

# Approximation methods for latent variable models

*Samuel Parsons*

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
**Doctor of Philosophy**  
of  
**University College London.**

Department of Statistical Science  
University College London

August 31, 2016



I, Samuel Parsons, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.



# Abstract

Modern statistical models are often intractable, and approximation methods can be required to perform inference on them. Many different methods can be employed in most contexts, but not all are fully understood. The current thesis is an investigation into the use of various approximation methods for performing inference on latent variable models.

Composite likelihoods are used as surrogates for the likelihood function of state space models (SSM). In chapter 3, variational approximations to their evaluation are investigated, and the interaction of biases as composite structure changes is observed. The bias effect of increasing the block size in composite likelihoods is found to balance the statistical benefit of including more data in each component. Predictions and smoothing estimates are made using approximate Expectation-Maximisation (EM) techniques. Variational EM estimators are found to produce predictions and smoothing estimates of a lesser quality than stochastic EM estimators, but at a massively reduced computational cost.

Surrogate latent marginals are introduced in chapter 4 into a non-stationary SSM with i.i.d. replicates. They are cheap to compute, and break functional dependencies on parameters for previous time points, giving estimation algorithms linear computational complexity. Gaussian variational approximations are integrated with the surrogate marginals to produce an approximate EM algorithm. Using these Gaussians as proposal distributions in importance sampling is found to offer a positive trade-off in terms of the accuracy of predictions and smoothing estimates made using estimators.

A cheap to compute model based hierarchical clustering algorithm is proposed

in chapter 5. A cluster dissimilarity measure based on method of moments estimators is used to avoid likelihood function evaluation. Computation time for hierarchical clustering sequences is further reduced with the introduction of short-lists that are linear in the number of clusters at each iteration. The resulting clustering sequences are found to have plausible characteristics in both real and synthetic datasets.

# Acknowledgements

This thesis would not have been possible without the help and support of more people than I can mention, but some names deserve particular acknowledgement. Firstly, my supervisor Dr Ricardo Silva has provided invaluable direction to my studies. Our discussions have been enlightening and his ideas have been inspiring. From before even the kernel of this thesis had been born, his advice and instruction have been essential components in my academic development.

I was very fortunate to receive a fully funded studentship from the Financial Computing Centre for Doctoral Training. The support they have given me is gratefully acknowledged. Professor Philip Treleaven in particular has provided support and advice that I greatly appreciate. I am also grateful to Professor Tom Fearn for giving me the support of the Statistical Science department at UCL. The departmental grant I received allowed me to continue my studies to their completion.

My friends and family have been unquestionably supportive of my studies, and I cannot thank them enough. Bogi, Ashish, and Emma in particular deserve mention for the debts of gratitude I owe them. I am lucky to have benefited so much from the pleasure of their company. Finally, I would like to thank my parents for their love and support over the years. It has been total and unwavering, and I appreciate it beyond words. I hope to make you proud.





# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                                | <b>17</b> |
| <b>2</b> | <b>Literature Review</b>                           | <b>23</b> |
| 2.1      | Approximate inference . . . . .                    | 23        |
| 2.1.1    | Inference . . . . .                                | 24        |
| 2.1.2    | Approximations . . . . .                           | 26        |
| 2.2      | Variational approximations . . . . .               | 27        |
| 2.2.1    | Variational EM . . . . .                           | 28        |
| 2.2.2    | Limitations and mitigations . . . . .              | 33        |
| 2.3      | Composite likelihoods . . . . .                    | 36        |
| 2.3.1    | Specialised models and applications . . . . .      | 42        |
| 2.4      | Monte Carlo . . . . .                              | 44        |
| 2.4.1    | Importance Sampling . . . . .                      | 47        |
| 2.4.2    | Particle filters . . . . .                         | 49        |
| 2.4.3    | Markov chain Monte Carlo . . . . .                 | 61        |
| 2.5      | Message passing in graphical models . . . . .      | 65        |
| 2.5.1    | Graphical models . . . . .                         | 68        |
| 2.5.2    | Belief propagation . . . . .                       | 72        |
| 2.5.3    | Belief propagation in graphs with cycles . . . . . | 74        |
| 2.6      | Trade-offs in Computational Statistics . . . . .   | 76        |
| 2.6.1    | Statistical trade-offs . . . . .                   | 78        |
| 2.6.2    | Computational trade-offs . . . . .                 | 79        |

|          |  |            |
|----------|--|------------|
| <b>3</b> | <b>Composite Likelihood Estimators in State Space Models</b>             | <b>89</b>  |
| 3.1      | The state space model . . . . .  | 90         |
| 3.1.1    | Approximate inference . . . . .  | 93         |
| 3.1.2    | Gaussian state space, Student-t observations . . . . .                   | 95         |
| 3.1.3    | Further model assumptions . . . . .                                      | 96         |
| 3.2      | Composite likelihoods . . . . .  | 100        |
| 3.3      | Variational approximations . . . . .                                     | 101        |
| 3.3.1    | Estimation algorithm . . . . .   | 103        |
| 3.4      | Stochastic EM . . . . .  | 109        |
| 3.4.1    | Estimating smoothed distributions . . . . .                              | 112        |
| 3.5      | Prediction . . . . .   | 113        |
| 3.6      | Synthetic data . . . . .   | 116        |
| 3.7      | Experiments . . . . .  | 117        |
| 3.8      | Results . . . . .  | 121        |
| 3.8.1    | Bias experiment . . . . .  | 122        |
| 3.8.2    | Variance experiment . . . . .  | 123        |
| 3.8.3    | Smoothing experiment . . . . .   | 124        |
| 3.8.4    | Prediction experiment . . . . .  | 126        |
| 3.9      | Discussion . . . . .   | 126        |
| <b>4</b> | <b>Integrating Composite Likelihood and Non-Likelihood Based Methods</b> | <b>133</b> |
| 4.1      | Problem outline . . . . .  | 134        |
| 4.2      | Gaussian-Poisson state space model . . . . .                             | 136        |
| 4.2.1    | Exploiting surrogate marginals . . . . .                                 | 137        |
| 4.3      | Surrogate marginals . . . . .  | 139        |
| 4.3.1    | Thresholding . . . . .   | 141        |
| 4.4      | Estimating maximum composite likelihood parameters . . . . .             | 143        |
| 4.4.1    | Expectation step . . . . .   | 145        |
| 4.4.2    | Maximisation step . . . . .  | 152        |
| 4.5      | Method of moments parameter estimates . . . . .                          | 153        |
| 4.6      | Experiments . . . . .  | 155        |

|                     |  |            |
|---------------------|--|------------|
| 4.6.1               | Synthetic data . . . . .   | 157        |
| 4.7                 | Results . . . . .  | 158        |
| 4.8                 | Discussion . . . . .   | 161        |
| <b>Appendices</b>   |  | <b>169</b> |
| 4.A                 | Grad vector for KL divergence from $q \in \mathcal{Q}_{\mathcal{N}}$ . . . . .       | 169        |
| 4.B                 | Hessian matrix for KL divergence from $q \in \mathcal{Q}_{\mathcal{N}}$ . . . . .    | 170        |
| 4.C                 | Grad vector for KL divergence from $q \in \mathcal{Q}_{\Pi\mathcal{N}}$ . . . . .    | 171        |
| 4.D                 | Hessian matrix for KL divergence from $q \in \mathcal{Q}_{\Pi\mathcal{N}}$ . . . . . | 171        |
| <b>5</b>            | <b>Applying the method of moments in hierarchical clustering</b>                     | <b>173</b> |
| 5.1                 | Parameter estimation . . . . .   | 176        |
| 5.1.1               | Fitting the latent process . . . . .   | 178        |
| 5.1.2               | Fitting the $\beta$ parameters . . . . .   | 181        |
| 5.2                 | Hierarchical clustering . . . . .  | 183        |
| 5.2.1               | Short-lists . . . . .  | 186        |
| 5.2.2               | Analysis of clustering sequence . . . . .  | 188        |
| 5.3                 | Experimental data . . . . .  | 190        |
| 5.3.1               | Synthetic data . . . . .   | 190        |
| 5.3.2               | TfL data . . . . .   | 191        |
| 5.4                 | Experiments . . . . .  | 192        |
| 5.5                 | Results . . . . .  | 193        |
| 5.6                 | Discussion . . . . .   | 193        |
| <b>6</b>            | <b>General Conclusions</b>   | <b>197</b> |
| <b>Bibliography</b> |  | <b>203</b> |



# List of Figures

|     |   |     |
|-----|---|-----|
| 2.1 | Variational approximation as a tractable sub-graph . . . . .          | 33  |
| 2.2 | Tractable sub-graph of factorial hidden Markov model . . . . .        | 36  |
| 2.3 | Example model for using a composite likelihood . . . . .              | 39  |
| 2.4 | Directed graph of the state space model . . . . .                     | 51  |
| 2.5 | Example of particle degeneracy . . . . .                              | 56  |
| 2.6 | Simple undirected graphical model . . . . .                           | 69  |
| 2.7 | Simple directed graphical model . . . . .                             | 71  |
| 3.1 | Directed graph of the state space model . . . . .                     | 91  |
| 3.2 | Directed graph of Gaussian Student-t state space model . . . . .      | 96  |
| 3.3 | Tractable sub-graph of Gaussian Student-t state space model . . . . . | 102 |
| 3.4 | Per-element variances in parameter estimates . . . . .                | 125 |
| 4.1 | Per-time RMSE for in-sample smoothing . . . . .                       | 164 |
| 5.1 | Average exit counts at Tube stations . . . . .                        | 174 |
| 5.1 | Reconstruction errors in hierarchical clustering . . . . .            | 193 |
| 5.2 | Plots of clustering characteristics . . . . .                         | 196 |



# List of Tables

|     |   |     |
|-----|---|-----|
| 1   | Table of notation . . . . .   | 16  |
| 3.1 | Running times for parameter estimation . . . . .  | 121 |
| 3.2 | Comparison of composite likelihood estimators from composite<br>likelihoods with overlapping and non-overlapping components . . . | 122 |
| 3.3 | Results of bias experiment . . . . .  | 123 |
| 3.4 | Results of variance experiment . . . . .  | 124 |
| 3.5 | Results of smoothing experiment . . . . .   | 125 |
| 3.6 | Results of prediction experiment . . . . .  | 127 |
| 4.1 | Times taken to compute parameter estimates . . . . .  | 158 |
| 4.2 | Differences in estimators across approximation methods . . . . .  | 165 |
| 4.3 | Results of smoothing experiment . . . . .   | 166 |
| 4.4 | Results of prediction experiment . . . . .  | 167 |

**Table 1:** Unless otherwise indicated at specific parts of the text, the following notational conventions are employed throughout the current thesis.

|                  |   |
|------------------|---|
| $X'$             | The transpose of $X$  |
| $X_{a:b}$        | The subset of $X$ corresponding to the interval of index values $a, \dots, b$ |
| $\mathcal{P}(X)$ | The probability of a random variable having the outcome $X$                   |
| $\mathcal{N}$    | The Gaussian distribution   |
| $\mathcal{U}$    | The uniform distribution  |
| $\mathcal{G}$    | The Gamma distribution  |
| $\mathcal{PO}$   | The Poisson distribution  |



## Chapter 1

# Introduction

One of the biggest challenges facing modern statisticians is the successful application of a given model to a real world problem and/or dataset. Whether the model in question is determined by a real world problem or by more abstract considerations, it is a ubiquitous feature of modern statistics that popular models have non-trivial implementation challenges. Furthermore, these challenges are largely computational in nature.

It is an unfortunate truth that statistical models largely belong to a spectrum whose endpoints are - at one end - those that are easy to implement but are poor models of reality, and - at the other end - those that are effective models of reality but are difficult to implement. In practice this means that a trade-off between utility and feasibility exists, and inference frameworks that lie at virtually all points on the trade-off spectrum will be valuable in some context somewhere. Much of a statistician's judgement, therefore, is often focussed on choosing an inference framework that optimises this trade-off in their given context.

A simple but illustrative example of such a trade-off is the modelling of heights of people in a mixed gender, adult population. A basic background in statistics provides the intuition to suggest a Gaussian distribution being used:

$$X_i \sim \mathcal{N}(X_i | \mu_X, \sigma_X^2) \tag{1.1}$$

where  $X_i$  denotes each member of the population, and each member's height is

drawn independently from the same Gaussian. Inference is particularly simple for this model; given a dataset  $\{X_i\}_{i=1}^N$ , the maximum likelihood estimates of  $(\mu_X, \sigma_X^2)$  are easily calculated as:

$$\begin{aligned}\hat{\mu}_X &= \frac{1}{N} \sum_{i=1}^N X_i \\ \hat{\sigma}_X^2 &= \frac{1}{N} \sum_{i=1}^N (X_i - \hat{\mu}_X)^2\end{aligned}\quad (1.2)$$

and inferences on the population distribution can then be made using these parameter estimates. If, for example, the boundaries of the central  $1 - \alpha$  of the population,  $\alpha \in (0, 1)$ , want to be estimated then they are simply:

$$\left(\hat{Q}_X\left(\frac{\alpha}{2}\right), \hat{Q}_X\left(1 - \frac{\alpha}{2}\right)\right) = \left(Q_N\left(\frac{\alpha}{2} \mid \hat{\mu}_X, \hat{\sigma}_X^2\right), Q_N\left(1 - \frac{\alpha}{2} \mid \hat{\mu}_X, \hat{\sigma}_X^2\right)\right) \quad (1.3)$$

where  $Q_Y(\alpha)$  is the inverse distribution function of  $Y$ , which for  $Y \sim \mathcal{N}$  is almost universally implemented on statistical programming platforms.

Such a choice, though, ignores the empirically observed structural difference in height distribution between males and females. A model that could incorporate this distinction would almost certainly model reality more effectively than a Gaussian distribution.

If, instead of just one Gaussian distribution being used to model the heights of all people, two Gaussians were used in a mixture model, then the guiding intuition of using a Gaussian distribution could be satisfied simultaneously with the concern of modelling a known artefact of the dataset in question. Such a distribution would (assuming gender is a binary valued characteristic possessed by all members of the population) look like:

$$X_i \sim \pi_M \mathcal{N}(X_i \mid \mu_M, \sigma_M^2) + (1 - \pi_M) \mathcal{N}(X_i \mid \mu_F, \sigma_F^2) \quad (1.4)$$

where  $\pi_M$  denotes the probability of an arbitrary person being male,  $(\mu_M, \sigma_M^2)$  are the parameters for the distribution of male heights, and  $(\mu_F, \sigma_F^2)$  are the parameters

for the distribution of female heights. The height of each person would be drawn independently from (1.4).

While (1.4) is a conceptually simple statistical model, if the gender of the person to whom each observation  $X_i$  belongs is unknown then its implementation is much more complicated than that of (1.1). Maximum likelihood parameter estimates are no longer available in closed form, and must be estimated iteratively:

$$\hat{\theta}^{(k+1)} = f\left(\{X_i\}_{i=1}^N, \hat{\theta}^{(k)}\right) \quad (1.5)$$

where  $\theta$  is a vector collecting all model parameters together and  $f(\cdot)$  is some function. Estimates of quantiles based on particular parameter estimates are also non-trivial:

$$\hat{Q}_X(\alpha) = \hat{F}_X^{-1}(\alpha) \quad (1.6)$$

where

$$\hat{F}_X(x) = \hat{\pi}_M F_{\mathcal{N}}(x|\hat{\mu}_M, \hat{\sigma}_M^2) + (1 - \hat{\pi}_M) F_{\mathcal{N}}(x|\hat{\mu}_F, \hat{\sigma}_F^2) \quad (1.7)$$

which, given only the distribution function  $F_{\mathcal{N}}$ , is a non-trivial function to invert.

This example illustrates how introducing more realism into a model generally increases both the theoretical and computational complexity of any inference that might be made from data. While the computational challenges introduced in this simple example are themselves not beyond the capabilities of modern technology, this is not generally true in a typical modern statistical setting; modern technology is not sufficient to successfully use a brute force approach towards the implementation of realistic statistical models. Furthermore, the requirement of statistical modelling that it be as realistic as possible means that maximising the effectiveness of inference under computational constraints is a permanent feature of research in statistical computing.

It is in this context that approximation methods, and choosing from amongst

them, become particularly relevant. Exact inference is not always possible in modern statistics and even when it is, it might take a lot of computing resources to perform. If a particular approximation method produces results that are sufficient for the context in question then it should be considered as a potential alternative to exact inference. When both the accuracy and computing requirements of an approximation are favourable then by implication exact inference becomes needlessly expensive.

The choices to be made regarding whether to employ an approximate inference framework, and if so then which one, thus boil down to trade-offs between computational costs and statistical benefits. Some of these trade-offs can be deduced analytically, and in such cases the preference between alternatives can be relatively clear cut.

Current statistical theory is not always sufficient to elucidate the optimal choice among methods though. Variational approximations to likelihood evaluation, for example, introduce a bias to parameter estimates that is not well understood (Jordan et al., 1999; Turner and Sahani, 2008). They are often much cheaper to evaluate than either unbiased approximations or the likelihood function itself though. It is difficult to know *a priori* whether the trade-off between compute times and accuracy favours them or not.

The current thesis is an investigation into the impacts on inference of using various approximation methods. Particular state space models that require approximations to be made are used as objects of inferential interest. The use of variational approximations to composite likelihoods is explored in chapter 3. In chapter 4 a method of moments approximation to the prior marginals is integrated with Gaussian variational approximations. A model based hierarchical clustering algorithm that avoids likelihood evaluation is developed in chapter 5.

The remaining content of the current thesis is outlined as follows. Chapter 2 is a self contained review of the literature on approximate methods featuring in the subsequent chapters. Descriptions of each method aim to be thorough, with a view to allowing their implementations in subsequent chapters not to be encumbered with

distracting detail. Occasionally, however, content from chapter 2 will be repeated later, when deemed appropriate for purposes of exposition.

In chapter 3, a stationary state space model with Student-t observations is the basis of approximate inference. Parameters are estimated via composite likelihoods, which are in turn evaluated approximately via variational approximations. The composite structure of each composite likelihood affects the finite sample bias and variance of parameter estimates, and the accuracy of variational approximations depends on the size of each component in the composite likelihood. The effects of using both methods in tandem is investigated to explore how they interact. The behaviour of parameter estimates, and of further inference made using them, is observed to find where optimal trade-off choices lie.

Chapters 4 and 5 use a non-stationary state space model with Poisson observations to further explore approximate inference in computationally challenging contexts. Composite likelihood evaluation is difficult when data is non-stationary, as marginal latent distributions have varying functional dependencies on parameters. Approximations to the latent marginals are introduced in chapter 4 that bypass this issue.

Additionally, closed form factorised variational approximations are not available for this model. Gaussian approximations are chosen instead as classes of tractable distributions under which to take expectations. Finding the optimal Gaussians for approximating posterior distributions can be quite expensive computationally, but is much less so if the Gaussians are constrained to be diagonal. The computational and statistical trade-offs between using general Gaussians and the restriction to diagonal Gaussians only is investigated.

Chapter 5 makes use of the same model as in chapter 4, only in a high dimensional setting. Likelihood based methods are not practical in such a context, so inference founded on the method of moments framework developed in chapter 4 is explored. A hierarchical clustering algorithm is developed, allowing a low dimensional extension to the original state space model to be exploited. Clusterings in the hierarchical sequence with sufficiently low dimensionality are used to fit parameters

and perform subsequent predictions and smoothing approximations.

Chapter 6 gathers the results and conclusions of chapters 3 - 5 together for a final summary of the thesis. The outcomes of experiments and the limitations of the findings are discussed, along with general conclusions and avenues of further research.

## **Chapter 2**

# **Literature Review**

This chapter will cover the current literature on the challenges described in Chap 1. There is a brief section giving a description of the tasks looking to be achieved in approximate inference. Following this, the major groups into which current methodologies can be placed will each be described, and example methodologies analysed in more detail. Currently popular approximation strategies, and where they fit in the literature on approximation methods, will therefore be available to the reader with self-contained descriptions. This should serve as a general outline to the relevance of the current thesis, as well as providing detail that will be referred to specifically in the thesis itself.

## **2.1 Approximate inference**

The current thesis is an investigation into the conditions under which different choices regarding the trade-off between utility and feasibility might be optimal. Various concepts regarding statistical models, inference, approximations, and measures of performance will be employed, giving rise to potential ambiguities. As such, it is important to define key terminology precisely. The following section will aim to clarify the subsequent use of terminology. For clarity it should be noted that the models considered in the current thesis are frequentist, with fixed value parameter estimates. The work in chapters 3 and 4 can in principle be extended to Bayesian frameworks.

### 2.1.1 Inference

Statistical inference is the task of reaching conclusions about a population of data after having seen only a sample of it, and of quantifying the appropriate amount of confidence that should be placed in them. The precise conclusions that can be reached depend on the modelling assumptions that are made regarding the population, but in general the output of an inference procedure is a posited distribution for either the data population itself or for some subset of the complete collection of random variables that are part of the model. Some of the fundamental objects that need to be computed of a statistical model include:

**Posterior distributions** Posterior distributions are data dependent distributions that take the form described by *Bayes' rule*:

$$\mathcal{P}(X | Y) = \frac{\mathcal{P}(X)\mathcal{P}(Y | X)}{\mathcal{P}(Y)} \quad (2.1)$$

where  $Y$  refers in this case to observed data, and  $X$  being the random variable of interest. In general this can be either a latent variable or a Bayesian parameter.

**Marginal distributions** Marginal distributions are distributions of a subset of variables that have a joint distribution. In general this requires jointly integrating out the unwanted variables from the joint distribution, i.e. if  $f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$  is the joint density of the variables of interest  $\mathbf{x}$  and nuisance variables  $\mathbf{y}$ , then

$$f_{\mathbf{X}}(\mathbf{x}) = \int f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})d\mathbf{y} \quad (2.2)$$

which may have a computational complexity that is exponential in the size of the model, depending on how it is parametrised. If further assumptions can be made about the independence structure of the joint model though, then the computational complexity of this operation can be significantly reduced. This idea will be expanded on in detail in chapter 2, sec 2.5.

**Predictive distributions** Predictive distributions are the modelled distributions of



new data, conditioned on the data that has already been seen. The nature of a predictive distribution depends on the choice made between a frequentist and a Bayesian model:

**Frequentist** For a frequentist model with parameter  $\theta$ , this is simply the modelled distribution with parameters estimated from the observed data:

$$Y^{\text{new}}|Y^{\text{old}} \sim \mathcal{P}(Y^{\text{new}} | Y^{\text{old}}, \hat{\theta}(Y^{\text{old}})) \quad (2.3)$$

where  $\hat{\theta}(Y^{\text{old}})$  is the estimate of  $\theta$  made using  $Y^{\text{old}}$ .

**Bayesian** In a Bayesian model, the prior distribution of the parameter  $\theta$  can be used to derive a *prior predictive distribution*:

$$f_Y(y^{\text{new}}) = \int f_{Y|\theta}(y^{\text{new}}|\theta)f_{\theta}(\theta)d\theta \quad (2.4)$$

or alternatively, a *posterior predictive distribution* can be derived from the prior and the observed data together:

$$f_{Y|Y^{\text{old}}}(y^{\text{new}} | Y^{\text{old}}) = \int f_{Y|Y^{\text{old}},\theta}(y^{\text{new}} | Y^{\text{old}}, \theta)f_{\theta|Y^{\text{old}}}(\theta|Y^{\text{old}})d\theta \quad (2.5)$$

**Clustering** A desired outcome of some statistical analyses is a many-to-one function mapping elements  $x \in \mathcal{X}$  of some population to a collection  $c \in \mathcal{C}$  of cluster labels:

$$k : x \mapsto k(x) \quad (2.6)$$

with  $|\mathcal{C}| \leq |\mathcal{X}|$ , and the equality only holding in the case of the *trivial clustering* where  $k(\cdot)$  is a bijection. Cluster membership will generally be on the basis of some shared statistical property.

Many algorithms to achieve this are employed in practice, and those that are of a statistical nature are subject to the utility / feasibility trade-off described above.

### 2.1.2 Approximations

When the calculation of a quantity of interest is intractable, an approximation of some sort has to be made. There are in general two strategies that can be pursued when making approximations:

**Surrogate quantities** Sometimes it is possible to calculate a different quantity to the one of interest, one that can be argued to be ‘similar enough’ that its value is still of interest. In this case, interest can be transferred to this surrogate quantity, with an acknowledgement that the general information available to base inference on is reduced correspondingly. An example of such a strategy is the use of composite likelihood estimators, described in detail in sec 2.3, which maximise a product of low dimensional marginal or conditional likelihoods as a surrogate for the data likelihood.

**Numerical approximations** If numerical methods can return, with acceptable computational cost, a value that is sufficiently close to the quantity of interest then this can often be the optimal approximation strategy. A particular method could be deterministic or (pseudo) random, and some of such methods will share the characteristic that, given enough time to run, an arbitrarily precise approximation to the quantity of interest can be made. An example of such a strategy is Monte Carlo integration, described in detail in sec 2.4, where samples values are drawn from a distribution and their empirical distribution is subsequently treated as representative of the underlying distribution itself.

Another example of a numerical approximation is making a variational approximation to the evaluation of a log likelihood, described in detail in sec 2.2. As the approximation made takes the form of a tractable alternative quantity, the view that this is a surrogate quantity could be held with some justification. The current thesis takes the view that as this approximation needs to be recalculated for each value of  $\theta$ , the log likelihood function is not being replaced but merely its evaluation approximated. As such it is categorised as a numerical approximation, with an acknowledgement of other equally valid

perspectives.

## 2.2 Variational approximations

This section is going to describe variational approximations, a thorough treatment is given in Wainwright and Jordan (2008).

Variational approximations are a general class of approximations that directly approximate the data likelihood with a lower bound. As such they can be used in all inference tasks. They are an approximation method that trades bias for computational complexity / viability. Variational methods in general will first be described, and then their application to parameter estimation will be focussed on. They are of particular utility when the joint distribution of variables belongs to an *exponential family*. If a random vector belongs to an exponential family then its joint distribution can be written as:

$$\mathcal{P}(Z) = \exp(\langle \theta, \phi(Z) \rangle - A(\theta)) \quad (2.7)$$

where  $Z = \{X, Y\}$  contains both latent variable  $X$  and observed variables  $Y$ ,  $\langle x, y \rangle$  denotes the standard inner product between  $x$  and  $y$ ,  $\theta = (\theta_1, \dots, \theta_n)'$  is a vector of parameters and  $\phi(X) = (\phi_1(X), \dots, \phi_n(X))'$  is a vector of functions on the realised values of the variables.  $A(\theta) = \log \int \exp(\langle \theta, \phi(x) \rangle) dx$  is the *log partition function* and ensures the distribution is normalised. When a distribution from an exponential family is represented as a Markov random field, the associated factorisation will correspond to the elements of  $\phi(x)$ .

Variational approximations involve expressing calculations involving (2.7) in a form in which variational calculus can be used. When computations of marginal or conditional distributions, or of the log partition function, are intractable, variational calculus can be used to find the closest approximation subject to a specific set of constraints. These constraints will be chosen to introduce tractability to the computation, and if they are well chosen they will permit solutions that reasonably approximate their target.

As these methods directly introduce incorrect solutions to intractable calcu-

lations, they are necessarily biased. The theoretical properties of the bias are not completely understood (Jordan et al., 1999; Turner and Sahani, 2008), but in practice these methods can in some cases provide satisfactory performance. By construction they are tractable, and if the approximation constraints are suitably chosen then trade-offs between bias and computational complexity will be beneficial to practitioners.

There is a wide variety of applications for which variational approximations can be used. One of these is approximate maximum likelihood estimation of parameters, which is the focus of the following section.

### 2.2.1 Variational EM

*Variational EM* (VEM) is a methodology for making approximate inference in models with latent variables, and it is a generalisation of the EM framework used in Dempster et al. (1977) for making maximum likelihood estimates. While the procedure described here is for parameter estimates in a frequentist model, its Bayesian analogue is obtained by simply absorbing the parameter  $\theta$  into the latent variable  $X$ .

VEM works by placing a lower bound on the data log-likelihood using Jensen's inequality, which is then maximised using variational calculus. Inference is performed using the lower bound rather than the actual log-likelihood. In the following exposition, the statistical model of interest contains latent variables  $X$  and observed variables  $Y$ . It is assumed that the *data likelihood*  $\mathcal{P}(Y | \theta) = \int \mathcal{P}(x, Y | \theta) dx$  is intractable and needs to be approximated to make estimates of the parameter  $\theta$ .

The lower bound is introduced (Hathaway, 1986; Neal and Hinton, 1998) through taking the expectation of the *total log-likelihood*  $\log \mathcal{P}(X, Y | \theta)$  with re-

spect to an arbitrary distribution  $q(X)$ :

$$\begin{aligned}
\log \mathcal{P}(Y | \theta) &= \log \int \mathcal{P}(x, Y | \theta) \, dx \\
&= \log \int q(x) \frac{\mathcal{P}(x, Y | \theta)}{q(x)} \, dx \\
&\geq \int q(x) \log \frac{\mathcal{P}(x, Y | \theta)}{q(x)} \, dx \\
&= \mathbb{E}_q[\log \mathcal{P}(X, Y | \theta)] + H[q] \\
&= L(q, \theta)
\end{aligned} \tag{2.8}$$

where  $x$  denotes a particular value of the random variable  $X$ , and  $H[q] = -\int q(x) \log q(x) \, dx$  is the *differential entropy* of  $q$ , and the inequality is an application of *Jensen's inequality*:

$$\mathbb{E}[\phi(X)] \geq \phi(\mathbb{E}[X]) \tag{2.9}$$

for all convex functions  $\phi$ , and it is noted that  $-\log$  is convex.

Maximising  $L(q, \theta)$  over distributions  $q$  can be achieved using variational calculus, which is from where the method gets its name. If  $q$  can be chosen without constraint then the lower bound  $L(q, \theta)$  is maximised by the posterior distribution of the latent variables:

$$\arg \max_q L(q, \theta) = \mathcal{P}(X | Y, \theta) \tag{2.10}$$

and in fact equals the data log-likelihood in this case:

$$L(\mathcal{P}(X | Y, \theta), \theta) = \log \mathcal{P}(Y | \theta) \tag{2.11}$$

which is the result underpinning the common and popular *Expectation-Maximisation* (EM) algorithm.

It is not always possible to evaluate  $\mathcal{P}(X | Y, \theta)$ , in which case some other  $q$  can be chosen to derive a lower bound. If some tractable class  $Q$  of distributions is cho-

sen to maximise  $L(q, \theta)$  over, then the lower bound can be evaluated. Finding this  $q$  is analogous to calculating the posterior  $\mathcal{P}(X | Y, \theta)$ , and as such is often referred to as ‘inference’. Estimating the marginal data likelihood with its lower bound is similarly known as ‘learning’. If  $\mathcal{P}(X | Y, \theta)$  can be approximated reasonably well by distributions in  $\mathcal{Q}$ , then the resulting approximate inference and learning can be reasonably accurate (Jordan et al., 1999).

Precisely what is meant by one distribution approximating another reasonably well can be expanded on by re-writing (2.8) in a different form:

$$\begin{aligned} L(q, \theta) &= \int q(x) \log \frac{P(x, Y | \theta)}{q(x)} dx \\ &= \int q(x) \log \frac{\mathcal{P}(x | Y, \theta)}{q(x)} dx + \int q(x) \log \mathcal{P}(Y | \theta) dx \\ &= \log \mathcal{P}(Y | \theta) - \text{KL} [q(X) || \mathcal{P}(X | Y, \theta)] \end{aligned} \quad (2.12)$$

where  $\text{KL} [q(X) || \mathcal{P}(X | Y, \theta)]$  is the *Kullback-Leibler (KL) divergence* from  $q(X)$  to  $\mathcal{P}(X | Y, \theta)$ :

$$\text{KL} [f(X) || g(X)] = \int f(x) \log \frac{f(x)}{g(x)} dx \quad (2.13)$$

and is easily shown to be non-negative,  $\text{KL} [f(X) || g(X)] \geq 0$  with equality if and only if  $f(X) = g(X)$  (Bishop, 2007). This non-negativity illustrates that the closer a distribution  $q(X)$  is to the latent posterior  $\mathcal{P}(X | Y, \theta)$  in terms of KL divergence, the tighter the lower bound  $L(q, \theta)$  is to the true data log-likelihood  $\log \mathcal{P}(Y | \theta)$ .

A common choice of constrained distribution class  $\mathcal{Q}$  is the set of all distributions over  $X$  with a given factorisation, where the factorisation is *a*) disjoint, and *b*) finer than in the original model:

$$\mathcal{Q}_S = \left\{ q(X) : q(X) = \prod_{i=1}^M q_i(S_i) \right\} \quad (2.14)$$

for given  $S = \{S_i\}_{i=1}^M, \bigcup_{i=1}^M S_i = X$ , with  $S_i \subset \tilde{S}_j$  for some  $\tilde{S}_j \in \tilde{S}$  of the original factorisation  $\tilde{S}$ . The factorisation  $S$  will generally be chosen to introduce tractability

to  $L(q, \theta)$  with some nominally minimal number of additional conditional independences.

By choosing the  $q(X) \in Q$  that minimises the KL divergence from  $q$  to  $\mathcal{P}(X | Y, \theta)$ :

$$q^*(X) = \arg \min_{q \in Q} \text{KL}[q(X) || \mathcal{P}(X | Y, \theta)] \quad (2.15)$$

the lower bound is the tightest it can be when  $q$  is constrained to belong to  $Q$ .

When  $Q = Q_S$  as above, the  $q \in Q_S$  that optimises the lower bound is found through variational calculus to have the following form:

$$\begin{aligned} \log q_i^*(S_i) &= c + \mathbb{E}_{q_{\setminus i}^*}[\log \mathcal{P}(X, Y | \theta)] \\ \Rightarrow q_i^*(S_i) &= \frac{\exp(\mathbb{E}_{q_{\setminus i}^*}[\log \mathcal{P}(X, Y | \theta)])}{\int \exp(\mathbb{E}_{q_{\setminus i}^*}[\log \mathcal{P}(X, Y | \theta)]) dS_i} \end{aligned} \quad (2.16)$$

where  $c$  is a constant with respect to  $S_i$  and  $q_{\setminus i}^*$  refers to all factors  $q_j^*$ ,  $j \neq i$ . When the complete log-likelihood is in the exponential family, the normalisation in (2.16) can often be performed by visual inspection and comparison with known exponential family distributions.

