-varying noise that can model more extreme price jumps. Hence, the AR-HGF model recovers smoother estimates than a

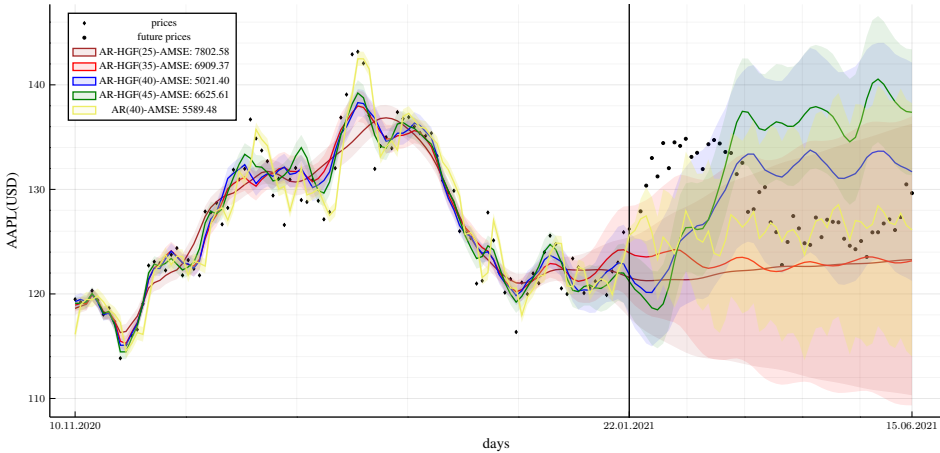


Figure 6.3: The figure plots the results of the prediction task. Diamond-shaped black dots indicate past prices, and circle-shaped dots indicate future prices that are to be predicted. Mean and standard deviation statistics of the smoothed past prices and predicted future prices are color-coded according to the AR order specified in the legend. An MSE based error for the predictions is presented in the legend as well.

standard AR model.

6.6 Discussion and Conclusions

We presented a message passing-based online inference method for joint state and parameter tracking in auto-regressive models with time-varying noise processes. The proposed AR-HGF model includes a dynamical prior on AR process noise variance, which can be extended hierarchically with variants of the HGF-like structures. With the combination of message update rules derived in Chapter 4 and [96], inference in the proposed model follows automatically.

An issue we have not addressed in our analysis is the determination of the order automatically. The order of an underlying AR process significantly affects the quality of the predictions. It is possible to assume a Dirichlet process prior to the order and obtain a posterior for the AR order. We plan to address this issue by extending ReactiveMP.jl with Dirichlet process structures in future research. Then, Dirichlet processes can be readily available to develop models with variable order, including variable order AR models with time-varying process noise.

We validated the proposed model on a T-step ahead prediction task and compared the results with an AR model with constant noise process variance. Due to its time-varying noise specification, the proposed model is more robust to overfitting and outliers than the standard AR model. The modularity of the FFG framework allowed us to create a complex

model such that inference is approximated automatically by message passing. Due to complex model specifications, the inference is considerably more challenging than a standard AR model. Nevertheless, we did not need to derive the inference equations by hand, thanks to the automated message passing framework.

CHAPTER 7

Discussion and Conclusions

"The future is uncertain. And the end is always near."

–The Doors

7.1 Contributions

This dissertation formulated a guideline framework for automated approximate message passing in hierarchical dynamical models by answering research questions **Q1-Q4**. In Chapter 3, we specified a constraint manipulation scheme to derive message-passing based algorithms on Forney-style factor graphs that paves the way for an efficient implementation of automated approximate inference. By changing the constraints on the local sub-graphs, we derived a variety of local message update rules as the stationary solutions of the constrained Bethe free energy from first principles (Section 3.4).

In Chapter 4, we demonstrated how message passing by constraint manipulation applies to hierarchical dynamical systems. We specifically focused on the hierarchical Gaussian filter, a time-series model for volatile processes where non-linear transforms couple the states in this process. We constructed a composite factor node (named GCV) representing the state transition distribution of the HGF that can be used as a plug-in module for any factor graph. We have derived various message update rules for the GCV node under multiple constraints. Combining derived update rules, we have implemented automated hybrid message passing for the variants of the HGF-like models in software packages ForneyLab.jl and ReactiveMP.jl. In summary, this chapter illustrated how message passing on factor graphs (as discussed in Chapter 3) applies to hierarchical dynamical models.

In Chapter 5, we extended the HGF model to account for context switches by augmenting it with a hidden Markov model governing a selection mechanism for the parameters of the

ordinary HGF model. We created a composite node (named GCSV) as a successor of GCV and derived closed-form message update rules for the GCSV node. We have verified the update equations for the SHGF model on a synthetic dataset and validated the SHGF model on stock prices.

In Chapter 6, we illustrated how the GCV node could be used as a plug-in module within the graphs of auto-regressive models to extend the auto-regressive models such that the deriving noise processes are time-varying. Message passing in the corresponding model leads to online state and parameter estimation in autoregressive models with time-varying process noise. We verified the proposed model and inference on a synthetic data set and validated on a financial modeling task. In the next section, we will detail the strengths and limitations of the contributions of this dissertation.

7.2 Strengths and Limitations

To address different dimensions of the research question **Q**, we decomposed **Q** into sub-parts and tried to address each question on its own. We posed the first question concerning understanding inference and finding approximations when exact solutions are not tractable. To that end, we presented the first research question:

Q1 How can approximate inference algorithms for probabilistic generative models be derived from first principles?

We formulated our answer to this question from the perspective of constrained minimization of the Bethe free energy. We showed how to obtain various local message updates by changing the Bethe free energy constraints. By combining these local updates, one can perform hybrid message passing. This dissertation advocates that a possible way to derive an approximate inference algorithm is via constraint manipulation. Celebrated algorithms such as BP, VMP, and EM were derived from specific constraints as local message updates instead of global algorithms. In Chapter 3, we explored a range of widespread constraints such as factorization constraints and form constraints and showed how these constraints blend in to derive approximate inference algorithms.

The main strengths of the advocated solution are the following:

- A modular way of generating algorithms by combining local updates
- Factorized computations of local updates for efficient implementation
- Performance evaluation for derived algorithms by computing Bethe free energy

This dissertation has not discussed universal inference algorithms based on sampling solutions. However, these algorithms can be derived from the minimization of constraint divergences. Message passing allows for efficient implementation of algorithms that do not require tabulated rules, and [148, 149] implemented these algorithms in `ForneyLab.jl` and `ReactiveMP.jl` toolboxes.

One of the limitations of the message passing framework is that if the underlying global function is not factorized, then the computations required for inference do not localize. Non-parametric models such as Gaussian and Dirichlet processes are inherently global specifications, and message passing can not straightforwardly leverage factorized computations. Although graphs can represent non-parametric models, they are not within the scope of message passing algorithms discussed in this dissertation and hence present a limitation to the current approach.

Due to the non-convexity of Bethe free energy, the minimization problem is not guaranteed to have a unique solution, and even if a solution exists, it might correspond to a local minimum. Sometimes, message passing iterations do not converge and enter limit cycles. Bethe free energy needs to be diagnosed for convergence. There are double-loop algorithms that can guarantee convergent algorithms [65]. However, we did not resort to these algorithms. Therefore, checking for convergence after a simulation ends is another limitation of the presented solution.

Q2 How can message passing algorithms for models of volatile systems be efficiently implemented?

In our attempt to answer **Q2**, we chose to work with a particular model called the hierarchical Gaussian filter, which is a non-linear hierarchical dynamical model. We have translated the HGF into the FFG framework by abstracting the state transitions to a composite factor node called the Gaussian with controlled variance. The GCV node represents a state transition distribution where the variance of state transitions is controlled by non-linearly transformed states that change over time. Time-dependent variance structure makes GCV capable of modeling volatile processes. We derived modular closed-form update rules for the GCV that blend with other local updates. We have added the proposed factor node and the rules in the software packages `ForneyLab.jl` and `ReactiveMP.jl` such that the GCV can be used as a plug-in factor for any factor graph. Specifically, the derived message update rules in these software packages allow GCV to be used in any combination of the following algorithm modes: Mean-field Variational Message Passing, Structured Variational Message Passing, Laplace Propagation, Expectation Propagation, and Expectation Maximization. Moreover, the message update rules for GCV support easy modifications such that they can be extended to include a different class of constraints.

Instead of giving a global analysis of how inference should be performed in a model of

volatile systems, we have isolated a structure responsible for volatility (GCV) and derived local functional relations among the hierarchically coupled states. The main merit of our solution to **Q2** is a readily available factor node (GCV) equipped with a variety of local message update rules corresponding to a different set of constraints and approximations.

Along with its strengths, the proposed solution to **Q2** comes with its limitations. A first limitation is the choice of non-linearity to couple hierarchical states in the GCV node. In section 4.5 we have motivated our choice for the exponential non-linearity and mentioned that this choice restricts the relation between hierarchically coupled states to first-order coupling functions in the GCV node. This limitation has consequences on the estimation of states for an HGF model. Information propagated from the lower layers to the upper layer goes through a non-linearity which is only accurate to first order. This limitation can be remedied by extending the GCV such that the exponential non-linearity is replaced by an arbitrary positive non-linearity that the designer can specify. In this case, message updates to higher layers are no longer closed-form solutions but rather messages that explicitly depend on the choice of non-linearity. Methods and approximations used in Chapter 4 would still apply and can be utilized for inference. Indeed, we have implemented this extension of the GCV in `ReactiveMP.jl`, and simulations with this implementation can be found in <https://github.com/biaslab/KernelMessagePassing>.

Another limitation is regarding the identification of states and parameters. In section 4.6, we assumed that states κ and z are uncorrelated and approximated their multiplication with a Gaussian distribution. Due to these assumptions, the message update rules for the states κ and z have symmetrical functional forms. A consequence of having symmetrical functional forms is that the estimates for the states κ and z might have swapped during inference if no further constraints are imposed. A remedy to this problem can be found by removing the limiting assumptions at the expense of computational complexity.

Q3 How can models of volatile environments be equipped with context switching dynamics?

We answered **Q3** by extending the GCV node with a switching mechanism to a new node named GCSV. In this extended version of the GCV, parameters are now vectors such that a categorical switching state selects an element of the parameter vector. We derived closed-form approximate update rules for GCSV and showed how GCSV could be used to construct a switching hierarchical Gaussian filter. The proposed solution for **Q3** inherits all the merits and limitations of the solution for **Q2**. In short, our answer to **Q3** produces a modular factor node (GCSV) equipped with a variety of local message update rules, which can be used in conjunction with numerous factor nodes in the literature.

A limitation of the proposed solution is determining the number of switches in an SHGF like model. The current formulation assumes that the number of switches is given. This

means that there is no automatic way to discover the number of switches in the given solution framework. We presented a free-energy-based comparison to determine a better model among possible candidates. In this comparison, the performance of a model with a particular number of switches is determined after inference is made and is benchmarked against an alternative model. This is not a principled approach. It is possible to assume a Dirichlet process prior to modeling the number of switches. Then, inference will automatically determine the number of switches in a more principled (but computationally more costly) way.

Q4 How can inference on auto-regressive models with time-varying noise processes be efficiently implemented?

Our answer to **Q4** illustrates the plug-and-play nature of the message passing approach to inference. We have used the GCV node as a driving noise process for auto-regressive models such that the innovation (or observation) noise can be time-varying. We have used the update rules derived in Chapter 4 in conjunction with the message update rules of [96] leading us to a hybrid inference scheme where states and parameters can be estimated jointly on autoregressive models with time-varying noise processes. The main strength of the proposed solution is that it is fully automated and robust to non-stationarities in the noise process when modeling auto-regressive processes.

We proposed the overarching research question **Q** that was formulated as:

Q *How can approximate inference and performance evaluation for hierarchical dynamical models be automated and efficiently implemented?*

. Altogether, the answers to the research questions **Q1-Q4** present a solution to **Q**. Collectively these answers formulate a guideline for automated approximate inference and performance evaluation for discrete-time hierarchical dynamical models. Our answer to **Q** is that: inference and performance evaluation can be automated and efficiently implemented by message passing on factor graphs for hierarchical dynamical models via locally approximated message update rules. A product of our answer to **Q** is a message-passing framework (software library) equipped with modular factor nodes with available message update rules to create hierarchical dynamical systems.

7.3 Outlook

This dissertation can be of interest to researchers in various ways. For researchers who are interested in deriving novel message passing algorithms, Chapter 3 gives a simple recipe that attempts to minimize the Bethe free energy by changing the constraint specification. A researcher might find the choice of Bethe free energy as a divergence measure to minimize as inconvenient for her application. Then she needs to apply the constraint manipulation recipe to minimize a divergence measure of her choice. For researchers who are interested in model development, Chapters 4,5 and 6 provide an extensive analysis of how to extend simple primitive building blocks to complex networks with hierarchical relations while retaining local message computations to perform inference. Perhaps the most significant impact of the dissertation can be on the practitioners of the HGF. This dissertation provides a flexible way to incorporate the HGF and variants of the HGF into the modeling of non-stationary processes.

APPENDIX A

Appendix

A.1 Free Energy Minimization by Variational Inference

In this section, we present a pedagogical example of inductive inference. After establishing an intuition, we apply the same principles to a more general context in the further sections. We follow Caticha [73, 150], who showed that a constrained free energy functional can be interpreted as a principled objective measure for inductive reasoning, see also [151, 152]. The calculus of variations offers a principled method for optimizing this free energy functional.

In this section we assume an example model

$$f(\mathbf{y}, \theta) = f_{\mathbf{y}}(\mathbf{y}, \theta) f_{\theta}(\theta), \quad (\text{A.1})$$

with observed variables \mathbf{y} and a single parameter θ .

We define the (variational) free energy (VFE) as

$$F[q, f] = \iint q(\mathbf{y}, \theta) \log \frac{q(\mathbf{y}, \theta)}{f(\mathbf{y}, \theta)} d\mathbf{y} d\theta. \quad (\text{A.2})$$

The goal is to find a posterior

$$q^* = \arg \min_{q \in \mathcal{Q}} F[q, f] \quad (\text{A.3})$$

that minimizes the free energy subject to some pre-specified constraints. These constraints may include form or factorization constraints on q (to be discussed later) or relate to observations of a signal \mathbf{y} .

As an example, assume that we obtained some measurements $\mathbf{y} = \hat{\mathbf{y}}$ and wish to obtain a posterior marginal belief $q^*(\theta)$ over the parameter. We can then incorporate the data in the form of a data constraint

$$\int q(\mathbf{y}, \theta) d\theta = \delta(\mathbf{y} - \hat{\mathbf{y}}), \quad (\text{A.4})$$

where δ defines a Dirac-delta. The *constrained* free energy can be rewritten by including Lagrange multipliers as

$$L[q, f] = F[q, f] + \gamma \left(\iint q(\mathbf{y}, \theta) d\mathbf{y} d\theta - 1 \right) + \int \lambda(\mathbf{y}) \left(\int q(\mathbf{y}, \theta) d\theta - \delta(\mathbf{y} - \hat{\mathbf{y}}) \right) d\mathbf{y}, \quad (\text{A.5})$$

where the first term specifies the (to be minimized) free energy objective, the second term a normalization constraint, and the third term the data constraint. Optimization of (A.5) can be performed using variational calculus.

Variational calculus considers the impact of a variation in $q(\mathbf{y}, \theta)$ on the Lagrangian $L[q, f]$. We define the variation as

$$\delta q(\mathbf{y}, \theta) \triangleq \epsilon \phi(\mathbf{y}, \theta),$$

where $\epsilon \rightarrow 0$, and ϕ is a continuous and differentiable “test” function. The fundamental theorem of variational calculus states that the stationary solutions q^* are obtained by setting $\delta L / \delta q = 0$, where the functional derivative $\delta L / \delta q$ is implicitly defined by [6, App. D]:

$$\left. \frac{dL[q + \epsilon \phi, f]}{d\epsilon} \right|_{\epsilon=0} = \iint \frac{\delta L}{\delta q}(\mathbf{y}, \theta) \phi(\mathbf{y}, \theta) d\mathbf{y} d\theta. \quad (\text{A.6})$$

Equation (A.6) provides a way to derive the functional derivative through ordinary differentiation. For example, we take the Lagrangian defined by (A.5) and work out the left hand side of (A.6):

$$\begin{aligned} \left. \frac{dL[q + \epsilon \phi, f]}{d\epsilon} \right|_{\epsilon=0} &= \left. \frac{dF[q + \epsilon \phi, f]}{d\epsilon} \right|_{\epsilon=0} + \left. \frac{d}{d\epsilon} \gamma \iint (q + \epsilon \phi) d\mathbf{y} d\theta \right|_{\epsilon=0} \\ &\quad + \left. \frac{d}{d\epsilon} \int \lambda(\mathbf{y}) \int (q + \epsilon \phi) d\theta d\mathbf{y} \right|_{\epsilon=0} \end{aligned} \quad (\text{A.7a})$$

$$\begin{aligned} &= \iint \left. \frac{d}{d\epsilon} \left((q + \epsilon \phi) \log \frac{(q + \epsilon \phi)}{f} \right) \right|_{\epsilon=0} d\mathbf{y} d\theta + \gamma \iint \left. \frac{d}{d\epsilon} (q + \epsilon \phi) \right|_{\epsilon=0} d\mathbf{y} d\theta \\ &\quad + \int \lambda(\mathbf{y}) \int \left. \frac{d}{d\epsilon} (q + \epsilon \phi) \right|_{\epsilon=0} d\theta d\mathbf{y} \end{aligned} \quad (\text{A.7b})$$

$$= \iint \underbrace{\left[\log \frac{q(\mathbf{y}, \theta)}{f(\mathbf{y}, \theta)} + 1 + \gamma + \lambda(\mathbf{y}) \right]}_{\delta L[q, f] / \delta q} \phi(\mathbf{y}, \theta) d\mathbf{y} d\theta. \quad (\text{A.7c})$$

Note that since (A.7c) has been written in similar form as (A.6), it is easy to identify the functional derivative. This procedure is one of many ways to obtain the functional derivatives [153].

Setting $\delta L[q, f]/\delta q = 0$ we find the stationary solution as

$$q^*(\mathbf{y}, \theta) = \exp(-1 - \gamma - \lambda(\mathbf{y})) f(\mathbf{y}, \theta) \quad (\text{A.8a})$$

$$= \frac{1}{Z} \exp(-\lambda(\mathbf{y})) f(\mathbf{y}, \theta), \quad (\text{A.8b})$$

with $Z = \iint \exp(-\lambda(\mathbf{y})) f(\mathbf{y}, \theta) d\mathbf{y} d\theta = \exp(\gamma + 1)$. In order to determine the Lagrange multipliers γ and $\lambda(\mathbf{y})$ we must substitute the stationary solution (A.8b) back into the constraints. The normalization constraint evaluates to

$$\frac{1}{Z} \iint \exp(-\lambda(\mathbf{y})) f(\mathbf{y}, \theta) d\mathbf{y} d\theta = 1. \quad (\text{A.9})$$

We find that (A.9) is always satisfied since $Z = \iint \exp(-\lambda(\mathbf{y})) f(\mathbf{y}, \theta) d\mathbf{y} d\theta$ by definition. Note however that the computation of the normalization constant still depends on the undetermined Lagrange multiplier $\lambda(\mathbf{y})$.

The data constraint evaluates to

$$\int q^*(\mathbf{y}, \theta) d\theta = \frac{1}{Z} \exp(-\lambda(\mathbf{y})) \int f(\mathbf{y}, \theta) d\theta = \delta(\mathbf{y} - \hat{\mathbf{y}}) \quad (\text{A.10})$$

which can be rewritten as

$$\frac{\exp(-\lambda(\mathbf{y}))}{Z} = \frac{\delta(\mathbf{y} - \hat{\mathbf{y}})}{\int f(\mathbf{y}, \theta) d\theta}. \quad (\text{A.11})$$

Equation (A.11) shows that $\lambda(\mathbf{y})$ can satisfy this constraint only if it is proportional to $\delta(\mathbf{y} - \hat{\mathbf{y}})$. Indeed, substitution of (A.11) into (A.8b) gives

$$q^*(\mathbf{y}, \theta) = \frac{f(\mathbf{y}, \theta)}{\int f(\mathbf{y}, \theta) d\theta} \delta(\mathbf{y} - \hat{\mathbf{y}}),$$

and the posterior for the parameters evaluates to

$$\begin{aligned} q^*(\theta) &= \int q^*(\mathbf{y}, \theta) d\mathbf{y} \\ &= \int \frac{f(\mathbf{y}, \theta)}{\int f(\mathbf{y}, \theta) d\theta} \delta(\mathbf{y} - \hat{\mathbf{y}}) d\mathbf{y} \\ &= \frac{f(\hat{\mathbf{y}}, \theta)}{\int f(\hat{\mathbf{y}}, \theta) d\theta} \\ &= \frac{f_{\mathbf{y}}(\hat{\mathbf{y}}, \theta) f_{\theta}(\theta)}{\int f_{\mathbf{y}}(\hat{\mathbf{y}}, \theta) f_{\theta}(\theta) d\theta}, \end{aligned}$$

which we recognize as Bayes rule.

The Bayes rule was derived here as a particular case of constrained variational free energy minimization when data constraints are present. This derivation of the Bayes rule seems unnecessarily tedious. Still, the value of this approach to inductive inference is that the same principle applies when other (not data) constraints on q are present.

A.2 Lagrangian optimization and the dual problem

With the addition of Lagrange multipliers to the Bethe functional, the resulting Lagrangian depends both on the variational distribution $q(\mathbf{s})$ and the Lagrange multipliers $\Psi(\mathbf{s})$. Formally, the introduction of the Lagrange multipliers allows us to rewrite the constrained optimization on the local polytope as an unconstrained optimization. We follow [65], and write

$$\min_{q \in \mathcal{L}(\mathcal{G})} F[q] = \min_q \max_{\Psi} L[q, \Psi].$$

Weak duality [154, Ch.5] then states that

$$\min_q \max_{\Psi} L[q, \Psi] \geq \max_{\Psi} \min_q L[q, \Psi].$$

The minimization with respect to q then yields a solution that depends on the Lagrange multipliers, as

$$q^*(\mathbf{s}; \Psi) = \arg \min_q L[q, \Psi].$$

For any given q the Lagrangian is concave in Ψ . Therefore, substituting q^* in the Lagrangian, the maximisation over $L[q^*, \Psi]$ yields the unique solution

$$\Psi^*(\mathbf{s}) = \arg \max_{\Psi} L[q^*, \Psi].$$

Stationary solutions are then given by

$$q^*(\mathbf{s}; \Psi^*) = \arg \min_{q \in \mathcal{L}(\mathcal{G})} F[q].$$

In the current paper we consider factorized q 's (e.g. (3.8)), and consider variations with respect to the individual factors. We then need to show that the combined stationary points of the individual factors also constitute a stationary point of the total objective.