As these distributions are coupled to each other through the expectation terms in (2.16), an iterative method can be used to find them. After initialising each  $q_i$ , the equations in (2.16) are implemented in turn, with the expectations taken using the newest estimate of each  $q_i$ . This is summarised in algorithm 2.1.

Finding the distribution in a given class  $Q$  that optimises the lower bound is the foundation of VEM, which is an iterative procedure. Once  $q_{\hat{\theta}^{(k)}}^*(X)$  has been found for a given estimate  $\hat{\theta}^{(k)}$  of  $\theta$ , the lower bound can be maximised over  $\theta$  by noting that only the first term in the right-hand side of (2.8) depends on  $\theta$ . Maximisation is therefore achieved by maximising the expected total log-likelihood (keeping the

**Algorithm 2.1** Finding the optimal  $q \in Q_S$

1. Initialise  $q_i^{(0)}$ ,  $i = 1, \dots, M$
2. Repeat until convergence:
  - (a) For each  $i$  in  $1, \dots, M$ :
    - i. Update estimate of  $q_i^*$  according to (2.16):

$$\log q_i^{(k+1)}(S_i) = c + \mathbb{E}_{q_{\setminus i}^{(l)}}[\log \mathcal{P}(X, Y | \theta)] \quad (2.17)$$

where

$$q_j^{(l)} = \begin{cases} q_j^{(k+1)} & j < i \\ q_j^{(k)} & j > i \end{cases} \quad (2.18)$$

distribution  $q$  fixed):

$$\begin{aligned} \hat{\theta}^{(k+1)} &= \arg \max_{\theta} L(q_{\hat{\theta}^{(k)}}^*, \theta) \\ &= \arg \max_{\theta} \mathbb{E}_{q_{\hat{\theta}^{(k)}}^*}[\log \mathcal{P}(X, Y | \theta)] \end{aligned} \quad (2.19)$$

The iterative scheme therefore consists of an expectation step, where the optimal  $q \in Q$  is chosen according to (2.15) and used to evaluate  $\mathbb{E}_q[\log \mathcal{P}(X, Y | \theta)]$ , followed by the maximisation step (2.19). This is summarised in algorithm 2.2. For the unconstrained class of all distributions over  $X$ , this procedure reduces to the standard EM algorithm and, on convergence, will return a local maximum of the data likelihood.

The simplest example of a tractable distribution of latent variables is the fully factorised distribution, also known in this context as the *mean-field approximation*. This is where the joint posterior distribution of all latent variables  $X$ , conditioned on observed data  $Y$ , completely factorises such that each factor contains only one variable. It corresponds to the sub-graph obtained by removing all edges between latent variables in the original graph.

As an illustrative example, consider a latent variable model where an ob-



**Algorithm 2.2** Variational EM

1. Initialise  $\hat{\theta}^{(0)}$
2. Repeat until convergence:
  - (a) Maximise lower bound  $L(q, \theta)$  over distributions  $q \in \mathcal{Q}$  according to (2.15) and evaluate  $\mathbb{E}_{q_k^*}[\log \mathcal{P}(X, Y | \theta)]$ :

$$q_k^*(X) = \arg \min_{q \in \mathcal{Q}} \text{KL} \left[ q(X) \parallel \mathcal{P}(X | Y, \hat{\theta}^{(k)}) \right] \quad (2.20)$$

- (b) Maximise lower bound with respect to  $\theta$ :

$$\hat{\theta}^{(k+1)} = \arg \max_{\theta} \mathbb{E}_{q_k^*}[\log \mathcal{P}(X, Y | \theta)] \quad (2.21)$$



**Figure 2.1:** Latent variable model (left) and the tractable sub-graph (right) corresponding to a variational approximation. Observed variables are shaded grey.

served variable  $Y$  is Gaussian with variance  $\sigma^2$  and mean given by the sum of two latent independent standard Gaussians,  $X_1$  and  $X_2$  i.e.  $Y | X \sim N(X_1 + x_2, \sigma^2)$ ,  $X_1, X_2 \sim^{\text{i.i.d.}} N(0, 1)$ .

The latent variables  $X_i$  are conditionally dependent given the observation  $Y$ , so their posterior distribution will not factorise between them. This fact is represented in an undirected graph of the distribution by an edge between them, so the three variables together form a factor in the joint distribution, see fig 2.1.

Approximating the true posterior distribution with a fully factorised distribution is equivalent to approximating the original graph with the sub-graph having all edges between the latent variables removed. The specific fully factorised distribution is chosen to minimise the KL divergence from it to the posterior as described above. For a detailed description of variational methods see Wainwright and Jordan (2008), and for a detailed study of variational approximations in a Bayesian setting, see Beal (2003).

## 2.2.2 Limitations and mitigations

In general, the bias introduced to parameter estimation using variational EM is not well understood. Some work has been done on specific models, in particular the state space model (Wang and Titterington, 2004; Turner and Sahani, 2008). The work in Wang and Titterington (2004) focusses on the linear Gaussian state space model in its one dimensional form. This model operates according to the following dynamics:

$$\begin{aligned}\mathcal{P}(X_1) &= \mathcal{N}(X_1 | \mu_0, \sigma_0^2) \\ \mathcal{P}(X_{t+1} | X_t) &= \mathcal{N}(X_{t+1} | \alpha X_t, \sigma_x^2) \\ \mathcal{P}(Y_t | X_t) &= \mathcal{N}(Y_t | \gamma X_t, \sigma_y^2) \quad t = 1, \dots, T\end{aligned}\tag{2.22}$$

For more details on the state space model, see sec 3.1 in chapter 3.

The authors assume all variances are equal i.e.  $\sigma_0^2 = \sigma_x^2 = \sigma_y^2 = \sigma^2$ , and that all parameters except  $\alpha$  are known. They aim to estimate  $\alpha \in (0, 1)$ . Under these conditions the true posterior can be calculated analytically, so no approximation is needed and exact EM can be used to obtain a maximum likelihood estimated. As the true posterior is known, it can be compared to its variational approximation and the KL divergence between them can be studied.

The class of variational approximation used in this study is the fully factorised product distributions  $Q = Q_S S = \cup_t X_t$  described above. The KL divergence  $\text{KL}[q^*(X) || \mathcal{P}(X | Y, \theta)]$  is calculated analytically, and its dependence on  $T$ ,  $\alpha$ ,  $\gamma$  and  $\sigma^2$  is investigated.

The authors find that  $\text{KL}[q^*(X) || \mathcal{P}(X | Y, \theta)] \rightarrow 0$  as  $t \rightarrow \infty$ , i.e the KL divergence will not go to zero as the number of observations increases. This implies that the variational EM estimate of  $\alpha$  is not consistent. It was also found that  $\text{KL}[q^*(X) || \mathcal{P}(X | Y, \theta)]$  was independent of  $\sigma^2$  and as such the bias will not go to zero even if the noise becomes very small. Furthermore, the KL divergence will not go to zero for any value of  $\gamma$  either. On the other hand, it was found that  $\text{KL}[q^*(X) || \mathcal{P}(X | Y, \theta)] \rightarrow 0$  as  $\alpha \rightarrow 0$ , so the bias will be small if  $\alpha$  is small.

The fully factorised approximation is not the only form the approximating distribution can take. Structure can be defined within the factors, which corresponds to keeping some but not all edges between latent variables from the original graph. It is such structured approximations that will be used in the current thesis. The modern study of the use of structured mean-field approximations originated with Saul and Jordan (1996), and developments in the area have continued since then. For details, see amongst others Wiegerinck (2000); Jaakkola (2001); Jaakkola and Jordan (1998).

An example of a structured approximation is studied in Barber and Wiegerinck (1999). In this paper, approximations to both undirected and directed graphs are described. The undirected graph that is approximated is that of a *Boltzmann machine*. A Boltzmann machine is made up of a vector  $X$  of binary variables, only some of which are in general observable. The probability of any particular configuration  $x$ , parametrised by a symmetric weight matrix  $W$ , is:

$$\mathcal{P}(X = x) = \frac{1}{Z} \exp\left(\sum_{i,j} W_{i,j} x_i x_j\right) \propto \exp(x' W x) \quad (2.23)$$

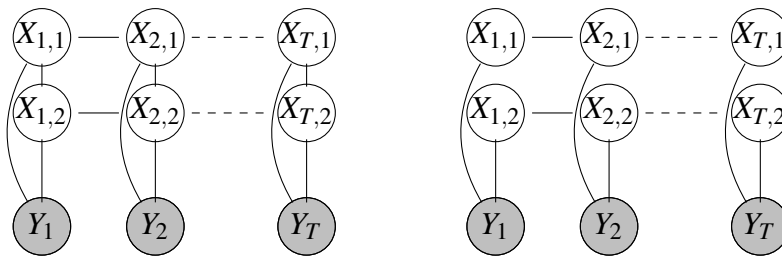
where  $Z$  is a normalisation constant known as the *partition function*. Calculating the partition function generally involves summing over  $2^{|X|}$  possible states, which becomes intractable for large  $|X|$ . One of the findings in this paper was a principled method of defining sub-graphs of the original (fully connected) graph, whose probabilities are tractable. These sub-graphs are then used to perform variational EM and learn the optimal parameters for the observed variables. The authors find that using such structured approximations can bring a considerable improvement in performance over using a fully factorised approximation.

Another example is Ghahramani and Jordan (1997). In this paper, the authors approximate the posterior distribution of hidden states in a *factorial hidden Markov model*, which is an extension to the *hidden Markov model*. The hidden Markov model is an example of a state space model, which are described in further detail in chapter 4. A factorial hidden Markov model makes use of multiple independent

Markov chains, and at each time point a linear combination of their values is used to determine the distribution of the observed variable. In this paper, for example, the observed variable is a Gaussian vector with mean given by a linear combination of the latent variables:

$$\mathcal{P}(Y_t | X_t) = \mathcal{N}(Y_t | WX_t, C) \quad (2.24)$$

where  $W$  is a weight matrix and  $C$  is a common covariance matrix. The authors approximate the posterior of the hidden variables with a distribution that factorises over the Markov chains at each time step, but does not factorise over time. This is equivalent to using a sub-graph that keeps only the edges over time between the nodes in each Markov chain as an approximation, see fig 2.2.



**Figure 2.2:** Factorial hidden Markov model (left) and the tractable sub-graph (right) corresponding to a variational approximation. Observed variables are shaded grey. Dashed lines indicate a repeating pattern until time  $T$ .

As the latent state space is finite, exact computation of the posterior is possible but very slow. Furthermore, the authors found that using exact EM was liable to over-fit the model to the data. Variational EM was found to offer similar performance and run time to using an MCMC approach (see sec 2.4.3 for more details on MCMC).

## 2.3 Composite likelihoods

This section is going to describe composite likelihoods and maximum composite likelihood estimators, and how they can provide a computationally feasible approach to inference that can have consistency guarantees and understood asymptotic estimator distributions. A thorough review of composite likelihood methods

can be found in Varin et al. (2011), which should be referred to for more detail on the theoretical results stated here.

Composite likelihoods are used as alternatives to data likelihoods for performing parameter estimation. They are generally used when evaluation of the data likelihood is either impossible or impractical. The first example of a composite likelihood being used in practice was in Besag (1974), in which a product of conditional likelihoods is used in the place of the data likelihood.

When performing parameter estimation, a common general method is to maximise an objective function of data and parameters with respect to the parameters. A common choice of objective function is the likelihood function, which returns the maximum likelihood estimator (MLE) as the result of maximisation. Maximum likelihood estimators are commonly used in part because they have desirable properties. They are consistent and have optimal statistical efficiency, i.e. they achieve the Cramér-Rao lower bound.

If evaluating or maximising the likelihood function is difficult or impossible then an alternative to the MLE can be used in its place. One possibility is to use composite likelihoods, which are products of marginal or conditional likelihoods of subsets of the data. An estimator that maximises such an objective function is known as a composite likelihood estimator.

Under suitable conditions, composite likelihood estimators are consistent, but will have a higher variance than, for example, maximum likelihood estimators. In addition to the standard conditions for MLE consistency, is the condition that the data in each component interacts with the parameters being estimated; otherwise any estimates would not depend on data. Unlike the MLE, maximum composite likelihood estimators do not achieve the Cramér-Rao lower bound for estimator variance. One motivation for using a composite likelihood objective function is as a trade-off of reduced computational costs / practical viability against an increase in estimator variance.

A composite likelihood is a surrogate for the likelihood function, made from the low dimensional marginal or conditional distributions of subsets of the data. A

set  $\{A_i | i = 1, \dots, k\}$  of  $k$  marginal or conditional data events are defined, each of which has an associated likelihood  $L_i(\theta | A_i)$  under the model being used for the full (intractable) joint distribution. A weighted product of these associated likelihoods,  $L_C(\theta | Y) = \prod_{i=1}^k L_i(\theta | A_i)^{w_i}$ , is known as a composite likelihood.

---

**Composite likelihood** A *composite likelihood* is a product

$$L_C(\theta | Y) = \prod_{i=1}^k L_i(\theta | A_i)^{w_i} \quad (2.25)$$

of low dimensional marginal or conditional distributions  $L_i(\theta | A_i)$  derived from an intractable joint distribution over  $Y$ .

---

A composite likelihood is used in inference just as a full likelihood normally would be. Wherever the full likelihood appears in a procedure for parameter estimation or for calculating the posterior distribution of latent variables, it is simply replaced with the composite likelihood. If the log-composite likelihood is denoted  $\ell_C(\theta | Y)$ , the composite likelihood estimator is then

$$\hat{\theta}_{cl} = \arg \max_{\theta} \ell_C(\theta | Y) \quad (2.26)$$

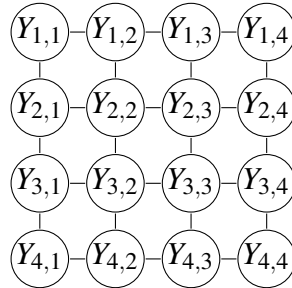
The non-negative weights  $w_i$  can be chosen to improve efficiency, if they are all equal then their particular common value does not affect the results of any inference performed. If unequal weights are chosen then they can be set to depend on the size of the sets  $A_i$ , or on the strength of the correlations of the data inside each set. It is common though, for them to be set to be equal.

Whilst the definition of a composite likelihood allows the use of both marginal and conditional densities in the same product, in practice it is not common for them to be mixed. The motivations for using either form are generally different from each other, so their appearance in the same product is unlikely.

The motivations for using a composite conditional likelihood can be illustrated with an example of spatial data. With spatial data, it might be possible to assume that each data point is strongly dependent on those data points close to it, and almost conditionally independent of those which are not close. A conditional likelihood for each data point could be far easier to calculate than a low-dimensional marginal, and it can be argued heuristically that the weak dependence between distant data points justifies taking the product of each conditional likelihood. If some data had a spatial grid structure as in fig 2.3, for example, and dependencies between points were only strong between neighbours, then the composite likelihood

$$L_C(\theta | Y) = \prod_{i,j} \mathcal{P}(Y_{i,j} | N(Y_{i,j}), \theta) \quad (2.27)$$

where  $N(Y_{i,j})$  denotes the horizontal and vertical neighbours of  $Y_{i,j}$ , would capture most of the interactions in the full joint distribution. In situations analogous to this, the use of a composite conditional likelihood should have relatively good performance.



**Figure 2.3:** A possible graphical model illustrating spatial arrangement of observed data, where each data point could be modelled to have strong dependencies only on those data that are adjacent on the grid

In the case of composite marginal likelihood, however, the motivations for its use are different. Pairwise composite marginal likelihoods, for example:

$$L_C(\theta | Y) = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathcal{P}(Y_i, Y_j | \theta) \quad (2.28)$$

can be appropriate if the full likelihood is difficult to compute and higher order cor-

relations are not of particular interest. In general, composite marginal likelihoods can be appropriate surrogate functions when the information contained in each subset of data regarding correlations is sufficient for the objective of the research. This perspective can be particularly pertinent if the true likelihood contains nuisance parameters that complicate estimation of the parameters of interest.

It is important to note that factors in a composite likelihood function need not themselves be tractable. Approximation method methods might be required for the calculation of the factors themselves. As described below, using a composite likelihood will increase the variance of an estimator. If data is segmented into  $n$  equal sized blocks then the variance inflation due to using a composite likelihood of parameters will diminish as the size of the blocks goes to the size of the dataset and the number  $n$  of blocks goes to 1. This statement is made more precisely below, but heuristically it implies an incentive to use larger factors in the composite likelihood. Any approximation method used for each factor will likely be less accurate or more computationally expensive for large factors though, and these considerations will affect the decision on how large the factors should be.

As each factor in the composite likelihood is a marginal or conditional density, the resulting estimating equation  $\nabla_{\theta} \log L_C(\theta | Y) = \nabla_{\theta} \ell_C(\theta | Y)$  is unbiased, i.e.

$$\mathbb{E}[\nabla_{\theta} \ell_C(\theta | Y)] = 0 \quad (2.29)$$

so subject to mild regularity conditions, notably among them that the data in each component is sufficient to make an estimate of each parameter in  $\theta$ , maximum composite likelihood estimators will, in the limit of infinite data, converge to the true value of  $\theta$ , i.e. they are consistent. Additionally, the use of a composite likelihood can be considered equivalent to using a misspecified model, and from this perspective it can be shown that the variance of maximum composite likelihood estimators will be higher than for maximum likelihood estimators.

In particular, if data is segmented into  $n$  equal sized blocks and the blocks are modelled as i.i.d replicates of each other then a central limit theorem can be stated, see Varin et al. (2011) sec 2.3. If  $u(\theta | Y) = \nabla_{\theta} \ell_C(\theta | Y)$  is the gradient of the



composite log-likelihood, and defining

$$H(\boldsymbol{\theta}) = \mathbb{E}[-\nabla_{\boldsymbol{\theta}} u(\boldsymbol{\theta} | Y)] \quad (2.30)$$

and

$$J(\boldsymbol{\theta}) = \text{var}[u(\boldsymbol{\theta} | Y)] \quad (2.31)$$

then the maximum composite likelihood estimator  $\hat{\boldsymbol{\theta}}_{cl}$  has an asymptotically normal distribution:

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{cl} - \boldsymbol{\theta}) \rightarrow_d N(0, G^{-1}(\boldsymbol{\theta})) \quad (2.32)$$

where  $G(\boldsymbol{\theta})$  is the *Godambe information matrix* (Godambe, 1960):

$$G(\boldsymbol{\theta}) = H(\boldsymbol{\theta})J^{-1}(\boldsymbol{\theta})H(\boldsymbol{\theta}) \quad (2.33)$$

In the case where the true likelihood is used, producing the mle as the estimator, then  $H(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) = I(\boldsymbol{\theta})$  is the Fisher information matrix, and the estimator asymptotically achieves the Cramér-Rao lower bound. As all estimator variances are greater than or equal to the Cramér-Rao lower bound, the asymptotic variance in the general composite likelihood case will, therefore, be greater than that of the mle.

In practice this means that a composite likelihood function that contains many ‘small’ factors, i.e. with each factor containing only a small subset of the total number of random variables, could easily produce an estimator with much higher variance than the mle. Computational concerns provide an incentive to use a composite likelihood that is easy to evaluate, but which may also suffer from high estimator variance. It is in this way that using composite likelihood forces a trade-off between computational complexity and variance.

Tests of alternative hypotheses analogous to likelihood ratio, Wald, and score tests that use maximum likelihood are available for use with composite likelihoods.

Molenberghs and Verbeke (2005) describe analogues to Wald and score tests. Various options have been proposed as analogues to the likelihood ratio test. The direct analogue:

$$W = 2[\ell_C(\hat{\theta}_{H1}) - \ell_C(\hat{\theta}_{H0})] \quad (2.34)$$

where  $\ell_C(\hat{\theta}_{H0})$  and  $\ell_C(\hat{\theta}_{H1})$  refer to the values of the log composite likelihood using the minimal and augmented parameter vectors respectively, has a non-standard asymptotic distribution, which can make computation difficult (Kent, 1982). Alternatives which have various approximate  $\chi^2$  distributions have been proposed by Geys et al. (1999); Rotnitzky and Jewell (1990); Varin (2008) amongst others.

Model comparisons can also be made using composite likelihood. Analogues of both the AIC and BIC have been derived, and they correspond very closely to their standard likelihood counterparts. In both cases, the number of parameters is replaced with the number of ‘effective’ parameters, defined as  $\dim \theta = \text{trace}(H(\theta)G(\theta)^{-1})$  with  $G(\theta)$  and  $H(\theta)$  as defined above. Derivations of these criteria can be found in Varin and Vidoni (2005) and Gao and Song (2010) for the AIC and BIC respectively.

A large amount of research has been performed investigating composite likelihood approaches, including the use of the AIC and BIC analogues for model averaging (Claeskens and Hjort, 2008). The interested reader is referred to Varin et al. (2011) for a detailed review.

### 2.3.1 Specialised models and applications

An early example of composite likelihood being used in practice is Besag (1974). In this paper, the author analyses spatial models with variables in the model lying on a lattice. Particular examples of such models include infection or yield studies of plants, with each plant placed in the lattice structure. Conditional likelihoods for each variable given its nearest neighbours were used to construct the surrogate objective function, as in the example (2.27) and fig 2.3, to avoid the computational challenges of constructing the full joint likelihood.

An example of a marginal composite likelihood being used in practice is Jöreskog and Moustaki (2001). Factor analysis is extended to the case of ordinal response variables, with each value given a probability of occurrence conditioned on the jointly normally distributed latent variables. One of their proposals is to replace the data likelihood with the product of all univariate and bivariate marginal likelihoods. These marginals are calculated using quadrature to integrate out the latent factors, and the computational cost of doing this is the motivation for not using the full data likelihood.

A more recent example is Vasdeskis et al. (2012), in which a random effects model for ordinal longitudinal data is modelled to include latent variables for each person / time point combination. They model data  $Y$  which can be indexed by person, item, and time, i.e.  $Y = \{Y_{j,i,t} \mid j = 1, \dots, n, i = 1, \dots, p, t = 1, \dots, T\}$ . To reduce the computational complexity of inference they replace the full likelihood with the product of all bivariate (over items and times) likelihoods, similarly to in the example (2.28):

$$L_C(\theta \mid Y) = \prod_{(j,i,t),(j',i',t')} \mathcal{P}(Y_{j,i,t}, Y_{j',i',t'} \mid \theta) \quad (2.35)$$

One feature of composite likelihoods that can negatively affect their computational profiles is the number of components that they contain. The number of bivariate marginals, for example, is quadratic in the size of the dataset. By choosing which components are used in a controlled probabilistic manner, a trade-off between computation time and statistical efficiency can be made. This idea is explored in Dillon and Lebanon (2010), and is discussed further in sec 2.6.2.2.

---

**Remark** Time series models offer an obvious method of defining the events  $\{A_i \mid i = 1, \dots, k\}$ . As the data in a time series model is by definition ordered by time, it can easily be broken up over time. Breaking up the full data  $\{Y_j \mid j = 1, \dots, T\}$  into sub-intervals  $A_i = \{Y_{I_i} \mid i = 1, \dots, k\}$  and taking the product of their likelihoods as the composite likelihood provides a surrogate for the full likelihood that can, for

models with other approximation methods available, be compared against benchmark methods.

---

**Remark** Papers in which composite likelihood and EM type methods are combined are not uncommon, and include Liang and Yu (2003); Varin et al. (2005); Gao and Song (2011). In Gao and Song (2011), the authors show that using EM in a composite likelihood produces an algorithm that shares the convergence properties of standard EM algorithms whilst having the reduced computational complexity of a composite likelihood.

It should be noted that if any latent variable appears in more than one component, then its expectation will differ in each component. The expectations in each component are conditioned on the data in that component only, and as such the optimal  $q^*$  distributions will not be equal.

It should also be noted that without some form of conditional independence structure between latent and observed variables, all latent variables would generally appear in all components. This can significantly increase the computational complexity of evaluating each component likelihood. By introducing a Markov structure to a model, where conditional independences allow most latent variables to be trivially integrated out of each component likelihood, this problem is avoided.

Any remaining latent variables in each component can be integrated out via, for example, taking expectations as in EM type algorithms or, if the dimension of the latent variable is low in each component, as in Jöreskog and Moustaki (2001), then quadrature is also an option. If the latent variables are modelled to have a Markov structure then the problem reduces to integrating out only the parents for each component likelihood. If each observed variable is modelled to have only few latent parents, whose marginals are known or easy to calculate, then the computational cost of evaluating the component likelihoods can be minimised.

In state space models (see sec 2.4.2 and chapter 3 sec 3.1), for example, if the

latent process is modelled as a stationary Gaussian process then calculations will be simplified massively. The marginal distribution of only the latent parents in a component factor can be easily calculated analytically in this case, and used in the calculation of each component likelihood.

---

## 2.4 Monte Carlo

*Monte Carlo* approximations are amongst the most widespread numerical integration techniques in current use, as they are generally easy to construct and can be applied in almost all contexts. Furthermore, their approximation error goes to zero as the number of samples use in the approximation increases, so given enough computing resources and/or time to run they can be used to approximate almost any quantity of interest. A thorough treatment of Monte Carlo methods can be found in Rubinstein and Kroese (2011), and a condensed course covering most of the same material is available in Kroese (2011).

As the running theme of the current thesis makes clear, computational constraints are often relevant in an applied context. This prevents Monte Carlo methods from being a panacea, as some intractable model structures can require a prohibitive number of samples to achieve an acceptable approximation error. In such cases Monte Carlo algorithms may not provide an optimal trade-off between computational costs and statistical efficiency.

A Monte Carlo approximation is an estimate of the expected value of a function of a random variable with a given distribution:

---

**Monte Carlo approximation** A *Monte Carlo* approximation to  $\mathbb{E}_\theta[f(X)]$ , the expected value of a function  $f$  of a random variable  $X$  distributed as  $\mathcal{P}(X | \theta)$ , is the sample mean of function values drawn either from  $\mathcal{P}(X | \theta)$  or from a sampling

distribution  $q(X)$  that approximates  $\mathcal{P}(X | \theta)$ :

$$\begin{aligned} \mathbb{E}_\theta[f(X)] &\approx \hat{E}_\theta[f] \\ &= \frac{1}{N} \sum_{i=1}^N f(X^{(i)}) \quad X^{(i)} \sim q(X), q(X) \approx \mathcal{P}(X | \theta) \end{aligned} \quad (2.36)$$

(2.37)

with approximations converging in distribution as the number of samples increases:  $\tilde{\mathcal{P}}_N(X) \rightarrow_d \mathcal{P}(X | \theta)$  as  $N \rightarrow \infty$ . When samples are drawn from the true distribution  $\mathcal{P}(X | \theta)$  then  $\hat{E}_\theta[f]$  is unbiased. The approximations are consistent:  $\hat{E}_\theta[f] \rightarrow \mathbb{E}_\theta[f(X)]$  as  $N \rightarrow \infty$ . The variance of the approximation is  $\text{var}(\hat{E}_\theta[f]) = \frac{1}{N} \text{var}(f(X))$ .

---

As  $f$  can be taken to be  $f = \mathbb{1}_{X^{(i)} \leq x}$ , the distribution function  $F_\theta(x) = \mathcal{P}(X \leq x | \theta)$  of one dimensional distributions and multi-dimensional generalisations  $F_\theta(A) = \mathcal{P}(X \in A | \theta)$  can be estimated. The density function can therefore be approximated by a weighted sum of Dirac delta functions centred at each sample:

$$f(x) = \frac{d}{dx} F_\theta(x) \approx \frac{1}{N} \sum_{i=1}^N \delta(x - X^{(i)}) \quad (2.38)$$

Ideally the samples  $X^{(i)}$  will be drawn from the distribution of interest, but in general this cannot be guaranteed. Some common strategies for overcoming this challenge are discussed in the following sections, but the topic of variance reduction deserves a mention beforehand.

Variance reduction techniques are, as the name suggests, methods of reducing the variance of  $\hat{E}_\theta[f]$ . Two commonly used methods are *antithetic sampling* and *control sampling* (Kroese, 2011). Antithetic sampling is a method available for any one dimensional distribution that is sampled using its inverse distribution function,

i.e.

$$X^{(i)} = F^{-1}(U^{(i)}) \quad U^{(i)} \sim \mathcal{U}(0, 1) \quad (2.39)$$

and simply involves taking both  $U^{(i,1)} = U^{(i)}$  and  $U^{(i,2)} = 1 - U^{(i)}$  as samples for each  $i$ . Both of the corresponding samples  $X^{(i,1)}, X^{(i,2)}$  will have the same marginal distribution but they will be negatively correlated, and if  $f$  is monotonic then so will  $f(X^{(i,1)}), f(X^{(i,2)})$ . If each  $U^{(i)}$  is drawn i.i.d from  $\mathcal{U}(0, 1)$  then the variance of  $\hat{E}_\theta[f]$  (approximated from  $M = \frac{N}{2}$  draws for comparative purposes) therefore becomes:

$$\begin{aligned} \text{var}(\hat{E}_\theta[f]) &= \text{var}\left(\frac{1}{N} \sum_{i=1}^M \left(f(X^{(i,1)}) + f(X^{(i,2)})\right)\right) \\ &= \frac{1}{N} \left(\text{var}(f(X)) + \text{cov}\left(f(X^{(i,1)}), f(X^{(i,2)})\right)\right) \\ &\leq \frac{1}{N} \text{var}(f(X)) \end{aligned} \quad (2.40)$$

where the last line is in reference to the variance of the original Monte Carlo estimate noted in the definition above. Antithetic sampling can be extended to elliptical multivariate distributions with known mean vector  $\mu$  by defining the antithetic sample to be  $X^{(i,2)} = \mu - X^{(i,1)}$ .

Control sampling is a related method to antithetic sampling, in that an extra term correlated to only one sample is added to the sum of sampled function values, but it can only be used in the restricted context of there being an additional function  $g(X)$  with known expected value and non-zero correlation with  $f(X)$ . In this context, the following estimator can be defined:

$$\hat{E}_\theta[f]^* = \alpha \mathbb{E}_\theta[g(X)] + \frac{1}{N} \sum_{i=1}^N \left(f(X^{(i)}) - \alpha g(X^{(i)})\right) \quad (2.41)$$

and is trivially shown to have the expected value of interest. The variance of  $\hat{E}_\theta[f]^*$

is:

$$\text{var}(\hat{E}_\theta[f]^*) = \frac{1}{N} (\text{var}(f(X)) + \alpha^2 \text{var}(g(X)) - 2\alpha \text{cov}(f(X), g(X))) \quad (2.42)$$

which can be minimised with respect to  $\alpha$ :

$$\alpha^* = \arg \min_{\alpha} \text{var}(\mathbb{E}[f^*]) = \frac{\text{cov}(f(X), g(X))}{\text{var}(g(X))} \quad (2.43)$$

giving an optimised variance of:

$$\text{var}(\hat{E}_\theta[f]^*) = \frac{1}{N} (1 - \rho^2 \text{var}(f(X))) \quad (2.44)$$

where  $\rho = \text{corr}(f(X), g(X)) \in [-1, 1]$ .

If the values of  $\text{cov}(f(X), g(X))$ ,  $\text{var}(g(X))$  are not known then they can be estimated from the samples. This method obviously relies heavily on the existence of a function  $g$  with known mean and non-negligible correlation  $c \text{corr}(f(X), g(X))$ , so its utility cannot be universally exploited.

Where possible, either or both of antithetic sampling and control sampling can be used in conjunction with the methods described below, particularly with importance sampling. The following sections describe various methods for making Monte Carlo estimates when the distribution of interest cannot be sampled from directly.

The following sections refer to both a function whose expected value is of interest and to the density function  $\frac{d\mathcal{P}}{dX}$ . Conventional notation commonly denotes both of these functions as  $f$ ; for clarity they will subsequently be denoted as:

- $h(x)$ : the function whose expected value  $\mathbb{E}_\theta[h(X)]$  is of interest.
- $f_\theta(x)$ : the density function  $f_\theta(x)$  of a distribution  $\mathcal{P}(X | \theta)$

### 2.4.1 Importance Sampling

*Importance sampling* (Kroese, 2011) can be summarised as the use of samples from a distribution other than the one of interest in a weighted sum that produces a consis-



tent estimator. When the distribution of interest can be sampled from directly then, if an appropriate *importance distribution* can be found, it is a method of variance reduction.

It can also be used when the distribution of interest cannot be sampled from, and it is in this context that it has relevance to the current thesis. All that is required to make consistent Monte Carlo estimates  $\hat{E}_\theta[h] \approx \mathbb{E}_\theta[h(X)]$  using importance sampling is *a*) knowledge, up to a proportional constant, of the density function  $f_\theta(x) \propto g(x)$  and *b*) an importance distribution that can be sampled from with a known, up to a proportional constant, density function  $p(x) \propto q(x)$ , and whose support contains that of  $f_\theta(x)$ :  $f_\theta(x) > 0 \Rightarrow p(x) > 0$ .

Approximating expectations under one distribution using samples drawn from another rests on the following identity:

$$\begin{aligned} \mathbb{E}_\theta[h(X)] &= \int f_\theta(x)h(x) dx \\ &= \int p(x)\frac{f_\theta(x)}{p(x)}h(x) dx \\ &= \mathbb{E}_p\left[\frac{f_\theta(X)}{p(X)}h(X)\right] \\ &\approx \frac{1}{N}\sum_{i=1}^N w_i h(X^{(i)}) \quad X^{(i)} \stackrel{\text{i.i.d.}}{\sim} p(X) \end{aligned} \quad (2.45)$$

where  $w_i = \frac{f_\theta(X^{(i)})}{p(X^{(i)})}$  are known as *importance weights*.