Consider a Lagrangian having multiple arguments, i.e.,

$$L[\mathbf{q}] = L[q_1, \dots, q_n, \dots, q_N] \tag{A.12}$$

$$\mathbf{q} \triangleq [q_1, \dots, q_N]^\top. \tag{A.13}$$

We want to determine the first total variation of the Lagrangian given by

$$\delta L = L[\mathbf{q} + \epsilon \boldsymbol{\phi}] - L[\mathbf{q}] \tag{A.14}$$

$$\boldsymbol{\phi}(\mathbf{s}) \triangleq [\phi_1(\mathbf{s}), \dots, \phi_N(\mathbf{s})]^\top. \tag{A.15}$$

By a Taylor series expansion on ϵ we obtain [26, Equation 23.2] [153, A.14]

$$L[\mathbf{q} + \epsilon \boldsymbol{\phi}] - L[\mathbf{q}] = \sum_{k=1}^K \frac{1}{k!} \frac{d}{d\epsilon^k} (L^k[\mathbf{q} + \epsilon \boldsymbol{\phi}]) \epsilon^k + \mathcal{O}(\epsilon^{K+1}). \tag{A.16}$$

Omitting all terms higher than the first order, we obtain the first variation as

$$\delta L = \frac{d}{d\epsilon} (L[\mathbf{q} + \epsilon\phi]) \epsilon. \quad (\text{A.17})$$

Rearranging the terms and letting ϵ vanish, we obtain the following expression

$$\lim_{\epsilon \rightarrow 0} \frac{\delta L}{\epsilon} = \frac{d}{d\epsilon} (L[\mathbf{q} + \epsilon\phi]) \Big|_{\epsilon=0}. \quad (\text{A.18})$$

Let us assume that the Frechet derivative exists [153] such that we can obtain the following integral representation¹

$$\frac{d}{d\epsilon} (L[\mathbf{q} + \epsilon\phi]) \Big|_{\epsilon=0} = \int \phi(\mathbf{s})^\top \frac{\delta L}{\delta \mathbf{q}} d\mathbf{s} \quad (\text{A.19})$$

where $\frac{\delta L}{\delta \mathbf{q}}$ is the variational derivative

$$\frac{\delta L}{\delta \mathbf{q}} = \left[\frac{\delta L}{\delta q_1}, \dots, \frac{\delta L}{\delta q_N} \right]^\top \quad (\text{A.20})$$

$$\delta q_n = \epsilon \phi_n(\mathbf{s}). \quad (\text{A.21})$$

This means that (A.19) can be written as [26, Equation 22.5]²

$$\lim_{\epsilon \rightarrow 0} \frac{\delta L}{\epsilon} = \frac{d}{d\epsilon} (L[\mathbf{q} + \epsilon\phi]) \Big|_{\epsilon=0} = \sum_n \int \phi(\mathbf{s}) \frac{\delta L}{\delta q_n} d\mathbf{s}. \quad (\text{A.22})$$

Fundamental theorem of variational calculus states that in order for a point to be stationary, the first variation needs to vanish. In order for the first variation to vanish it is sufficient to have vanishing of the variational derivatives

$$\frac{\delta L}{\delta q_n} = 0 \text{ for every } n = 1, \dots, N. \quad (\text{A.23})$$

Vanishing of individual variational derivatives will mean that that the local stationary points will also correspond to a global stationary point.

A.3 Local free energy example for a deterministic node

Theorem 3.8 tells us how to evaluate the node-local free energy for a deterministic node. As an example, consider the node function $f_a(y, x) = \delta(y - \text{sgn}(x))$, with $y \in \{-1, 1\}$ and $x \in \mathbb{R}$ as depicted in Fig. A.1. Interestingly, there is information loss in this node

¹It should be noted that this integral expression is not always possible for a generic Lagrangian. That is why we need to assume that the Frechet derivative exists.

²Here we use a more generic Lagrangian and our notation is different than in [26], however the expression is motivated again by a Taylor series expansion on ϵ .

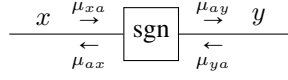


Figure A.1: Messages around a “sign” node.

because the “sign” mapping is not bijective. Given an incoming Bernoulli distributed message $\mu_{ya}(y) = \text{Ber}(y|p)$, the backward outgoing message is derived as

$$\begin{aligned} \mu_{ax}(x) &= \int \mu_{ya}(y) \delta(y - \text{sgn}(x)) dy \\ &= \begin{cases} p & \text{if } x \geq 0 \\ 1 - p & \text{if } x < 0. \end{cases} \end{aligned}$$

Given a Gaussian distributed incoming message $\mu_{xa}(x) = \mathcal{N}(x|m, \vartheta)$, the resulting belief then becomes

$$\begin{aligned} q_x(x) &= \frac{\mu_{xa}(x) \mu_{ax}(x)}{\int \mu_{xa}(x) \mu_{ax}(x) dx} \\ &= \begin{cases} \frac{p}{p+\Phi-2p\Phi} \mathcal{N}(x|m, \vartheta) & \text{if } x \geq 0 \\ \frac{1-p}{p+\Phi-2p\Phi} \mathcal{N}(x|m, \vartheta) & \text{if } x < 0, \end{cases} \end{aligned}$$

with $\Phi = \int_{-\infty}^0 \mathcal{N}(x|m, \vartheta) dx$. We define a truncated Gaussian distribution as

$$\mathcal{T}(x|m, \vartheta, a, b) = \begin{cases} \frac{1}{\Phi(a,b;m,\vartheta)} \mathcal{N}(x|m, \vartheta) & \text{if } a \leq x \leq b, \\ 0 & \text{otherwise,} \end{cases}$$

with $\Phi(a, b; m, \vartheta) = \int_a^b \mathcal{N}(x|m, \vartheta) dx$. This leads to

$$q_x(x) = \underbrace{\frac{p(1-\Phi)}{p+\Phi-2p\Phi}}_K \mathcal{T}(x|m, \vartheta, -\infty, 0) + \underbrace{\frac{(1-p)\Phi}{p+\Phi-2p\Phi}}_{1-K} \mathcal{T}(x|m, \vartheta, 0, \infty),$$

as a truncated Gaussian mixture.

The node-local free energy then evaluates to

$$\begin{aligned} F[q_a, f_a] &= -H[q_x] = \int_{-\infty}^0 q_x(x) \log q_x(x) dx + \int_0^{\infty} q_x(x) \log q_x(x) dx \\ &= -KH[\mathcal{T}(m, \vartheta, -\infty, 0)] + K \log K - (1-K)H[\mathcal{T}(m, \vartheta, 0, \infty)] + (1-K) \log(1-K) \\ &= -KH[\mathcal{T}(m, \vartheta, -\infty, 0)] - (1-K)H[\mathcal{T}(m, \vartheta, 0, \infty)] - H[\text{Ber}(K)], \end{aligned}$$

as a weighted sum of entropies.

A.4 Proofs

A.4.1 Proof of Lemma 3.1

Proof. We apply the variation $\epsilon\phi_b$ to q_b , and as discussed in Appendix A.1 we can identify the functional derivative $\delta L_b/\delta q_b$ through ordinary differentiation as

$$\left. \frac{dL_b[q_b + \epsilon\phi_b, f_b]}{d\epsilon} \right|_{\epsilon=0} = \int \left(\overbrace{\log \frac{q_b(\mathbf{s}_b)}{f_b(\mathbf{s}_b)} + 1 + \psi_b - \sum_{i \in \mathcal{E}(b)} \lambda_{ib}(s_i)}^{\delta L_b/\delta q_b} \right) \phi_b(\mathbf{s}_b) d\mathbf{s}_b.$$

Setting the functional derivative to zero and identifying

$$\mu_{ib}(s_i) = \exp(\lambda_{ib}(s_i)) \quad (\text{A.24})$$

$$\psi_b = \log \int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b - 1 \quad (\text{A.25})$$

yields the stationary solutions (3.18) in terms of Lagrange multipliers that are to be determined. \square

A.4.2 Proof of Lemma 3.2

Proof. We follow the same procedure as in Appendix A.4.1, where we apply a variation $\epsilon\phi_j$ to q_j (instead of q_b), and identify the functional derivative $\delta L_j/\delta q_j$ through

$$\left. \frac{dL_j[q_j + \epsilon\phi_j]}{d\epsilon} \right|_{\epsilon=0} = \int \left(\overbrace{-\log q_j(s_j) - 1 + \psi_j + \sum_{a \in \mathcal{V}(j)} \lambda_{ja}(s_j)}^{\delta L_j/\delta q_j} \right) \phi_j(s_j) ds_j.$$

Because the TFFG is terminated, each edge has degree 2 and the node-induced edge set has only 2 factors, which we denote by f_b and f_c . Then, setting the functional derivative to zero and identifying

$$\mu_{ja}(s_j) = \exp(\lambda_{ja}(s_j)) \quad (\text{A.26})$$

$$\psi_j = -\log \int \mu_{jb}(s_j) \mu_{jc}(s_j) ds_j + 1 \quad (\text{A.27})$$

yields the stationary solution of (3.20) in terms of the Lagrange multipliers. \square

A.4.3 Proof of Theorem 3.1

Proof. The local polytope of (3.14) constructs the Lagrangians of (3.17) and (3.19). Substituting the stationary solutions from Lemma 3.1 and 3.2 in the marginalization constraint,

$$q_j(s_j) = \int q_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus j},$$

we obtain the following relation

$$\frac{\mu_{jb}(s_j)\mu_{jc}(s_j)}{Z_j} = \frac{1}{Z_b} \int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_{b \setminus j},$$

where we defined the following normalization constants to ensure that the computed marginals are normalized:

$$Z_j = \int \mu_{jb}(s_j)\mu_{jc}(s_j) ds_j$$

$$Z_b = \int f_b(\mathbf{s}_b) \prod_{i \in \mathcal{E}(b)} \mu_{ib}(s_i) d\mathbf{s}_b.$$

Extracting μ_{jb} from the integral

$$\frac{\mu_{jb}(s_j)\mu_{jc}(s_j)}{Z_j} = \frac{\mu_{jb}(s_j)}{Z_b} \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}(s_i) d\mathbf{s}_{b \setminus j},$$

$$\mu_{jc}(s_j) = \frac{Z_j}{Z_b} \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}(s_i) d\mathbf{s}_{b \setminus j} \quad (\text{A.28})$$

and cancelling μ_{jb} on both sides then yields the condition on the functional form of the message μ_{jc} .

We now need to show that the fixed points of (3.25) satisfy (A.28). Let us assume that the fixed points exist, such that $\mu_{jc}^{(k)} = \mu_{jc}^{(k+1)}$ for some k . Then we want to show that at the fixed points the following equality holds:

$$\mu_{jc}^{(k)}(s_j) = \frac{Z_j^{(k)}}{Z_b^{(k)}} \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j}.$$

Substituting (3.25), we need to show that

$$\mu_{jc}^{(k)}(s_j) = \frac{Z_j^{(k)}}{Z_b^{(k)}} \mu_{jc}^{(k+1)}(s_j).$$

Since $\mu_{jc}^{(k)} = \mu_{jc}^{(k+1)}$, we can rearrange

$$\mu_{jc}^{(k)} \left(1 - \frac{Z_j^{(k)}}{Z_b^{(k)}} \right) = 0.$$

From Z_b , we obtain

$$\begin{aligned} Z_b^{(k)} &= \int \mu_{jb}^{(k)}(s_j) \int f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j} ds_j \\ &= \int \mu_{jb}^{(k)}(s_j) \mu_{jc}^{(k+1)}(s_j) ds_j \\ &= \int \mu_{jb}^{(k)}(s_j) \mu_{jc}^{(k)}(s_j) ds_j \\ &= Z_j^{(k)}, \end{aligned}$$

which implies that the fixed points satisfy the desired condition. This proves that the stationary solutions to the BFE within the local polytope can be obtained as fixed points of the sum-product update equations. \square

A.4.4 Proof of Lemma 3.3

Proof. Substituting the definition of (3.32), we can re-write the second term of Lagrangian (3.30) as

$$\begin{aligned} \int \left\{ \prod_{n \in l(b)} q_b^m(\mathbf{s}_b^m) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b &= \int q_b^m(\mathbf{s}_b^m) \left(\int \left\{ \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^m(\mathbf{s}_b^m) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b^{\setminus m} \right) d\mathbf{s}_b^m \\ &= \int q_b^m(\mathbf{s}_b^m) \log \tilde{f}_b^m(\mathbf{s}_b^m) d\mathbf{s}_b^m. \end{aligned}$$

We apply the variation $\epsilon \phi_b^m$ to q_b^m , and identify the functional derivative $\delta L_b^m / \delta q_b^m$, as

$$\frac{dL_b^m[q_b^m + \epsilon \phi_b^m]}{d\epsilon} \Big|_{\epsilon=0} = \int \left(\overbrace{\log \frac{q_b^m(\mathbf{s}_b^m)}{\tilde{f}_b^m(\mathbf{s}_b^m)} + 1 + \psi_b^m - \sum_{i \in m} \lambda_{ib}(s_i)}^{\delta L_b^m / \delta q_b^m} \right) \phi_b^m(\mathbf{s}_b^m) d\mathbf{s}_b^m,$$

whose functional form we recognize from Appendix A.4.1. Setting the functional derivative to zero and again identifying $\mu_{ib}(s_i) = \exp \lambda_{ib}(s_i)$, yields the stationary solutions of (3.31). \square

A.4.5 Proof of Theorem 3.2

Proof. The local polytope of (3.33) constructs the Lagrangians L_b^m and L_j as (3.30) and (3.19) respectively. We substitute the stationary solutions of Lemma (3.3) and (3.2) in the local marginalization constraint (3.29b), which yields

$$q_j(s_j) = \int q_b^m(\mathbf{s}_b^m) d\mathbf{s}_{b \setminus j}^m.$$

Following the structure of the proof in Appendix A.4.3, we obtain the following condition for the stationary solutions in terms of messages:

$$\begin{aligned} \frac{\mu_{jb}(s_j)\mu_{jc}(s_j)}{Z_j} &= \frac{\mu_{jb}(s_j)}{Z_b^m} \int \tilde{f}_b^m(\mathbf{s}_b^m) \prod_{\substack{i \in m \\ i \neq j}} \mu_{ib}(s_i) d\mathbf{s}_{b \setminus j}^m \\ \frac{\mu_{jc}(s_j)}{Z_j} &= \frac{1}{Z_b^m} \int \tilde{f}_b^m(\mathbf{s}_b^m) \prod_{\substack{i \in m \\ i \neq j}} \mu_{ib}(s_i) d\mathbf{s}_{b \setminus j}^m. \end{aligned} \quad (\text{A.29})$$

Now we want to show that the fixed points of the message updates (3.36) satisfy (A.29). Let us assume that the fixed points exists for some k such that $\mu_{jc}^{(k+1)} = \mu_{jc}^{(k)}$. Then we will show that the fixed points satisfy

$$\frac{\mu_{jc}^{(k)}(s_j)}{Z_j^{(k)}} = \frac{1}{Z_b^{m,(k)}} \int \tilde{f}_b^{m,(k)}(\mathbf{s}_b^m) \prod_{\substack{i \in m \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j}^m. \quad (\text{A.30})$$

Similar to Appendix A.4.3, it will suffice to show that $Z_b^{m,(k)} = Z_j^{(k)}$ at the fixed points. Arranging the order of integration in normalization constant $Z_b^{m,(k)}$, we obtain

$$\begin{aligned} Z_b^{m,(k)} &= \int \mu_{jb}^{(k)}(s_j) \int \tilde{f}_b^{m,(k)}(\mathbf{s}_b^m) \prod_{\substack{i \in m \\ i \neq j}} \mu_{ib}^{(k)}(s_i) d\mathbf{s}_{b \setminus j}^m ds_j \\ &= \int \mu_{jb}^{(k)}(s_j) \mu_{jc}^{(k+1)}(s_j) ds_j \\ &= \int \mu_{jb}^{(k)}(s_j) \mu_{jc}^{(k)}(s_j) ds_j \\ &= Z_j^{(k)}. \end{aligned}$$

By the same line of reasoning as in Appendix A.4.3, this shows that the fixed points of the message updates (3.36) leads to stationary distributions of the Bethe free energy with structured factorization constraints. \square

A.4.6 Proof of Corollary 3.1

Proof. For a fully factorized local variational distribution (3.41), the augmented node function $\tilde{f}_b^m(\mathbf{s}_b^m)$ of (3.32) reduces to

$$\tilde{f}_j(s_j) = \exp \left(\int \left\{ \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} q_i(s_i) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_{b \setminus j} \right). \quad (\text{A.31})$$

The message of (3.36) then reduces to

$$\mu_{jc}(s_j) = \tilde{f}_j(s_j),$$

which after substitution recovers (3.43). \square

A.4.7 Proof of Lemma 3.4

Proof. When we apply the variation $\epsilon \phi_b$ to q_b and identify the functional derivative $\delta L_b / \delta q_b$, we recover the result from Appendix A.4.1, which leads to a solution of the form (3.47). \square

A.4.8 Proof of Theorem 3.3

Proof. We construct the Lagrangian of (3.46), which by Lemma 3.4 leads to a solution of the form (3.47). Substituting this solution in the constraint of (3.45) leads to

$$\left[\int f_b(\mathbf{s}_b) \overbrace{\prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}(s_i)}^{\mu_{bj}(s_j)} d\mathbf{s}_{b \setminus j} \right] \mu_{jb}(s_j) = \delta(s_j - \hat{s}_j). \quad (\text{A.32})$$

This equation is then satisfied by (3.50), which proves the theorem. \square

A.4.9 Proof of Lemma 3.5

Proof. The proof follows directly from Appendix A.4.1, with $\tilde{f}_b(\mathbf{s}_b; \hat{\mathbf{s}}_b)$ substituted for $f_b(\mathbf{s}_b)$. \square

A.4.10 Proof of Theorem 3.4

Proof. Given the result of Lemma 3.5, the proof follows Appendix A.4.3, where Laplace propagation chooses the expansion point to be the fixed point $\hat{\mathbf{s}}_b = \arg \max \log q_b(\mathbf{s}_b)$.

For all second-order fixed points of the Laplace iterations, it holds that $\hat{\mathbf{s}}_b$ is a fixed point if and only if it is a local optimum of q_b . The proof is then concluded by Lemma 1 in [155]. \square

A.4.11 Proof of Lemma 3.6

Proof. We note that the Lagrange multiplier η_{jb} does not depend on s_j because the expectation removes all the functional dependencies on s_j . Furthermore, the expectations of $T_j(s_j)$ have the same dimension as the function $T_j(s_j)$. This means that the dimension of η_{jb} needs to be compatible with that of $T_j(s_j)$ so that we can write the constraint as an inner product.

We apply the variation $\epsilon\phi_b$ to q_b , and identify the functional derivative $\delta L_b/\delta q_b$, as

$$\left. \frac{dL_b[q_b + \epsilon\phi_b, f_b]}{d\epsilon} \right|_{\epsilon=0} = \int \left(\overbrace{\log \frac{q_b(\mathbf{s}_b)}{f_b(\mathbf{s}_b)} + 1 + \psi_b - \sum_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \lambda_{ib}(s_i) - \eta_{jb}^\top T_j(s_j)}^{\delta L_b/\delta q_b} \right) \phi_b(\mathbf{s}_b) d\mathbf{s}_b.$$

Setting the functional derivative to zero and identifying $\mu_{ib}(s_i) = \exp \lambda_{ib}(s_i)$ for $i \neq j$ and identifying $\mu_{jb}(s_j) = \exp(\eta_{jb}^\top T_j(s_j))$ yields the functional form of the stationary solution as (3.62). \square

A.4.12 Proof of Lemma 3.7

Proof. We follow a similar procedure as in Appendix A.4.11 and apply the variation $\epsilon\phi_j$ to q_j , which identifies the functional derivative $\delta L_j/\delta q_j$, as

$$\left. \frac{dL[q_j + \epsilon\phi_j]}{d\epsilon} \right|_{\epsilon=0} = \int \left(\overbrace{-\log q_j(s_j) - 1 + \psi_j + \sum_{a \in \mathcal{V}(j)} \eta_{ja}^\top T_j(s_j)}^{\delta L_j/\delta q_j} \right) \phi_j(s_j) ds_j.$$

Setting the functional derivative to zero and following the same procedure as in Appendix A.4.2, yields (3.64). \square

A.4.13 Proof of Theorem 3.5

Proof. Substituting the stationary solutions given by Lemma 3.6 and 3.7 into the moment matching constraint (3.59), we obtain the following condition:

$$\begin{aligned} \frac{1}{\bar{Z}_j} \int T_j(s_j) \exp([\eta_{jb} + \eta_{jc}]^\top T_j(s_j)) ds_j &= \\ \frac{1}{\bar{Z}_j} \int T_j(s_j) \overbrace{\exp(\eta_{jb}^\top T_j(s_j))}^{\mu_{jb}(s_j)} \left[\int \overbrace{f_b(\mathbf{s}_b) \prod_{\substack{i \in \mathcal{E}(b) \\ i \neq j}} \mu_{ib}(s_i)}^{\tilde{\mu}_{jc}(s_j)} d\mathbf{s}_{b \setminus j} \right] ds_j &= \\ = \int T_j(s_j) \tilde{q}_j(s_j) ds_j, & \end{aligned}$$

where we recognize the sum-product message $\tilde{\mu}_{jc}(s_j)$, which we multiply by the incoming exponential family message $\mu_{jb}(s_j)$ and normalize to obtain $\tilde{q}_j(s_j)$. Defining $\eta_j = \eta_{jb} + \eta_{jc}$, normalization constants are given by

$$Z_j(\eta_j) = \int \exp(\eta_j^\top T_j(s_j)) \, ds_j$$

$$\tilde{Z}_j = \int \exp(\eta_{jb}^\top T_j(s_j)) \tilde{\mu}_{jc}(s_j) \, ds_j.$$

Computing the moments allows us to determine the exponential family parameter by solving the following equation [37, Proposition 3.1]

$$\nabla_{\eta_j} \log Z_j(\eta_j) = \int \tilde{q}_j(s_j) T_j(s_j) \, ds_j.$$

Suppose you obtain a solution to this equation denoted by $\tilde{\eta}_j$ then this allows us to approximate the sum-product message $\tilde{\mu}_{jc}(s_j)$ by an exponential family message whose parameter is given by

$$\eta_{jc} = \tilde{\eta}_j - \eta_{jb}.$$

Now let us assume that the fixed points of the sum-product iterations $\tilde{\mu}_{jc}^{(k)}(s_j) = \tilde{\mu}_{jc}^{(k+1)}(s_j)$ and the incoming exponential family messages $\mu_{jb}^{(k)}(s_j) = \mu_{jb}^{(k+1)}(s_j)$ exist for some k . Then, we need to show that the existence of these fixed points implies the existence of the fixed points of $\mu_{jc}^{(k+1)} = \mu_{jc}^{(k)}$.