If either or both of  $f_\theta(x), p(x)$  are only known up to a proportional constant, i.e.  $f_\theta(x) \propto g(x)$  or  $p(x) \propto q(x)$ , which is a common situation in practice, then using normalised importance weights produces a biased but consistent estimator:

$$\begin{aligned} z_i &= \frac{g(x_i)}{q(x_i)} \\ \tilde{z}_i &= \frac{z_i}{\sum_i z_i} \\ &= \frac{w_i}{\sum_i w_i} = \tilde{w}_i \\ \sum_{i=1}^N \tilde{w}_i h(X^{(i)}) &\xrightarrow{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N w_i h(X^{(i)}) \end{aligned} \quad (2.46)$$

Whilst the derivations (2.45), (2.46) are theoretically valid, in practice the results of using importance sampling can be varied. If the proposal distribution  $p(X)$  does not share areas of high density with  $f_\theta$  then the importance weights can have high variance. In an extreme example, if an area of  $\text{supp } p = \{x: p(x) > 0\}$  has very low density under  $p$  but very high density under  $f_\theta$  then any finite sample could possibly contain some (but probably not many) samples from this area, and the weights of these samples would dominate the approximation  $\hat{E}_\theta[h]$ .

This concept can be formalised with the notion of the *effective sample size* associated with a given set of normalised importance weights  $w_i$ . Defined as:

$$N_{\text{eff}} = \frac{1}{\sum_i w_i^2} \quad (2.47)$$

we have  $N_{\text{eff}} = N$  for the equal weighting  $w_i = w_j \forall i, j$  and  $N_{\text{eff}} \rightarrow 1$  as the number of negligible weights increases. The effect on the variance of  $\hat{E}_\theta[h]$  of a low  $N_{\text{eff}}$  can be illustrated with an (unrealistic) example where the importance weights are fixed and such that  $N_{\text{eff}} = M < N$ :

$$\begin{aligned} \text{var}(\hat{E}_\theta[h]) &= \text{var}_p \left( \sum_{i=1}^N w_i h(X^{(i)}) \right) \\ &= \sum_{i=1}^N w_i^2 \text{var}_p(h(X^{(i)})) \\ &= \frac{1}{M} \text{var}_p(h(X)) \end{aligned} \quad (2.48)$$

which clearly increases as  $N_{\text{eff}} \downarrow 1$ . True normalised importance weights would obviously co-vary with  $h(X)$  so (2.48) would not strictly hold, but it still serves an illustrative purpose.

## 2.4.2 Particle filters

*Particle filters* are a form of Monte Carlo approximation to the posterior distribution of latent variables in a *state space model*. State space models are described in more detail in sec 4, but they can be quickly described as time series models with a Markov latent process  $\{X_t\}_{t=1}^T$  and observations  $\{Y_t\}_{t=1}^T$  at each time that are

conditionally independent given the latent process:

$$\begin{aligned}
X_1 &\sim \mathcal{P}(X_1 \mid \boldsymbol{\theta}) \\
X_t \mid X_{1:t-1} &\sim X_t \mid X_{t-1} \sim \mathcal{P}(X_t \mid X_{t-1}, \boldsymbol{\theta}) \quad t \in 2, \dots, T \\
Y_t \mid X_{1:T} &\sim Y_t \mid X_t \sim \mathcal{P}(Y_t \mid X_t, \boldsymbol{\theta}) \\
(Y_{t_1} \perp Y_{t_2}) \mid X_{1:T} & \quad 1 \leq t_1, t_2 \leq T \quad (2.49)
\end{aligned}$$

where the colon in subscripts  $t_1 : t_2$  indicates the interval of time periods  $t_1, \dots, t_2$ . The latent process takes values in some space  $\mathcal{X}$ , and observations take values in some space  $\mathcal{Y}$ . All distributions in (2.49) are assumed to have density functions  $f_{\boldsymbol{\theta}}(x_1), f_{\boldsymbol{\theta}}(x_t \mid x_{t-1}), f_{\boldsymbol{\theta}}(y_t \mid x_t)$ .

Common objects of inferential interest in the state space model are the sequence of *filtered* distributions:

$$\mathcal{P}(X_t \mid Y_{1:t}, \boldsymbol{\theta}) \quad t \in 1, \dots, T \quad (2.50)$$

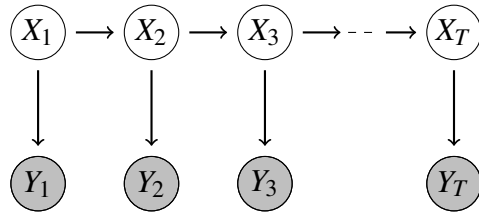
and the *smoothed* distributions:

$$\mathcal{P}(X_t \mid Y_{1:T}, \boldsymbol{\theta}) \quad t \in 1, \dots, T \quad (2.51)$$

and estimating either of these has computational complexity that, without the modelled Markov property of the latent process, increases exponentially with  $T$ , the number of observations.

The Markov property of the latent process allows the filtered distributions  $\mathcal{P}(X_t \mid Y_{1:t})$  for  $t \in 1, \dots, T$  to be estimated with a cost that is linear in  $T$ . This can be seen from the graphical representation of the state space model, shown in fig 2.4. The graph, when moralised (see sec 2.5.1.2 for a definition), is clearly a tree, so message passing algorithms to calculate both the filtered and smoothed distributions can be constructed.

Calculating the filtered distributions can be achieved via a two-stage recursive



**Figure 2.4:** The directed graph of the state space model. Observed variables are shaded grey, dashed line indicates repeated pattern until  $t = T$ .

procedure, which is composed of a *prediction step*:

$$f_{\theta}(x_t | Y_{1:t-1}) = \int f_{\theta}(x_{t-1} | Y_{1:t-1}) f_{\theta}(x_t | x_{t-1}) dx_{t-1} \quad (2.52)$$

that makes use of the filtered distribution at the previous time point. After this there is an *update step*:

$$\begin{aligned} f_{\theta}(x_t | Y_{1:t}) &= \frac{f_{\theta}(x_t | Y_{1:t-1}) f_{\theta}(Y_t | x_t)}{\int f_{\theta}(x_t | Y_{1:t-1}) f_{\theta}(Y_t | x_t) dx_t} \\ &= \frac{f_{\theta}(x_t | Y_{1:t-1}) f_{\theta}(Y_t | x_t)}{f_{\theta}(Y_t | Y_{1:t-1})} \end{aligned} \quad (2.53)$$

Furthermore, if the quantities in the denominator of the right hand side of (2.53) are stored then the data likelihood can also be calculated:

$$L(\theta | Y_{1:T}) = f_{\theta}(Y_1) \prod_{t=2}^T f_{\theta}(Y_t | Y_{1:t-1}) \quad (2.54)$$

where

$$f_{\theta}(Y_1) = \int f_{\theta}(x_1) f_{\theta}(Y_1 | x_1) dx_1 \quad (2.55)$$

The smoothed distributions are calculated by passing messages back again from time  $T$  to the beginning. Similarly to how the conditioning on previous observations can be passed through messages to later variables, passing the conditioning on all data is achieved via another two-stage process. Giving the same names to the two steps as for the filtering procedure for purposes of analogy, we first define the

‘predict’ step:

$$f_{\theta}(x_t | Y_{1:t}, x_{t+1}) = \frac{f_{\theta}(x_t | Y_{1:t}) f_{\theta}(x_{t+1} | x_t)}{\int f_{\theta}(x_t | Y_{1:t}) f_{\theta}(x_{t+1} | x_t) dx_t} \quad (2.56)$$

which further conditions  $X_t$  on  $X_{t+1}$ . Next the ‘update’ step:

$$f_{\theta}(x_t | Y_{1:T}) = \int f_{\theta}(x_t | Y_{1:t}, x_{t+1}) f_{\theta}(x_{t+1} | Y_{1:T}) dx_{t+1} \quad (2.57)$$

passes on the conditioning on all data from  $X_{t+1}$  to  $X_t$ .

When all variables are continuous, the predict and update steps in both the filtering and smoothing procedures can only be calculated analytically if all distributions in (2.49) are linear-Gaussian. In this context they are known together as the *Kalman filter and smoother*. Deterministic approximations, for example the *extended Kalman filter* (EKF) (Jazwinski, 1970) and the *unscented Kalman filter* (UKF) (Julier and Uhlmann, 2004), can be made in other state space models but they are not particularly effective for models that are highly non-linear and/or non-Gaussian.

A more general framework for filtering and smoothing makes use of a sampling construct known as a *particle*. A particle is a sample that propagates stochastically according to the dynamics of a state space model, and represents a single trajectory through the state space. Early particle based methods developed in a variety of fields, including computational physics, molecular chemistry, and genetics. Their statistical foundations were not researched until relatively recently though, beginning with Del Moral (1996). Further foundational research into methodology and the theoretical properties particle methods followed quickly after, for example Crisan and Lyons (1997, 1999); Del Moral and Guionnet (1999b, 2001, 1999a).

Research into theoretical properties and methodological practices has continued ever since. Developing methodologies include adaptive particle filters (Del Moral et al., 2012), backward methods (Del Moral et al., 2010a), and island type methods (Vergé et al., 2015).

A particle filter conditions particles on each new observation as they propagate,

and approximates the sequence of filtered distributions  $\{X_t | Y_{1:t}\}_{t=1}^T$ . Smoothed distributions can also be approximated using particle methods, a detailed description of some algorithms follows in sec 2.4.2.3. When the dimension of the state space is low, as discussed in sec 2.4.2.4, then particle methods can be effective at applied filtering and smoothing tasks. They are used in a variety of applied fields, including target tracking and positioning (Zhou et al., 2004; Gustafsson et al., 2002; Zhang et al., 2013), data fusion (Khaleghi et al., 2013; Castanedo, 2013), and finance (Chib et al., 2006; Aihara et al., 2009).

### 2.4.2.1 Sequential importance sampling

In general the conditional distributions  $X_1 | Y_1, \theta$  and  $X_t | X_{t-1}, Y_t, \theta$  for  $t \in 2, \dots, T$  cannot be sampled from directly, and a common method of overcoming this is to use a recursive form of importance sampling known as *sequential importance sampling*.

Sequential importance sampling makes use of proposal distributions and normalised importance weights, similarly to regular importance sampling. The proposal distributions  $q(X_t | X_{t-1}, Y_t)$  can be considered to represent, in some heuristic sense, the prediction step of the filtering procedure, and calculating the importance weights similarly represents the updating step. The importance weights are calculated recursively such that estimated quantities are asymptotically unbiased:

$$\begin{aligned}
 w_t^{(i)} &\propto \frac{\prod_{\tau=1}^t f_{\theta}(X_{\tau}^{(i)} | X_{1:\tau-1}^{(i)}, Y_{1:\tau})}{\prod_{\tau=1}^t q(X_{\tau}^{(i)} | X_{\tau-1}^{(i)}, Y_{\tau})} \\
 &= \frac{\prod_{\tau=1}^t f_{\theta}(X_{\tau}^{(i)} | X_{\tau-1}^{(i)}, Y_{\tau})}{\prod_{\tau=1}^t q(X_{\tau}^{(i)} | X_{\tau-1}^{(i)}, Y_{\tau})} \\
 &\propto w_{t-1}^{(i)} \frac{f_{\theta}(X_t^{(i)} | X_{t-1}^{(i)}) f_{\theta}(Y_t | X_t^{(i)})}{q(X_t^{(i)} | X_{t-1}^{(i)}, Y_t)} \quad (2.58)
 \end{aligned}$$

If the prior transition distribution is used as the proposal distribution,  $q(X_t | X_{t-1}^{(i)}, Y_t) = \mathcal{P}(X_t | X_{t-1}^{(i)}, \theta)$ , then recursive update simplifies to:

$$w_t^{(i)} \propto w_{t-1}^{(i)} f_{\theta}(Y_t | X_t^{(i)}) \quad (2.59)$$

### 2.4.2.2 Sequential importance resampling

Though the prior will generally be relatively easy to sample from, and is indeed a popular choice of proposal distribution, it will not necessarily be an effective choice. A poor proposal distribution will produce samples with a low effective number of particles  $N_{\text{eff}}$ , see sec 2.4.1 for details, and a correspondingly high variance. A common method (Del Moral et al., 2012) of overcoming this is to re-sample the particles at time  $t$  according to the importance weights whenever  $N_{\text{eff}}$  falls below a given threshold  $N_{\text{min}}$ :

$$\begin{aligned} N_{\text{eff}} < N_{\text{min}} &\Rightarrow X_t^{*(i)} \overset{\text{i.i.d.}}{\sim} R_t \\ X_t^{(i)} &\leftarrow X_t^{*(i)} \end{aligned} \quad (2.60)$$

where  $R_t$  is a distribution over the collection of original particles  $\{X_t^{(i)}\}_{i=1}^N$ , giving re-sampling probability mass equal to the weights  $w_t^{(i)}$ :

$$\mathcal{P}\left(X_t^{*(i)} = X_t^{(j)}\right) = w_t^{(j)} \quad (2.61)$$

and after re-sampling, each importance weight is reset such that they are all equal:

$$w_t^{(i)} \leftarrow \frac{1}{N} \quad \forall i \quad (2.62)$$

The method described above is summarised in algorithm 2.3. If, after running a particle filter using this method, particles are persistently being re-sampled then the utility of the chosen proposal distribution should be questioned.

Some authors recommend an enforced re-sampling regime, where re-sampling is performed at every time step and the entire particle paths up to time  $t$  are re-

**Algorithm 2.3** Particle filter with sequential importance re-sampling

1. Sample  $N$  particle values  $X_1^{(i)}$  for  $t = 1$  from a proposal distribution  $q(X_1)$  and calculate importance weights:

$$w_1^{(i)} \propto \frac{f_{\theta}(X_1^{(i)} | Y_1)}{q(X_1^{(i)})} \quad (2.63)$$

For each  $t \in 2, \dots, T$ :

{

2. Sample each particle value  $X_t^{(i)}$  from a proposal distribution  $q(X_t | X_{t-1}^{(i)}, Y_t)$  and calculate importance weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{f_{\theta}(X_t^{(i)} | X_{t-1}^{(i)}) f_{\theta}(Y_t | X_t^{(i)})}{q(X_t^{(i)} | X_{t-1}^{(i)}, Y_t)} \quad (2.64)$$

3. Calculate the effective number of particles:

$$N_{\text{eff}} = \frac{1}{\sum_i w_t^{(i)2}} \quad (2.65)$$

4. If  $N_{\text{eff}}$  is below the threshold  $N_{\text{min}}$  then re-sample particles according to their importance weights, and reset the weights:

$$\begin{aligned} N_{\text{eff}} < N_{\text{min}} &\Rightarrow X_t^{*(i)} \stackrel{\text{i.i.d.}}{\sim} R_t \\ X_t^{(i)} &\leftarrow X_t^{*(i)} \\ w_t^{(i)} &\leftarrow \frac{1}{N} \end{aligned} \quad (2.66)$$

}

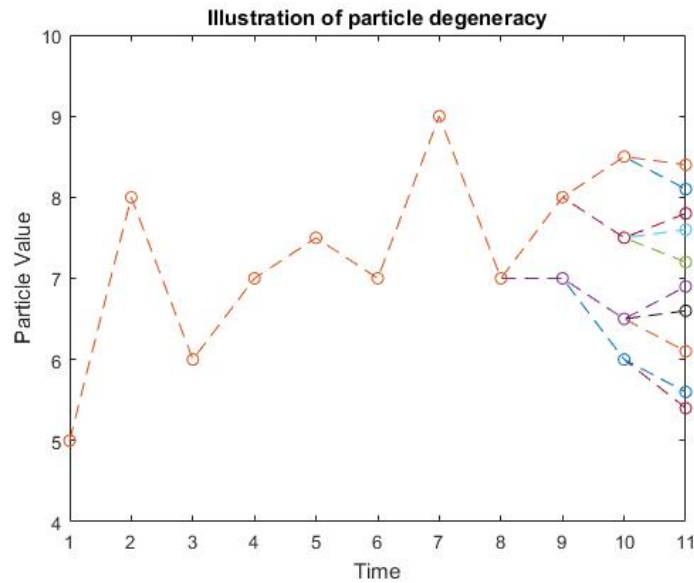


sampled according to  $R_t$ :

$$\begin{aligned}
 N_{\text{eff}} < N_{\text{min}} &\Rightarrow X_{1:t}^{*(i)} \overset{\text{i.i.d.}}{\sim} R_t \\
 X_{1:t}^{(i)} &\leftarrow X_{1:t}^{*(i)} \\
 w_t^{(i)} &\leftarrow \frac{1}{N}
 \end{aligned} \tag{2.67}$$

Such a regime implements a smoothing of the approximate distributions of previous time points on all currently available data, i.e. after re-sampling at time  $t$ , the particle values  $\{X_\tau^{(i)}\}_{i=1}^N$  for  $\tau < t$  approximate the smoothed distributions  $\mathcal{P}(X_\tau | Y_{1:t})$ . This would be desirable, except that the re-sampling at each time point compounds on itself and results in a phenomenon known as *particle degeneracy*. This is where the particle approximations to the smoothed distributions  $X_\tau | Y_{1:t}$  for all but the most recent times  $\tau$  have only one unique value. An example of particle degeneracy is shown in fig 2.5.

**Figure 2.5:** An illustrative example of particle degeneracy. Each line represents a particle path when the whole path of each particle is re-sampled at each time point. The repeated re-sampling results in approximate distributions containing only one unique value for all but the most recent time points.



### 2.4.2.3 Smoothing

Alternative methods of particle based smoothing that don't suffer from degeneracy have been developed in recent years. A popular family of these are known as *forward-backward* methods, which proceed through the filtering process as above and then use various approximations to the smoothing messages described previously in sec 2.4.2.

Two of the more popular variants will be described here, illustrating the trade-off between estimator variance and computing complexity. The first is known as *forward-filtering backward-sampling* (Godsill et al., 2004; Douc et al., 2009), and produces a new set of particle paths  $\{\tilde{X}_{1:T}^{(i)}\}_{i=1}^N$  which approximate the smoothed distribution  $X_{1:T} | Y_{1:T}, \theta$ . The new paths are produced in backward time, and are generated by re-sampling from the filtered distributions at each time  $t$  according to probabilities conditioned on the new particle's value at time  $t + 1$ .

As particles are re-sampled at all time points, all weights are equal;  $w_{t_1}^{(i)} = w_{t_2}^{(j)} = \frac{1}{N} \forall i, j \in 1, \dots, N$  and  $t_1, t_2 \in 1, \dots, T$ . For each smoothed particle  $\tilde{X}_{t+1}^{(i)}$  at time  $t + 1$ , its particle value at time  $t$  is sampled from the set of filtered particles at time  $t$  with probabilities that approximate the conditional structure of the true smoothed distributions:

$$\mathcal{P}(X_{t:t+1} | Y_{1:T}, \theta) = \mathcal{P}(X_t | Y_{1:t}, X_{t+1}, \theta) \mathcal{P}(X_{t+1} | Y_{1:T}, \theta) \quad (2.68)$$

by approximately sampling from

$$\mathcal{P}(X_t | Y_{1:t}, X_{t+1}, \theta) = \frac{\mathcal{P}(X_t | Y_{1:t}, \theta) \mathcal{P}(X_{t+1} | X_t, \theta)}{\int \mathcal{P}(X_t | Y_{1:t}, \theta) \mathcal{P}(X_{t+1} | X_t, \theta) dX_t} \quad (2.69)$$

as per (2.56). This approximate sampling is achieved by using the particle approximations to the distributions in (2.69):

$$\mathcal{P}\left(\tilde{X}_t^{(i)} = X_t^{(j)} | \tilde{X}_{t+1}^{(i)}\right) = \frac{w_t^{(j)} f_{\theta}\left(\tilde{X}_{t+1}^{(i)} | X_t^{(j)}\right)}{\sum_{k=1}^N w_t^{(k)} f_{\theta}\left(\tilde{X}_{t+1}^{(i)} | X_t^{(k)}\right)} \quad (2.70)$$

This procedure of sampling the value of each particle path in backward time

**Algorithm 2.4** Forward-filtering backward-sampling

1. Produce a sequence of particle approximations  $\{X_t^{(i)}, w_t^{(i)}\}_{t=1}^T$  to  $\mathcal{P}(X_t | Y_{1:t}, \theta)$  as per algorithm 2.3.
2. Re-sample  $\tilde{X}_T^{(i)}$  from  $X_T^{(i)} \sim \text{i.i.d. } R_T$  and reset  $w_T^{(i)} \leftarrow \frac{1}{N} \forall i$ .

For each  $t \in T - 1, \dots, 1$ :

3. For each particle  $\tilde{X}_{t+1}^{(i)}$ , sample its value at time  $t$  from  $\{X_t^{(j)}\}_{j=1}^N$  with probabilities as per (2.70):

$$\mathcal{P}(\tilde{X}_t^{(i)} = X_t^{(j)} | \tilde{X}_{t+1}^{(i)}) = \frac{w_t^{(j)} f_\theta(\tilde{X}_{t+1}^{(i)} | X_t^{(j)})}{\sum_{k=1}^N w_t^{(k)} f_\theta(\tilde{X}_{t+1}^{(i)} | X_t^{(k)})} \quad (2.71)$$

and reset  $w_t^{(i)} \leftarrow \frac{1}{N} \forall i$ .

by conditioning on its next value in forward time produces, after sampling the particle values for  $t = 1$ , a set of  $N$  particle paths that approximate the joint smoothed posterior  $\mathcal{P}(X_{1:T} | Y_{1:T}, \theta)$ . The computational cost is  $\mathcal{O}(N^2T)$ , i.e. it increases quadratically with the number of particles. If all particles are re-sampled in the filtering procedure then this can be reduced to  $\mathcal{O}(NT)$ , but as this itself costs  $\mathcal{O}(N)$  per time step the total cost grows similarly either way.

One alternative, but related, method is *forward-filtering backward-smoothing* (Doucet et al., 2000), which produces a sequence of approximate smoothed marginals  $\{\mathcal{P}(X_t | Y_{1:T}, \theta)\}_{t=1}^T$ . Rather than re-sampling from the filtered particles at time  $t$ , particles are re-weighted using the updated weights of *all* particles at time  $t + 1$ . The cost of such re-weighting is  $\mathcal{O}(N^2T)$ .

By using all particles at time  $t + 1$  to re-weight every particle at time  $t$ , the variance of estimated quantities is reduced as compared to forward-filtering backward-sampling. This variance improvement comes at the cost of losing estimates of  $\mathcal{P}(X_{1:T} | Y_{1:T}, \theta)$ , the joint distribution over all time of the smoothed distributions. Prior to each re-weighting, the pairwise approximate smoothed marginals  $\mathcal{P}(X_{t:t+1} | Y_{1:T}, \theta)$  are available, so expectations of time-pairwise functions  $\mathbb{E}[h(X_t, X_{t+1})]$  can

be calculated. Expectations of functions of latent variables with greater time lags cannot though.

To illustrate the principles of forward-filtering backward-smoothing, first assume that  $X_{t+1}^{(i)} \sim^{\text{approx.}} \mathcal{P}(X_{t+1} | Y_{1:T}, \theta)$  has already been re-weighted with weights  $\tilde{w}_{t+1}^{(i)}$ . Note that  $X_T^{(i)} \sim^{\text{approx.}} \mathcal{P}(X_T | Y_{1:T}, \theta)$  by construction of the filtering procedure so we set

$$\tilde{w}_T^{(i)} \leftarrow w_T^{(i)} \quad (2.72)$$

The re-weighting of the particles at time  $t$  is constructed such that the particles at each time approximate the marginal posterior  $\mathcal{P}(X_t | Y_{1:T}, \theta)$  defined recursively as per (2.57) by:

$$\begin{aligned} f_\theta(x_t | Y_{1:T}) &= \int f_\theta(x_t | Y_{1:t}, x_{t+1}) f_\theta(x_{t+1} | Y_{1:T}) dx_{t+1} \\ &\approx \sum_{j=1}^N f_\theta(x_t | Y_{1:t}, X_{t+1}^{(j)}) \tilde{w}_{t+1}^{(j)} \end{aligned} \quad (2.73)$$

where the conditional densities in (2.73) are for the distributions (2.69) in the forward-filtering backward-sampling procedure described above. The approach taken to approximating to  $\mathcal{P}(X_t | Y_{1:t}, X_{t+1}, \theta)$  by the forward-filtering backward-smoothing procedure differs here though; rather than re-sampling using the approximate probabilities in (2.70) and resetting their weights, the particles and weights from the filtering procedure are substituted into the right hand side of (2.69):

$$\begin{aligned} f_\theta(x_t | Y_{1:t}, X_{t+1}^{(j)}) &= \frac{f_\theta(x_t | Y_{1:t}) f_\theta(X_{t+1}^{(j)} | x_t)}{\int f_\theta(x_t | Y_{1:t}) f_\theta(X_{t+1}^{(j)} | x_t) dx_t} \\ &\approx \sum_{i=1}^N \left( \frac{w_t^{(i)} f_\theta(X_{t+1}^{(j)} | X_t^{(i)})}{\sum_{k=1}^N w_t^{(k)} f_\theta(X_{t+1}^{(j)} | X_t^{(k)})} \right) \delta(x_t - X_t^{(i)}) \end{aligned} \quad (2.74)$$

Inserting (2.74) into (2.73) gives:

$$f_{\theta}(x_t | Y_{1:T}) \approx \sum_{i=1}^N \sum_{j=1}^N \left( \frac{w_t^{(i)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(i)}) \tilde{w}_{t+1}^{(j)}}{\sum_{k=1}^N w_t^{(k)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(k)})} \right) \delta(x_t - X_t^{(i)}) \quad (2.75)$$

which implies the following update rule for weights  $w_t^{(i)}$ :

$$\tilde{w}_t^{(i)} \leftarrow \sum_{j=1}^N \left( \frac{w_t^{(i)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(i)}) \tilde{w}_{t+1}^{(j)}}{\sum_{k=1}^N w_t^{(k)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(k)})} \right) \quad (2.76)$$

**Algorithm 2.5** Forward-filtering backward-smoothing

1. Produce a sequence of particle approximations  $\{X_t^{(i)}, w_t^{(i)}\}_{t=1}^T$  to  $\mathcal{P}(X_t | Y_{1:t}, \theta)$  as per algorithm 2.3.
2. Set  $\tilde{w}_T^{(i)} \leftarrow w_T^{(i)} \forall i$ .  
For each  $t \in T - 1, \dots, 1$ :
3. Re-weight the filtered particles  $\{X_t^{(i)}\}_{i=1}^N$  according to (2.76):

$$\tilde{w}_t^{(i)} \leftarrow \sum_{j=1}^N \left( \frac{w_t^{(i)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(i)}) \tilde{w}_{t+1}^{(j)}}{\sum_{k=1}^N w_t^{(k)} f_{\theta}(X_{t+1}^{(j)} | X_t^{(k)})} \right) \quad (2.77)$$

Other particle based smoothing algorithms have been developed (Briers et al., 2010; Fearnhead et al., 2010), in particular *generalised two-filter* approaches. These algorithms have similar computational costs to forward-backward methods and can sometimes produce estimates  $\hat{E}h$  with superior variance properties, but are also generally more challenging to derive.

#### 2.4.2.4 Convergence

Asymptotic results for the particle filtering and smoothing procedures described above have been published (see Crisan and Doucet (2002); Del Moral (2004); Poyiadjis et al. (2011) among others). For particle approximations to the filtered distributions, convergence of  $\hat{E}_{\theta}[h]$  for bounded test functions  $h : \mathcal{X}^T \rightarrow [-1, 1]$  is with

$\sqrt{N}$ :

$$\mathbb{E} \left[ \left\| \int h(x_{1:T}) (f_\theta(x_{1:T} | Y_{1:T}) - \hat{f}_\theta(x_{1:T} | Y_{1:T})) dx_{1:T} \right\|^p \right]^{\frac{1}{p}} \leq \frac{A_{\theta,T,p}}{N^{\frac{1}{2}}} \quad (2.78)$$

where  $\hat{f}_\theta(x_{1:T} | Y_{1:T})$  is the particle approximation to  $f_\theta(x_{1:T} | Y_{1:T})$  and  $A_{\theta,T,p} < \infty$  is a constant with respect to  $N$  parametrised by its indices. For both of the smoothing procedures presented here, it can be shown (Del Moral et al., 2010a,b; Douc et al., 2009) that the variance of estimated *smooth additive functionals*

$$\hat{E}[S_T] \approx \int S_T(x_{1:T}) f_\theta(x_{1:T} | Y_{1:T}) dx_{1:T} \quad (2.79)$$

where

$$S_T(x_{1:T}) = \sum_{t=1}^{T-1} s_t(x_{t:t+1}) \quad (2.80)$$

is bounded:

$$\text{var}(\hat{E}[S_T]) \leq B_\theta \frac{T}{N} \quad (2.81)$$

for some  $B_\theta < \infty$ .

It is unfortunately the case that both of the constants  $A_{\theta,T,p}, B_\theta$  explode with the dimension of  $\mathcal{X}$ , which leaves particle methods currently restricted to state space models with low dimensional state spaces.

### 2.4.3 Markov chain Monte Carlo

Possibly the most widespread of all Monte Carlo methods is known as *Markov chain Monte Carlo* (MCMC). MCMC algorithms can be used when the distribution of interest has known (up to a proportionality constant) density function, but cannot be sampled from directly. Instead, samples are drawn from a Markov chain designed to have the distribution of interest as its stationary distribution. As more samples are drawn from the chain, their distributions approximate the distribution of interest ever more closely, and estimates  $\hat{E}_\theta[h]$  converge to their true values.

A (time-homogeneous) *Markov chain* is a sequence of distributions defined on a space  $\mathcal{X}$  with the (first order) Markov property:

$$\mathcal{P}(X_{t+1} | X_{1:t}) = \mathcal{P}(X_{t+1} | X_t) \quad (2.82)$$

such that  $\mathcal{P}(X_{t_1+1} | X_{t_1}) = \mathcal{P}(X_{t_2+1} | X_{t_2}) \forall t_1, t_2$  is constant over time. Under certain conditions a Markov chain can have a *stationary distribution*  $\pi(X)$ , such that if  $X_1 = \pi(X)$  then all marginal distributions  $X_t$  will share the distribution:

$$\begin{aligned} f_1(x_1) = \pi(x_1) &\Rightarrow f_t(x_t) = \int f_{t-1}(x_{t-1})f(x_t | x_{t-1}) \mathrm{d}x_{t-1} \\ &= \int \pi(x_{t-1})f(x_t | x_{t-1}) \mathrm{d}x_{t-1} \\ &= \pi(x_t) \end{aligned} \quad (2.83)$$

A sufficient condition for a Markov chain to have a stationary distribution is for it to be *reversible*, which means that finite dimensional distributions do not depend on the direction of time. Formally:

---

**Reversible Markov chain** A Markov chain in equilibrium is *reversible* if, for all  $n, t_1, \dots, t_n, \tau$ :

$$\mathcal{P}(X_{t_1}, \dots, X_{t_n}) = \mathcal{P}(X_{\tau-t_1}, \dots, X_{\tau-t_n}) \quad (2.84)$$


---

and this property can be equivalently expressed in the form of *detailed balance equations*:

$$\pi(x)f(x' | x) = \pi(x')f(x | x') \quad x, x' \in \mathcal{X} \quad (2.85)$$

thus being in equilibrium is equivalent to the marginal distribution at all times being equal to  $\pi$ . Any distribution  $\pi$  that solves the detailed balance equations will be a

stationary distribution for the Markov chain in question.

A Markov chain is *ergodic* if all states are aperiodic and *positive recurrent*; expected return times are finite for all states. If the Markov chain is ergodic then it has a unique stationary distribution  $\pi(X)$ , and from an arbitrary initial distribution  $\mathcal{P}(X_1)$ , the marginal distributions  $\mathcal{P}(X_t)$  converge to  $\pi(X)$ :

$$\begin{aligned} f_t(x) &= \int f_{t-1}(x_{t-1})f(x | x_{t-1}) dx_{t-1} \\ &\rightarrow_d \pi(x) \quad \text{as } t \rightarrow \infty \end{aligned} \quad (2.86)$$

By constructing an ergodic Markov chain with the distribution of interest as its stationary distribution, samples drawn from each conditional distribution in the chain will approximate samples drawn from the distribution of interest ever more closely.

The well known and popular *Metropolis-Hastings* (Metropolis et al., 1953; Hastings, 1970; Rubinstein and Kroese, 2011) algorithm draws samples conditionally from distributions conditioned on the last sample, and accepts them with probability constructed to conform to detailed balance equations:

$$\begin{aligned} \tilde{X}_{t+1} &\sim q(\tilde{X}_{t+1} | X_t) \\ X_{t+1} &= \begin{cases} \tilde{X}_{t+1} & : \mathcal{P}(X_{t+1} = \tilde{X}_{t+1}) = \min\left(1, \frac{\pi(\tilde{X}_{t+1})q(X_t | \tilde{X}_{t+1})}{\pi(X_t)q(\tilde{X}_{t+1} | X_t)}\right) \\ X_t & : \mathcal{P}(X_{t+1} = X_t) = 1 - \mathcal{P}(X_{t+1} = \tilde{X}_{t+1}) \end{cases} \end{aligned} \quad (2.87)$$

where  $\pi(X)$  is the distribution of interest. The resulting Markov chain is ergodic, so the distribution of samples at each step converges to  $\pi(X)$ . As  $\pi$  appears in both the numerator and denominator of the acceptance probability in (2.87), it only needs to be known up to a constant of proportionality. This can be particularly useful if samples from a conditional distribution  $\mathcal{P}(X | Y)$  are required, but only the joint distribution  $\mathcal{P}(X, Y)$  is known.

In practice, the first  $n$  samples are often discarded, as convergence to  $\pi$  cannot be observed and drawing  $X_1$  from a region of high  $\pi$  density cannot be assumed. These  $n$  samples are known as the *burn in*, and  $n$  is often chosen heuristically or



arbitrarily.