By moment-matching, we have

$$\begin{aligned} \eta_{jc}^{(k+1)} &= \tilde{\eta}_j^{(k+1)} - \eta_{jb}^{(k+1)} \\ &= \tilde{\eta}_j^{(k)} - \eta_{jb}^{(k)} \\ &= \eta_{jc}^{(k)}, \end{aligned}$$

which proves the existence of the fixed point of μ_{jc} if $\tilde{\mu}_{jc}$ and $\mu_{jb}(s_j)$ have fixed points. \square

A.4.14 Proof of Theorem 3.6

Proof. The proof follows directly from substituting the Laplace-approximated factor-function (3.53) in the naive mean-field result of Corollary. 3.1. \square

A.4.15 Proof of Theorem 3.7

Proof. In order to obtain the optimal parameter value θ_j^* , we view the free energy as a function of θ_j . Because there are two node-local free energies that depend upon θ_j , this leads

to

$$\begin{aligned}
\theta_j^* &= \arg \min_{\theta_j} \left(F[q_b, f_b; \theta_j] + F[q_c, f_c; \theta_j] \right) \\
&= \arg \max_{\theta_j} \left(\int \left\{ \delta(s_j - \theta_j) \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^n(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_b) d\mathbf{s}_b \right. \\
&\quad \left. + \int \left\{ \delta(s_j - \theta_j) \prod_{\substack{n \in l(c) \\ n \neq m}} q_c^n(\mathbf{s}_c^n) \right\} \log f_c(\mathbf{s}_c) d\mathbf{s}_c \right) \\
&= \arg \max_{\theta_j} \left(\int \left\{ \prod_{\substack{n \in l(b) \\ n \neq m}} q_b^n(\mathbf{s}_b^n) \right\} \log f_b(\mathbf{s}_{b \setminus j}, \theta_j) d\mathbf{s}_{b \setminus j} \right. \\
&\quad \left. + \int \left\{ \prod_{\substack{n \in l(c) \\ n \neq m}} q_c^n(\mathbf{s}_c^n) \right\} \log f_c(\mathbf{s}_{c \setminus j}, \theta_j) d\mathbf{s}_{c \setminus j} \right) \\
&= \arg \max_{s_j} \left(\log \mu_{bj}(s_j) + \log \mu_{cj}(s_j) \right),
\end{aligned}$$

where in the last step we replaced θ_j by s_j for convenience. Here we recognize μ_{bj} and μ_{cj} as the structured variational updates of Theorem 3.2. Identification of the fixed points can then be obtained by [87, Corollary 2]. For a rigorous discussion on convergence of the EM algorithm we refer to [156, Corollary 32], [37, Chapter 6] and [87, Section 3]. \square

A.4.16 Proof of Theorem 3.8

Proof. Substituting for $q_a(\mathbf{s}_a)$, the node-local free energy becomes

$$\begin{aligned}
F[q_a, f_a] &= \int q_a(\mathbf{s}_a) \log \frac{q_{j|a}(s_j | \mathbf{s}_{a \setminus j})}{f_a(\mathbf{s}_a)} d\mathbf{s}_a + \int q_a(\mathbf{s}_a) \log q_{a \setminus j}(\mathbf{s}_{a \setminus j}) d\mathbf{s}_a \\
&= \int q_{a \setminus j}(\mathbf{s}_{a \setminus j}) q_{j|a}(s_j | \mathbf{s}_{a \setminus j}) \log \frac{q_{j|a}(s_j | \mathbf{s}_{a \setminus j})}{f_a(\mathbf{s}_a)} d\mathbf{s}_a \\
&\quad + \int q_{a \setminus j}(\mathbf{s}_{a \setminus j}) q_{j|a}(s_j | \mathbf{s}_{a \setminus j}) \log q_{a \setminus j}(\mathbf{s}_{a \setminus j}) d\mathbf{s}_a \\
&= \int q_{a \setminus j}(\mathbf{s}_{a \setminus j}) \left[\int q_{j|a}(s_j | \mathbf{s}_{a \setminus j}) \log \frac{q_{j|a}(s_j | \mathbf{s}_{a \setminus j})}{f_a(\mathbf{s}_a)} ds_j \right] d\mathbf{s}_{a \setminus j} \\
&\quad + \int q_{a \setminus j}(\mathbf{s}_{a \setminus j}) \log q_{a \setminus j}(\mathbf{s}_{a \setminus j}) d\mathbf{s}_{a \setminus j} \\
&= \mathbb{E}_{q_{a \setminus j}} [D[q_{j|a} \| f_a]] - H[q_{a \setminus j}],
\end{aligned}$$

where the first term expresses an expected Kullback-Leibler divergence, and the second term a negative entropy. The only possibility for the local free energy to become finite, is when $q_{j|a}(s_j|\mathbf{s}_{a\setminus j}) = f_a(\mathbf{s}_a) = \delta(s_j - g_a(\mathbf{s}_{a\setminus j}))$. We then have:

$$F[q_a, f_a] = \begin{cases} -H[q_{a\setminus j}] & \text{if } q_{j|a}(s_j|\mathbf{s}_{a\setminus j}) = \delta(s_j - g_a(\mathbf{s}_{a\setminus j})) \\ \infty & \text{otherwise.} \end{cases}$$

□

A.4.17 Proof of Theorem 3.9

Proof. The proof is similar to Appendix A.4.16. Substituting for $q_a(\mathbf{s}_a)$, the node-local free energy becomes

$$\begin{aligned} F[q_a, f_a] &= \int q_a(\mathbf{s}_a) \log \frac{q_a(\mathbf{s}_a)}{f_a(\mathbf{s}_a)} d\mathbf{s}_a \\ &= \int q_a(s_i, s_j, s_k) \log \frac{q_{ik|j}(s_i, s_k|s_j)}{f_a(s_i, s_j, s_k)} ds_i ds_j ds_k + \int q_j(s_j) \log q_j(s_j) ds_j \\ &= \mathbb{E}_{q_j} [D[q_{ik|j} \| f_a]] - H[q_j]. \end{aligned}$$

In contrast to Appendix A.4.16, here we have a joint belief within the divergence with a single conditioning variable. Conditioning on s_j (or by symmetry s_i or s_k), already determines the realization of the other variables. Therefore, we have:

$$F[q_a, f_a] = \begin{cases} -H[q_j] & \text{if } q_{ik|j}(s_i, s_k|s_j) = \delta(s_j - s_i) \delta(s_j - s_k) \\ \infty & \text{otherwise.} \end{cases}$$

□

References

- [1] P. Lamkin, “Wearable Tech Market To Double By 2021,” forbes, <https://www.forbes.com/sites/paullamkin/2017/06/22/wearable-tech-market-to-double-by-2021/>, last accessed on 3-4-2019. [Online]. Available: <https://www.forbes.com/sites/paullamkin/2017/06/22/wearable-tech-market-to-double-by-2021/>
- [2] R. C. Grinold, “The fundamental law of active management,” *The Journal of Portfolio Management*, vol. 15, no. 3, pp. 30–37, 1989. [Online]. Available: <https://jpm.pm-research.com/content/15/3/30>
- [3] S. Jansen, *Machine learning for algorithmic trading predictive models to extract signals from market and alternative data for systematic trading strategies with Python, second edition*, 2nd ed., 2020, ISBN: 9781839216787 1839216786 9781839217715 1839217715.
- [4] W. F. Sharpe, “Mutual Fund Performance,” *The Journal of Business*, vol. 39, no. 1., pp. 119–138, 1966. [Online]. Available: <http://www.jstor.org/stable/2351741>
- [5] R. C. Conant and W. R. Ashby, “Every good regulator of a system must be a model of that system,” *Intl. J. Systems Science*, pp. 89–97, 1970.
- [6] 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>
- [7] E. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003. [Online]. Available: <http://bayes.wustl.edu/etj/prob/book.pdf>
- [8] G. F. Cooper, “The computational complexity of probabilistic inference using bayesian belief networks,” *Artificial Intelligence*, vol. 42, no. 2, pp. 393–405, 1990. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/000437029090060D>
- [9] D. Rudoy, T. F. Quatieri, and P. J. Wolfe, “Time-Varying Autoregressions in Speech: Detection Theory and Applications,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 977–989, May 2011, arXiv: 0911.1697. [Online]. Available: <http://arxiv.org/abs/0911.1697>
- [10] C. Berninger, A. Stöcker, and D. Rügamer, “A Bayesian Time-Varying Autoregressive Model for Improved Short- and Long-Term Prediction,” *arXiv:2006.05750 [q-fin, stat]*, Jun. 2020, arXiv: 2006.05750. [Online]. Available: <http://arxiv.org/abs/2006.05750>
- [11] M. Cassidy and W. Penny, “Bayesian nonstationary autoregressive models for biomedical signal analysis,” *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 10, pp. 1142–1152, Oct. 2002.
- [12] O. Jakubov, P. Kovar, P. Kacmarik *et al.*, “Distributed Extended Kalman Filter for Position, Velocity, Time Estimation in Satellite Navigation Receivers.” *Radioengineering*, vol. 22,

- no. 3, 2013. [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=12102512&AN=90458159&h=JgukY%2BIJhky2wXeAWvLTsdi4C%2FEB4OCWsFMjKB7hfSnxGBSoaUzib1m0C863f50zrQrwN7qmSJJdt4T3qksAA%3D%3D&crl=c>
- [13] F. Deng, C. Bao, and W. Kleijn, “Sparse Hidden Markov Models for Speech Enhancement in Non-Stationary Noise Environments,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1973–1987, Nov. 2015.
- [14] C. M. Carvalho and H. F. Lopes, “Simulation-based sequential analysis of Markov switching stochastic volatility models,” *Computational Statistics & Data Analysis*, vol. 51, no. 9, pp. 4526–4542, May 2007. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167947306002349>
- [15] C. J. Darwin and R. P. Carlyon, “Chapter 11 - Auditory Grouping,” in *Hearing*, ser. Handbook of Perception and Cognition, B. C. J. Moore, Ed. San Diego: Academic Press, 1995, pp. 387–424. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780125056267500133>
- [16] A. S. Bregman and others, *Auditory scene analysis*. Cambridge, ma: mit press, 1990, vol. 10. [Online]. Available: http://webpages.mcgill.ca/staff/Group2/abregm1/web/pdf/2004_%20Encyclopedia-Soc-Behav-Sci.pdf
- [17] H. Attias and C. E. Schreiner, “Temporal low-order statistics of natural sounds,” in *Advances in neural information processing systems*, 1997, pp. 27–33. [Online]. Available: <http://papers.nips.cc/paper/1262-temporal-low-order-statistics-of-natural-sounds.pdf>
- [18] J. H. McDermott and E. P. Simoncelli, “Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis,” *Neuron*, vol. 71, no. 5, pp. 926–940, Sep. 2011. [Online]. Available: [http://www.cell.com/neuron/abstract/S0896-6273\(11\)00562-9](http://www.cell.com/neuron/abstract/S0896-6273(11)00562-9)
- [19] J. Pearl, “Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach,” in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, ser. AAAI’82. Pittsburgh, Pennsylvania: AAAI Press, 1982, pp. 133–136. [Online]. Available: <http://www.aaai.org/Papers/AAAI/1982/AAAI82-032.pdf>
- [20] T. Dean, “Scalable Inference in Hierarchical Generative Models.” in *ISAIM*. Citeseer, 2006.
- [21] J. W. Tukey, “The future of data analysis,” *The annals of mathematical statistics*, vol. 33, no. 1, pp. 1–67, 1962.
- [22] J.-T. Chien and P.-K. Yang, “Bayesian Factorization and Learning for Monaural Source Separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 185–195, Jan. 2016.
- [23] M. Betancourt, “A Conceptual Introduction to Hamiltonian Monte Carlo,” *arXiv:1701.02434 [stat]*, Jul. 2018, arXiv: 1701.02434. [Online]. Available: <http://arxiv.org/abs/1701.02434>
- [24] M. J. Beal, “Variational algorithms for approximate Bayesian inference,” Ph.D. dissertation, University of London, 2003. [Online]. Available: <http://www.cse.buffalo.edu/faculty/mbeal/papers/beal03.pdf>
- [25] C. Zhang, J. Butepage, H. Kjellstrom *et al.*, “Advances in Variational Inference,” *arXiv:1711.05597 [cs, stat]*, Nov. 2017, arXiv: 1711.05597. [Online]. Available: <http://arxiv.org/abs/1711.05597>
- [26] C. Lanczos, *The variational principles of mechanics*. Courier Corporation, 2012.
- [27] M. I. Jordan and T. J. Sejnowski, Eds., *Graphical models: foundations of neural computation*, ser. Computational neuroscience. Cambridge, Mass: MIT Press, 2001.
- [28] G. Forney, “Codes on graphs: normal realizations,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001, conference Name: IEEE Transactions on Information Theory. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/910573>
- [29] H. Ge, K. Xu, and Z. Ghahramani, “Turing: A Language for Flexible Probabilistic Inference,” in *International Conference on Artificial Intelligence and Statistics*, Mar. 2018, pp. 1682–1690, iSSN: 1938-7228 Section: Machine Learning. [Online]. Available: <http://proceedings.mlr.press/v84/ge18b.html>