As the samples  $\{X_t\}$  are clearly correlated, it could be argued that the variance over estimates  $\hat{E}_\theta[h]$  might become significantly inflated if all samples are used. This has been shown (Geyer, 1992; MacEachern and Berliner, 1994) to be a spurious argument, however. Some authors do recommend the use of *sub-sampling* in the case of making variance estimates though. This involves using only every  $i^{\text{th}}$  sample, where the lag  $i$  empirical auto-correlation  $\hat{\rho}_i$  is negligible:

$$X_j^* = X_{n+ij} \quad j = 1, \dots, \tilde{N} = \left\lfloor \frac{N-n}{i} \right\rfloor$$

$$\hat{E}_\theta[h] = \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} h(X_j^*) \quad (2.88)$$

**Algorithm 2.6** Metropolis-Hastings

1. Draw  $X_1 \sim q_1(X_1)$  with  $\text{supp}(q_1) = \text{supp}(\pi)$ .
2. For  $t \in 2, \dots, n+1$ :
  - (a) Draw  $\tilde{X}_{t+1} \sim q(\tilde{X}_{t+1} | X_t)$  and accept with probability as in (2.87):

$$X_{t+1} = \begin{cases} \tilde{X}_{t+1} & : \mathcal{P}(X_{t+1} = \tilde{X}_{t+1}) = \min\left(1, \frac{\pi(\tilde{X}_{t+1})q(X_t|\tilde{X}_{t+1})}{\pi(X_t)q(\tilde{X}_{t+1}|X_t)}\right) \\ X_t & : \mathcal{P}(X_{t+1} = X_t) = 1 - \mathcal{P}(X_{t+1} = \tilde{X}_{t+1}) \end{cases} \quad (2.89)$$

3. Discard  $\{X_t\}_{t=1}^n$ . Restart time index:  $X_1 \leftarrow X_{n+1}$
4. For  $t \in 2, \dots, N$ :
  - (a) Draw  $\tilde{X}_{t+1} \sim q(\tilde{X}_{t+1} | X_t)$  and randomly accept as in step 2a.

The Metropolis-Hastings algorithm is summarised in algorithm 2.6. In theory, any conditional distribution  $q(x | x')$  with  $\text{supp}(q(x | x')) = \text{supp}(\pi(x))$  can be used as a proposal distribution, but if most proposed new samples  $\tilde{X}_{t+1}$  are rejected then samples will both be highly correlated and move slowly through the sample space. Many samples would therefore be needed for estimates  $\hat{E}_\theta[h]$  to have low variance. Even if most proposed new samples are accepted, if they are always very close to

their conditioning sample then they will still move slowly through the sample space, and many samples would be needed for estimates to have low variance. Markov chains constructed as above suffering from this pathology are said to be *slow mixing*.

Optimising the proposal distributions with respect to mixing speed is a current area of active academic research. Innovations such as *Langevin adjusted MCMC* (Roberts and Tweedie, 1996; Atchadé, 2006), *Hamiltonian MCMC* (Duane et al., 1987; Neal, 1993; Livingstone et al., 2016), and *Riemannian MCMC* (Girolami and Calderhead, 2011; Fraccaro et al., 2016) are very promising. The performance of MCMC algorithms, in particular in high dimensional settings, is improving rapidly.

A particular form of Metropolis-Hastings MCMC known as *Gibbs sampling* (Geman and Geman, 1984) has acceptance probability 1, and draws a sample from each dimension  $i$  of the  $m$  dimensional target distribution in turn from their conditional distributions

$$X_{n+1,i} \sim \pi_{i|\setminus i}(X_{n+1,i} | X_{n+1,1}, \dots, X_{n+1,i-1}, X_{n,i+1}, \dots, X_{n,m}) \quad (2.90)$$

where

$$\pi_{i|\setminus i}(X_i | X_{\setminus i}) = \frac{\pi(X)}{\int \pi(X) dx_i} \quad (2.91)$$

is the conditional distribution  $X_i | X_{\setminus i}$  implied by the target distribution  $\pi(X)$ .

Approximating distributions using MCMC is a general method that can be widely applied and can be arbitrarily accurate. Furthermore, research providing improvements to the computational costs does not appear to be tailing off. Whilst the progress made on improving MCMC algorithms is both exciting and fruitful, current algorithms can suffer slow run and/or development times, and are still somewhat vulnerable to the curse of dimensionality. Alternative methods of approximating inference, including those investigated in the current thesis, remain essential components of computational statistics as a whole.

## 2.5 Message passing in graphical models

This section is going to describe the graphical model approach to model description, message passing on trees and approximate extensions to graphs with cycles. The example of belief propagation is used to illustrate the principles of message passing. Belief propagation on trees, defined in sec 2.5.1.1 below, can produce exact inference when the prescribed sums or integrals are tractable. As such is not directly relevant to the current thesis, but the approximation methods used for graphs with cycles are most easily described after having described the tree based method first.

Message passing is a general method for reducing the computational burden of inference on models whose variables have a known conditional independence structure. It is a concept that is exploited throughout the current thesis, particularly so in chapter 3. Many calculations of interest in statistical inference require the joint distribution over all variables to be summed or integrated over many of those variables, as required for latent variable models for instance. This is a task that, if performed naively, clearly has exponentially increasing computational complexity in the size of the variable set.

In general, the focus of the current thesis is on models with continuous random variables. The current section will focus on models with discrete variables, as that is the context in which much of the content developed. The concepts can transferred to the continuous setting by simply replacing sums with integrals, assuming that the integrals in question can be calculated.

Conditional independences can significantly reduce the computational burden though. They allow quantities that are notionally a function of all variables to be decomposed as a product of component functions. As such, sums or integrals over many variables can be decomposed into components of much lower dimension. The way these components interact with each other can be determined by the application in question, and in general this will be through the value of auxiliary functions known as *messages*.

*Message passing* is any iterative procedure that updates the value of these message as a function of the current value of the other messages, until all messages have

converged to a stationary point. In its classic formulation for exact computations, e.g. Pearl (1988), it is essentially a type of dynamic programming. Either these converged messages themselves, or easily computable functions of them, will return the quantities of inferential interest. In general these messages will require significantly less computation time than a naive attempt at summing or integrating out the unneeded variables.

To clarify the above description, it is necessary to employ a formalism for describing the conditional independence structure of a statistical model. A common and effective choice of formalism is the *graphical model*. It is first useful though, to quickly define what is meant by *conditional independence*:

---

**Conditional independence** Two sets of variables,  $X$  and  $Y$ , are *conditionally independent* given a third set of variables,  $Z$ , if their joint conditional probability factorises between them:

$$\mathcal{P}(X, Y | Z) = \mathcal{P}(X | Z)\mathcal{P}(Y | Z) \quad (2.92)$$

or, equivalently,

$$\mathcal{P}(X | Y, Z) = \mathcal{P}(X | Z) \quad (2.93)$$

and is denoted

$$(X \perp Y) | Z \quad (2.94)$$


---

This definition extends to multiple sets of variables. Suppose, for example, that

a joint distribution over the sets of variables  $A, B, C, D$  can be factorised as follows:

$$\begin{aligned}\mathcal{P}(A, B, C, D) &= \mathcal{P}(A)\mathcal{P}(B | A)\mathcal{P}(C | A, B)\mathcal{P}(D | A, B, C) \\ &= \mathcal{P}(A)\mathcal{P}(B | A)\mathcal{P}(C | A)\mathcal{P}(D | C)\end{aligned}\quad (2.95)$$

then the following conditional independence relationships are implied:

$$\begin{aligned}(B \perp C) | A \\ (A \perp D) | C \\ (B \perp D) | C \\ (B \perp D) | A\end{aligned}\quad (2.96)$$

More generally, a joint distribution may be factorised into factors that are not necessarily conditional distributions, for example the distribution of a collection  $V$  of variables might be factorised:

$$\mathcal{P}(V = V') = \frac{1}{Z} \prod_{i=1}^C f_i(S_i = S'_i) \quad S_i \subset V, i = 1, \dots, C \quad (2.97)$$

where the  $f_i$  are non-negative functions and  $Z$  is a normalisation constant enforcing the constraint  $\sum_{V=V'} \mathcal{P}(V') = 1$ . Such a factorisation still encodes the conditional independences of the distribution, which can be decoded with a graphical model.

### 2.5.1 Graphical models

A powerful method of describing conditional independences between variables with a given joint distribution is to use *graphs*. Graphs are mathematical objects that encode pairwise relations between a discrete set of objects. Formally, a (undirected) graph is defined as:

---

**Undirected graph** An *undirected graph*  $G$  is an ordered pair  $(V, E)$  of a set of nodes  $v \in V$  and edges  $(u, v) \in E \Rightarrow u, v \in V$  between them. A *sub-graph*  $G' = (V' \subset V, E' \subset E)$  s.t.  $(u, v) \in E' \Rightarrow u, v \in V'$  is a subset of the nodes in  $G$  with some

of their edges from  $G$ . An *induced sub-graph*  $G' = (V' \subset V, E' \subset E)$  s.t.  $u, v \in V', (u, v) \in E \Rightarrow (u, v) \in E'$  is a subset of nodes in  $G$  with *all* of their edges from  $G$ .

---

If two nodes share an edge then they are said to be *neighbours*, and the set of neighbours of a given node  $v$  is denoted  $N(v)$ :

$$u \in N(v) \Leftrightarrow (u, v) \in E \quad (2.98)$$

An *undirected graphical model* represents the factorisation structure of a joint distribution in the form of an undirected graph:

---

**Undirected graphical model** An *undirected graphical model* of a joint distribution  $\mathcal{P}(V)$  over variables  $v \in V$  is a graph  $G = (V, E)$  where an edge  $(u, v) \in E$  exists between variables  $u$  and  $v$  if and only if they share a factor in  $\mathcal{P}$ . For clarity, it is assumed that factors cannot be further factorised. Equivalently, if a joint distribution  $\mathcal{P}(V)$  can be maximally factorised as:

$$\mathcal{P}(V) = \prod_{i=1}^C f_i(S_i) \quad S_i \subset V, i = 1, \dots, C \quad (2.99)$$

where a maximal factorisation is such that it cannot be further factorised:

$$\nexists \{f_{i,j}(S_{i,j})\}_{j=1}^{C_i} : f_i(S_i) = \prod_{j=1}^{C_i} f_{i,j}(S_{i,j}) \quad S_{i,j} \subset S_i, i = 1, \dots, C, j = 1, \dots, C_i \quad (2.100)$$

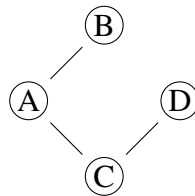
then (Lauritzen, 1996)

$$(u, v) \in E \Leftrightarrow \exists 1 \leq i \leq C : u, v \in S_i \quad (2.101)$$


---

If a subset  $S \subset V$  of variables is fully connected, i.e. all of them have an edge with all of the others, then  $S$  is known as a *clique*. If  $S$  is not properly contained in any other clique then it is known as a *maximal clique*. The maximal cliques of a graph imply a minimal factorisation; the joint distribution will factorise according to the maximal cliques, and those factors may themselves factorise further.

It is common to represent a graph  $G = (V, E)$  graphically, with nodes  $v \in V$  placed inside circles and an edge drawn between nodes  $u$  and  $v$  if  $(u, v) \in E$ . The graphical model representing the conditional independences in (2.95) is shown in fig 2.6.



**Figure 2.6:** Graphical representation of the conditional independences in (2.95)

A valuable feature of graphical models is that they can clarify if any additional conditional independence statements are implied by a given set of conditional independences. First define a *path* from node  $u$  to node  $v$  as a sequence of edges, each of which shares a node with adjacent edges in the sequence, that starts at node  $u$  and ends at node  $v$ , for example

$$(u, x) \rightarrow (x, y) \rightarrow (y, z) \rightarrow (z, v) \quad (2.102)$$

would be a path from  $u$  to  $v$  that goes through nodes  $x, y, z$ . Two (sets of) nodes  $A, B$  are conditionally independent given another (set of) node(s)  $C$  if no path between  $A$  and  $B$  exists that does not go through  $C$ . From the graphical model in fig 2.6 therefore, it is easily seen that the factorisation (2.95) implies  $(B \perp D) | A$  in addition to the conditional independences in (2.96). Graphical models provide a powerful method for determining all of the conditional independences implied by a given factorisation.

### 2.5.1.1 Trees

If a given undirected graph  $G$  has the property that for all nodes  $v \in V$ , no path that starts and ends at  $v$  exists that has no repeated edges, then  $G$  is known as a *tree*. The nodes on a tree that are connected to only one other node are known as *leaves*.

---

**Tree** A graph  $G$  is a *tree* if all paths that start and end at the same node contain at least one repeated edge.

---

A graph that is not a tree is known as a *graph with cycles*. Trees have various unique properties, one of which is the fact that any two nodes are connected by precisely one path. A related property is that if any one edge is removed from a tree then there will be two disconnected components, both of which are trees.

These properties in particular are very useful in message passing algorithms, and allow their results to be exact. Message passing can still be conducted on graphs with cycles, see sec 2.5.3 below, but any algorithm that uses it will only produce approximate results.

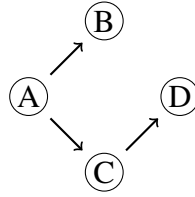
### 2.5.1.2 Directed graphs

A *directed graph*  $G = (V, D)$  is a graph whose edges  $(u \rightarrow v) \in D$  are directed from one of their nodes to the other. The source of the arrow is known as the *parent* and its target is known as the *child*. Directed graphs can be used in the context of graphical models in the form of a *Directed Acyclic Graph*, or DAG. A directed graph is acyclic if no *directed path*, i.e. a path that always travels in the direction of its component edges, exists that starts and stops at the same node.

A given factorisation of a joint distribution can be represented as a DAG when each factor contains only a node and its parents. A joint distribution that is represented by an undirected graph is commonly known as a *Markov Random Field* (MRF), and one represented by a DAG is commonly known as a *Bayesian network*.

Fig 2.7 shows the directed graph of (2.95). Determining the conditional independence relationships in a joint distribution from a DAG is slightly more compli-





**Figure 2.7:** Directed graphical representation of the conditional independences in (2.95)

cated than from an undirected graph, as the directions of each edge in a path need to be taken into account, but a given DAG has an equivalent undirected graph. A given DAG  $G = (V, D)$  can be converted into its equivalent undirected graph  $G' = (V, E)$  by *moralising* it. Moralising is achieved by converting all directed edges into undirected ones, and adding an undirected edge between each pair of parents that share a common child, i.e.  $G' = (V, E)$ , where

$$\begin{aligned} (u \rightarrow w) \in D &\Rightarrow (u, w) \in E \\ (u \rightarrow w), (v \rightarrow w) \in D &\Rightarrow (u, v) \in E \end{aligned} \quad (2.103)$$

As every DAG has an equivalent undirected graph, it is simpler for the following exposition to exclusively focus on undirected graphs.

## 2.5.2 Belief propagation

Given a graphical model  $G = (V, E)$  of a joint distribution  $\mathcal{P}(V)$ , it is a common task to try and find the marginal distribution of some subset  $S \subset V$  of variables. If all variables are discrete then this task can naively be completed by summing out the other variables:

$$\mathcal{P}(S = S') = \sum_{V': S=S'} \mathcal{P}(V = V') \quad (2.104)$$

where  $S', V'$  are vector-valued realisations of the variable vectors  $S, V$  respectively. For the case of continuous variables sums can simply be replaced by integrals.

This operation clearly has exponential computational complexity in the number of nodes, so it is not a practical foundation for finding marginal distributions for tree structured models. A popular framework with linear complexity in the number

of nodes that uses message passing to find the marginals of tree structured joint distributions is known as *belief propagation*. For ease of exposition, the following description focusses on the case of finding the marginal distribution of a single variable but it is trivially expanded to the case of more than one variable.

Belief propagation exploits the factorisation of  $\mathcal{P}(V)$  and its properties as a tree to reduce the computational complexity of its execution. As  $G$  is a tree, all of its cliques contain at most 2 nodes:

$$\mathcal{P}(V = V') \propto \prod_{v \in V} f_v(v') \prod_{(u,v) \in E} f_{u,v}(u', v') \quad (2.105)$$

where  $u, v$  refer to specific nodes and  $u', v'$  refer to their particular values. This notational convention will be maintained throughout this section, whereby  $z'$  refers to a specific value of variable  $z$ . The marginal distribution of a single variable  $x \in V$  is therefore a specific sum of products:

$$\begin{aligned} \mathcal{P}(x = x') &\propto \sum_{V': x=x'} \mathcal{P}(V = V') \\ &= \sum_{V': x=x'} \prod_{v \in V} f_v(v') \prod_{(u,v) \in E} f_{u,v}(u', v') \end{aligned} \quad (2.106)$$

The tree structure of  $G$  allows this sum to be decomposed in a powerful way. As mentioned in sec 2.5.1.1, if any single edge is removed from a tree then there will be two disconnected components, both of which will be trees. If the edge between  $x$  and any of its neighbours  $u$  is removed, therefore, the disconnected component containing  $u$  will be a tree, hereafter denoted  $T_x(u)$ ,  $u \in N(x)$ . This is equivalent to saying that among the factors  $f_{u,v}(u, v)$  in (2.106), there is precisely one that contains precisely one node from  $T_x(u)$  and that factor is  $f_{x,u}(x, u)$ .

The sum of products in (2.106) can therefore be decomposed and rearranged

as a product of sums:

$$\begin{aligned}
\mathcal{P}(x = x') &\propto \sum_{V': x=x'} f_x(x') \prod_{u \in N(x)} f_{x,u}(x', u') \prod_{v \in V(T_x(u))} f_v(v') \prod_{(v,w) \in E(T_x(u))} f_{v,w}(v', w') \\
&= f_x(x') \prod_{u \in N(x)} \sum_{V(T_x(u))'} f_{x,u}(x', u') \prod_{v \in V(T_x(u))} f_v(v') \prod_{(v,w) \in E(T_x(u))} f_{v,w}(v', w') \\
&\propto f_x(x') \prod_{u \in N(x)} M_{x,u}(x') \tag{2.107}
\end{aligned}$$

where

$$M_{x,u}(x') = \frac{1}{Z} \sum_{V(T_x(u))'} f_{x,u}(x', u') \prod_{v \in V(T_x(u))} f_v(v') \prod_{(v,w) \in E(T_x(u))} f_{v,w}(v', w') \tag{2.108}$$

with  $Z$  enforcing the constraint  $\sum_{x'} M_{x,u}(x') = 1$ , and  $V(T_x(u)), E(T_x(u))$  refer to the nodes and edges of  $T_x(u)$  respectively, and each term in the product of the final line of (2.107) is called a *message* from  $u$  to  $x$ .

It is here that the message passing structure of belief propagation becomes clear. As  $T_x(u)$  is itself a tree, each of the  $|N(x)|$  sums of products in (2.107) can themselves be decomposed into further products of sums, each of which is a message of the same form:

$$\begin{aligned}
\mathcal{P}(x = x') &\propto f_x(x') \prod_{u \in N(x)} M_{x,u}(x') \\
&= f_x(x') \prod_{u \in N(x)} \sum_{u'} f_{x,u}(x', u') f_u(u') \sum_{(V_{x,u} \setminus u)'} \prod_{v \in V_{x,u} \setminus u} f_v(v') \prod_{(v,w) \in E_{x,u}} f_{v,w}(v, w) \\
&\propto f_x(x') \prod_{u \in N(x)} \sum_{u'} f_{x,u}(x', u') f_u(u') \prod_{v \in N(u) \setminus x} M_{u,v}(u') \tag{2.109}
\end{aligned}$$

where  $V_{x,u}, E_{x,u}$  is shorthand for  $V(T_x(u)), E(T_x(u))$ .

All marginal distributions can therefore be calculated easily given knowledge of all messages. All messages  $\{M_{a,b}(a'), M_{b,a}(b') : (a, b) \in E\}$  can be calculated together iteratively from an arbitrary initialisation, with the updates determined by

their implied stationary point in (2.109):

$$M_{a,b}^{(k+1)}(a') = \frac{1}{Z} \sum_{b'} f_{a,b}(a', b') f_b(b') \prod_{c \in N(b) \setminus a} M_{b,c}^{(k)}(b') \quad (2.110)$$

It can be shown (Pearl, 1988) that by using the updates in (2.110), the messages  $M_{a,b}^{(k)}(a')$  will converge to the true messages  $M_{a,b}(a')$  in a finite number of iterations. Optimal scheduling (Yedidia et al., 2003) can be employed to minimise the number of updates required for convergence, and is described in algorithm 2.7.

**Algorithm 2.7** Belief propagation

1. Initialise messages  $M_{a,b}^{(0)}(a'), M_{b,a}^{(0)}(b')$  for all  $(a, b) \in E$ .
2. Designate one node as the *root* of the tree.
3. Repeat until convergence:
  - (a) Update the message from each leaf node  $l$ , i.e. the nodes with only one neighbour, to their neighbour  $a$  according to (2.110):

$$M_{a,l}^{(k+1)}(a') = \frac{1}{Z} \sum_{l'} f_{a,l}(a', l') f_l(l') \prod_{c \in N(l) \setminus a} M_{l,c}^{(k)}(l') \quad (2.111)$$

with  $N(l) \setminus a = \emptyset$ .

- (b) Update the messages from the neighbour of each leaf to their non-leaf neighbours
  - (c) Update messages along each path from a leaf towards the root according to (2.110). Where paths join, wait until the messages from all paths have been updated before updating messages along joined path.
  - (d) Update all messages again, only in the reverse order.
4. Calculate any marginal distribution by normalising (2.107):

$$\mathcal{P}(x = x') \propto f_x(x') \prod_{u \in N(x)} M_{x,u}(x'), \quad \sum_{x'} \mathcal{P}(x = x') = 1 \quad (2.112)$$

The procedure is known as belief propagation as beliefs regarding the distribution at each node are propagated through the graph to update beliefs in other nodes. It is an exact procedure when the graph is a tree, due to the unique paths between

nodes. The message from one node to another can only be passed through one path, so it is received only once per iteration. If the graph contains cycles though, messages between some nodes will be received through more than one path, and this duplicity of information prevents the procedure from being exact.

### 2.5.3 Belief propagation in graphs with cycles

While the message updates in (2.110) are derived on the assumption that the graph of  $\mathcal{P}(V)$  is a tree, the assumption is not necessary to implement them. As such, belief propagation algorithms can be implemented on graphs with cycles, though the resulting marginals will not be exact. Naively implementing belief propagation on a graph with cycles is commonly known as *loopy belief propagation*.

Much research has been conducted into and using loopy belief propagation, for example Frey et al. (2001); Weiss (2000); Ihler et al. (2005); Freeman et al. (2000); McEliece et al. (1998). In Freeman et al. (2000), a study on computer vision analysis, high level information known as the scene is inferred from low level data contained in an image. For example, the details of a 3D shape might be inferred from its 2D image. To do this, the scene and its image are modelled in patches, with each patch corresponding to a node on a graph.

Edges are modelled between each scene patch and its corresponding image patch, and between neighbouring scene patches. The edges between scene patches form cycles, so belief propagation becomes loopy. The quality of the approximate marginals found using loopy belief propagation on this graph in Freeman et al. (2000) was sufficient for their purposes.

Another interesting application was in coding theory (McEliece et al., 1998), and in particular an algorithm known as turbo decoding. Turbo decoding had been introduced to information theorists in 1993 and was known for its impressive performance, but there was little understanding as to why it performed so well. The authors showed in McEliece et al. (1998) that it was an implementation of loopy belief propagation, which provided scope for a deeper understanding of its theoretical properties.

Loopy belief propagation can also be modified by introducing weights to each

of the edges. These weights adjust the strength each node has in the message updating formula (2.110), which can reduce bias while keeping computational complexity low. A particular example is *tree re-weighted belief propagation* (Wainwright et al., 2003), where the weights are determined by spanning trees.

A spanning tree is a sub-graph of a graph with cycles that contains all nodes but not all edges, such that it is a tree. A given graph has a finite number of spanning trees, and the weights in Wainwright et al. (2003) are proportional to the number of spanning trees in which they appear. More details on re-weighted belief propagation can be found in Roosta et al. (2008).

## 2.6 Trade-offs in Computational Statistics

Choices between different approximation methods are influenced by both theoretical and practical concerns. The need to even make a choice highlights the fundamentally pragmatic approach that must be taken, but choosing between comparable methods is aided significantly with a profile of both their theoretical properties and their implementation requirements.

As the current chapter has detailed, there is a large variety of approximation methods available in modern statistics. To provide some detail as to how different methods relate to each other, the briefly described delineation between surrogate quantities and numerical approximations in sec 2.1.2 is expanded on now. This delineation is useful in that it determines which methods ‘compete’ with each other as options for an approximate inference solution.

In short, approximate inference can be performed by making choices regarding *what quantity* is being evaluated, and/or by making choices regarding *how* it is evaluated. Any complete approximate inference algorithm has to address both of these questions; numerical approximations compete amongst each other as methods for evaluating a given (surrogate) quantity, and (surrogate) quantities - with full consideration given to their evaluation - compete amongst each other as alternative quantities of interest. The word ‘surrogate’ has been placed in parentheses to emphasise that one or more of the alternatives might be the original quantity of interest,

e.g. the full data likelihood, evaluated with a particular numerical approximation.

The methods whose implementations are explored in the current thesis are composite likelihoods, variational approximations, and, being viewed as a surrogate quantity, the method of moments. Under the dichotomy of surrogate quantities and numerical approximations, variational approximations lie separate to the other two in the numerical approximations side of the division. As such they do not directly ‘compete’ with the other two as a choice to be made; once a decision has been made on what the target quantity is to be, variational methods might be one of the options available to approximately estimate it.

Surrogate quantities are therefore approximations of a more abstract nature than numerical approximations. They represent the choice of what to compute, whereas numerical approximations represent choices of how to compute. The two decisions are not independent of each other, as pragmatic concerns regard only their compound effects. A particular composite likelihood might have components too small to capture correlations of interest, for example, whereas a variational approximation to the full data log-likelihood might capture them but only approximately. In between these two extremes, perhaps a range of component sizes that can be approximated with varying accuracy could be considered for undertaking any inference. This particular choice is investigated in chapter 3.

Having articulated which approximation methods might compete with each other as choices in a given application, it is pertinent to describe the features they have that should influence any decisions regarding their use. The profile of an approximation method includes both statistical and computational characteristics, and while computational costs might seem a frustrating presence they are undeniably relevant from a pragmatic perspective. The following sections detail the kind of trade-offs that are made when using approximate inference, and as such should provide some perspective for the investigations and discussions in subsequent chapters.

### **2.6.1 Statistical trade-offs**

Possibly the most well known example of a trade-off between the theoretical properties of statistical estimators is between bias and variance. If the inherent variance

of an estimator (the variance due to variance in the data) is high then inference is subject to the risk of over-fitting; significantly more data than is available would be needed to confidently return accurate estimates. It is sometimes possible to instead estimate a surrogate quantity with less variance, but such a choice will often have to be paid for in the form of bias.

An example of this is the well known *LASSO* model (Tibshirani, 1996; Hastie et al., 2009), which in its original formulation is a restricted form of linear regression. Estimated parameter vectors are constrained to have bounded  $\mathcal{L}_1$  norm:

$$\hat{\beta} = \arg \min_{\beta: \|\beta\|_1 \leq c} \sum_{i=1}^N (Y_i - \mathbf{x}'_i \beta)^2 \quad (2.113)$$

which naturally reduces their variance. The resulting estimates are also biased, but if the bias is small then the trade-off could be judged worthwhile.

Trade-offs of this variety are often, as in the example of the *LASSO* model, not being made due to computational considerations; they are purely statistical. In the *LASSO* example, if a particular linear regression model is highly parametrised and/or only few observations are available for model fitting, then parameter estimates can have high variance. In particular, some of the estimated regression coefficients can have an absolute value far larger than would be in the case of infinite data. Computing the standard maximum likelihood estimates is not difficult computationally, but a surrogate quantity is chosen for estimation because the bias-variance trade-off is deemed more optimal than the unbiased-high variance alternative.

That being said, computational concerns are not irrelevant in many instances of bias-variance trade-offs being made. In the case of approximating component likelihoods of various sizes, a trade-off between bias and variance is also being made. The reasons for the bias and variance in this example are fundamentally computational though, as it can be assumed that with unlimited computing resources a numerical approximation the maximum likelihood estimate with arbitrary accuracy could be made. Such an estimate would be consistent and have optimal variance,



thus outperforming a composite likelihood estimator with components of any size.

It is only when computational constraints demand that inexact numerical approximations to (component) likelihoods be made, and when their bias increases with component size, that a bias-variance trade-off comes into being. For some trade-offs that are between statistical qualities of estimators, therefore, the motivation for the trade-off is driven by computational constraints.

The subject matter of the current thesis is trade-offs that are driven by computational constraints, whether explicit or not. Where the computational challenges are explicit in the trade-off, for example when choosing between numerical approximations whose bias decreases with algorithm run-time, then the optimal choice is generally determined by the availability of computing resources. When implicit though, the exact nature of the trade-off can be subtle yet significant.

### 2.6.2 Computational trade-offs

The challenges of computing a quantity of interest introduces an extra dimension to the trade-offs that can be made when an approximation is necessary. Storage and run-time requirements can mean particular algorithms are particularly compute expensive, or they can scale such that an algorithm is only of practical use in a small-scale setting.

The particle methods described in chapter 2, sec 2.4.2 are a good example of this challenges of scaling. Convergence results are such that the number of particles needed to make estimates with a certain variance must, for example, grow linearly with the length of the chain. This can naively be expressed as  $\text{var}(\hat{E}h) = \mathcal{O}(T)$ , but after noting that there is exponential explosion with the dimension of the latent space  $\mathcal{X}$  it can be written more accurately as  $\text{var}_{\mathcal{X}}(\hat{E}h) = \mathcal{O}(T \exp(\dim(\mathcal{X})))$ .

There can be applied settings where these asymptotic results are not relevant, as sufficient computing resources exist to achieve the task in hand, for example a state space model with sufficiently few latent dimensions for particle methods to be viable. They must always be a consideration when looking for a general solution to an inference problem though, and when deciding which approximations are suitable at different scales.

The asymptotic computational complexity of a given approximation is therefore a matter of relevance to modern statisticians. This relevance has led to the development of *learnability* theory. Beginning with Valiant (1984), the idea that the solution to a problem can be *learnable* has gained traction amongst academics and is now well established. Given a metric that can quantify the error  $\|\theta - \hat{\theta}\|$  of an approximation to the quantity  $\theta$ , then the following definition can be made:

---

**PAC Learnability** An inference problem is *PAC learnable* if an algorithm exists that, given  $\delta, \varepsilon > 0$  and sufficient data, can return with probability  $1 - \delta$  an approximate solution with error at most  $\varepsilon$ , i.e.

$$\mathcal{P}(\|\theta - \hat{\theta}\| \leq \varepsilon) = 1 - \delta \quad (2.114)$$

If the amount of data required is polynomial in  $\frac{1}{\delta}, \frac{1}{\varepsilon}$  then the problem is *efficiently PAC learnable*. An upper bound on the amount of data required is known as the *sample complexity* of the algorithm.

---

Whilst the concept of learnability is relatively abstract, its underlying principle is powerful and intuitive. If the sample complexity of an algorithm, or a class of algorithms, can be given then a basis for comparing algorithms can be made. Choosing between algorithms, or within a class of algorithms, can thus be formalised with a view to making optimal choices.

The sample complexity of an algorithm provides a convenient method for articulating the statistical and computational trade-offs inherent to the algorithm, and also a method of comparing alternative algorithms. Knowing how much the data requirements scale with increasing precision, along with the scaling costs of processing the data, gives an insight into its practicality in a given application, and/or its quality relative to an alternative method.

### 2.6.2.1 Method of moments

One method that can be presented in the context of learnability is the method of moments. When parameters can be expressed as known functions of theoretical moments of data, estimators can be constructed by substituting sample moments for their theoretical counterparts in these functions. The law of large numbers can be used to justify this; sample moments converge to their theoretical values as the amount of data increases.

The implied consistency of method of moments estimators shows that in this context the task of parameter estimation is learnable. The cost of storing / processing data samples may become a significant concern though. In this case, there is a trade-off between the accuracy of an estimator and the costs of estimation. If sample complexity bounds can be derived for a particular algorithm, then the trade-off can be articulated precisely and an informed decision can be made regarding how much data to collect / process.

An example of this is in Anandkumar et al. (2014). Low order sample moments are used to estimate the parameters in exchangeable latent variable models. The models considered are assumed to have low order moment tensors with low rank decompositions, with decomposition analogous to the eigenvalue decompositions of symmetric matrices, i.e. for second and third order moments they assume the decompositions:

$$\begin{aligned}\mathbb{E}[X \otimes X] &= \sum_{i=1}^k w_i \cdot v_i \otimes v_i \\ \mathbb{E}[X \otimes X \otimes X] &= \sum_{i=1}^k \tilde{w}_i \cdot \tilde{v}_i \otimes \tilde{v}_i \otimes \tilde{v}_i\end{aligned}\tag{2.115}$$

where  $\otimes$  denotes the outer product,  $w_i, \tilde{w}_i > 0$ , and  $v_i, \tilde{v}_i \in \mathbb{R}^n$  are eigenvectors that comprise the ground truth moments.

In Anandkumar et al., the authors further assume that model parameters are a function of the eigenvalues and eigenvectors for the second and third order mo-

ments:

$$\theta = f(\mathbf{w}, \mathbf{v}, \tilde{\mathbf{w}}, \tilde{\mathbf{v}}) \quad (2.116)$$

with for some function  $f$ , thus motivating their algorithm for estimating the decomposition structure from sample moments.

The authors present an algorithm for estimating the decompositions from sample moments using a power method, where multiple randomly initialised unit vectors  $\hat{\mathbf{v}}_i^{(0)}$ ,  $i = 1, \dots, L$  are iterated through the tensor and re-normalised:

$$\begin{aligned} \hat{\mathbf{v}}_i^{(k+1)} &= \frac{T \hat{\mathbf{v}}_i^{(k)}}{\|T \hat{\mathbf{v}}_i^{(k)}\|} \\ &\propto \sum_{j_2, \dots, j_p} T_{j_2, \dots, j_p} \hat{\mathbf{v}}_{i, j_2}^{(k)} \cdots \hat{\mathbf{v}}_{i, j_p}^{(k)} \end{aligned} \quad (2.117)$$

that approximately finds the largest eigenvalue  $\hat{w}$  and associated eigenvector  $\hat{\mathbf{v}}$ . These are then ‘deflated’ from the sample moment tensor:

$$\tilde{T} = T - \hat{w} \cdot \hat{\mathbf{v}} \otimes \hat{\mathbf{v}} \otimes \hat{\mathbf{v}} \quad (2.118)$$

and the process is repeated to estimate the modelled  $k$  eigenvalues and eigenvectors.

Sample complexity bounds are calculated on the accuracy of the algorithm, by which the authors showed the task is efficiently PAC learnable. The bounds are too complicated to be both complete and concisely described but, subject to constraints on  $\varepsilon$  and the number  $L$  of random initialisations in (2.117), the amount of data needed for an estimated decomposition to have error that shrinks with  $\varepsilon$  is  $\mathcal{O}\left(\log k + \log \log\left(\frac{\lambda_{\max}}{\varepsilon}\right)\right)$ .

In addition to the sample complexity being calculated, analysis of the computational complexity showed the cost of the algorithm to be  $\mathcal{O}(k^{5+\delta}(\log(k) + \log \log(\frac{1}{\varepsilon})))$ . This shows that the size the dataset is asymptotically not a dominating concern, but rather the modelled rank  $k$  of the tensor decomposition is, and this is partly due to the  $L$  random initialisations in (2.117). The trade-offs for this

algorithm are therefore related to number of modelled eigenvalues and the precision with which they are estimated.

### 2.6.2.2 Stochastic composite likelihood

As mentioned in sec 2.3.1, the research in Dillon and Lebanon (2010) explores the use of randomly selecting which of the possible components in a composite likelihood are actually used for parameter estimation. The authors describe a procedure they call *stochastic composite likelihood* to select the components in composite likelihoods composed of low dimensional conditional distributions. The distributions studied are Markov random fields with graphical structure such that the full data likelihood is intractable. For i.i.d. observations from such a distribution, the authors propose a composite likelihood surrogate composed of conditional distributions:

$$\ell_C(\boldsymbol{\theta} | Y) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k w_j \log \mathcal{P}(Y_{A_j}^{(i)} | Y_{B_j}^{(i)}, \boldsymbol{\theta}) \quad (2.119)$$

where  $w_j > 0$ , and  $A_j, B_j$  are subsets of the variables  $V$ , and  $A_j \cap B_j = \emptyset$ , and  $A_j \neq \emptyset$ . The sizes of each  $A_j, B_j$  determine the computational complexity of calculating each likelihood object in (2.119). The stochastic element of stochastic composite likelihood is introduced with a random binary  $k$  vector for each data point:

$$\begin{aligned} \ell_{SC}(\boldsymbol{\theta} | Y, w, \boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k w_j Z_j^{(i)} \log \mathcal{P}(Y_{A_j}^{(i)} | Y_{B_j}^{(i)}, \boldsymbol{\theta}) \\ Z^{(i)} &\stackrel{\text{i.i.d.}}{\sim} \mathcal{P}(Z | \boldsymbol{\lambda}), \quad \boldsymbol{\lambda}_j = \mathbb{E}[Z_j] > 0 \end{aligned} \quad (2.120)$$

Furthermore, the joint distribution of the elements of  $Z^{(i)}$  can also be defined. This allows, for example, the probability of including a component  $\log \mathcal{P}(Y_{A_j}^{(i)} | Y_{B_j}^{(i)}, \boldsymbol{\theta})$  to be reduced if a component  $\log \mathcal{P}(Y_{A_l}^{(i)} | Y_{B_l}^{(i)}, \boldsymbol{\theta})$  with  $A_j \subset A_l$  had already been included in  $\ell_{SC}$  for data point  $i$ .

By controlling the weights  $w_j$  and the distribution of inclusion vectors  $Z^{(i)}$ , the asymptotic statistical efficiency and computational costs of the (asymptotically

unbiased) maximum stochastic likelihood estimator

$$\hat{\theta}_{\text{mscl}} = \arg \max_{\theta} \ell_{\text{SC}}(\theta | Y, w, \lambda) \quad (2.121)$$

can be controlled continuously. In the asymptotic analysis of the estimators, the dependency of the estimates on particular realisations of the component inclusion vectors  $Z^{(i)}$  can be ignored in favour of the inclusion probabilities  $\lambda_j$ . As the trade-offs being examined in the paper are between computational costs and statistical efficiencies, and as computational costs are independent of the weights  $w_j$ , the authors propose making a pragmatic choice for  $\lambda$  with respect to the available computing resources, and optimising the estimator with respect to the  $w_j$  iteratively:

$$\begin{aligned} \hat{\theta}_{\text{mscl}}^{(k)} &= \arg \max_{\theta} \ell_{\text{SC}}(\theta | Y, w^{(k)}, \lambda) \\ w^{(k+1)} &= \arg \min_w -\log |\hat{H} \hat{J}^{-1} \hat{H}| = \log |\hat{J}| - 2 \log |\hat{H}| \end{aligned} \quad (2.122)$$

where  $(\hat{H} \hat{J}^{-1} \hat{H})^{-1}$  is the finite sample estimate of the asymptotic estimator covariance matrix. The authors approximate these terms using the product of diagonal elements as a proxy for determinants:

$$\begin{aligned} \log |\hat{H}| &\approx \sum_{r=1}^p \log \sum_{j=1}^k w_j \lambda_j K_{r,r}^{(j,j)} \\ \log |\hat{J}| &\approx \sum_{r=1}^p \log \sum_{j=1}^k \sum_{l=1}^k w_j w_l \lambda_j \lambda_l K_{r,r}^{(j,l)} \end{aligned} \quad (2.123)$$

where

$$K^{(j,l)} = \text{cov}_{\theta}(\nabla_{\theta} \log \mathcal{P}(Y_{A_j} | Y_{B_j}, \theta), \nabla_{\theta} \log \mathcal{P}(Y_{A_l} | Y_{B_l}, \theta)) \quad (2.124)$$

with elements  $K_{s,t}^{(j,l)}$ . The efficiency of the estimator is therefore maximised with respect to the available computing resources. Fitting maximum stochastic composite likelihood (mcs) estimators to models with both synthetic and real data showed the trade-off to be controlled effectively. Furthermore, in real data that was poorly

modelled by the Markov random field in question, mcs1 estimators were found to, in effect, ‘self-regularise’; the full data maximum likelihood estimator would have poorer predictive performance for some datasets than the mscl estimators. This was hypothesised to be due to empirical inconsistencies between the data and the modelled conditional independence structure of the model in question being ‘overlooked’ by components that had  $|A_j| \ll |V|$ .

### 2.6.2.3 Convex relaxations

On a different scale of comparison is Chandrasekaran and Jordan (2013). Here, an entire class of algorithms has its profile examined with a view to choosing between members in any given application. Specifically, the de-noising of a sequence of observations  $Y_i \in \mathbb{R}^p$  modelled as

$$\mathbf{Y}_i = \mathbf{x}^* + \sigma \mathbf{z}_i, \quad \mathbf{z}_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_p(0, 1) \quad (2.125)$$

where  $\mathcal{N}_p(0, 1)$  is the standard  $p$ -dimensional Gaussian and  $\mathbf{x}^* \in \mathcal{S} \subset \mathbb{R}^p$  for a known subset  $\mathcal{S}$  of  $\mathbb{R}^p$ . It is assumed that  $p$  is large. The task of inference is to estimate  $\mathbf{x}^*$ .

An intuitive estimator could be the projection of the sample mean onto  $\mathcal{S}$ , but this is computationally challenging for arbitrary  $\mathcal{S}$ . The general inference procedure they suggest is to project the sample mean onto a convex set  $\mathcal{C} \supset \mathcal{S}$ :

$$\hat{\mathbf{x}}_n(\mathcal{C}) = \arg \min_{\mathbf{x} \in \mathcal{C}} \|\bar{\mathbf{Y}} - \mathbf{x}\| \quad (2.126)$$

where the estimator is subscripted with the number of samples  $n$ . If the convex set  $\mathcal{C}$  is itself defined as the intersection of tractable convex sets, then the complexity of the minimisation in (2.126) increases with the number of these tractable sets in the intersection. The intersection of many sets can approximate an arbitrary set  $\mathcal{S}$  more closely, but is more expensive to minimise over.

The authors show that for an estimator to have a mean squared error

$\mathbb{E} \left[ \|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\|^2 \right] \leq 1$  then a sample complexity of

$$n \geq \sigma^2 g(T_{\mathcal{C}}(\mathbf{x}^* \cap B^p)) \quad (2.127)$$

is required, where

$$g(Z) = \mathbb{E}_{\mathbf{v}} \left[ \sup_{\mathbf{a} \in Z} \langle \mathbf{a}, \mathbf{v} \rangle^2 \right] \quad \mathbf{v} \sim \mathcal{N}(0, I_p) \quad (2.128)$$

is the *squared Gaussian complexity* of the set  $Z \in \mathbb{R}^p$ , with  $I_p$  the  $p \times p$  identity matrix,  $B^p$  is the unit ball, and  $T_{\mathcal{C}}(\mathbf{x}^*)$  is the *tangent cone* at  $\mathbf{x}^*$  relative to  $\mathcal{C}$ :

$$T_{\mathcal{C}}(\mathbf{x}^*) = \text{cone}\{\mathbf{b} - \mathbf{x}^* : \mathbf{b} \in \mathcal{C}\} \quad (2.129)$$

where the *cone* of a convex set is its closure under positive linear combinations. This result formalises the intuition that for  $\mathcal{C}_j \supset \mathcal{C}_k \supset \mathcal{S}$ :

$$\mathbb{E} \left[ \|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C}_j)\|^2 \right] \geq \mathbb{E} \left[ \|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C}_k)\|^2 \right] \quad (2.130)$$

The result (2.127) motivates the authors to suggest that in cases where the amount of data exceeds the sample complexity bound of (2.127) for a simple convex set  $\mathcal{C}$ , then the computational benefits of using the estimator  $\hat{\mathbf{x}}_n(\mathcal{C})$  come at an acceptable cost. Here a simple convex set is defined as the intersection of only few tractable convex sets.

#### 2.6.2.4 Conclusions

As all of the examples above show, if statistical efficiency and computational complexity can be expressed for a given algorithm, then optimising an algorithm with regards to trade-offs can be formally justified. Whilst these complexities are not always tractable, even establishing learnability can be a positive grounding for a proposed algorithm. Furthermore, any complexity rates that are calculated can give insight into when an algorithm becomes impractical or impossible to implement.

Even when complexities are calculated though, they might not be perfectly



illustrative of an algorithms statistical or computational profile. Rates of change are valuable for informing choices as data or parametrised modelling assumptions go towards infinity, but in real world applications limiting behaviour might not have come to dominate an algorithm's performance.



## Chapter 3

# Composite Likelihood Estimators in State Space Models

This chapter is an investigation into the trade-off between different component sizes in approximate composite likelihood inference in state space models. A specific model with a linear-Gaussian state space and Student-t observations is chosen to represent the challenges involved in such approximate inference. Posterior distributions of the latent variables in this model are intractable, and when the latent dimensionality is too high Monte Carlo methods of approximation are impractical. Deterministic approximations provide a natural alternative.

The deterministic approximation made here is the variational approximation. Variational approximations are detailed in chapter 2, sec 2.2, but to summarise quickly they place lower bounds on data likelihood evaluations. Inference is performed using the lower bound as a proxy for evaluations of the likelihood.

The lower bounds are made in latent variable models by taking the expectation of the full log-likelihood with respect to a methodically chosen tractable distribution. The closer this distribution is to the true latent posterior, where proximity here has a specific meaning outlined below, the tighter the corresponding lower bound will be. When data are correlated, as in state space and other time series models, the tightness of the lower bound will generally loosen as more data are observed. In particular, it is shown in the derivations below that updates to the variational approximations are equivalent to Gibbs type updates that have no uncertainty in the

value of the conditioning latent variables. This implicitly fails to propagate uncertainty, see Turner and Sahani (2008) for a detailed discussion of the issue. There could consequently be significant bias introduced into variational approximations, as the impact of failing to propagate uncertainty could be uneven over time and/or over iterations. As stated in Turner and Sahani, it is not so much the tightness of variational lower bounds that is important but the consistency of their tightness over different parameter settings that affects their bias.

The use of composite likelihoods is made in an effort to counter this effect. By using a composite likelihood whose components are the marginals of fixed length subsets of the data, the quality of the variational approximations will no longer decline with increasing amounts of observed data. Unfortunately, this comes at the cost of estimator variance that increases as the component lengths get shorter.

A composite likelihood with shorter components, therefore, benefits from more accurate variational approximations to each component likelihood, but suffers from increased estimator variance. The trade-off between these two concerns, regarding both the quality of estimations and their computational costs, is now investigated.

### 3.1 The state space model

State space models are an actively used class of models in signal processing, finance, and robotics amongst other applied disciplines, and are actively researched within the statistics community. The model consists of an autoregressive latent state process which propagates through time, and the conditional distributions of observations given the state process.

The state space model of  $T$  sequential observations, also defined in chapter 2 section 2.4.2 but repeated here for clarity, consists of a latent state process  $\{X_t\}_{t=1}^T$  taking values in some space  $\mathcal{X}$ , and observations  $\{Y_t\}_{t=1}^T$  taking values in some space  $\mathcal{Y}$ . The state process is first order Markov and the observations are conditionally

independent given the state process:

$$X_1 \sim \mathcal{P}(X_1 | \theta)$$

$$X_t | X_{1:t-1} \sim X_t | X_{t-1} \sim \mathcal{P}(X_t | X_{t-1}, \theta) \quad t \in 2, \dots, T$$

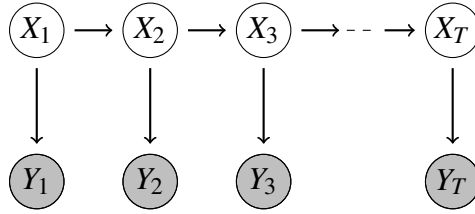
$$Y_t | X_{1:T} \sim Y_t | X_t \sim \mathcal{P}(Y_t | X_t, \theta) \quad (3.1)$$

$$(3.2)$$

where the colon in subscripts  $t_1 : t_2$  indicates the interval of time periods  $t_1, \dots, t_2$ .

The distributions in (3.2) are assumed to have densities  $f_\theta(x_1), f_\theta(x_t | x_{t-1}), f_\theta(y_t | x_t)$ .

The directed graph of the state space model is shown in fig 3.1.



**Figure 3.1:** The directed graph of the state space model. Observed variables are shaded grey, dashed line indicates repeated pattern until  $t = T$ .

Inference in state space models exploits the tree structure illustrated in fig 3.1 with the use of message passing algorithms, and is usually directed towards one of:

*Filtering:* Producing a sequence of estimated latent posterior distributions conditioned on all currently available observations

$$\mathcal{P}(X_t | Y_{1:t}, \theta) \quad t \in 1, \dots, T \quad (3.3)$$

For  $t = 1$  this is achieved by conditioning  $X_1 | Y_1$ :

$$f_\theta(x_1 | Y_1) = \frac{f_\theta(x_1)f_\theta(Y_1 | x_1)}{\int f_\theta(x_1)f_\theta(Y_1 | x_1) dx_1} \quad (3.4)$$

For  $t = 2, \dots, T$  this is achieved by calculating, either exactly or approximately, the following two-step recursion with each new observation:

1. Predict:

$$f_{\theta}(x_{t+1} | Y_{1:t}) = \int f_{\theta}(x_t | Y_{1:t}) f_{\theta}(x_{t+1} | x_t) dx_t \quad (3.5)$$

2. Update:

$$\begin{aligned} f_{\theta}(x_{t+1} | Y_{1:t+1}) &= \frac{f_{\theta}(x_{t+1} | Y_{1:t}) f_{\theta}(Y_{t+1} | x_{t+1})}{\int f_{\theta}(x_{t+1} | Y_{1:t}) f_{\theta}(Y_{t+1} | x_{t+1}) dx_{t+1}} \\ &= \frac{f_{\theta}(x_{t+1} | Y_{1:t}) f_{\theta}(Y_{t+1} | x_{t+1})}{f_{\theta}(y_{t+1} | y_{1:t})} \end{aligned} \quad (3.6)$$

*Smoothing:* Producing a sequence of estimated latent posterior distributions conditioned on all observations

$$\mathcal{P}(X_t | Y_{1:T}, \theta) \quad t \in 1, \dots, T \quad (3.7)$$

This procedure is completed in backward time. For  $t = T$  the smoothed distribution equals the filtered distribution by construction. For  $t = T - 1, \dots, 1$  this is achieved by calculating, either exactly or approximately, another two-step recursion, with each step given the names of their filtering equivalent for purposes of analogy:

1. ‘Predict’:

$$f_{\theta}(x_t | Y_{1:t}, x_{t+1}) = \frac{f_{\theta}(x_t | Y_{1:t}) f_{\theta}(x_{t+1} | x_t)}{\int f_{\theta}(x_t | Y_{1:t}) f_{\theta}(x_{t+1} | x_t) dx_t} \quad (3.8)$$

2. ‘Update’:

$$f_{\theta}(x_t | Y_{1:T}) = \int f_{\theta}(x_t | Y_{1:t}, x_{t+1}) f_{\theta}(x_{t+1} | Y_{1:T}) dx_{t+1} \quad (3.9)$$

*Prediction:* Producing a sequence of predictive distributions of as yet unseen observations

$$\mathcal{P}(Y_{t+n} | Y_{1:t}, \theta) \quad t \in 1, \dots, T \quad (3.10)$$

This is also achieved by, after completing the filtering procedure for time  $t$ , calculating either exactly or approximately another two-stage process:

1. Perform  $n$  compositions of the filtering predict step:

$$f_{\theta}(x_{t+n} | Y_{1:t}) = \int f_{\theta}(x_t | Y_{1:t}) \prod_{i=1}^n f_{\theta}(x_{t+i} | x_{t+i-1}) dx_{t:t+n-1} \quad (3.11)$$

2. Data prediction:

$$f_{\theta}(y_{t+n} | Y_{1:t}) = \int f_{\theta}(x_{t+n} | Y_{1:t}) f_{\theta}(y_{t+n} | x_{t+n}) dx_{t+n} \quad (3.12)$$

### 3.1.1 Approximate inference

All of the calculations described in (3.5) - (3.12) can only be calculated analytically if (assuming continuous latent variables) all of the densities  $f_{\theta}(x_1), f_{\theta}(x_t | x_{t-1}), f_{\theta}(y_t | x_t)$  are linear Gaussian, i.e. if

$$\begin{aligned} X_1 &\sim \mathcal{N}(X_1 | \mu_1, \Sigma_1) \\ X_t | X_{t-1} &\sim \mathcal{N}(X_t | A_t X_{t-1}, V_t) \\ Y_t | X_t &\sim \mathcal{N}(Y_t | C_t X_t, W_t) \end{aligned} \quad (3.13)$$

with  $A_t, C_t$  linear mappings  $A_t: \mathcal{X} \rightarrow \mathcal{X}: X_{t-1} \mapsto A_t X_{t-1}, C_t: \mathcal{X} \rightarrow \mathcal{Y}: X_t \mapsto C_t X_t$ . The filtering and smoothing algorithms in this instance are known as *Kalman filtering / smoothing* (Kalman, 1960; Kalman and Bucy, 1961; Rauch et al., 1965), and return the exact (Gaussian) filtered and smoothed distributions.

In all other state space models approximations have to be made. When the dimension of  $\mathcal{X}$  is low then particle methods are commonly used, but the estimates they produce require the number of particles to grow with the number of observations, and they also suffer variance explosion with increasing  $\mathcal{X}$  dimensionality. For long chains of data observations and/or high dimensional  $\mathcal{X}$  therefore, alternative methods are necessary.

Other Monte Carlo techniques can be substituted in many contexts, but these

will all generally suffer the curse of dimensionality; the variance of estimates will explode exponentially with the length of the observation chain. This is somewhat mitigated with MCMC algorithms, whose samples converge on the regions of posterior high density, but the run time / computational cost of a low variance MCMC estimate can still be prohibitively high.

Deterministic methods are an alternative with relatively untapped potential. For models with only mild non-linearities and Gaussian noise the well known *extended Kalman filter* (EKF) (Jazwinski, 1970) and *unscented Kalman filter* (UKF) (Julier and Uhlmann, 2004) provide accurate and computationally cheap approximations to the filtered distributions (3.3). They are both used in Gaussian state space models with conditional distributions of the form:

$$\begin{aligned} X_t | X_{t-1} &\sim \mathcal{N}(X_t | g_X(X_{t-1}), V_t) \\ Y_t | X_t &\sim \mathcal{N}(Y_t | g_Y(X_t), W_t) \end{aligned} \quad (3.14)$$

with  $g_X, g_Y$  non-linear functions of the state process. The EKF approximates these functions by taking first order Taylor expansions around the filtered mean:

$$\begin{aligned} g_X(X_t) &\approx g_X(\mathbb{E}[X_t | Y_{1:t}]) + G_X(X_t - \mathbb{E}[X_t | Y_{1:t}]) \\ g_Y(X_t) &\approx g_Y(\mathbb{E}[X_t | Y_{1:t}]) + G_Y(X_t - \mathbb{E}[X_t | Y_{1:t}]) \end{aligned} \quad (3.15)$$

where  $G_X = \left. \frac{dg_X}{dX} \right|_{X=\mathbb{E}[X_t | Y_{1:t}]}$ ,  $G_Y = \left. \frac{dg_Y}{dX} \right|_{X=\mathbb{E}[X_t | Y_{1:t}]}$  are the Jacobian matrices of partial derivatives evaluated at  $X = \mathbb{E}[X_t | Y_{1:t}]$ . The UKF takes a different approach to the approximation, and produces the *unscented transform* of the filtered mean  $\mathbb{E}[X_t | Y_{1:t}]$ , which is a deterministic set of points that are propagated through the functions  $f_X, f_Y$ . These propagated points are then used to estimate the mean and variance of the distributions in (3.5) and (3.6).

Both the EKF and UKF are useful and effective approximations in the restricted model class (3.14), but the EKF performs badly when the functions  $g_X, g_Y$  are highly non-linear and they can both suffer from numerical stability issues. An alternative deterministic approximation framework is the method of *variational ap-*



proximations.

In the current context, a variational approximation to a posterior distribution of interest, i.e. (3.3), (3.7), (3.10), is simply an optimally chosen distribution  $q^*$  over  $\mathcal{X}^T$  from a restricted class  $Q$  of tractable distributions. Optimally chosen here means that  $q^*$  minimises the *Kullback-Leibler divergence* from  $q \in Q$  to the posterior:

$$\text{KL}[q || p(x | Y)] = \int q(x) \log \frac{q(x)}{p(x | Y)} dx \quad (3.16)$$

i.e.

$$q^* = \arg \min_{q \in Q} \text{KL}[q(X) || p(X | Y)] \quad (3.17)$$

for observed data  $Y$ . More details on variational approximations can be found in chapter 2, sec 2.2.

### 3.1.2 Gaussian state space, Student-t observations

The specific state space model investigated in this chapter is time-homogeneous, has a Gaussian state space with linear transitions, and has Student-t observations:

$$\begin{aligned} X_1 &\sim \mathcal{N}(X_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ X_t | X_{t-1} &\sim \mathcal{N}(X_t | AX_{t-1} + \mathbf{b}, V) \\ Y_t | X_t &\sim \mathcal{S}(Y_t | \mathbf{v}, CX_t + \mathbf{d}, W) \end{aligned} \quad (3.18)$$

where  $\dim \mathcal{X} = k$ ,  $\dim \mathcal{Y} = p$

It is convenient here to make use of the formulation of the Student-t distribution as an infinite mixture of Gaussians:

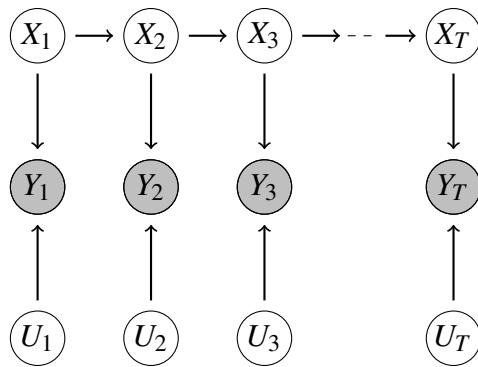
$$Z \sim \mathcal{S}(Z | \mathbf{v}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \Leftrightarrow f_{\theta}(z) = \int_0^{\infty} \mathcal{N}\left(z \mid \boldsymbol{\mu}, \frac{1}{u}\boldsymbol{\Sigma}\right) \mathcal{G}\left(u \mid \frac{\mathbf{v}}{2}, \frac{\mathbf{v}}{2}\right) du \quad (3.19)$$

and its implied conditional form:

$$Z \sim \mathcal{S}(Z | \mathbf{v}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \Leftrightarrow Z | U \sim \mathcal{N}\left(Z \mid \boldsymbol{\mu}, \frac{1}{U}\boldsymbol{\Sigma}\right), \quad U \sim \mathcal{G}\left(U \mid \frac{\mathbf{v}}{2}, \frac{\mathbf{v}}{2}\right) \quad (3.20)$$

where  $\mathcal{G}(\cdot | \alpha, \beta)$  is the Gamma distribution with shape parameter  $\alpha$  and rate parameter  $\beta$ , to give an alternative formulation of the model (3.18):

$$\begin{aligned}
 X_1 &\sim \mathcal{N}(X_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\
 X_t | X_{t-1} &\sim \mathcal{N}(X_t | AX_{t-1} + \mathbf{b}, V) \\
 U_t &\sim \mathcal{G}\left(U_t \mid \frac{\mathbf{v}}{2}, \frac{\mathbf{v}}{2}\right) \\
 Y_t | X_t, U_t &\sim \mathcal{N}\left(Y_t \mid CX_t + \mathbf{d}, \frac{1}{U_t}W\right)
 \end{aligned} \tag{3.21}$$



**Figure 3.2:** The directed graph of the Gaussian-Student state space model with alternative representation (3.21). Observed variables are shaded grey, dashed line indicates repeated pattern until  $t = T$ .

Student-t distributed variables have a density that is superficially similar to a Gaussian density, but it has heavier tails. This places more probability mass in the tails, and as such the Student-t distribution can be a more appropriate modelling assumption than a Gaussian in many applied fields. Financial time-series modelling, for example, is well known for the empirical heavy tails of modelled data. An effective use of approximation methods in the implementation of (3.21) therefore has benefit to practitioners as well as providing insight into approximation methods generally.

### 3.1.3 Further model assumptions

To construct a parameter estimation algorithm for the above model, the following further assumptions are made on the model: *a*) the  $k \times T$  dimensional latent process  $X_{1:T}$  is composed of  $k$  independent  $T$  dimensional stationary latent processes

$X_{1:T}^i$ ,  $i = 1, \dots, k$ , each with the common marginal distribution  $X_t^i \sim \mathcal{N}(0, 1)$ ,  $t = 1, \dots, T$ , but with unique auto-correlation parameters  $\text{corr}(X_t^i, X_{t+1}^i) = \rho_i$ ; and  $b$ ) each element  $Y_t^j$ ,  $j \in 1, \dots, p$  of the observed data  $Y_t$  is conditionally independent of all other elements given the latent variables  $X_t, U_t$ .

The stationarity assumption on  $X_{1:T}$ , which implies that  $X_t \sim \mathcal{N}(\mu, \Sigma)$  at all times  $t$ , is a common assumption in practice as it restricts the size of the parameter vector. It can be justified if the data itself appear to be stationary. As the latent precision scaling variables  $U_{1:T}$  are i.i.d., assuming stationarity of  $X_{1:T}$  is equivalent to assuming stationarity of  $Y_{1:T}$ . The additional assumption, implied above, that  $X_t \sim \mathcal{N}(0, I_k)$  at each time  $t$ , where  $I_k$  is the  $k \times k$  identity matrix, is also made. Furthermore, it is made without loss of generality. Regarding the zero mean, any mean vector  $\mu$  can be absorbed into the intercept vector  $\mathbf{d}$ :

$$\begin{aligned} X_t &= \tilde{X}_t + \mu & \tilde{X}_t &\sim \mathcal{N}(0, \Sigma) \\ \Rightarrow CX_t + \mathbf{d} &= C(\tilde{X}_t + \mu) + \mathbf{d} \\ \Rightarrow \tilde{\mathbf{d}} &= \mathbf{d} + C\mu \end{aligned} \tag{3.22}$$

This also implies the latent intercept vector  $\mathbf{b} = 0$ . Regarding the unit covariance matrix, this can be assumed without loss of generality as the linear transformation of a standard normal that gives  $\Sigma$  as a covariance matrix can be absorbed into  $C$ :

$$\begin{aligned} X_t &= L\tilde{X}_t & \tilde{X}_t &\sim \mathcal{N}(0, I_k) \\ \Rightarrow CX_t &= CL\tilde{X}_t \\ \Rightarrow \tilde{C} &= CL \end{aligned} \tag{3.23}$$

where  $L$  is any matrix such that  $LL' = \Sigma$ , which holds for example if  $L$  defines the Cholesky decomposition of  $\Sigma$ .

Modelling the latent process as  $k$  independent processes has the implication that the transition matrix  $A$  is diagonal. As each dimension is assumed to have unit variance at each time, the parameters on the diagonal of  $A$  equal the temporal

correlations for each dimension:

$$\begin{aligned} A &= \text{diag}(\boldsymbol{\rho}) \quad \boldsymbol{\rho} = (\rho_1, \dots, \rho_k)' \\ \rho_i &= \text{corr}(X_t^i, X_{t+1}^i) \end{aligned} \quad (3.24)$$

where the superscript  $i$  denotes the element of the process at each time  $t$ . This allows the joint distribution of  $X_{t:t+1}^i$  to be parametrised by the single parameter  $\boldsymbol{\rho}^i$ :

$$\begin{aligned} X_{t:t+1}^i &\sim \mathcal{N}(X_{t:t+1}^i \mid \boldsymbol{\rho}^i) \\ \Leftrightarrow X_{t:t+1}^i &\sim \mathcal{N}(X_{t:t+1}^i \mid \mathbf{0}, \Sigma^i) \quad \Sigma^i = \begin{pmatrix} 1 & \rho^i \\ \rho^i & 1 \end{pmatrix} \end{aligned} \quad (3.25)$$

and implies a diagonal conditional covariance matrix  $V$ :

$$\begin{aligned} V &= I_k - \text{diag}(\boldsymbol{\rho}^2) \\ \boldsymbol{\rho}^2 &= (\rho_1^2, \dots, \rho_k^2)' \end{aligned} \quad (3.26)$$

As noted above, in addition to the structure assumed of  $X_{1:T}$ , each element  $Y_t^j$ ,  $j = 1, \dots, p$  of  $Y_t$  is assumed to be conditionally independent given the latent variables. This implies that the conditional covariance matrix of the conditionally Gaussian data  $Y_t$  is diagonal:

$$W = \text{diag}(\mathbf{w}), \quad \mathbf{w} = (w_1, \dots, w_p)' \quad (3.27)$$

This assumption was made for ease of exposition, though removing the restriction and allowing a general conditional covariance matrix is not a challenging extension. After making all of the above assumptions, the model therefore becomes:

$$\begin{aligned} X_1 &\sim \mathcal{N}(X_1 \mid \mathbf{0}, I_k) \\ X_{t+1} \mid X_t &\sim \mathcal{N}(X_{t+1} \mid \text{diag}(\boldsymbol{\rho})X_t, I_k - \text{diag}(\boldsymbol{\rho}^2)) \\ Y_t \mid X_t &\sim \mathcal{S}(Y_t \mid \mathbf{v}, CX_t + \mathbf{d}, \text{diag}(\mathbf{w})) \end{aligned} \quad (3.28)$$

and the parameters of the model are:

$$\begin{aligned}
\boldsymbol{\rho} &= (\rho_i \in (-1, 1) : i = 1, \dots, k) \\
\mathbf{v} &> 0 \\
\mathbf{C} &= (c_{i,j} \in \mathbb{R} : i = 1, \dots, p, j = 1, \dots, k) \\
\mathbf{d} &= (d_i \in \mathbb{R} : i = 1, \dots, p) \\
\mathbf{w} &= (w_i > 0 : i = 1, \dots, p)
\end{aligned} \tag{3.29}$$

These assumptions on  $X_{1:T}$  thus make the model (3.21) comparable to the factorial hidden Markov model of Ghahramani and Jordan (1997), discussed in chapter 2, sec 2.2.2. Whereas the approximations to the posterior in Ghahramani and Jordan (1997) factorise over latent Markov chains, the only factorisation considered here is between  $X_{1:T}$  and  $U_{1:T}$ .