- [30] E. Bingham, J. P. Chen, M. Jankowiak *et al.*, “Pyro: Deep Universal Probabilistic Programming,” *Journal of Machine Learning Research*, vol. 20, no. 28, pp. 1–6, 2019. [Online]. Available: <http://jmlr.org/papers/v20/18-403.html>
- [31] J. V. Dillon, I. Langmore, D. Tran *et al.*, “TensorFlow Distributions,” *arXiv:1711.10604 [cs, stat]*, Nov. 2017, arXiv: 1711.10604. [Online]. Available: <http://arxiv.org/abs/1711.10604>
- [32] T. Minka, J. Winn, J. Guiver *et al.*, “Infer.NET 2.6, <http://research.microsoft.com/infernet>,” 2014. [Online]. Available: <http://research.microsoft.com/infernet>
- [33] M. Cox, T. van de Laar, and B. de Vries, “A factor graph approach to automated design of Bayesian signal processing algorithms,” *International Journal of Approximate Reasoning*, vol. 104, pp. 185–204, Jan. 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0888613X18304298>
- [34] D. Bagaev, “Reactive Message Passing for Scalable Bayesian Inference,” p. 40, 2021.
- [35] T. Minka, “Divergence Measures and Message Passing,” Tech. Rep., 2005.
- [36] C. D. Mathys, “Hierarchical Gaussian filtering,” Ph.D. dissertation, Diss., Eidgenössische Technische Hochschule ETH Zuerich, Nr. 20909, 2012. [Online]. Available: <http://e-collection.library.ethz.ch/view/eth:6419>
- [37] M. J. Wainwright and M. I. Jordan, “Graphical Models, Exponential Families, and Variational Inference,” *Foundations and Trends in Machine Learning*, vol. 1, no. 12, pp. 1–305, Nov. 2008. [Online]. Available: <https://www.nowpublishers.com/article/Details/MAL-001>
- [38] D. Barber, T. Cemgil, and S. Chiappa, *Bayesian Time Series Models*. Cambridge University Press, 2011.
- [39] 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>
- [40] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available: <https://www.jstor.org/stable/2236703>
- [41] H.-A. Loeliger, “An introduction to factor graphs,” *Signal Processing Magazine, IEEE*, vol. 21, no. 1, pp. 28–41, 2004. [Online]. Available: <https://ieeexplore.ieee.org/document/1267047>
- [42] J. Winn and C. M. Bishop, “Variational message passing,” *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 661–694, 2005. [Online]. Available: <http://www.jmlr.org/papers/volume6/winn05a/winn05a.pdf>
- [43] J. S. Yedidia, “Understanding Belief Propagation and its Generalizations,” Nov. 2001.
- [44] —, “An Idiosyncratic Journey Beyond Mean Field Theory,” in *Advanced Mean Field Methods*, 2000, pp. 37–49. [Online]. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6281795>
- [45] J. Dauwels, “On Variational Message Passing on Factor Graphs,” in *IEEE International Symposium on Information Theory*, Jun. 2007, pp. 2546–2550. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/4557602>
- [46] D. Zhang, W. Wang, G. Fettweis *et al.*, “Unifying Message Passing Algorithms Under the Framework of Constrained Bethe Free Energy Minimization,” *arXiv:1703.10932 [cs, math]*, Mar. 2017, arXiv: 1703.10932. [Online]. Available: <http://arxiv.org/abs/1703.10932>
- [47] T. van de Laar, I. enöz, A. Özçelikkale *et al.*, “Chance-Constrained Active Inference,” *arXiv preprint arXiv:2102.08792*, 2021.
- [48] A. J. Smola, S. V. N. Vishwanathan, and E. Eskin, “Laplace propagation,” in *NIPS*, 2004, pp. 441–448.
- [49] J. S. Yedidia, “Generalized belief propagation and free energy minimization,” 2002.
- [50] J. S. Yedidia, W. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/1459044>

- [51] T. P. Minka, "Expectation Propagation for Approximate Bayesian Inference," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 362–369. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2074022.2074067>
- [52] T. Heskes, "Stable fixed points of loopy belief propagation are local minima of the bethe free energy," in *Advances in neural information processing systems*, 2003, pp. 359–366.
- [53] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=910572
- [54] M. Hoffman, D. M. Blei, C. Wang *et al.*, "Stochastic Variational Inference," *arXiv:1206.7051 [cs, stat]*, vol. 14, no. 4, pp. 1303–1347, Jun. 2012, arXiv: 1206.7051. [Online]. Available: <http://arxiv.org/abs/1206.7051>
- [55] E. Archer, I. M. Park, L. Buesing *et al.*, "Black box variational inference for state space models," *arXiv:1511.07367 [stat]*, Nov. 2015, arXiv: 1511.07367. [Online]. Available: <http://arxiv.org/abs/1511.07367>
- [56] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.
- [57] M. Chertkov and V. Y. Chernyak, "Loop Calculus in Statistical Physics and Information Science," *Physical Review E*, vol. 73, no. 6, Jun. 2006, arXiv: cond-mat/0601487. [Online]. Available: <http://arxiv.org/abs/cond-mat/0601487>
- [58] A. Weller, K. Tang, T. Jebara *et al.*, "Understanding the Bethe approximation: When and how can it go wrong?" in *UAI*, 2014, pp. 868–877.
- [59] S. Korl, "A factor graph approach to signal modelling, system identification and filtering," Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, 2005.
- [60] S. Särkkä, *Bayesian Filtering and Smoothing*. London ; New York: Cambridge University Press, Oct. 2013.
- [61] J. C. Sibel, "Region-based approximation to solve inference in loopy factor graphs: decoding LDPC codes by Generalized Belief Propagation," Ph.D. dissertation.
- [62] T. Minka, "From hidden markov models to linear dynamical systems," *Vision and Modeling group, Media Lab, MIT, Tech. Rep.* 531, 1999.
- [63] H.-A. Loeliger, J. Dauwels, J. Hu *et al.*, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.
- [64] H.-A. Loeliger, L. Bolliger, C. Reller *et al.*, "Localizing, forgetting, and likelihood filtering in state-space models," in *Information Theory and Applications Workshop, 2009*, Feb. 2009, pp. 184–186.
- [65] T. Heskes, "Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies," *Journal of Artificial Intelligence Research*, vol. 26, pp. 153–190, 2006.
- [66] M. E. Khan and W. Lin, "Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models," *arXiv:1703.04265 [cs]*, Mar. 2017, arXiv: 1703.04265. [Online]. Available: <http://arxiv.org/abs/1703.04265>
- [67] B. Logan and P. Moreno, "Factorial HMMs for acoustic modeling," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998*, vol. 2, May 1998, pp. 813–816 vol.2.
- [68] M. D. Hoffman and D. M. Blei, "Structured Stochastic Variational Inference," *arXiv:1404.4114 [cs]*, Apr. 2014, arXiv: 1404.4114. [Online]. Available: <http://arxiv.org/abs/1404.4114>
- [69] R. Singh, J. Ling, and F. Doshi-Velez, "Structured Variational Autoencoders for the Beta-Bernoulli Process," p. 9.

- [70] R. Bamler and S. Mandt, “Structured Black Box Variational Inference for Latent Time Series Models,” *arXiv:1707.01069 [cs, stat]*, Jul. 2017, arXiv: 1707.01069. [Online]. Available: <http://arxiv.org/abs/1707.01069>
- [71] C. Zhang, Z. Yuan, Z. Wang *et al.*, “Low Complexity Sparse Bayesian Learning Using Combined BP and MF with a Stretched Factor Graph,” *Signal Processing*, vol. 131, pp. 344–349, Feb. 2017, arXiv: 1602.07762. [Online]. Available: <http://arxiv.org/abs/1602.07762>
- [72] M. P. Wand, “Fast Approximate Inference for Arbitrarily Large Semiparametric Regression Models via Message Passing,” *Journal of the American Statistical Association*, vol. 112, no. 517, pp. 137–168, Jan. 2017, arXiv: 1602.07412. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/01621459.2016.1197833>
- [73] A. Caticha, *Entropic Inference and the Foundations of Physics*. EBEB-2012, the 11th Brazilian Meeting on Bayesian Statistics, 2012. [Online]. Available: <https://www.twirpx.com/file/1706750/>
- [74] J. Pearl, “A probabilistic calculus of actions,” in *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 454–462.
- [75] O. Zoeter and T. Heskes, “Gaussian Quadrature Based Expectation Propagation,” *Tenth International Workshop on Artificial Intelligence and Statistics*, p. 9, 2005.
- [76] I. Arasaratnam and S. Haykin, “Cubature Kalman Filters,” *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009, conference Name: IEEE Transactions on Automatic Control.
- [77] S. Sarkka, “Bayesian estimation of time-varying systems: discrete-time systems,” 2012, course lecture notes.
- [78] A. Gelman, A. Vehtari, P. Jylänki *et al.*, “Expectation propagation as a way of life,” *arXiv preprint arXiv:1412.4869*, 2014. [Online]. Available: <http://arxiv.org/abs/1412.4869>
- [79] M. P. Deisenroth and S. Mohamed, “Expectation Propagation in Gaussian Process Dynamical Systems: Extended Version,” *arXiv:1207.2940 [cs, stat]*, Jul. 2012. [Online]. Available: <http://arxiv.org/abs/1207.2940>
- [80] Y. W. Teh, L. Hasenclever, T. Lienart *et al.*, “Distributed Bayesian Learning with Stochastic Natural-gradient Expectation Propagation and the Posterior Server,” *arXiv preprint arXiv:1512.09327*, 2015. [Online]. Available: <http://arxiv.org/abs/1512.09327>
- [81] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning.” MIT Press, 2006. [Online]. Available: <http://mitpress.mit.edu/books/chapters/026218253Xchap2.pdf>
- [82] M. Cox, “Robust Expectation Propagation in Factor Graphs Involving Both Continuous and Binary Variables,” in *26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, p. 5.
- [83] K. J. Friston, L. Harrison, and W. Penny, “Dynamic causal modelling,” *Neuroimage*, vol. 19, no. 4, pp. 1273–1302, 2003, publisher: Elsevier.
- [84] C. D. Mathys, J. Daunizeau, K. J. Friston *et al.*, “A Bayesian foundation for individual learning under uncertainty,” *Frontiers in Human Neuroscience*, vol. 5, 2011. [Online]. Available: <http://www.frontiersin.org/Journal/10.3389/fnhum.2011.00039/full>
- [85] K. Friston, J. Kilner, and L. Harrison, “A free energy principle for the brain,” *Journal of Physiology, Paris*, vol. 100, no. 1-3, pp. 70–87, Sep. 2006.
- [86] K. Friston, “The free-energy principle: a rough guide to the brain?” *Trends in Cognitive Sciences*, vol. 13, no. 7, pp. 293–301, Jul. 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S136466130900117X>
- [87] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. [Online]. Available: <http://www.jstor.org/stable/2984875>
- [88] J. Dauwels, A. Eckford, S. Korl *et al.*, “Expectation maximization as message passing-part I: Principles and gaussian messages,” *arXiv preprint arXiv:0910.2832*, 2009. [Online]. Available: <http://arxiv.org/abs/0910.2832>