---

**Remark** As composite likelihood estimators are to be approximated, the latent variables indexed by time points not in each component  $t_c : t_c + n - 1$  (see sec 3.2 below) must be integrated out before variational approximations to posterior distributions can be made. This is most easily achieved by exploiting the stationarity assumption on (and the Gaussianity of)  $X_{1:T}$ :

$$\mathcal{P}(X_{t_c:t_c+n-1}) = \mathcal{N}(X_{t_c} | 0, I_k) \prod_{t=t_c+1}^{t_c+n-1} \mathcal{N}(X_t | AX_{t-1}, V) \tag{3.30}$$

for all components of size  $n$ . As the latent process  $X_{1:T}$  is composed of  $k$  independent processes  $X_{1:T}^i$ , this distribution is equal to the product of  $k$  Gaussian vectors of length  $n$ :

$$\mathcal{P}(X_{t_c:t_c+n-1} | \boldsymbol{\theta}) = \prod_{i=1}^k \mathcal{N}(X_{1:n}^i | 0, \Lambda_i^{-1}) \tag{3.31}$$

where

$$\Lambda_i = \frac{1}{1 - \rho_i^2} \begin{pmatrix} 1 & -\rho_i & 0 & \cdots & 0 \\ -\rho_i & 1 + \rho_i^2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 + \rho_i^2 & -\rho_i \\ 0 & \cdots & 0 & -\rho_i & 1 \end{pmatrix} \quad (3.32)$$

The determinant  $|\Lambda_i|$  of each precision matrix  $\Lambda_i$  is required for the estimation of the corresponding correlation parameter  $\rho_i$ . The tri-diagonal form of each  $\Lambda_i$  allows its determinant to be calculated analytically and gives it a concise closed form:

$$|\Lambda_i| = (1 - \rho_i^2)^{-(n-1)} \quad (3.33)$$

### 3.2 Composite likelihoods

The following is a brief section defining notation particular to composite likelihoods. Each component in a composite likelihood will be a marginal likelihood of  $n$  contiguous data observations. For a given component size  $n$ , two different structures will be considered: *a*) every possible such component being included:

$$L_C^{n,\cup}(\theta) = \prod_{t_c=1}^{T-n+1} \mathcal{P}(Y_{t_c:t_c+n-1} \mid \theta) \quad (3.34)$$

where the superscript  $n$  indicates the size of each component, and *b*) only disjoint components being included:

$$L_C^{n,\dot{\cup}}(\theta) = \prod_{t_c=1}^{\lfloor \frac{T}{n} \rfloor} \mathcal{P}(Y_{(t_c-1)n+1:t_cn} \mid \theta) \quad (3.35)$$

When the component structure is not pertinent to the discussion the superscripts will be dropped, and the index set of the data belonging to the  $c^{\text{th}}$  component will be denoted  $M_c$ . For example,

$$Y_{M_c} = \{Y_t : t \in M_c\} \quad (3.36)$$

denotes the observed data belonging to the  $c^{\text{th}}$  component.

### 3.3 Variational approximations

Maximum likelihood estimation of the model (3.21) is intractable, due to the non-Gaussian conditional distributions  $\mathcal{P}(Y_t | X_t, U_t)$ . Variational approximations are chosen as a numerical approximation to the likelihood, with an investigation into the effect on the bias of the size of the composite likelihood blocks. Approximations to maximum composite likelihood estimators are proposed as surrogate quantities, with the effect of having  $M$  components of size  $n$  on the bias and variance of estimators being investigated:

$$L_C(\theta) = \prod_{c=1}^C f_{\theta}(Y_{M_c}) \quad (3.37)$$

A variational approximation to the smoothed latent joint distribution  $\mathcal{P}(X_{M_c}, U_{M_c} | Y_{M_c})$  is now derived, with a view to placing a lower bound on the composite log-likelihood as a sum of lower bounds on each component log-likelihood. To keep indices cleaner, the  $t^{\text{th}}$  time point in the interval  $M_c$  associated to component  $c$  will be denoted with the subscript  $(c, t)$ .

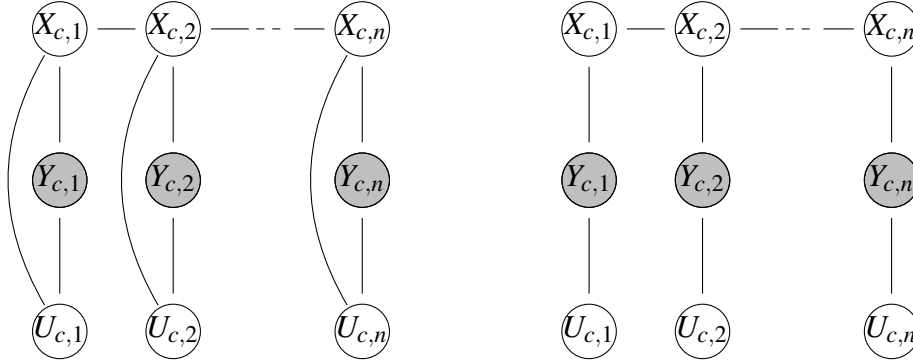
As described in detail in chapter 2, sec 2.2, variational approximations in latent variable models make use of a tractable class  $Q$  of distributions. A common choice for  $Q$ , and the choice made here, is for  $Q$  to consist of all distributions that factorise over some subsets of latent dimensions.

The choice of factorising subsets made here follows that of Svensén and Bishop (2005), and keeps the Gaussian dimensions in one factor and the Gamma dimen-

sions in another:

$$Q = \{q(X_{M_c}, U_{M_c}) : q(X_{M_c}, U_{M_c}) = q_X(X_{M_c})q_U(U_{M_c})\} \quad (3.38)$$

This is equivalent to defining  $Q$  as the class of distributions derived from (3.21) whose undirected graphs have had the edges  $(X_t, U_t)$  removed, see fig 3.3. As de-



**Figure 3.3:** The moralised graph of the Gaussian-Student state space model with alternative representation (3.21) (left), and the graph of distributions with the factorisation defined in (3.38) (right). Observed variables are shaded grey, dashed line indicates repeated pattern until  $t = (c, n)$ .

scribed in chapter 2, sec 2.2, the  $q^* = \operatorname{argmin}_{q \in Q} \text{KL}[q || \mathcal{P}(X_{M_c}, U_{M_c} | Y_{M_c})]$  that minimises the KL divergence between  $q \in Q$  and  $\mathcal{P}(X_{M_c}, U_{M_c} | Y_{M_c})$  is found using the following coupled equations:

$$\begin{aligned} \log q_X^*(X_{M_c}) &= \mathbb{E}_{q_U^*}[\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c})] + \text{constant} \\ \log q_U^*(U_{M_c}) &= \mathbb{E}_{q_X^*}[\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c})] + \text{constant} \end{aligned} \quad (3.39)$$

and as the prior distributions  $X_{M_c}, U_{M_c}$  are both conjugate to  $Y_{M_c} | X_{M_c}, U_{M_c}$ , the optimal  $q_X^*, q_U^*$  in (3.39) will belong to the same parametrised families as the priors, i.e.

$$\begin{aligned} q_X^* &= \mathcal{N}(\mu^*, \Sigma^*) \\ q_U^* &= \mathcal{G}(\alpha^*, \beta^*) \end{aligned} \quad (3.40)$$

for some parameters  $\mu^*, \Sigma^*, \alpha^*, \beta^*$ . The full component log-likelihood  $\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c})$



is:

$$\begin{aligned}\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c}) &= \log \mathcal{P}(X_{M_c}) \mathcal{P}(U_{M_c}) \mathcal{P}(Y_{M_c} | X_{M_c}, U_{M_c}) \\ &= \log \mathcal{P}(X_{M_c}) + \log \mathcal{P}(U_{M_c}) + \log \mathcal{P}(Y_{M_c} | X_{M_c}, U_{M_c})\end{aligned}\quad (3.41)$$

### 3.3.1 Estimation algorithm

The parameter estimation algorithm returns a variational approximation to the maximum composite likelihood estimator for a given set of components. This is achieved through an iterative procedure of repeatedly *a*) optimising the lower bound on each component log-likelihood  $\log L(\theta | Y_{M_c})$  given the current parameter estimate  $\hat{\theta}^{(k)}$ :

$$q_c^* = \arg \min_{q \in \mathcal{Q}} \text{KL} \left[ q || \mathcal{P}(X_{M_c}, U_{M_c} | Y_{M_c}, \hat{\theta}^{(k)}) \right] \quad (3.42)$$

and *b*) updating the parameter estimate by maximising the lower bound:

$$\hat{\theta}^{(k+1)} = \arg \max_{\theta} \sum_{c=1}^C \mathbb{E}_{q_c^*} [\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c} | \theta)] \quad (3.43)$$

until parameter estimates have converged.

#### 3.3.1.1 Optimising the lower bound

The lower bound on the (component) log-likelihood of the data covering the time interval  $M_c = t_c, \dots, t_c + n - 1$  given by

$$\log L(\theta | Y_{M_c}) \geq \mathbb{E}_{q_X, q_U} [\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c} | \theta)] \quad (3.44)$$

is optimised by initialising  $q_X, q_U$  to some  $q_X^{(0)}, q_U^{(0)}$  and updating each of them iteratively according to the coupled equations (3.39):

$$\begin{aligned}\log q_X^{(k+1)}(X_{M_c}) &= \mathbb{E}_{q_U^{(k)}} [\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c} | \theta)] + \text{constant} \\ \log q_U^{(k+1)}(U_{M_c}) &= \mathbb{E}_{q_X^{(k+1)}} [\log \mathcal{P}(X_{M_c}, U_{M_c}, Y_{M_c} | \theta)] + \text{constant}\end{aligned}\quad (3.45)$$

and as  $\mathcal{P}(X_{M_c} | \theta)$ ,  $\mathcal{P}(U_{M_c} | \theta)$  are both conjugate to  $\mathcal{P}(Y_{M_c} | X_{M_c}, U_{M_c}, \theta)$ , the form of  $q_X^{(k)}$ ,  $q_U^{(k)}$  will stay constant across iterations. This implies that one of  $q_X, q_U$  should be initialised to some member of their prior family:

$$\begin{aligned} q_X^{(0)}(X_{M_c}) &= \mathcal{N}(X_{M_c} | \mu^{(0)}, \Sigma^{(0)}) \quad \text{or} \\ q_U^{(0)}(U_{M_c}) &= \prod_{t \in M_c} \mathcal{G}(U_t | \alpha_t^{(0)}, \beta_t^{(0)}) \end{aligned} \quad (3.46)$$

By taking the expectations in (3.45) and matching terms, the parameters identifying the updated distributions  $q_X^{(k+1)}$ ,  $q_U^{(k+1)}$  can be determined analytically.

As the model (3.21) is a tree, its graphical structure can be exploited to reduce message passing in the context of updating variational distributions as per (3.45). *Variational message passing* (Winn and Bishop, 2005) updates the expected sufficient statistics needed to compute (3.45) in a recursive procedure that is similar to Kalman smoothing type algorithms.

In the linear-Gaussian context, a Kalman smoother algorithm is a two stage algorithm composed of *a*) a forward running filter that alternates between a predict step and an update step to compute the (exact) filtered distributions  $\mathcal{P}(X_t | Y_{1:t}, \theta)$ , and *b*) a backward running smoother that computes the exact smoothed distributions  $\mathcal{P}(X_{t:t+1} | Y_{1:T}, \theta)$ . When updating the variational distributions  $q_X^{(k)}$  according to (3.45), the procedure is essentially equivalent to running a Kalman smoother on a model whose conditional data distributions are not time homogeneous:

$$Y_t | X_t, \mathbb{E}_{q_U^{(k)}}[U_t], \theta \sim \mathcal{N} \left( Y_t | CX_t + \mathbf{d}, \frac{1}{\mathbb{E}_{q_U^{(k)}}[U_t]} \text{diag}(\mathbf{w}) \right) \quad (3.47)$$

For fixed  $q_U^{(k)}$  this is a closed form message passing algorithm with a forward running filter and backward running smoother, both of which are completed analytically. To run the filter for times  $t \in M_c = t_c, \dots, t_c + n - 1$ , the filtered distribution

at time  $t_c$  is first required:

$$\begin{aligned}\tilde{q}_X^{(k+1)}(X_{t_c}) &= \mathcal{N}(X_{t_c} \mid \tilde{\mathbf{m}}_{t_c}, \tilde{\mathbf{S}}_{t_c}) \\ \tilde{\mathbf{m}}_{t_c} &= \tilde{\mathbf{S}}_{t_c} (C' W_{t_c}^{-1} (Y_{t_c} - \mathbf{d})) \\ \tilde{\mathbf{S}}_{t_c} &= (I + C' W_{t_c}^{-1} C)^{-1}\end{aligned}\quad (3.48)$$

where the shorthand

$$W_t = \frac{1}{\mathbb{E}_{q_U^{(k)}}[U_t]} \text{diag}(\mathbf{w}) \quad (3.49)$$

will be used from here on. The filtering procedure is then completed by alternating between the predict step:

$$\begin{aligned}\tilde{q}_X^{(k+1)}(X_{t_c+i+1}) &= \mathcal{N}(X_{t_c+i+1} \mid \tilde{\mathbf{m}}_{t_c+i+1}, \tilde{\mathbf{S}}_{t_c+i+1}) \\ \tilde{\mathbf{m}}_{t_c+i+1} &= A \tilde{\mathbf{m}}_{t_c+i} \\ \tilde{\mathbf{S}}_{t_c+i+1} &= V + A \tilde{\mathbf{S}}_{t_c+i} A'\end{aligned}\quad (3.50)$$

and the update step:

$$\begin{aligned}\tilde{q}_X^{(k+1)}(X_{t_c+i+1}) &= \mathcal{N}(X_{t_c+i+1} \mid \tilde{\mathbf{m}}_{t_c+i+1}, \tilde{\mathbf{S}}_{t_c+i+1}) \\ \tilde{\mathbf{m}}_{t_c+i+1} &= \tilde{\mathbf{S}}_{t_c+i+1} \left( C' W_{t_c+i+1}^{-1} (Y_{t_c+i+1} - d) + \tilde{\mathbf{S}}_{t_c+i+1}^{-1} \tilde{\mathbf{m}}_{t_c+i+1} \right) \\ \tilde{\mathbf{S}}_{t_c+i+1} &= \left( \tilde{\mathbf{S}}_{t_c+i+1}^{-1} + C' W_{t_c+i+1}^{-1} C \right)^{-1}\end{aligned}\quad (3.51)$$

for  $i = 1, \dots, n-1$ . The smoothing procedure (going backward through time) is completed by alternating between the backward predict and backward update steps. For clarity it should be noted that the smoothed distribution at time  $t_c + n - 1$  is equivalent to the filtered distribution at that time. The smoothing procedure can be

implemented by calculating the auxiliary parameters

$$\begin{aligned}\tilde{A}_{t_c+i} &= \left( \tilde{S}_{t_c+i-1}^{-1} + A'V^{-1}A \right)^{-1} A'V^{-1} \\ \tilde{\mathbf{b}}_{t_c+i} &= \left( \tilde{S}_{t_c+i-1}^{-1} + A'V^{-1}A \right)^{-1} \tilde{S}_{t_c+i-1}^{-1} \tilde{\mathbf{m}}_{t_c+i-1}\end{aligned}\quad (3.52)$$

for  $i = n-1, \dots, 1$  and using them to obtain both of *a*) the variational marginals:

$$\begin{aligned}q_X^{(k+1)}(X_{t_c+i-1}) &= \mathcal{N}(X_{t_c+i-1} \mid \mathbf{m}_{t_c+i-1}, S_{t_c+i-1}) \\ \mathbf{m}_{t_c+i-1} &= \tilde{A}_{t_c+i} \mathbf{m}_{t_c+i} + \tilde{\mathbf{b}}_{t_c+i} \\ S_{t_c+i-1} &= \left( \tilde{S}_{t_c+i-1}^{-1} + A'V^{-1}A \right)^{-1} + \tilde{A}_{t_c+i} S_{t_c+i} \tilde{A}'_{t_c+i}\end{aligned}\quad (3.53)$$

and *b*) the cross-time covariances:

$$\text{cov}_{q_X^{(k+1)}}(X_{t_c+i-1}, X'_{t_c+i}) = \tilde{A}_{t_c+i} S_{t_c+i}\quad (3.54)$$

where

$$\text{cov}_{q_X^{(k+1)}}(X_{t_c+i-1}, X'_{t_c+i})_{a,b} = \mathbb{E}_{q_X^{(k+1)}}[X_{t_c+i-1}^a X_{t_c+i}^b] - \mathbf{m}_{t_c+i-1,a} \mathbf{m}_{t_c+i,b}\quad (3.55)$$

is the covariance between the  $a^{\text{th}}$  dimension of  $X_{t_c+i-1}$  and the  $b^{\text{th}}$  dimension of  $X_{t_c+i}$  with respect to  $q_X^{(k+1)}$ .

Once the updated distribution  $q_X^{(k+1)}$  has been calculated, the update to  $q_U^{(k)}$  can be expressed far more concisely:

$$\begin{aligned}\alpha_{c,t}^{(k+1)} &= \frac{\mathbf{v}}{2} + \frac{p}{2} \\ \beta_{c,t}^{(k+1)} &= \frac{\mathbf{v}}{2} + \frac{1}{2} \mathbb{E}_{q_X^{(k+1)}} \left[ (Y_{c,t} - CX_{c,t} - \mathbf{d})' W^{-1} (Y_{c,t} - CX_{c,t} - \mathbf{d}) \right]\end{aligned}\quad (3.56)$$

These updates are applied iteratively until the distributions have converged, with convergence considered to have occurred when the change in parameter vector

$\phi^{(k)} = \left( \text{vec} \left( \mathbf{m}_{t_c:t_c+n-1}^{(k)} \right)', \text{vec} \left( S_{t_c:t_c+n-1}^{(k)} \right)', \alpha_{t_c:t_c+n-1}^{(k)'} , \beta_{t_c:t_c+n-1}^{(k)'} \right)'_{t_c=1}^M$  falls below

a pre-specified level:

$$\left\| \boldsymbol{\phi}^{(k+1)} - \boldsymbol{\phi}^{(k)} \right\| < \delta \quad (3.57)$$

for some small  $\delta > 0$ .

### 3.3.1.2 Updating parameter estimates

Maximising the lower bound to the composite log-likelihood with respect to  $C, \mathbf{d}, W$  can be completed analytically. Unlike when optimising the variational distributions  $q_X, q_U$ , it is convenient to represent  $X, Y$  in matrix form. It is also convenient to augment  $C$  with an extra column containing  $\mathbf{d}$ , and augment  $X_t$  with an extra dimension fixed at the value 1:

$$\begin{aligned} C^+ &= (\mathbf{d}, C) \\ X_t^+ &= (1, X_t)' \\ \Rightarrow CX_t + \mathbf{d} &= C^+ X_t^+ \end{aligned} \quad (3.58)$$

which gives updated estimates of  $C^+, W$  as:

$$C^{+(k+1)} = (Y' \mathbb{E}[\text{diag}(U) X^+]) (\mathbb{E}[X^{+'} \text{diag}(U) X^+])^{-1} \quad (3.59)$$

where expectations are with respect to variational distributions  $q^*$  optimised with respect to parameter estimates  $\hat{\boldsymbol{\theta}}^{(k)}$ , and  $Y, X^+, U$  depend on the structure of the

composite likelihood being used:

$$Y = \begin{pmatrix} Y_{1,1}^1 & \cdots & Y_{1,1}^p \\ \vdots & & \\ Y_{1,n}^1 & \cdots & Y_{1,n}^p \\ \vdots & & \\ Y_{M,1}^1 & \cdots & Y_{M,1}^p \\ \vdots & & \\ Y_{M,n}^1 & \cdots & Y_{M,n}^p \end{pmatrix} \quad X^+ = \begin{pmatrix} 1 & X_{1,1}^1 & \cdots & X_{1,1}^k \\ \vdots & \vdots & & \vdots \\ 1 & X_{1,n}^1 & \cdots & X_{1,n}^k \\ \vdots & \vdots & & \vdots \\ 1 & X_{M,1}^1 & \cdots & X_{M,1}^k \\ \vdots & \vdots & & \vdots \\ 1 & X_{M,n}^1 & \cdots & X_{M,n}^k \end{pmatrix} \quad U = \begin{pmatrix} U_{1,1} \\ \vdots \\ U_{1,n} \\ \vdots \\ U_{M,1} \\ \vdots \\ U_{M,n} \end{pmatrix} \quad (3.60)$$

The updated estimate of  $W$  is:

$$\hat{W}^{(k+1)} = \text{diag} \left( \frac{1}{Mn} \mathbb{E} \left[ \left( Y - X^+ \hat{C}^{+(k+1)'} \right)' \text{diag}(U) \left( Y - X^+ \hat{C}^{+(k+1)'} \right) \right] \right) \quad (3.61)$$

where the  $\text{diag}$  operator acting on a square matrix  $Z$  returns a column vector containing the diagonal elements of  $Z$ , and expectations are again with respect to variational distributions optimised with respect to parameter estimates  $\hat{\theta}^{(k)}$ .

Estimates of correlation parameters  $\rho_i$  and the Student-t degrees of freedom parameter  $\nu$  cannot be updated analytically. As each dimension of the latent process  $X_{1:T}$  is independent in the prior distribution  $\mathcal{P}(X_{1:T} | \theta)$ , each estimate  $\hat{\rho}_i^{(k)}$  is updated independently as, recalling the form of  $|\Lambda_i|$  given in equation (3.33), the (numerical) maximiser:

$$\hat{\rho}_i^{(k+1)} = \arg \max_{\rho_i \in (-1,1)} \left( M(n-1) \log(1 - \rho_i^2) + \sum_{c=1}^M \mathbb{E} \left[ X_{c:c+n-1}^i{}' \Lambda_i X_{c:c+n-1}^i \right] \right) \quad (3.62)$$

The updated estimate of  $\nu$  is the (numerical) maximiser:

$$\begin{aligned} \hat{\nu}^{(k+1)} = \arg \max_{\nu > 0} & \left( \frac{\nu}{2} \log \left( \frac{\nu}{2} \right) - \log \left( \Gamma \left( \frac{\nu}{2} \right) \right) \right) \\ & + \sum_{c=1}^M \sum_{t=c}^{c+n-1} \left( \frac{\nu}{2} - 1 \right) \mathbb{E}[\log(U_{c,t})] - \frac{\nu}{2} \mathbb{E}[U_{c,t}] \end{aligned} \quad (3.63)$$

---

**Remark** As has been found elsewhere, see Liu and Rubin (1995); Fernández and Steel (1999) for example, estimation of the degrees of freedom parameter  $\nu$  has been challenging in the current thesis. Though the update equation (3.63) is correct, it was found in practice that  $\hat{\nu}^{(k+1)} > \hat{\nu}^{(k)}$  at every update iteration, without convergence. Such a feature clearly forces the estimation algorithm to become unstable and non-viable. As an undesirable, but unavoidable, pragmatic solution to this problem, estimates of  $\nu$  were randomly initialised but not updated in subsequent update iterations, i.e.  $\hat{\nu}^{(k)} = \hat{\nu}^{(0)}$  for all  $k$ . It should be noted that issues along these lines were not mentioned in the discussions of Svensén and Bishop (2005).

---

Parameter estimates should be updated iteratively until convergence, which can be declared when the change in the parameter vector  $\hat{\theta}^{(k)} = (\text{vec}(\hat{C}^{+(k)})', \hat{W}^{(k)'}, \hat{\rho}^{(k)'})'$  is less than some pre-specified level:

$$\left\| \hat{\theta}^{(k+1)} - \hat{\theta}^{(k)} \right\| < \delta \quad (3.64)$$

for some small  $\delta > 0$ . The variational EM algorithm is summarised in algorithm 3.1.

### 3.4 Stochastic EM

To provide a benchmark against which the bias introduced by using variational approximations for parameter estimation can be compared, estimates are also made using the *stochastic EM* algorithm. This algorithm makes an unbiased approximation to the log composite likelihood at each iteration by using MCMC to approximate taking the posterior expectation. Gibbs sampling is used to draw samples

**Algorithm 3.1** Variational EM for maximum composite likelihood estimators

1. Initialise parameter estimates  $\hat{\theta}^{(0)}$ .
2. Repeat until convergence of parameter estimates  $\hat{\theta}^{(k)}$ :
  - (a) Optimise lower bound by finding  $q_{X,U}^*(X_{M_c}, U_{M_c}) = q_X^*(X_{M_c})q_U^*(U_{M_c})$  for each  $M_c$  in the composite likelihood:
    - i. Initialise  $q_U(U_{M_c}) = \prod_{t=1}^n \mathcal{G}(U_{c,t} | \alpha_{c,t}^{(0)}, \beta_{c,t}^{(0)})$ .
    - ii. Repeat until convergence of  $\mu_c^{(k)}, \Sigma_c^{(k)}, \alpha_{c,t}^{(k)}, \beta_{c,t}^{(k)}$ :
      - A. Update  $q_X$  according to the filtering / smoothing procedure for each component described in equations (3.48) and (3.50) - (3.53).
      - B. Update  $q_U$  according to (3.56).
  - (b) Update parameter estimate  $\hat{\theta}^{(k)}$  according to (3.59), (3.61), (3.62):

$$\begin{aligned}
 C^{+(k+1)} &= (Y' \mathbb{E}[\text{diag}(U)X^+])(\mathbb{E}[X^{+'} \text{diag}(U)X^+])^{-1} \\
 \hat{W}^{(k+1)} &= \text{diag} \left( \frac{1}{Mn} \mathbb{E} \left[ \left( Y - X^+ \hat{C}^{+(k+1)'} \right)' \text{diag}(U) \left( Y - X^+ \hat{C}^{+(k+1)'} \right) \right] \right) \\
 \hat{\rho}_i^{(k+1)} &= \arg \max_{\rho_i \in (-1,1)} \left( M(n-1) \log(1 - \rho_i^2) + \sum_{c=1}^M \mathbb{E} \left[ X_{c:c+n-1}^i {}' \Lambda_i X_{c:c+n-1}^i \right] \right)
 \end{aligned} \tag{3.65}$$

where  $\hat{C}^+$  is the estimate of  $C^+ = (\mathbf{d}, C)$ , and  $X^+$  is the augmented latent  $X$  variable arranged according to (3.60).

approximately from  $\mathcal{P}(X_{M_c}, U_{M_c} | Y_{M_c}, \hat{\theta}^{(k)})$ :

$$\begin{aligned}
 X_{M_c}^{(k+1)} &\sim \mathcal{P}(X_{M_c} | Y_{M_c}, U_{M_c}^{(k)}, \hat{\theta}^{(j)}) \\
 U_{M_c}^{(k+1)} &\sim \mathcal{P}(U_{M_c} | Y_{M_c}, X_{M_c}^{(k+1)}, \hat{\theta}^{(j)})
 \end{aligned} \tag{3.66}$$

with posterior conditional distributions derived almost identically to the  $q$  distribution updates in sec 3.3.1.1. Indeed they are equivalent to those updates only with sampled values taking the place of the expectations in (3.48) and (3.50) - (3.53) and (3.56). As mentioned in the introduction to the current chapter, and as discussed in Turner and Sahani (2008), the equivalent formulation of the variational update



effectively means that the uncertainty in  $U_{1:T}$  is not propagated to  $X_{1:T}$  during variational update iterations, and vice versa.

When the  $X_{M_c}^{(k)}$  samples are being drawn according to (3.66), a procedure similar to that for updating the lower bound in sec 3.3.1.1 can be constructed. This procedure is essentially a particle filter with only one sample being drawn and no need for any resampling, as the target distributions can be drawn from directly. In the following exposition, the shorthand  $W_t$  is redefined to become:

$$W_t^{(k)} = \frac{1}{U_t^{(k)}} \text{diag}(\mathbf{w}) \quad (3.67)$$

giving the filtered distribution at time  $t_c$ :

$$\begin{aligned} \mathcal{P}\left(\tilde{X}_{t_c}^{(k+1)} \mid Y_{t_c}, U_{t_c}^{(k)}, \hat{\theta}^{(j)}\right) &= \mathcal{N}\left(\tilde{X}_{t_c}^{(k+1)} \mid \tilde{\mathbf{m}}_{t_c}, \tilde{S}_{t_c}\right) \\ \tilde{\mathbf{m}}_{t_c} &= \tilde{S}_{t_c} (C' \cdot W_{t_c}^{(k)-1} (Y_{t_c} - \mathbf{d})) \\ \tilde{S}_{t_c} &= (I + C' \cdot W_{t_c}^{(k)-1} \cdot C)^{-1} \end{aligned} \quad (3.68)$$

the filtered distribution at time  $t_c + i$ :

$$\begin{aligned} \mathcal{P}\left(\tilde{X}_{t_c+i}^{(k+1)} \mid \tilde{X}_{t_c+i-1}^{(k+1)}, Y_{t_c+i}, U_{t_c+i}^{(k)}, \hat{\theta}^{(j)}\right) &= \mathcal{N}\left(\tilde{X}_{t_c+i}^{(k+1)} \mid \tilde{\mathbf{m}}_{t_c+i}, \tilde{S}_{t_c+i}\right) \\ \tilde{\mathbf{m}}_{t_c+i} &= \tilde{S}_{t_c+i} \left( C' \cdot W_{t_c+i}^{(k)-1} (Y_{t_c+i} - \mathbf{d}) \right. \\ &\quad \left. + V^{-1} A \tilde{X}_{t_c+i-1}^{(k+1)} \right) \\ \tilde{S}_{t_c+i} &= \left( V^{-1} + C' \cdot W_{t_c+i}^{(k)-1} \cdot C \right)^{-1} \end{aligned} \quad (3.69)$$

for  $i = 1, \dots, n-1$ , and the smoothed distribution at time  $t_c + i - 1$ :

$$\begin{aligned} \mathcal{P}\left(X_{t_c+i-1}^{(k+1)} \mid Y_{t_c:t_c+i-1}, U_{t_c+i-1}^{(k)}, X_{t_c+i}^{(k+1)}, \hat{\theta}^{(j)}\right) &= \mathcal{N}\left(X_{t_c+i-1}^{(k+1)} \mid \tilde{A}_{t_c+i} X_{t_c+i}^{(k+1)} + \tilde{\mathbf{b}}_{t_c+i}, S_{t_c+i-1}\right) \\ \tilde{A}_{t_c+i} &= S_{t_c+i-1} \cdot A' \cdot V^{-1} \\ \tilde{\mathbf{b}}_{t_c+i} &= S_{t_c+i-1} \cdot \tilde{S}_{t_c+i-1}^{-1} \tilde{\mathbf{m}}_{t_c+i-1} \\ S_{t_c+i-1} &= \left( \tilde{S}_{t_c+i-1}^{-1} + A' \cdot V^{-1} \cdot A \right)^{-1} \end{aligned} \quad (3.70)$$

for  $i = n - 1, \dots, 1$ , noting that the smoothed distribution  $X_{t_c+n-1}^{(k+1)}$  at time  $t_c + n - 1$  equals the filtered distribution  $\tilde{X}_{t_c+n-1}^{(k+1)}$ . The  $U_{M_c}^{(k+1)}$  samples are drawn conditioned on  $X_{M_c}^{(k+1)}$ :

$$\begin{aligned} \mathcal{P}\left(U_{M_c}^{(k+1)} \mid Y_{M_c}, X_{M_c}^{(k+1)}, \hat{\theta}^{(j)}\right) &= \prod_{t=1}^n \mathcal{G}\left(U_{c,t}^{(k+1)} \mid \alpha_{c,t}^{(k+1)}, \beta_{c,t}^{(k+1)}\right) \\ \alpha_{c,t}^{(k+1)} &= \frac{\nu}{2} + \frac{p}{2} \\ \beta_{c,t}^{(k+1)} &= \frac{\nu}{2} + \frac{1}{2} \left(Y_{c,t} - CX_{c,t}^{(k+1)} - \mathbf{d}\right)' W^{-1} \left(Y_{c,t} - CX_{c,t}^{(k+1)} - \mathbf{d}\right) \end{aligned} \quad (3.71)$$

All moments required for parameter estimate updates are estimated using the sample moments from the Gibbs sampling procedure (3.66). After the burn in samples have been discarded, samples drawn using (3.68) - (3.71) are used to make Monte Carlo estimates of  $\mathbb{E}_{X,U|Y} \left[ \log \mathcal{P}\left(X_{M_c}, U_{M_c}, Y_{M_c} \mid \hat{\theta}^{(j)}\right) \right]$ :

$$\mathbb{E}_{X,U|Y} \left[ \log \mathcal{P}\left(X_{M_c}, U_{M_c}, Y_{M_c} \mid \hat{\theta}^{(j)}\right) \right] \approx \frac{1}{N} \sum_{i=1}^N \log \mathcal{P}\left(X_{M_c}^{(i)}, U_{M_c}^{(i)}, Y_{M_c} \mid \hat{\theta}^{(j)}\right) \quad (3.72)$$

Apart from this alternative form of approximate expectation taking, the stochastic EM algorithm is identical to algorithm 3.1. The sample moments replace the variational expectations in (3.59) - (3.63) used for VEM parameter updates to form essentially identical update equation equations. The stochastic EM algorithm is summarised in algorithm 3.2.

### 3.4.1 Estimating smoothed distributions

After parameters estimates  $\hat{\theta}$  have been made by either variational EM via algorithm 3.1 or stochastic EM via algorithm 3.2, they can be used to estimate the smoothed distributions  $\mathcal{P}(X_{1:T}, U_{1:T} \mid Y_{1:T}, \hat{\theta})$ . The method is identical to that used to obtain the MCMC samples for the expectation step of stochastic EM. Instead of using the samples to estimate the expected sufficient statistics needed to maximise log composite likelihood, however, the samples are used as a direct approximation to the smoothed posterior distribution of the latent variables.

**Algorithm 3.2** Stochastic EM for maximum composite likelihood estimators

1. Initialise parameter estimates  $\hat{\theta}^{(0)}$ .
2. Repeat until convergence of parameter estimates  $\hat{\theta}^{(k)}$ :
  - (a) Approximate the expected log composite likelihood  $\sum_{c=1}^M \mathbb{E}_{X,U|Y} \log \left( \mathcal{P} \left( X_{M_c}, U_{M_c}, Y_{M_c} \mid \hat{\theta}^{(k)} \right) \right)$  using Gibbs sampling:
    - i. Initialise  $U_{M_c}^{(0)}$ ,  $c = 1, \dots, M$ .
    - ii. Repeat  $N$  times:
      - A. Sample  $X_{M_c}^{(k+1)} \mid Y_{M_c}, U_{M_c}^{(k)}$  according to (3.68) - (3.70).
      - B. Sample  $U_{M_c}^{(k+1)} \mid Y_{M_c}, X_{M_c}^{(k+1)}$  according to (3.71).
    - iii. Discard the first  $N_{\text{burn-in}}$  samples.
  - (b) Update parameter estimate  $\hat{\theta}^{(k)}$  according to:

$$\begin{aligned}
 \hat{C}^{+(k+1)} &= \left( Y' \overline{\text{diag}(U) X^+} \right) \left( \overline{X^{+'} \text{diag}(U) X^+} \right)^{-1} \\
 \hat{W}^{(k+1)} &= \text{diag} \left( \frac{1}{Mn} \overline{(Y - X^+ \hat{C}^{+(k+1)'})' \text{diag}(U) (Y - X^+ \hat{C}^{+(k+1)'})} \right) \\
 \hat{\rho}_i^{(k+1)} &= \arg \max_{\rho_i \in (-1, 1)} \left( M(n-1) \log(1 - \rho_i^2) + \sum_{c=1}^M \overline{X_{c:c+n-1}^i \Lambda_i X_{c:c+n-1}^i} \right)
 \end{aligned} \tag{3.73}$$

where over-lines indicate sample moments,  $\hat{C}^+$  is the estimate of  $C^+ = (\mathbf{d}, C)$ , and  $X^+$  is the augmented latent  $X$  variable arranged according to (3.60).