- [89] P. Bouvrie, J. Angulo, and J. Dehesa, “Entropy and complexity analysis of Dirac-delta-like quantum potentials,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 11, pp. 2215–2228, Jun. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378437111001464>
- [90] J. Dauwels, S. Korl, and H.-A. Loeliger, “Expectation maximization as message passing,” in *International Symposium on Information Theory, 2005. ISIT 2005. Proceedings*, Sep. 2005, pp. 583–586.
- [91] M. Cox, T. van de Laar, and B. de Vries, “ForneyLab.jl: Fast and flexible automated inference through message passing in Julia,” in *International Conference on Probabilistic Programming*, Boston, MA, Apr. 2018.
- [92] J. Bezanson, A. Edelman, S. Karpinski *et al.*, “Julia: A Fresh Approach to Numerical Computing,” *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/141000671>
- [93] I. Şenöz and B. de Vries, “Online Variational Message Passing in the Hierarchical Gaussian Filter,” in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2018, pp. 1–6.
- [94] C. D. Mathys, “Uncertainty, precision, and prediction errors,” in *UCL Computational Psychiatry Course*, 2014.
- [95] I. Şenöz and B. de Vries, “Online Message Passing-based Inference in the Hierarchical Gaussian Filter,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2020, pp. 2676–2681, ISSN: 2157-8117.
- [96] A. Podusenko and W. M. Kouw, “Message Passing-based Inference for Time-Varying Autoregressive Models,” *Entropy*, vol. 23, no. 6, p. 34, Jun. 2021, number: 6 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1099-4300/23/6/683>
- [97] M. Welling, “On the Choice of Regions for Generalized Belief Propagation,” *arXiv:1207.4158 [cs]*, Jul. 2012, arXiv: 1207.4158. [Online]. Available: <http://arxiv.org/abs/1207.4158>
- [98] M. Welling, T. P. Minka, and Y. W. Teh, “Structured Region Graphs: Morphing EP into GBP,” *arXiv:1207.1426 [cs]*, p. 11, Jul. 2012, arXiv: 1207.1426. [Online]. Available: <http://arxiv.org/abs/1207.1426>
- [99] H.-A. Loeliger, “Factor Graphs and Message Passing Algorithms – Part 1: Introduction,” 2007, http://www.crm.sns.it/media/course/1524/Loeliger_A.pdf, last accessed on 3-4-2019. [Online]. Available: http://www.crm.sns.it/media/course/1524/Loeliger_A.pdf
- [100] S. Iglesias, C. Mathys, K. H. Brodersen *et al.*, “Hierarchical Prediction Errors in Midbrain and Basal Forebrain during Sensory Learning,” *Neuron*, vol. 80, no. 2, pp. 519–530, Oct. 2013. [Online]. Available: [https://www.cell.com/neuron/abstract/S0896-6273\(13\)00807-6](https://www.cell.com/neuron/abstract/S0896-6273(13)00807-6)
- [101] J. Daunizeau, H. E. M. den Ouden, M. Pessiglione *et al.*, “Observing the Observer (II): Deciding When to Decide,” *PLoS ONE*, vol. 5, no. 12, p. e15555, Dec. 2010. [Online]. Available: <https://dx.plos.org/10.1371/journal.pone.0015555>
- [102] M. P. Paulus, Q. J. M. Huys, and T. V. Maia, “A Roadmap for the Development of Applied Computational Psychiatry,” *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, vol. 1, no. 5, pp. 386–392, Sep. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2451902216300350>
- [103] C. D. Mathys, “TAPAS - Translational Algorithms for Psychiatry-Advancing Science,” Apr. 2018, original-date: 2017-01-14T13:14:23Z. [Online]. Available: <https://github.com/translationalneuromodeling/tapas>
- [104] J. Dauwels, S. Korl, and H.-a. Loeliger, “Particle Methods as Message Passing,” in *IEEE International Symposium on Information Theory*. IEEE, Jul. 2006, pp. 2052–2056. [Online]. Available: <http://ieeexplore.ieee.org/document/4036329>
- [105] S. T. Tokdar and R. E. Kass, “Importance sampling: a review,” *WIREs Computational Statistics*, vol. 2, no. 1, pp. 54–60, 2010, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.56>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.56>

- [106] T. v. d. Laar, "Automated design of Bayesian signal processing algorithms," Jun. 2019. [Online]. Available: <https://research.tue.nl/en/publications/automated-design-of-bayesian-signal-processing-algorithms>
- [107] L. A. Aroian, "The Probability Function of the Product of Two Normally Distributed Variables," *The Annals of Mathematical Statistics*, vol. 18, no. 2, pp. 265–271, Jun. 1947. [Online]. Available: <https://projecteuclid.org/euclid.aoms/1177730442>
- [108] J. P. Boyd, "Finding the Zeros of a Univariate Equation: Proxy Rootfinders, Chebyshev Interpolation, and the Companion Matrix," *SIAM Review*, vol. 55, no. 2, pp. 375–396, Jan. 2013. [Online]. Available: <http://epubs.siam.org/doi/10.1137/110838297>
- [109] P. Sebah and X. Gourdon, "Newtons method and high order iterations," p. 10.
- [110] M. Welling and Y. W. Teh, "Bayesian Learning via Stochastic Gradient Langevin Dynamics," p. 8.
- [111] E. Ruli, N. Sartori, and L. Ventura, "Improved Laplace Approximation for Marginal Likelihoods," *Electronic Journal of Statistics*, vol. 10, no. 2, Jan. 2016, arXiv: 1502.06440. [Online]. Available: <http://arxiv.org/abs/1502.06440>
- [112] G. H. Golub and J. H. Welsch, "Calculation of Gauss Quadrature Rules," *Mathematics of Computation*, vol. 23, Apr. 1969. [Online]. Available: <https://www.jstor.org/stable/2004418>
- [113] I. B. Yildiz, K. von Kriegstein, and S. J. Kiebel, "From Birdsong to Human Speech Recognition: Bayesian Inference on a Hierarchy of Nonlinear Dynamical Systems," *PLoS Computational Biology*, vol. 9, no. 9, p. e1003219, Sep. 2013. [Online]. Available: <http://dx.plos.org/10.1371/journal.pcbi.1003219>
- [114] R. Ranganath, D. Tran, and D. M. Blei, "Hierarchical Variational Models," *Journal of Machine Learning Research*, vol. 48, p. 10, 2016.
- [115] K. Friston, "Hierarchical models in the brain," *PLOS Computational Biology*, vol. 4, no. 11, pp. 1–24, 11 2008. [Online]. Available: <https://doi.org/10.1371/journal.pcbi.1000211>
- [116] E. Moench, S. Ng, and S. M. Potter, "Dynamic hierarchical factor models," Federal Reserve Bank of New York, Staff Reports 412, 2009. [Online]. Available: <https://ideas.repec.org/p/fip/fednsr/412.html>
- [117] C. D. Mathys, E. I. Lomakina, J. Daunizeau *et al.*, "Uncertainty in perception and the Hierarchical Gaussian Filter," *Frontiers in Human Neuroscience*, vol. 8, Nov. 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4237059/>
- [118] X. Liu, D. Margaritis, and P. Wang, "Stock market volatility and equity returns: Evidence from a two-state markov-switching model with regressors," *Journal of Empirical Finance*, vol. 19, no. 4, pp. 483–496, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0927539812000382>
- [119] H. Malagon-Vina, S. Ciocchi, J. Passecker *et al.*, "Fluid network dynamics in the prefrontal cortex during multiple strategy switching," *Nature Communications*, vol. 9, 2017.
- [120] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural computation*, vol. 12, no. 4, pp. 831–864, 2000. [Online]. Available: <https://doi.org/10.1162/089976600300015619>
- [121] E. Fox, E. Sudderth, M. Jordan *et al.*, "Nonparametric bayesian learning of switching linear dynamical systems," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2009, pp. 457–464. [Online]. Available: <https://proceedings.neurips.cc/paper/2008/file/950a4152c2b4aa3ad78bd6b366cc179-Paper.pdf>
- [122] J. Daunizeau, K. Friston, and S. Kiebel, "Variational bayesian identification and prediction of stochastic nonlinear dynamic causal models," *Physica D: Nonlinear Phenomena*, vol. 238, no. 21, pp. 2089–2118, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167278909002425>
- [123] V. P. Jilkov and X. R. Li, "Online bayesian estimation of transition probabilities for markovian jump systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1620–1630, 2004.
- [124] E. Fox, E. B. Sudderth, M. I. Jordan *et al.*, "Bayesian nonparametric inference of switching dynamic linear models," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1569–1585, 2011.

- [125] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012. [Online]. Available: <http://www.cs.ucl.ac.uk/staff/d.barber/bmrl/>
- [126] A. Doucet, N. de Freitas, K. Murphy *et al.*, “Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks,” *arXiv:1301.3853 [cs, stat]*, Jan. 2013, arXiv: 1301.3853. [Online]. Available: <http://arxiv.org/abs/1301.3853>
- [127] T. van de Laar, “Automated Design of Bayesian Signal Processing Algorithms,” Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, 2019.
- [128] S. Ghosh, F. M. Delle Fave, and J. Yedidia, “Assumed density filtering methods for learning bayesian neural networks,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [129] M. Tables, M. Abramowitz, I. Stegun *et al.*, “Handbook of mathematical functions with formulas, graphs,” 1971.
- [130] L. A. Aroian, “The probability function of the product of two normally distributed variables,” *The Annals of Mathematical Statistics*, vol. 18, no. 2, pp. 265–271, 1947. [Online]. Available: <http://www.jstor.org/stable/2235783>
- [131] H. Akaike, “Autoregressive model fitting for control,” *Annals of the Institute of Statistical Mathematics*, vol. 23, no. 1, pp. 163–180, Dec. 1971. [Online]. Available: <https://doi.org/10.1007/BF02479221>
- [132] D. C. Hill, D. McMillan, K. R. W. Bell *et al.*, “Application of Auto-Regressive Models to U.K. Wind Speed Data for Power System Impact Studies,” *IEEE Transactions on Sustainable Energy*, vol. 3, no. 1, pp. 134–141, Jan. 2012.
- [133] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0304407686900631>
- [134] G. Barnett, R. Kohn, and S. Sheather, “Bayesian estimation of an autoregressive model using markov chain monte carlo,” *Journal of Econometrics*, vol. 74, no. 2, pp. 237–254, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0304407695017445>
- [135] C. Andrieu, M. Davy, and A. Doucet, “Efficient particle filtering for jump markov systems. application to time-varying autoregressions,” *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1762–1770, 2003.
- [136] S. J. Roberts and W. D. Penny, “Variational Bayes for generalized autoregressive models,” *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2245–2257, Sep. 2002.
- [137] S. M. Tahir, A. Z. Shaameri, and S. H. S. Salleh, “Time-varying autoregressive modeling approach for speech segmentation,” in *Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat.No.01EX467)*, vol. 2, Aug. 2001, pp. 715–718 vol.2.
- [138] Y. J. Chu, S. C. Chan, Z. G. Zhang *et al.*, “A new regularized TVAR-based algorithm for recursive detection of nonstationarity and its application to speech signals,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, Aug. 2012, pp. 361–364, iSSN: 2373-0803.
- [139] M. J. Paulik, N. Mohankrishnan, and M. Nikiforuk, “A time varying vector autoregressive model for signature verification,” in *Proceedings of 1994 37th Midwest Symposium on Circuits and Systems*, vol. 2, Aug. 1994, pp. 1395–1398 vol.2.
- [140] K. Kostoglou, A. D. Robertson, B. J. MacIntosh *et al.*, “A Novel Framework for Estimating Time-Varying Multivariate Autoregressive Models and Application to Cardiovascular Responses to Acute Exercise,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 11, pp. 3257–3266, Nov. 2019, conference Name: IEEE Transactions on Biomedical Engineering.
- [141] K. B. Eom, “Analysis of Acoustic Signatures from Moving Vehicles Using Time-Varying Autoregressive Models,” *Multidimensional Systems and Signal Processing*, vol. 10, no. 4, pp. 357–378, Oct. 1999. [Online]. Available: <https://doi.org/10.1023/A:1008475713345>

- [142] Y. I. Abramovich, N. K. Spencer, and M. D. E. Turley, "Time-Varying Autoregressive (TVAR) Models for Multiple Radar Observations," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1298–1311, Apr. 2007, conference Name: IEEE Transactions on Signal Processing.
- [143] Z. G. Zhang, Y. S. Hung, and S. C. Chan, "Local Polynomial Modeling of Time-Varying Autoregressive Models With Application to Time-Frequency Analysis of Event-Related EEG," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 557–566, Mar. 2011, conference Name: IEEE Transactions on Biomedical Engineering.
- [144] Huan Wang, L. Bai, Jianmei Xu *et al.*, "EEG recognition through Time-varying Vector Autoregressive Model," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Aug. 2015, pp. 292–296.
- [145] K. Sharman and B. Friedlander, "Time-varying autoregressive modeling of a class of nonstationary signals," in *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 9, Mar. 1984, pp. 227–230.
- [146] Yuanjin Zheng and Zhiping Lin, "Time-varying autoregressive system identification using wavelets," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, vol. 1, Jun. 2000, pp. 572–575 vol.1, iISSN: 1520-6149.
- [147] A. Podusenko, W. M. Kouw, and B. de Vries, "Online Variational Message Passing in Hierarchical Autoregressive Models," p. 6, Jun. 2020, iISSN: 2157-8117.
- [148] S. Akbayrak and B. de Vries, "Reparameterization Gradient Message Passing," in *submitted to EUSIPCO*, 2019, p. 5.
- [149] S. Akbayrak and I. Bocharov, "Extended Variational Message Passing for Automated Approximate Bayesian Inference," p. 34, 2021.
- [150] A. Caticha, "Relative Entropy and Inductive Inference," *AIP Conference Proceedings*, vol. 707, pp. 75–96, 2004, arXiv: physics/0311093. [Online]. Available: <http://arxiv.org/abs/physics/0311093>
- [151] P. A. Ortega and D. A. Braun, "A Minimum Relative Entropy Principle for Learning and Acting," *J. Artif. Intell. Res.* 2010, pp. 475–511, 2010.
- [152] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on Information Theory*, vol. 26, no. 1, pp. 26–37, Jan. 1980, conference Name: IEEE Transactions on Information Theory.
- [153] E. Engel and R. M. Dreizler, *Density Functional Theory: An Advanced Course*, ser. Theoretical and Mathematical Physics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. [Online]. Available: <http://link.springer.com/10.1007/978-3-642-14090-7>
- [154] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.
- [155] S. Ahn, M. Chertkov, and J. Shin, "Gauging Variational Inference," *arXiv:1703.01056 [stat]*, p. 14, Mar. 2017, arXiv: 1703.01056. [Online]. Available: <http://arxiv.org/abs/1703.01056>
- [156] V. H. Tran, "Copula Variational Bayes inference via information geometry," *arXiv:1803.10998 [cs, math, stat]*, Mar. 2018, arXiv: 1803.10998. [Online]. Available: <http://arxiv.org/abs/1803.10998>

List of Publications

Journal Articles

1. Thijs van de Laar, İsmail Şenöz, Ayça Özçelikkale, Henk Wymeersch , "Chance Constrained Active Inference", *Neural Computation*, October 2021.
2. İsmail Şenöz, Thijs van de Laar, Dmitry Bagaev, and Bert de Vries, "Variational Message Passing and Local Constraint Manipulation in Factor Graphs", *Entropy* 23, no. 7: 807, June 2021.
3. Semih Akbayrak, İsmail Şenöz, Alp Sarı and Bert de Vries, " Probabilistic Programming with Stochastic Variational Message Passing", *Journal of Approximate Reasoning* (in review).

Conference Articles

1. İsmail Şenöz and Bert de Vries, "Online Variational Message Passing in the Hierarchical Gaussian Filter", *IEEE International Workshop on Machine Learning for Signal Processing*, November 2018.
2. İsmail Şenöz, Albert Podusenko, Wouter Kouw and Bert de Vries, "Bayesian Joint State and Parameter Tracking in Autoregressive Models", *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, PMLR 120:95-104, June 2020.
3. İsmail Şenöz and Bert de Vries, "Online Message Passing-based Inference in the Hierarchical Gaussian Filter", *IEEE International Symposium on Information Theory*, August 2020.
4. Bart van Erp, İsmail Şenöz and Bert de Vries, "Variational Log-Power Spectral Tracking for Acoustic Signals", *IEEE Statistical Signal Processing Workshop*, August 2021.

5. İsmail Şenöz, Albert Podusenko, Semih Akbayrak, Christoph Mathys and Bert de Vries, "The Switching Hierarchical Gaussian Filter", IEEE International Symposium on Information Theory, September 2021.
6. Albert Podusenko, Bart van Erp, Dmitry Bagaev, İsmail Şenöz and Bert de Vries, "Message Passing-Based Inference in the Gamma Mixture Model ", IEEE International Workshop on Machine Learning for Signal Processing, November 2021.
7. Semih Akbayrak, İsmail Şenöz and Bert de Vries, "Adaptive Importance Sampling Message Passing", IEEE International Symposium on Information Theory, June 2022.
8. Albert Podusenko, Bart van Erp, Dmitry Bagaev, İsmail Şenöz, Bert de Vries, "Message Passing-Based Inference in Switching Autoregressive Models", In 30th European Signal Processing Conference (EUSIPCO 2022) - Proceedings, August 2022.
9. Alp Sarı, Semih Akbayrak, İsmail Şenöz and Bert de Vries, "Adaptive Optimizer Design for Constrained Variational Inference ", Symposium on Information Theory and Signal Processing in the Benelux (SITB), 2022.
10. Albert Podusenko, Semih Akbayrak, İsmail Şenöz, Maarten Schoukens and Wouter Kouw, "Message-Passing-Based System Identification for NARMAX Models", IEEE Conference on Decision and Control (in review).

Acknowledgements

In 2016 while I was an M.Sc. student, I took the Adaptive Information Processing course from **Bert de Vries** and **Tjalling Tjalkens** as an introduction to machine learning and information theory. While attending the classes, I realized that I wanted to work on these subjects for my M.Sc. graduation. I was extremely late to start looking for projects and was awfully behind schedule to finish my M.Sc. I was stressfully looking for a graduation projection. I approached **Bert** to see if he had a project. For some reason, he accepted to be my advisor. Since then **Bert** has been a part of my life, and I am thankful for having him as a Ph.D. advisor. His free-spirited attitude towards life and research, however controversial it may seem on some occasions, is truly inspiring. His stories are always entertaining, and the conversations never get dull. Once again **Bert**, I am grateful that you chose me as a Ph.D. candidate in BIASlab. I did not have the opportunity to work with **Tjalling** on projects; however, I learned much from him during the classes, and from the intriguing questions he would ask in the seminars. Talking to him is enlightening on so many levels, so thank you **Tjalling**.

I would like to thank the members of the defense committee **Chris Mathys**, **Frans Willems**, **Justin Dauwels**, **Paul Van den Hof**, **Tom Heskes** and **Thijs van de Laar** for reading this dissertation and providing me with valuable feedback to improve the manuscript. A special thanks to my co-promoter **Chris** for having me as a guest researcher in Aarhus for three months and providing me insider information on the HGF and feedback for paper drafts. **Thijs**, it was a pleasure working with you during my Ph.D. It would have been impossible for me to navigate through the source code of ForneyLab without you. I am fascinated by how you clarify things and thankful for all the insightful discussions. You helped me greatly in stressful times, and you are always great fun to talk to. Thank you for the great times.

Many thanks to **GN Advanced Sciences** for supporting this research financially.

Thanks to **Jan** for creating an ideal working environment. I would like to thank the support staff of the SPS group **Anja**, **Carla**, **Emerald** and **Judith**, for their help with all non-technical matters.

I have been a part of BIASlab since 2016, and I have had the pleasure of meeting ex-

cellent colleagues and good friends. **Wouter**, you are a great mentor, and I appreciate your companionship in challenging and happy times. I will never forget the all-nighter we pulled together for the CDC submission. **Magnus**, you are a hilarious man with a great sense of humor and an excellent capability to communicate. Seeing you passionately lecture on active inference is instructional for anyone who aspires to teach. **Martin**, your positive attitude is an excellent fuel for many interesting conversations and, need I say, many crazy nights. Together with **Magnus** you made it to my home country and managed to charm my family and friends. Thanks for the memorable times. **Bart**, it is a great pleasure working with you. I admire your work ethic and wit. Thank you for all the great effort you put into the research we work on together. Also, many thanks to newer members of BIASlab **Alp**, **Hoang**, **Jim**, **Mykola**, **Sepideh** and **Tim** for their great team spirit.

Semih, you are an excellent researcher with tremendous intuition. I am thankful for all the detailed explanations of sampling and non-conjugate inference-related concepts. Your bachelor party was one of the most entertaining nights I had. It is a great pleasure working with you and having you as a friend. **Dmitry** I am grateful for all the significant contributions you make to the projects we work on. The way you code is poetry, and ReactiveMP is a great example. It is a pleasure working with you, but it is a greater pleasure jamming with you.

I am grateful for all the friends that I met during my Ph.D. and I am thankful to be surrounded by **Berk**, **Bora**, **Branislav**, **Burcu**, **Daniela**, **Dilara**, **Denizhan**, **Dilge**, **Eylem**, **Gizem**, **Gönenç**, **Jesse**, **Jiali**, **Mert**, **Naz**, **Ozan**, **Raquel**, **Seren**, **Tanya**, **Tunç** and **Yankı**. I enjoyed every trip we took to get away from the rainy lowlands, every picnic gathering and every other activity.

To my friends from the university days **Barışcan**, **Batuhan**, **Elif**, **Seren** and **Yiğit**. To all my friends who have been there with me since high school **Arhan**, **Beril**, **Cem**, **Emir**, **Hanzade**, **İdil**, **Simay** and **Simge**. The amount of memories and adventures we shared are far too long to list here. I am grateful to have all of you by my side.

My dearest friends **Albert**, **Omar** and **Yunus** thank you for being there all the time. **Omar**, what a marvelous friend you are. Your kindness is unbounded, and you are a gentle soul. **Albert**, I think the most valuable thing I got out of the Ph.D. is befriending you. It would have been impossible to get through the last four years without your friendship. **Yunus**, you deserve a special thanks because you endure all my shenanigans while living with me, which I must admit is not for the faint-hearted. You are one of a kind person, and I am grateful for all the time we spent together.

Dear **Elif**, thank you for always supporting me through this adventure, even at times when you were not close by. I would not have been the way I am without our experiences together.

I am grateful for all the support from my extended family **Güzide**, **Nejat**, **Fikret**, **Mehmet**, **Züleyha**, **Ebru**, **Yasin**, **Chris**, **Bilge**, **Güzide Su**, **Zeynep** and **Yiğit**.

On 21 March 2021, I lost my uncle **Hasan**. He was an extraordinary man with a loving heart. He showed me that love does not know boundaries. The rest of my family and I owe him so much. It would have been impossible for us to grow as a family without his efforts and love. May he rest in peace.

To my lovely sister **Ayşegül**. Your determinism and courage inspire me every day. You never stop supporting me through hardships, and it is reassuring to have you always looking out for me. It is amazing to see you grow to be an independent, strong woman. Thank you for all the love and support.

To my lovely aunts **Ayşe** and **Şenay**. It is hard to describe my gratitude towards you with words. I am eternally grateful for all the love you have poured in with me. It would not have been possible for me to reach this point without your support. Thank you for showing me the utmost kindness and spoiling me with unconditional love.

And last but not least, I would like to express my gratitude to my mom **Mürşide** and to my dad **Mehmet**. Nothing would have been possible without their endless love and support. I am grateful for all their sacrifices and for creating a loving family atmosphere. Thank you, Mom, for being such a resilient figure in my life and always ensuring that everything is alright. Thank you, Dad, for always providing the best of everything I can hope for.

Biography

İsmail Şenöz was born on May 2, 1992 in Izmir, Turkey. He received his B.Sc. degrees in Electrical and Electronics Engineering and Mathematics from Koç University, Istanbul, in 2015. He received his M.Sc. degree in Electrical Engineering from the Eindhoven University of Technology (TU/e) in 2017, where he completed the M.Sc. program in the Bayesian Intelligent Autonomous Systems Lab group.

In November 2017, İsmail started working towards a Ph.D. degree in the Signal Processing Systems group at the Eindhoven University of Technology under the supervision of Bert de Vries. His research mainly focused on approximate inference techniques for hierarchical dynamical systems. This dissertation includes some of the main results of this Ph.D. research.

Since 2021, İsmail has been working as a researcher in the Bayesian Intelligent Autonomous Systems Lab at the TU/e. His research interests include approximate inference methods, stochastic differential equations, and time-series modeling. Aside from his academic interests, İsmail enjoys playing the guitar and playing chess.