## 3.5 Prediction

Performing  $n$ -step ahead prediction is completed with parameter estimates  $\hat{\theta}$  by using a particle filter algorithm similar to that used in the expectation step of stochastic EM, described in sec 3.4. Gibbs sampling is used to obtain samples  $\{X_t^{(k)}, U_t^{(k)}\}_{k=1}^N$ , drawn approximately from the filtered distributions  $\mathcal{P}(X_t, U_t \mid X_{t-1}, Y_t, \hat{\theta})$ . Predictions are made via auxiliary particles  $\tilde{X}_t$  that are generated at each time  $t$ . These auxiliary particles are propagated forward  $n$  times according to the dynamics of the

model:

$$\begin{aligned} \tilde{X}_t^{(k)} &= X_t^{(k)} \\ \tilde{X}_{t+i}^{(k)} | \tilde{X}_{t+i-1}^{(k)}, Y_t, \hat{\theta} &\sim \mathcal{N}\left(\tilde{X}_{t+i}^{(k)} | \hat{A}\tilde{X}_{t+i-1}^{(k)}, \hat{V}\right), \quad i = 1, \dots, n \end{aligned} \quad (3.74)$$

Each auxiliary particle  $\tilde{X}_{t+n}^{(k)}$  makes a prediction  $\hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta} = \mathbb{E}[Y_{t+n} | \tilde{X}_{t+n}^{(k)}, U_{t+n}^{(k)}, \hat{\theta}]$  for  $Y_{t+n}$ . The collection  $\{\hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta}\}_{k=1}^N$  of particle predictions together make up the (distributional) prediction  $\hat{Y}_{t+n} | Y_{1:t}, \hat{\theta}$ .

The Gibbs sampling procedure used to obtain the samples  $\{X_1^{(k)}, U_1^{(k)}\}_{k=1}^N$  for the first time step differs from other times, as the samples are not conditioned on any previous particles. Conditional distributions are used to alternately draw samples  $X_1^{(k+1)} | Y_1, U_1^{(k)}, \hat{\theta}$  and  $U_1^{(k+1)} | Y_1, X_1^{(k+1)}, \hat{\theta}$  as per equations (3.68) and (3.71):

$$\begin{aligned} \mathcal{P}\left(X_1^{(k+1)} | Y_1, U_1^{(k)}, \hat{\theta}^{(j)}\right) &= \mathcal{N}\left(X_1^{(k+1)} | \mathbf{m}_1, S_1\right) \\ \mathbf{m}_1 &= S_1(C' \cdot W_1^{(k)-1}(Y_1 - \mathbf{d})) \\ S_1 &= (I + C' \cdot W_1^{(k)-1} \cdot C)^{-1} \\ \mathcal{P}\left(U_1^{(k+1)} | Y_1, X_1^{(k+1)}, \hat{\theta}^{(j)}\right) &= \mathcal{G}\left(U_1^{(k+1)} | \alpha_1^{(k+1)}, \beta_1^{(k+1)}\right) \\ \alpha_1^{(k+1)} &= \frac{\hat{v}}{2} + \frac{p}{2} \\ \beta_1^{(k+1)} &= \frac{\hat{v}}{2} + \frac{1}{2} \left(Y_1 - CX_1^{(k+1)} - \mathbf{d}\right)' W^{-1} \left(Y_1 - CX_1^{(k+1)} - \mathbf{d}\right) \end{aligned} \quad (3.75)$$

to obtain  $\tilde{N}$  samples. After the burn-in samples are discarded, the  $N$  remaining samples constitute the filtered particle set at  $t = 1$ .

The prediction procedure then goes through the iterative process of *a*) predicting the observation  $Y_{t+n}$  and *b*) propagating the particles forward one time step and conditioning them on  $Y_{t+1}$ . As described above, predictions are made by first propagating each filtered particle  $X_t^{(k)} | X_{t-1}^{(k)}, Y_t, \hat{\theta}$  forward  $n$  times, with each propagation

following the dynamics of the fitted model:

$$\begin{aligned} \tilde{X}_t^{(k)} &= X_t^{(k)} \\ \tilde{X}_{t+i}^{(k)} | \tilde{X}_{t+i-1}^{(k)}, Y_t, \hat{\theta} &\sim \mathcal{N}\left(\tilde{X}_{t+i}^{(k)} | \hat{A}\tilde{X}_{t+i-1}^{(k)}, \hat{V}\right), \quad i = 1, \dots, n \end{aligned} \quad (3.76)$$

After all auxiliary particles  $\tilde{X}_{t+n}^{(k)}$  have been drawn, particle predictions are made as the conditional means  $\mathbb{E}[Y_{t+n} | \tilde{X}_{t+n}^{(k)}, U_{t+n}^{(k)}, \hat{\theta}]$ :

$$\hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta} = \hat{C}\tilde{X}_{t+n}^{(k)} + \hat{\mathbf{d}} \quad (3.77)$$

noting that these quantities are independent of  $U_{t+n}^{(k)}$ . The distributional prediction  $\hat{Y}_{t+n} | Y_{1:t}, \hat{\theta}$  is the collection of these particle predictions:

$$\hat{Y}_{t+n} = \{\hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta}\}_{k=1}^N \quad (3.78)$$

After each prediction has been made, the auxiliary particles drawn as per (3.76) are discarded. The filtered particles  $X_t^{(k)}$  are then propagated forward once and, particle by particle, conditioned via Gibbs sampling on the next observation. This is achieved via sampling another set of auxiliary particles  $\{\tilde{X}_{t+1}^{(k,l)}, U_{t+1}^{(k,l)}\}_{l=1}^{N_{\text{burn-in}}+1}$  for each particle  $X_t^{(k)}$ :

$$\begin{aligned} \mathcal{P}\left(\tilde{X}_{t+1}^{(k,l+1)} | X_t^{(k)}, Y_{t+1}, U_{t+1}^{(k,l)}, \hat{\theta}\right) &= \mathcal{N}\left(\tilde{X}_{t+1}^{(k,l+1)} | \mathbf{m}_{t+1}, S_{t+1}\right) \\ \mathbf{m}_{t+1} &= S_{t+1} \left( \hat{C}' \cdot W_{t+1}^{(k,l)-1} (Y_{t+1} - \hat{\mathbf{d}}) \right. \\ &\quad \left. + \hat{V}^{-1} \hat{A} X_t^{(k)} \right) \\ S_{t+1} &= \left( \hat{V}^{-1} + \hat{C}' \cdot W_{t+1}^{(k,l)-1} \cdot \hat{C} \right)^{-1} \\ \mathcal{P}\left(U_{t+1}^{(k,l+1)} | Y_{t+1}, \tilde{X}_{t+1}^{(k,l+1)}, \hat{\theta}\right) &= \mathcal{G}\left(U_{t+1}^{(k,l+1)} | \alpha_{t+1}^{(k,l+1)}, \beta_{t+1}^{(k,l+1)}\right) \\ \alpha_{t+1}^{(k,l+1)} &= \frac{\hat{V}}{2} + \frac{p}{2} \\ \beta_{t+1}^{(k,l+1)} &= \frac{\hat{V}}{2} + \frac{1}{2} \left( Y_{t+1} - \bar{Y}_{t+1}^{(k,l+1)} \right)' \hat{W}^{-1} \left( Y_{t+1} - \bar{Y}_{t+1}^{(k,l+1)} \right) \\ \bar{Y}_{t+1}^{(k,l+1)} &= \hat{C}\tilde{X}_{t+1}^{(k,l+1)} + \hat{\mathbf{d}} \end{aligned} \quad (3.79)$$

with  $\tilde{U}_{t+1}^{(k,0)}$  randomly initialised. Only  $N_{\text{burn-in}} + 1$  samples are drawn in this fashion, and the last one is kept as the filtered particle  $X_{t+1}^{(k)}$ :

$$X_{t+1}^{(k)} = \tilde{X}^{(k, N_{\text{burn-in}}+1)} \quad (3.80)$$

As the predictions  $\hat{Y}_{t+n} | Y_{1:t}, \hat{\theta}$  do not depend on  $U_t^{(k)}$ , the particles  $U_t^{(k)}$  are not kept. The prediction algorithm is summarised in algorithm 3.3.

**Algorithm 3.3**  $n$ -step ahead prediction

1. Obtain  $N$  particles  $X_1^{(k)} | Y_1, \hat{\theta}$  as per (3.75).
2. For each  $t \in 1, \dots, T - n$ :
  - (a) Propagate the auxiliary particles  $\tilde{X}_t^{(k)}$  forward  $n$  times as per (3.76).
  - (b) Predict  $Y_{t+n}$  as the distribution of conditional means  $\mathbb{E}[Y_{t+n} | \tilde{X}_{t+n}^{(k)}, U_{t+n}^{(k)}, \hat{\theta}]$

$$\begin{aligned} \hat{Y}_{t+n} | Y_{1:t}, \hat{\theta} &= \{\hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta}\}_{k=1}^N \\ \hat{Y}_{t+n}^{(k)} | Y_{1:t}, \hat{\theta} &= \hat{C}X_{t+n}^{(k)} + \hat{\mathbf{d}} \end{aligned} \quad (3.81)$$

- (c) Propagate the particles  $X_t^{(k)}$  forward one time step and condition on  $Y_{t+1}$  by generating the auxiliary particle set  $\{\tilde{X}_{t+1}^{(k,l)}\}_{l=1}^{N_{\text{burn-in}}+1}$  as per (3.79) and (3.80), taking  $X_{t+1}^{(k)} = \tilde{X}_{t+1}^{(k, N_{\text{burn-in}}+1)}$ .

## 3.6 Synthetic data

Each of the 100 samples of synthetic data drawn will contain  $T = 500$  observations, each with  $p = 25$  dimensions. They will be drawn from the generative model (3.21), using a latent Gaussian process with  $k = 10$  dimensions. The Gaussian process correlation parameters  $\rho_i$  are each drawn independently from  $\mathcal{U}(-1, 1)$ , the degrees of freedom parameter is drawn from an exponential distribution  $\mathcal{E}(5)$ , and the individual elements of the emission parameters  $C, \mathbf{d}$  are each drawn i.i.d. from standard Gaussians  $\mathcal{N}(0, 1)$ . After parameters are drawn they are kept constant across all draws of synthetic data. The composite likelihood structures used in experiments

have component sizes 2, 10, 50, 500.

### 3.7 Experiments

Synthetic data is used to test the results of fitting parameters via VEM (algorithm 3.1) and SEM (algorithm 3.2) for a variety of composite likelihood structures. Experiments investigating the effects on bias and variance are made, as well as on the quality of smoothing estimates and prediction. From here on, to ease descriptions all estimators that approximately maximise composite likelihoods by making variational approximations are referred to as VEM estimators. Similarly, estimators that use stochastic approximations to maximise composite likelihoods are referred to as SEM estimators.

**Disjoint components:** The first experiment to be conducted examines the effect of including all possible component in a composite likelihood, as compared to only disjoint components. The experiment is conducted for components of size  $n = 2$  on the bias of estimators. Estimators that maximise

$$L_C^{2,\cup}(\theta) = \prod_{t_c=1}^{499} \mathcal{P}(Y_{t_c:t_c+1} \mid \theta) \quad (3.82)$$

and

$$L_C^{2,\dot{\cup}}(\theta) = \prod_{t_c=1}^{250} \mathcal{P}(Y_{2t_c-1:2t_c} \mid \theta) \quad (3.83)$$

are computed and compared.

Other effects are investigated on disjoint composite structures only. The composite structures being compared have disjoint components of sizes  $n \in \{2, 10, 50, 500\}$ . For clarity it should be noted that the composite likelihood estimator with component size  $n = T = 500$  is equivalent to the standard maximum likelihood estimator. The other experiments conducted are on:

**Bias:** To examine the effect of variational approximations and component size on the bias of estimators, the model (3.21) will be fit via both VEM and SEM

for all component sizes  $n$  to one common set of synthetic data. Each set of fitted parameters will be compared, along with the time taken to fit each of them. The bias from using the particular component structure is separated from the effect of making a variational approximation by observing the difference between the two estimators. The behaviour of estimates as component size changes will be observed. The values

$$\begin{aligned}\delta_{m,n}^{\text{VEM}} &= \|\hat{\theta}_m^{\text{VEM}} - \hat{\theta}_n^{\text{VEM}}\|, & m, n \in 2, 10, 50, 500 \\ \delta_{m,n}^{\text{SEM}} &= \|\hat{\theta}_m^{\text{SEM}} - \hat{\theta}_n^{\text{SEM}}\|, & m, n \in 2, 10, 50, 500 \\ \delta_{m,n}^{\text{VEM/SEM}} &= \|\hat{\theta}_m^{\text{SEM}} - \hat{\theta}_n^{\text{VEM}}\|, & m, n \in 2, 10, 50, 500\end{aligned}\quad (3.84)$$

where superscripts indicate the approximate EM method used, and subscripts indicate the component size of the estimator, will be recorded and analysed.

**Variance:** The effect of choosing different composite structures on the variance of VEM parameter estimates will also be examined. The model will be fitted to multiple, independent draws of synthetic data, all drawn from the same distribution, via VEM for each component size  $n$ . There will be 100 i.i.d. samples of synthetic data. The variances of estimators approximated with VEM for all component sizes  $n$  are observed. The values

$$\text{var}(\hat{\theta}_{n,i}^{\text{VEM}}), \quad n \in 2, 10, 50, 500, \quad i \in 1, \dots, |\theta| \quad (3.85)$$

where the second subscript indicates the element of  $\hat{\theta}$  whose variance is being recorded, will be compared to each other.

**Smoothing:** Smoothing is undertaken for a set of estimated parameters by taking MCMC samples to approximate the latent posterior as per sec 3.4.1. All parameter estimates made in the bias experiments above are used to generate approximate smoothed distributions. Estimates of both in-sample and out-of-sample smoothed distributions are made. In-sample smoothing estimates the smoothed distribution of the latent variables for the dataset to which param-



eters were fitted. Out-of-sample smoothing estimates the smoothed distribution for one dataset other than the one to which estimated parameters were fitted, but generated using the same parameters. Two different root mean squared error (RMSE) loss functions are used to measure performance. The first loss function is:

$$L_n^{(1),j} = \sqrt{\frac{1}{T(k+1)} \sum_{t=1}^T \left( \sum_{k'=1}^k \left( X_{t,k'} - \bar{X}_{t,k'}^{n,j} \right)^2 + \left( U_t - \bar{U}_t^{n,j} \right)^2 \right)}$$

$$n \in \{2, 10, 50, 500\}, \quad j \in \{\text{VEM}, \text{SEM}\} \quad (3.86)$$

where  $\bar{X}_{t,k'}^{n,j}$  and  $\bar{U}_t^{n,j}$  are the means of the smoothing estimates for each dimension of the latent space at each time:

$$\bar{X}_{t,k'}^{n,j} = \frac{1}{N} \sum_{i=1}^N \hat{X}_{t,k'}^{(i),n,j}$$

$$\bar{U}_t^{n,j} = \frac{1}{N} \sum_{i=1}^N \hat{U}_t^{(i),n,j} \quad (3.87)$$

This loss function measures how close the mean of each smoothing estimate is to the true value of its corresponding latent variable. The second loss function is:

$$L_n^{(2),j} = \sqrt{\frac{1}{NT(k+1)} \sum_{i=1}^N \sum_{t=1}^T \left( \sum_{k'=1}^k \left( X_{t,k'} - \hat{X}_{t,k'}^{(i),n,j} \right)^2 + \left( U_t - \hat{U}_t^{(i),n,j} \right)^2 \right)}$$

$$n \in \{2, 10, 50, 500\}, \quad j \in \{\text{VEM}, \text{SEM}\} \quad (3.88)$$

This loss function measures how close the individuals particles in the smoothing estimates are to the true value of their corresponding latent variables. The two loss functions are related:

$$L_n^{(2),j} = \sqrt{\left( L_n^{(1),j} \right)^2 + \frac{1}{T(k+1)} \sum_{t=1}^T \left( \sum_{k'=1}^k \text{var}(\hat{X}_{t,k'}^{n,j}) + \text{var}(\hat{U}_t^{n,j}) \right)} \quad (3.89)$$

where  $\text{var}(\hat{X}_{t,k}^{n,j})$  and  $\text{var}(\hat{U}_t^{n,j})$  are the variances of the smoothing estimates for each dimension of the latent variable at each time. This illustrates how  $L_n^{(2),j}$  measures the spread of particles in addition to the accuracy of the particle means at each time. It also implies that  $L_n^{(2),j} \geq L_n^{(1),j} \forall n, j$ .

Prediction: One step ahead prediction as per (3.10), i.e. with  $n = 1$ , is undertaken.

Similarly to the smoothing experiment, two different RMSE loss functions are used to measure performance. The first loss function is:

$$L_n^{(1),j} = \sqrt{\frac{1}{(T-1)p} \sum_{t=2}^T \sum_{p'=1}^p \left( Y_{t,p'} - \bar{Y}_{t,p'}^{n,j} \right)^2}$$

$$n \in 2, 10, 50, 500, \quad j \in \{\text{VEM}, \text{SEM}\} \quad (3.90)$$

where  $\bar{Y}_{t,p'}^{n,j}$  is dimension  $p'$  of the mean prediction of  $Y_t$  made using  $\hat{\theta}_n^j$ :

$$\bar{Y}_{t,p'}^{n,j} = \frac{1}{N} \sum_{i=1}^N \hat{Y}_{t,p'}^{(i),n,j} \quad (3.91)$$

This loss function measures how close the mean particle predictions for each time are to the data. The second loss function is:

$$L_n^{(2),j} = \sqrt{\frac{1}{N(T-1)p} \sum_{i=1}^N \sum_{t=2}^T \sum_{p'=1}^p \left( Y_{t,p'} - \hat{Y}_{t,p'}^{(i),n,j} \right)^2}$$

$$n \in 2, 10, 50, 500, \quad j \in \{\text{VEM}, \text{SEM}\} \quad (3.92)$$

where  $\hat{Y}_{t,p'}^{(i),n,j}$  is dimension  $p'$  of the prediction of  $Y_t$  made using  $\hat{\theta}_n^j$  by particle  $i$  of  $N$ . This loss function measures how close individual particle predictions for each time are to the data. The two loss functions are related:

$$L_n^{(2),j} = \sqrt{\left( L_n^{(1),j} \right)^2 + \frac{1}{(T-1)p} \sum_{t=2}^T \sum_{p'=1}^p \text{var}(\hat{Y}_{t,p'}^{n,j})} \quad (3.93)$$

where  $\text{var}(\hat{Y}_{t,p'}^{n,j})$  is the variance of the distributional prediction  $\hat{Y}_{t,p'}^{n,j}$ . This im-

**Table 3.1:** Running times for approximate parameter estimation algorithm for estimators using VEM and SEM, and with various composite likelihood structures. Times are in seconds, and are rounded to the nearest second.

| $\hat{\theta}$ | Non-overlapping blocks |     |     |     |       |       |       |       | Overlapping blocks |        |
|----------------|------------------------|-----|-----|-----|-------|-------|-------|-------|--------------------|--------|
|                | VEM                    |     |     |     | SEM   |       |       |       | VEM                | SEM    |
| $n$            | 2                      | 10  | 50  | 500 | 2     | 10    | 50    | 500   | 2                  | 2      |
| Seconds        | 116                    | 128 | 131 | 133 | 9,496 | 9,383 | 9,467 | 9,574 | 216                | 19,219 |

plies that  $L_n^{(2),j} \geq L_n^{(1),j} \forall n, j$ .

Predictions are made using the procedure described in algorithm 3.3. Both in-sample prediction and out-of-sample predictions are made for each set of parameter estimates. The same dataset used for the out-of-sample smoothing test is used for the out-of-sample prediction. For reference, predictions using the true parameters that generated the synthetic data are also made for both the in-sample and out-of-sample data, and both loss functions are computed for each of these predictions also.

All experiments are written and performed in MATLAB.

### 3.8 Results

The running times for performing parameter estimation using variational and stochastic approximations to expected sufficient statistics are shown in table 3.1. The time taken to calculate estimators with all possible components of size  $n = 2$  scales with the total number  $M = n(T - n + 1)$  of processed observations.

The difference in estimating parameters using a composite likelihood  $L_C^{n,\cup}(\theta | Y)$  constructed from all possible components of size  $n$ , and a composite likelihood  $L_C^{n,\dot{\cup}}(\theta | Y)$  is investigated for the component size  $n = 2$ . Parameters are fitted to the same dataset using each composite structure, making approximations with both VEM and SEM. The differences in these estimators are shown in table 3.2.

As can be seen from these tables, the composite structure (either all possible components or only disjoint components) has a negligible effect on the estimates. The method of approximating expected sufficient statistics is far more significant.

This result is taken to support the use of disjoint components in further experiments.

**Table 3.2:** Differences in maximum composite likelihood parameter estimates made using all possible components of size 2 ( $\cup$ ) and only disjoint components of size 2 ( $\dot{\cup}$ ), and with expectations approximated via VEM and SEM. Differences in norm are in left table, maximum per-element differences are in right table. The full parameter vector  $\theta$  contains 310 elements.

| (a) $\ \hat{\theta}_2^i - \hat{\theta}_2^j\ $ |            | (b) $\max(\text{abs}(\hat{\theta}_{2,k}^i - \hat{\theta}_{2,k}^j))$ |                  |
|---|------------|---|------------------|
|   | VEM $\cup$ | SEM $\cup$  |                  |
| VEM $\dot{\cup}$                              | 1.48       | 8.85  | VEM $\dot{\cup}$ |
| SEM $\dot{\cup}$                              | 8.98       | 2.29  | SEM $\dot{\cup}$ |

### 3.8.1 Bias experiment

The differences in parameter estimates made using disjoint components of sizes  $n \in 2, 10, 50, 500$  and via VEM and SEM are shown in table 3.3. The results in table 3.3 show all pairwise differences in estimated parameter vectors. Each quadrant of table 3.3 highlights different features of the bias experiment. The top left and bottom right quadrants record the differences between estimators within each of the VEM and SEM classes respectively as component size changes. The bottom left and top right quadrants record the differences between estimators across the two classes. As  $\hat{\theta}_{500}^{\text{SEM}}$  is equivalent to the maximum likelihood estimator for the dataset, it can be considered to be the ‘gold standard’ among the estimators being compared. The differences between  $\hat{\theta}_{500}^{\text{SEM}}$  and all other estimators (rightmost column of table 3.3) are therefore particularly relevant.

Within both the VEM and SEM classes of estimators, the norms of the differences go to 0 as the component sizes  $n$  approaches  $T$ . For example,  $\delta_{2,10}^{\text{VEM}} > \delta_{10,50}^{\text{VEM}}$  and  $\delta_{10,50}^{\text{SEM}} > \delta_{50,500}^{\text{SEM}}$ . This suggests that estimators within each class will converge as component size increases, subject to sufficient data being available for such component sizes to exist.

The error in SEM estimators appears to increase more rapidly as component size decreases than that of the VEM estimators. This suggests that the choice of component size is less significant when using VEM estimators than when using

SEM estimators. Each of the columns in the top right quadrant of table 3.3, and the rightmost column in particular, support this suggestion, as each SEM estimator is almost equally far from all VEM estimators. The (entire) rightmost column shows the distances of all estimators from  $\hat{\theta}_{500}^{\text{SEM}}$ . It shows all VEM estimators being similarly far from the gold standard  $\hat{\theta}_{500}^{\text{SEM}}$ , while the SEM estimators converge on the gold standard as the component size increases. The bias introduced with the variational approximations therefore seems to dominate any effect from making a particular choice of component size.

**Table 3.3:** Results of bias experiment. All pairwise differences in parameter estimates as (disjoint) component size  $n$  changes, with  $n \in 2, 10, 50, 500$ , for VEM estimators and SEM estimators.

|     | $n$ | VEM   |       |       |       | SEM   |       |       |       |
|-----|-----|-------|-------|-------|-------|-------|-------|-------|-------|
|     |     | 2     | 10    | 50    | 500   | 2     | 10    | 50    | 500   |
| VEM | 2   | 0     | 4.83  | 5.54  | 5.77  | 8.90  | 21.25 | 31.47 | 38.47 |
|     | 10  | 4.83  | 0     | 1.18  | 1.34  | 10.46 | 21.04 | 31.36 | 38.38 |
|     | 50  | 5.54  | 1.18  | 0     | 0.39  | 10.81 | 20.94 | 31.28 | 38.31 |
|     | 500 | 5.77  | 1.34  | 0.39  | 0     | 10.95 | 20.92 | 31.27 | 38.30 |
| SEM | 2   | 8.90  | 10.46 | 10.81 | 10.95 | 0     | 17.69 | 28.16 | 35.24 |
|     | 10  | 21.25 | 21.04 | 20.94 | 20.92 | 17.69 | 0     | 15.49 | 24.40 |
|     | 50  | 31.47 | 31.36 | 31.28 | 31.27 | 28.16 | 15.49 | 0     | 9.63  |
|     | 500 | 38.47 | 38.38 | 38.31 | 38.30 | 35.24 | 24.40 | 9.63  | 0     |

### 3.8.2 Variance experiment

The results of the variance experiment are shown in fig 3.4 (p. 125), and summarised in table 3.4 (p. 124). Recall that only VEM estimators are computed in this experiment, as it is the interaction effect of variational approximations and composite likelihoods being explored. There is no clear pattern across all parameters. Elements of  $\hat{\rho}$  and  $\hat{\mathbf{d}}$  tend to have lower variance with smaller component sizes. Elements of  $\hat{\mathbf{w}}$  do not have a particularly strong pattern but the trend is generally in the opposite direction; variances are lower with larger component sizes. Elements of  $\hat{C}$  have no discernible pattern, indeed there is very little difference in their variances as component size changes. Table 3.4 shows the average variance for each parameter  $\hat{\rho}$ ,  $\hat{\mathbf{w}}$ ,  $\hat{C}$ ,  $\hat{\mathbf{d}}$  for each component size, to illustrate the relative differences in variances

across parameters.

**Table 3.4:** Average variance of elements from each parameter across changing component size  $n$ .

| $n$                | 2     | 10    | 50    | 500   |
|--------------------|-------|-------|-------|-------|
| $\hat{\rho}$       | 0.057 | 0.072 | 0.077 | 0.078 |
| $\hat{\mathbf{w}}$ | 0.054 | 0.046 | 0.043 | 0.043 |
| $\hat{C}$          | 2.316 | 2.301 | 2.320 | 2.332 |
| $\hat{\mathbf{d}}$ | 1.063 | 1.648 | 1.951 | 2.045 |

### 3.8.3 Smoothing experiment

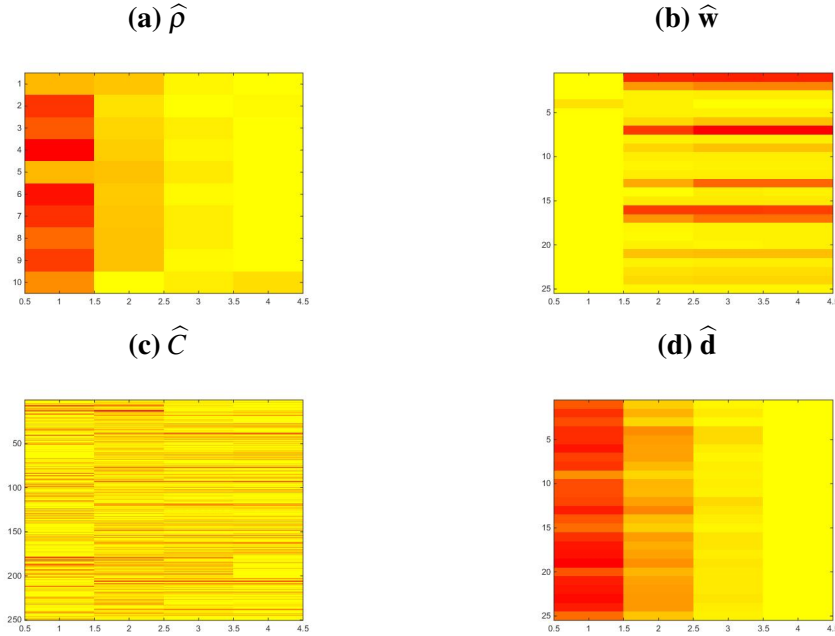
The results are shown in table 3.5. Several patterns can be seen, showing differences between VEM and SEM estimators, and between performance as measured by the two loss functions described in equations (3.86) and (3.88). All SEM estimators  $\hat{\theta}_n^{\text{SEM}}$  perform better than the corresponding VEM estimator  $\hat{\theta}_n^{\text{VEM}}$ . For other patterns, it is easiest to discuss the behaviour of VEM and SEM estimators separately.

For VEM estimators, losses as measured by both loss functions increase as component size  $n$  increases. The difference  $L_n^{(2),j} - L_n^{(1),j}$  stays roughly constant though. This indicates that the spread of particles stays the same while the particle means become less accurate.

For SEM estimators, the two loss functions  $L_n^{(2),j}$  and  $L_n^{(1),j}$  move in opposite directions as the component size  $n$  increases:  $L1$  loss goes down and  $L2$  loss goes up. The difference between them therefore increases with  $n$ . This indicates that the accuracy of the particle means improves while the spread of the particles widens.

Smoothing estimates made using the true parameters are significantly more accurate than those made using any of the different estimators. The spread of particles, as measured by the difference  $L_n^{(2),j} - L_n^{(1),j}$ , is larger than any of the spreads of VEM estimators but smaller than any of the spreads of SEM estimators. All observed patterns hold for both in-sample and out-of-sample data.

**Figure 3.4:** Heat maps of the normalised per-element variances of the elements in each parameter  $\hat{\rho}$ ,  $\hat{\mathbf{w}}$ ,  $\hat{C}$ ,  $\hat{\mathbf{d}}$ . Estimates are made using VEM, and each of the 4 columns in each plot corresponds to the variance vector of parameters estimated using a composite likelihood with component sizes  $n \in 2, 10, 50, 500$ . Each row of each plot is normalised by the largest variance in the row. Darker colours correspond to lower values.



**Table 3.5:** Root mean squared error (RMSE) of smoothing estimates made using estimated parameters and true parameters, on in-sample (first three rows) and out-of-sample (last three rows) data. Rows labelled  $L1$  use the first loss function  $L_n^{(1),j}$  described in sec 3.7, equation (3.86). Rows labelled  $L2$  use the second loss function  $L_n^{(2),j}$  described in sec 3.7, equation (3.88). Rows labelled Diff show the difference  $L_n^{(2),j} - L_n^{(1),j}$  between the two loss functions for each estimator  $\hat{\theta}_n^j$  and dataset (In/Out). The RMSE of smoothing estimates made using the true parameters are shown in the right-most column for reference.

| $n$  | VEM   |       |       |       | SEM   |       |       |       | $\theta^{\text{true}}$ |
|------|-------|-------|-------|-------|-------|-------|-------|-------|------------------------|
|      | 2     | 10    | 50    | 500   | 2     | 10    | 50    | 500   |                        |
| In   |       |       |       |       |       |       |       |       |                        |
| $L1$ | 1.717 | 1.755 | 1.768 | 1.771 | 1.088 | 1.034 | 1.005 | 1.002 | 0.181                  |
| $L2$ | 1.727 | 1.765 | 1.778 | 1.780 | 1.120 | 1.334 | 1.384 | 1.381 | 0.256                  |
| Diff | 0.010 | 0.010 | 0.010 | 0.010 | 0.111 | 0.299 | 0.380 | 0.379 | 0.075                  |
| Out  |       |       |       |       |       |       |       |       |                        |
| $L1$ | 1.762 | 1.780 | 1.812 | 1.815 | 1.146 | 1.061 | 1.011 | 1.009 | 0.175                  |
| $L2$ | 1.771 | 1.808 | 1.820 | 1.823 | 1.251 | 1.356 | 1.382 | 1.387 | 0.251                  |
| Diff | 0.008 | 0.009 | 0.009 | 0.009 | 0.105 | 0.295 | 0.371 | 0.378 | 0.077                  |

### 3.8.4 Prediction experiment

The results are shown in table 3.6. As with the smoothing experiment, several patterns can be seen. Differences can be observed between VEM and SEM estimators, and between performance as measured by the two loss functions described in equations (3.90) and (3.92).

Predictions made using VEM estimators  $\hat{\theta}_n^{\text{VEM}}$  outperform their corresponding SEM estimators  $\hat{\theta}_n^{\text{SEM}}$  when performance is measured by  $L1$  loss. The opposite is true when using  $L2$  loss though. For both VEM and SEM estimators, performance improves as component size  $n$  increases. This pattern holds for both loss functions. For other patterns, it is again easiest to discuss the behaviour of VEM and SEM estimators separately.

For VEM estimators, the spread of particles as measured by the difference  $L_n^{(2),j} - L_n^{(1),j}$  stays almost constant as component size  $n$  changes. There is in fact a consistent decrease with increasing  $n$ , but the effect is only slight. The improvement in  $L2$  performance with increasing  $n$  can largely be explained by the corresponding improvement in  $L1$  performance.

For SEM estimators, the spread of particles decreases rapidly with increasing component size  $n$ . The spreads of particles in predictions made using  $\hat{\theta}_{50}^{\text{SEM}}$  and  $\hat{\theta}_{500}^{\text{SEM}}$  are negligible; performance as measured by  $L1$  and  $L2$  losses are almost identical.

Predictions made using the true parameters have the best performance as measured by  $L1$  loss, but the worst performance as measured by  $L2$  loss. The spreads of particles in these predictions are therefore larger than those for any estimator. The mean prediction of particles is more accurate than for any estimator, but individual particle predictions are more widely spread than for any estimator. All patterns hold for both in-sample and out-of-sample data.

## 3.9 Discussion

The first notable result from the experiments is the small difference between complete composite likelihood estimators  $\hat{\theta}^{\cup}$  and disjoint-component-only estimators



**Table 3.6:** Root mean squared error (RMSE) of one-step-ahead predictions made using estimated parameters and true parameters, on in-sample (first three rows) and out-of-sample (last three rows) data. The rows labelled L1 use the first loss function  $L_n^{(1),j}$  described in sec 3.7, equation (3.90). The rows labelled L2 use the second loss function  $L_n^{(2),j}$  described in sec 3.7, equation (3.92). Rows labelled Diff show the difference  $L_n^{(2),j} - L_n^{(1),j}$  between the two loss functions for each estimator  $\hat{\theta}_n^j$  and dataset (In/Out). The RMSE of predictions made using the true parameters are shown in the right-most column for reference.

| $n$  | VEM   |       |       |       | SEM   |       |       |       | $\theta^{\text{true}}$ |
|------|-------|-------|-------|-------|-------|-------|-------|-------|------------------------|
|      | 2     | 10    | 50    | 500   | 2     | 10    | 50    | 500   |                        |
| In   |       |       |       |       |       |       |       |       |                        |
| L1   | 5.934 | 5.880 | 5.867 | 5.864 | 6.494 | 6.334 | 6.333 | 6.333 | 5.357                  |
| L2   | 7.314 | 7.228 | 7.197 | 7.188 | 6.919 | 6.459 | 6.333 | 6.333 | 7.442                  |
| Diff | 1.380 | 1.348 | 1.330 | 1.324 | 0.425 | 0.125 | 0.001 | 0.000 | 2.085                  |
| Out  |       |       |       |       |       |       |       |       |                        |
| L1   | 5.936 | 5.907 | 5.895 | 5.891 | 6.670 | 6.470 | 6.489 | 6.489 | 5.366                  |
| L2   | 7.316 | 7.251 | 7.212 | 7.210 | 7.083 | 6.593 | 6.489 | 6.489 | 7.449                  |
| Diff | 1.380 | 1.343 | 1.325 | 1.319 | 0.414 | 0.123 | 0.001 | 0.000 | 2.083                  |

$\hat{\theta}^{\dot{U}}$ . This is a particularly useful result, as in general the complete composite likelihood is much more expensive to maximise. Indeed, the times taken to compute each of these estimators for component size  $n = 2$ , shown in table 3.1, indicate a linear scaling with the number of processed observations  $M = n(T - n + 1)$ .

This number  $M$  is quadratic in  $n$ , and peaks at  $n = 250$  with  $M = 62,750$  processed observations. With regard to the experiments of the current chapter, component sizes of  $n = 2, 10, 50, 500$  would each have  $M = 998, 4910, 22,550, 500$  processed observations respectively. Such an increase in processing requirements would almost certainly translate into significantly increased compute times for the component sizes  $n = 10, 50$  in the subsequent experiments. Being able to experimentally justify the use of disjoint-only composite likelihoods allows a massive reduction in the run time of experiments.

For both the VEM and SEM algorithms, as they exploit message passing techniques the computational complexity of approximating expectations for one component is linear in its size  $n$ . When disjoint-only composite likelihoods are used, therefore, the running time of each algorithm will be roughly constant across component sizes, i.e. the only significant differences in run time are between the VEM

and the SEM estimators. Table 3.1 shows that computing SEM estimators increases computational cost by a factor of just over 70. For applications that don't require the improved accuracy that is achieved with SEM estimators, this extra cost is very difficult to justify.

The results of the bias experiment provide some interesting insight into the bias effect of variational approximations on estimators. The initial hypothesis of the current chapter would predict that a larger block size in a composite likelihood would result in a larger bias from making a variational approximation. This was not shown to hold in the bias experiment. Instead, the bias of VEM estimators remains almost constant across block sizes, and actually reduces slightly as block size increases.

One possible explanation for this is that the cost of making increasingly bad approximations to posteriors is balanced by the benefit of conditioning on increasing amounts of data. If this is the case, then it is quite remarkable how well balanced these two effects are. Regardless of the cause though, the consequence of making variational approximations in a composite likelihood context seems to be a bias to estimators that does not change with block size.

The results of the smoothing and prediction experiments illustrate the relevance of the application for which inference is being performed. In prediction tasks, if only a point prediction is required, VEM estimators produce better predictions than SEM estimators. If a predictive distribution is required, or any form of smoothing estimate, then SEM estimators will have better performance in these tasks.

Regarding smoothing estimates, in both the in-sample and out-of-sample tests all VEM smoothing estimates had high  $L1$  loss and consistently tight distributions of particles around their means. Furthermore, the  $L1$  loss increased slightly with increasing component size  $n$ . This is in contrast to the bias experiment, which showed estimators  $\hat{\theta}_n^{\text{VEM}}$  getting slightly closer to the gold standard  $\hat{\theta}_{500}^{\text{SEM}}$  with increasing  $n$ . The potential explanation for this offered above was that the cost of using increasingly bad approximations to posteriors was balanced pretty well by the benefit of having more data in each of the posteriors being approximated.

This could also be the explanation here, only with the net effect in smoothing estimates going in the opposite direction, i.e. their quality gets slightly worse as  $n$  increases. Perhaps the different contexts of bias estimation and smoothing produce different patterns of net effect, while the underlying balance of costs and benefits to changing component size largely holds true.

The pattern of results for SEM smoothing estimates was markedly different to that for VEM smoothing estimates.  $L1$  loss decreased with increasing  $n$ , while  $L2$  loss increased; the accuracy of mean particles increased with increasing component size but particles also became more widely distributed. This is a curious result, particularly when considering the performance of the true parameters as well. The  $L1$  loss for SEM estimators gets increasingly close to that for  $\hat{\theta}^{\text{true}}$  as  $n$  increases, but the  $L2$  loss gets increasingly far.

In the prediction experiment, the results are slightly more difficult to interpret. The two loss functions lead to opposite conclusions regarding the relative performance of VEM and SEM estimators. Predictions made via VEM estimators have lower  $L1$  loss than those made via SEM estimators, but higher  $L2$  loss. The mean particles of VEM predictions were more accurate than their corresponding SEM predictions, but the predictive distributions were also significantly more widely spread.

The spread of particles in VEM predictions, as measured by the difference between  $L2$  and  $L1$  losses, decreased slowly with increasing  $n$ .  $L1$  loss also decreased slowly with increasing  $n$ , and at a similarly slow rate to the decrease in spread. The spread of particles in SEM predictions, however, rapidly goes to zero with increasing  $n$ . For component sizes  $n = 50$  and  $n = 500$ , the spread is negligible; individual particle predictions are very close to each other.

The effect of changing component size  $n$  on the performance of VEM estimators in this experiment is not very pronounced, similarly to the bias and smoothing experiments. In this experiment, the net effect of increasing component size on predictive performance is slightly positive; using larger components to estimate parameters appears to both improve the predictive quality of particle means, and

narrow the spread of particles around their means.

It is strange that SEM estimators have consistently better performance than VEM estimators at smoothing, without this also being the case for making predictions. In particular the mean predictions of particles from SEM estimators have less predictive power than their VEM counterparts. SEM estimators make unbiased approximations to posteriors, so it should be expected that they would show better performance than VEM estimators at all tasks.

It is possible that the prediction task for this data is particularly hard. Even using the true parameters does not produce unambiguously superior predictions. Predictions made using  $\theta^{\text{true}}$  have the lowest  $L1$  loss of all predictive distributions, but the highest  $L2$  loss. The performance of SEM predictions seems particularly difficult to explain when compared to the performance of the true parameters.

Looking at the smoothing and prediction experiments together, it seems that particle based approximate distributions can have multiple features to their profiles as component size  $n$  changes. The accuracy of the particle means can improve or get worse with increasing  $n$ , and the spread of particles can get tighter, less tight, or stay roughly constant. VEM and SEM estimators do not share many features in either experiment. The only trend that holds for both types of estimator is in the prediction experiment, where the predictive power of estimators improved when component size is increased:  $L_{n_1}^{(i),j} < L_{n_2}^{(i),j}$  for  $n_1 > n_2$  and both of  $i = 1$  and  $i = 2$  and both of  $j = \text{VEM}$  and  $j = \text{SEM}$ .

VEM estimators produce smoothing estimates with very tight spreads of particles, but predictive distributions with loose spreads. SEM estimators produce smoothing estimates whose particles have increasing spreads with increasing component size  $n$ , but predictive distributions whose particles have spreads that vanish with increasing  $n$ . The accuracy of particle means for VEM smoothing estimates gets worse increasing  $n$ , but it improves for predictive distributions. With SEM estimators, the accuracy of particle means for both the smoothing estimates and the predictive distributions they produce follow the same trend. In both cases the accuracy improves with increasing  $n$ .

These results make it difficult to assess whether the hypothesis of this chapter regarding variational approximations holds true or not. The application in question clearly affects the efficacy of the different estimators in the experiments. The bias, smoothing, and prediction experiments suggest there might be two opposing effects to increasing component size on variational approximations: a benefit from having more data in each component, and a cost from using increasingly poor approximations to posteriors. If this is the case then the two effects are pretty well balanced in all experiments, but with the stronger effect of the two depending on the experiment in question.

The results of the experiments cannot prove such a conclusion, though they do provide supportive evidence. The net effect of changing component size on VEM estimators is small in all applications that were experimented with. Why the net effect shows opposite trends in smoothing to those of the bias and prediction experiments is an interesting question. Perhaps the natures of the tasks themselves have a structural impact on the measured quality of estimators.

If the results of the current chapter are to be used for deciding when variational approximations would be appropriate, then the recommendations are clear. In the specific context of making point predictions, VEM estimators produce better predictions than SEM estimators. In all other contexts, SEM estimators outperform their VEM counterparts. Whether the performance improvement is worth the significant increase in computation costs will depend on the specific context in question.



## Chapter 4

# Integrating Composite Likelihood and Non-Likelihood Based Methods

As the content of chapter 3 illustrated, one of the most computationally challenging aspects of developing an algorithm for making approximate composite likelihood parameter estimates in state space models is the (approximate) evaluation of the log composite likelihood. Assuming the dimensionality of  $\mathcal{Y}$  is too large for quadrature to be undertaken, either a deterministic approximation or a Monte Carlo estimate has to be made.

When the latent process is not modelled to be stationary there is the further problem of calculating the latent marginals for each component of the composite likelihood. For a linear Gaussian latent process they can be calculated analytically, but maximising the log composite likelihood with respect to parameters will be challenging. Furthermore, the functional dependence of each component likelihood on the parameters becomes increasingly complex with time.

The use of surrogate marginals, estimated via the method of moments, is proposed in the current chapter to overcome this challenge. These surrogates will be used to find tractable approximations to the latent posterior for each component in the composite likelihood. Evaluating log composite likelihoods in this way is practical and computationally feasible, and the cost of estimating the surrogates is negligible.

In addition, the use of Gaussian distributions as tractable approximations to

latent posteriors is also investigated. The classes  $Q$  of tractable distributions used in variational approximations are often chosen to be distributions that factorise over subsets of the latent variables. For general state space models, factorisations that produce tractable approximations to the latent posterior do not necessarily exist. By instead choosing  $Q$  to be a parametric family of distributions, their existence is obviously guaranteed.

Once the closest Gaussians to the (composite) latent posterior are found, the statistical benefits of their use is investigated. They are used to make variational approximations to log composite likelihoods, and are also used as importance distributions for Monte Carlo estimates using importance sampling. The bias of estimated parameters, and the quality of predictions made using those parameters, will be weighed against the computational costs of their estimation.

## 4.1 Problem outline

The specific problem this chapter aims to investigate is the estimation of component likelihoods of a state space model, for use in the approximation of maximum composite likelihood parameter estimation. A Gaussian-Poisson state space model will be the subject of inference, as it has both simple and complex features. Its computational challenges are sufficient for an approximate inference regime of some description to be necessary, while some necessary calculations remain analytically tractable. The investigations of this chapter therefore have an illustrative context for inference that is not weighed down with excessive complications.

The state space model is extended to allow multiple i.i.d. realisations of data, implying that each component in the composite likelihood now comprises a product over the realisations:

$$L_C(\theta | Y) = \prod_{c=1}^C \prod_{d=1}^D \mathcal{P}(Y_{M_c, :, d} | \theta) \quad (4.1)$$

where  $Y_{t,s,d}$  denotes the  $s^{\text{th}}$  element of the data at time  $t$  in the  $d^{\text{th}}$  i.i.d. realisation, and the colon operator denotes the vector valued index comprising all possible index values. Each component  $c$  in (4.1) is the product of  $D$  marginal data likelihoods,



each of which comprises  $n$  contiguous data observations from the same time interval but in different realisations  $d$ . The vector valued index  $M_c = t_c, \dots, t_c + n - 1$  for some  $t_c \in 1, \dots, T - n + 1$  denotes the common time interval for these contiguous data.

Each component log likelihood will be estimated by variational approximations and also by Monte Carlo estimates. As mentioned above, factorised distributions will not form the classes of tractable distributions in the current chapter. Instead, variational approximations will be implemented using two different Gaussian classes as tractable distributions; *a*) the class of general Gaussians of appropriate dimension:

$$\mathcal{Q}_{\mathcal{N}} = \{\mathcal{N}(\mu, \Sigma) : \dim \mu = \dim \mathcal{X}^n\} \quad (4.2)$$

where  $n$  is the number of observations in each component of the composite likelihood, and *b*) the class of Gaussians of appropriate dimension having a diagonal covariance matrix:

$$\mathcal{Q}_{\Pi\mathcal{N}} = \{\mathcal{N}(\mu, \Sigma) : \dim \mu = \dim \mathcal{X}^n, \Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_{\dim \mathcal{X}^n}^2)\} \quad (4.3)$$

Variational EM estimates will be made by using these closest Gaussians as variational distributions. Stochastic EM estimates will be made by using them as importance distributions for Monte Carlo estimates made using importance sampling. The trade-off between the costs of performing each KL divergence minimisation and the resulting impact on the quality of the lower bound will be investigated for both forms of parameter estimate.

The effects of increasing  $\dim \mathcal{X}$  will also be investigated. Inference will be performed on synthetic data of varying latent dimension. The computational cost of finding the closest Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$  for each  $\dim \mathcal{X}$  will be observed, along with the quality of parameter estimates, and of predictions made using them.

## 4.2 Gaussian-Poisson state space model

The state space model that is to be the subject of inference is a particular Gaussian-Poisson model. The model is an extension of the previously defined state space model of chapter 3, in that it allows for multiple i.i.d. realisations of data sharing a common structure:

$$\begin{aligned}
 \mathcal{P}(X_{1,:d} \mid \theta) &= \mathcal{N}(X_{1,:d} \mid \mu_1, \Sigma_1), \quad d \in 1, \dots, D \\
 \mathcal{P}(X_{t+1,:d} \mid X_{t,:d}, \theta) &= \mathcal{N}(X_{t+1,:d} \mid A_t X_{t,:d} + \mathbf{b}_t, V_t), \quad t \in 1, \dots, T-1 \\
 \mathcal{P}(Y_{t,:d} \mid x_{t,:d}, \theta) &= \prod_{s=1}^S \mathcal{P}(Y_{t,s,d} \mid X_{t,s,d}) \\
 &= \prod_{s=1}^S \mathcal{PO}(Y_{t,s,d} \mid \exp(X_{t,s,d})) \tag{4.4}
 \end{aligned}$$

where  $\dim \mathcal{X} = \dim \mathcal{Y} = S$ , and subscripts are as described in each 4.1 above. For each time  $t$ , and each data realisation  $d$ , therefore, both  $X_{t,:d}$  and  $Y_{t,:d}$  are  $S$  dimensional variables. When in the following discourse the colon operator is used with bounds in a subscript, i.e.  $a : b$ , this is meant to denote the interval of index values  $a, \dots, b$ . Note that different parameters control the transitions at each time point, so the latent process is not time-homogeneous and therefore non-stationary. It should also be noted that no parameters control the conditional data distributions given the latent variable. All parametric flexibility in the model is achieved through controlling the dynamics of the latent process.

A model such as this could be used to model neural spiking activity, as in for example Buesing et al. (2012); Byron et al. (2009); Smith and Brown (2003); Kulkarni and Paninski (2007); Byron et al. (2005). Similar models to (4.4) are used in these examples, though the latent process is often modelled to be stationary. An example application where non-stationarity has to be explicitly modelled is the exit counts for subway stations over the course of a day. For data like this, it is reasonable to expect some temporal dynamics to be operating, and for those dynamics to change throughout the day. A dataset of this kind is used in the content of chapter 5, and is described in more detail in that chapter.

As mentioned above, the latent marginals  $\mathcal{P}(X_{M_c, :, d} \mid \theta)$  need to be calculated to evaluate each component  $\sum_{d=1}^D \log \mathcal{P}(Y_{M_c, :, d} \mid \theta)$  of the log composite likelihood. The latent process in (4.4) is linear Gaussian so they can be calculated analytically, but the non-stationarity adds a complication. Calculating the functional dependence on  $\theta$  of each component in the composite likelihood becomes non-trivial. Furthermore, the functional dependence gets increasingly complex for components that are further forward in time.

Surrogate marginals  $\tilde{\mathcal{P}}(X_{t, :, d}) = \mathcal{N}(X_{t, :, d} \mid \tilde{\mu}_t, \tilde{\Sigma}_t)$  can be used to overcome this problem. If the parameters  $\tilde{\mu}_t, \tilde{\Sigma}_t$  are set to the method of moments estimators of the true parameters  $\mu_t, \Sigma_t$  (derived in sec 4.3), then by the law of large numbers they will converge to the true marginals as the amount of data increases:

$$\begin{aligned} (\tilde{\mu}_t, \tilde{\Sigma}_t) &= (\hat{\mu}_t^{\text{m.m.}}, \hat{\Sigma}_t^{\text{m.m.}}) \\ \Rightarrow \tilde{\mathcal{P}}(X_{t, :, d}) &\rightarrow_{\text{dist.}} \mathcal{P}(X_{t, :, d} \mid \mu_t, \Sigma_t) \quad \text{as } D \rightarrow \infty \end{aligned} \quad (4.5)$$

where  $(\hat{\mu}_t^{\text{m.m.}}, \hat{\Sigma}_t^{\text{m.m.}})$  are the method of moments estimators of  $(\mu_t, \Sigma_t)$ . Using such surrogate marginals is the approach taken here; multiple data realisations allow the latent marginals  $\mathcal{P}(X_{t, :, d} \mid \theta)$  to be approximated and thus keep the functional dependence of component log likelihoods  $\sum_{d=1}^D \log \mathcal{P}(Y_{M_c, :, d} \mid \theta)$  on parameters  $\theta$  constant across components.

Synthetic data will be used in experiments to explore the quality of estimators computed using surrogate marginals, and to compare the trade-offs with each choice of tractable distribution class. As the data are synthetically generated, the computational complexity of finding optimal distributions from  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$  can be controlled by choosing either or both of  $S$  and  $n$ . As such, the component size  $n$  will be kept fixed at  $n = 2$ , and  $S$  will be varied in experiments.

### 4.2.1 Exploiting surrogate marginals

If a composite likelihood approach is used to estimate the parameters in (4.4), then each of the components in the composite likelihood will have a functional depen-

dence on the parameters of all previous transitions:

$$\mathcal{P}(Y_{t:t+1,::} | \boldsymbol{\theta}) = \mathcal{P}(Y_{t:t+1,::} | \boldsymbol{\theta}_{1:t}) \quad (4.6)$$

where  $\boldsymbol{\theta}_t = (A_t, \mathbf{b}_t, V_t)$  denotes the parameters governing the  $t^{\text{th}}$  transition. This functional dependence can cause difficulties in both the expectation and maximisation steps of an approximate EM algorithm. Using surrogate marginals  $\tilde{\mathcal{P}}(X_{t,::,d})$  breaks these dependencies in the latent process:

$$\begin{aligned} \tilde{\mathcal{P}}(X_{t:t+1,::,d} | \boldsymbol{\theta}) &= \tilde{\mathcal{P}}(X_{t,::,d}) \mathcal{N}(X_{t+1,::,d} | A_t X_{t,::,d} + \mathbf{b}_t, V_t) \\ &= \tilde{\mathcal{P}}(X_{t:t+1,::,d} | \boldsymbol{\theta}_t) \end{aligned} \quad (4.7)$$

and consequently the dependencies in the components of the (approximate) composite likelihood:

$$\begin{aligned} \tilde{\mathcal{P}}(Y_{t:t+1,::} | \boldsymbol{\theta}) &= \prod_{d=1}^D \int \tilde{\mathcal{P}}(X_{t:t+1,::,d} | \boldsymbol{\theta}_t) \prod_{s=1}^S \mathcal{P}(Y_{t:t+1,s,d} | X_{t:t+1,s,d}, \boldsymbol{\theta}_t) dX_{t:t+1,::,d} \\ &= \tilde{\mathcal{P}}(Y_{t:t+1,::} | \boldsymbol{\theta}_t) \end{aligned} \quad (4.8)$$

The surrogate marginals therefore use exogenously provided parameter estimates to disconnect pieces of a model, in order to facilitate further parameter estimation. This idea has been suggested recently in Halpern and Sontag (2013), who use the general concept in the context of noisy-OR models. In the current context, their use can overcome difficulties otherwise encountered in both steps of an approximate EM algorithm.

In the expectation step, the closest Gaussians from each of the classes  $\mathcal{Q}_N, \mathcal{Q}_{\Pi N}$  are used to approximate the latent posteriors  $\mathcal{P}(X_{t,t+1,::,d} | Y_{t,t+1,::,d}, \boldsymbol{\theta})$ . A first step of this procedure is to compute the latent priors  $\mathcal{P}(X_{t,t+1,::,d} | \boldsymbol{\theta})$ . The Markov property of the latent process determines that the dependence on previous parameters  $\boldsymbol{\theta}_{1:t-1}$  is encoded in the latent marginal  $\mathcal{P}(X_{t,::,d} | \boldsymbol{\theta})$ , and as the latent process is linear-Gaussian, calculating  $\mathcal{P}(X_{t,::,d} | \boldsymbol{\theta})$  is a tractable operation.

Particularly in the early iterations of an approximate EM algorithm, though,

current estimates of previous parameters  $\theta_{1:t-1}$  might be significantly inaccurate. The latent marginals  $\mathcal{P}(X_{t,:},d \mid \theta)$  derived from these parameters would inherit such inaccuracies, which could slow down the improvements to estimates  $\hat{\theta}_t$  at each EM iteration. This effect would increase for parameters  $\theta_t$  as  $t$  increases.

The surrogate marginals  $\tilde{\mathcal{P}}(X_{t,:},d)$  do not have any functional dependence on previous parameters  $\theta_{1:t-1}$ , so they will not suffer from this pathology. If the surrogates are unbiased estimates of the true marginals, then substituting them in place of the current ‘true’ marginals  $\mathcal{P}(X_{t,:},d \mid \theta)$  can be justified and should bring a faster convergence rate to the estimation algorithm.

In the maximisation step, the functional dependence of component likelihoods on previous parameters brings a computational challenge. As all components are functionally dependent on  $\theta_1$ , all but one of the components dependent on  $\theta_2$  etc., the computational cost of each maximisation step is quadratic in the number of components in the composite likelihood:

$$\hat{\theta}_t = \arg \max_{\theta_t} \sum_{d=1}^D \sum_{\tau=t}^T \log \mathcal{P}(Y_{\tau:\tau+1,:},d \mid \theta_{1:\tau}) \quad (4.9)$$

By breaking the functional dependencies of each component on previous parameters, this cost is reduced to linear:

$$\hat{\theta}_t = \arg \max_{\theta_t} \sum_{d=1}^D \log \tilde{\mathcal{P}}(Y_{t:t+1,:},d \mid \theta_t) \quad (4.10)$$

### 4.3 Surrogate marginals

In this section the surrogate marginals  $\mathcal{P}(X_{t,:},d) = \mathcal{N}(X_{t,s},d \mid \tilde{\mu}_t, \tilde{\Sigma}_t)$  will be derived. Parameter thresholding is part of the procedure for their determination, but the naive form of their derivation is first described and the thresholding discussed subsequently. The derivation presented here largely follows that of Buesing et al. (2012), though the details of parameter thresholding differ slightly.

As discussed in sec 4.2, the surrogate parameters are the method of moments estimators of the true parameters. Their derivation is therefore based on the moment

equations:

$$\mathbb{E}[Y_{t,s,d}] = \exp\left(\mu_{t,s} + \frac{1}{2}\sigma_{t,s}^2\right) \quad (4.11)$$

$$\mathbb{E}[Y_{t,s,d}^2 - Y_{t,s,d}] = \exp(2\mu_{t,s} + 2\sigma_{t,s}^2) \quad (4.12)$$

$$\mathbb{E}[Y_{t,s_1,d}Y_{t,s_2,d}] = \mathbb{E}[Y_{t,s_1,d}]\mathbb{E}[Y_{t,s_2,d}]\exp(\sigma_{(t,s_1),(t,s_2)}), \quad s_1 \neq s_2 \quad (4.13)$$

where  $\sigma_{(t,s_1),(t,s_2)} = \text{cov}(X_{t,s_1,d}, X_{t,s_2,d})$  is the covariance between two different dimensions of the latent process at time  $t$ , and  $\sigma_{t,s}^2 = \sigma_{(t,s),(t,s)}$  is the variance of dimension  $s$  at time  $t$ .

Assuming the moments of  $Y_{t,:d}$  were known exactly then equations (4.11 - 4.13) could be used to exactly infer the moments of  $X_{t,:d}$ . The covariance  $\sigma_{(t,s_1),(t,s_2)}$  between two different dimension is trivially read from (4.13):

$$\sigma_{(t,s_1),(t,s_2)} = \log\left(\frac{\mathbb{E}[Y_{t,s_1,d}Y_{t,s_2,d}]}{\mathbb{E}[Y_{t,s_1,d}]\mathbb{E}[Y_{t,s_2,d}]}\right) \quad (4.14)$$

and for the calculation of the means  $\mu_{t,s}$  and variances  $\sigma_{t,s}^2$  it is convenient to first define the quantities

$$\begin{aligned} z_1(t,s) &= \log(\mathbb{E}[Y_{t,s,d}]) \\ z_2(t,s) &= \log(\mathbb{E}[Y_{t,s,d}^2 - Y_{t,s,d}]) \end{aligned} \quad (4.15)$$

which clarify the calculations for the mean and variance of each dimension  $X_{t,s,d}$ :

$$\begin{aligned} \mu_{t,s} &= 2z_1(t,s) - \frac{1}{2}z_2(t,s) \\ \sigma_{t,s}^2 &= z_2(t,s) - 2z_1(t,s) \end{aligned} \quad (4.16)$$

The naive method of moments estimators of  $\mu_{t,s}$ ,  $\sigma_{t,s}^2$ ,  $\Sigma_{(t,s_1),(t,s_2)}$  are made by

replacing the expected values in (4.14), (4.16) with sample moments from the data:

$$\begin{aligned}\tilde{\mu}_{t,s} &= 2\tilde{z}_1(t,s) - \frac{1}{2}\tilde{z}_2(t,s) \\ \tilde{\sigma}_{t,s}^2 &= \tilde{z}_2(t,s) - 2\tilde{z}_1(t,s) \\ \tilde{\sigma}_{(t,s_1),(t,s_2)} &= \log\left(\frac{\overline{Y_{t,s_1}Y_{t,s_2}}}{\overline{Y_{t,s_1}}\overline{Y_{t,s_2}}}\right)\end{aligned}\quad (4.17)$$

where

$$\begin{aligned}\tilde{z}_1(t,s) &= \log(\overline{Y_{t,s}}) \\ \tilde{z}_2(t,s) &= \log(\overline{Y_{t,s}^2} - \overline{Y_{t,s}}^2)\end{aligned}\quad (4.18)$$

and over-lines indicate sample means. The naive surrogate covariance matrix is constructed from the variance and covariance parameters:

$$\tilde{\Sigma}_t = \{\tilde{\sigma}_{t,s_1,s_2}\} = \begin{cases} \tilde{\sigma}_{t,s}^2 & s_1 = s_2 = s \\ \tilde{\sigma}_{(t,s_1),(t,s_2)} & s_1 \neq s_2 \end{cases}\quad (4.19)$$

As the linear-Gaussianity of the auto-regression in (4.4) implies that all latent marginals are Gaussian, using Gaussians with parameters as in (4.17) as surrogate marginals will ensure that the surrogates converge to the true marginals as  $D \rightarrow \infty$ .

### 4.3.1 Thresholding

Though defining surrogate parameters as in (4.17) ensures convergence of the surrogates to the true marginals, there are still practical considerations when calculating the parameters in practice. As the data are conditionally Poisson, there is a non-zero probability that some or all of  $z_1(t,s), z_2(t,s), \overline{Y_{t,s_1}Y_{t,s_2}}$  at any time  $t$  are 0. As  $\log(0)$  is undefined, this would cause any algorithm using such parameters to fail. Furthermore, the positive definiteness of  $\tilde{\Sigma}_t$  is not guaranteed when using sample means in place of expected values, so the estimated covariance matrix for each surrogate must be projected onto the space of positive definite matrices.

The first of these problems is avoided by thresholding the sample means:

$$\begin{aligned}\widehat{\bar{Y}}_{t,s} &= \max(\bar{Y}_{t,s}, \delta) \\ \widehat{\overline{Y_{t,s_1} Y_{t,s_2}}} &= \max(\overline{Y_{t,s_1} Y_{t,s_2}}, \delta)\end{aligned}\quad (4.20)$$

for some small  $\delta > 0$ , and then observing that the requirement

$$\tilde{\sigma}_{t,s}^2 > 0 \Rightarrow \log(\overline{Y^2}_{t,s} - \bar{Y}_{t,s}) > 2 \log(\bar{Y}_{t,s}) \quad (4.21)$$

can be enforced similarly:

$$\widehat{\overline{Y^2}}_{t,s} = \max(\overline{Y^2}_{t,s}, \widehat{\bar{Y}}_{t,s} + \widehat{\bar{Y}}_{t,s}^2 + \delta) \quad (4.22)$$

for some small  $\delta > 0$ .

These thresholded sample means are used in place of the true sample means:

$$\begin{aligned}\widehat{\mu}_{t,s} &= 2\widehat{z}_1(t,s) - \frac{1}{2}\widehat{z}_2(t,s) \\ \widehat{\sigma}_{t,s}^2 &= \widehat{z}_2(t,s) - 2\widehat{z}_1(t,s) \\ \widehat{\sigma}_{(t,s_1),(t,s_2)} &= \log\left(\frac{\widehat{\overline{Y_{t,s_1} Y_{t,s_2}}}}{\widehat{\bar{Y}}_{t,s_1} \widehat{\bar{Y}}_{t,s_2}}\right) \\ \widehat{S}_t = \{\widehat{\sigma}_{t,s_1,s_2}\} &= \begin{cases} \widehat{\sigma}_{t,s}^2 & s_1 = s_2 = s \\ \widehat{\sigma}_{(t,s_1),(t,s_2)} & s_1 \neq s_2 \end{cases}\end{aligned}\quad (4.23)$$

where  $\widehat{z}_1(t,s), \widehat{z}_2(t,s)$  are the thresholded equivalents to  $\tilde{z}_1(t,s), \tilde{z}_2(t,s)$ .

As mentioned above, even after this thresholding, the positive definiteness of  $\tilde{\Sigma}_t$  is not guaranteed. A projection onto the space of positive definite matrices of appropriate dimension is therefore required. Following Buesing et al. (2012), the projection is chosen to minimise the Frobenius norm of the difference between  $\tilde{\Sigma}_t$  and its projection. This is achieved by thresholding the eigenvalues of  $\tilde{\Sigma}_t$ :

$$\widehat{S}_t = V \cdot \text{diag}(\tilde{D}) \cdot V' \quad (4.24)$$



where

$$\widehat{S}_t = V \cdot \text{diag}(D) \cdot V' \quad (4.25)$$

is the eigenvalue decomposition of  $\widehat{S}_t$ , and

$$\widetilde{D}_s = \max(D_s, \delta) \quad (4.26)$$

for some small  $\delta > 0$ .

This projection will in general modify all entries of  $\widehat{S}_t$ , which is not necessarily desirable. When applied in the experiments of the current chapter, the modifications to the variance parameters  $\sigma_{t,s}^2$  after projection were found in practice to be causing unacceptable numerical errors in subsequent inference. To counteract this issue, a rescaling of the projection is performed, such that the resulting diagonal equals the diagonal of the original  $\widehat{S}_t$ :

$$\begin{aligned} \widehat{\Sigma}_t &= \text{diag}(R) \cdot V \cdot \text{diag}(\widetilde{D}) \cdot V' \cdot \text{diag}(R) \\ R_s &= \sqrt{\frac{\widehat{\sigma}_{t,s}^2}{\widehat{S}_{t,s,s}}} \end{aligned} \quad (4.27)$$

where  $\widehat{S}_{t,s,s}$  are the elements of the diagonal of  $\widehat{S}_t$ . The thresholded parameters  $\widehat{\mu}_t = \widehat{\mu}_{t,:}, \widehat{\Sigma}_t$  are used as the parameters for the surrogate marginals,:

$$\tilde{P}(X_{t,:,d}) = \mathcal{N}(X_{t,:,d} \mid \widehat{\mu}_t, \widehat{\Sigma}_t) \quad (4.28)$$

as detailed in algorithm 4.1.

## 4.4 Estimating maximum composite likelihood parameters

The surrogate marginals derived in sec 4.3 are used to develop approximations to maximum composite likelihood parameter estimates. As discussed in sec 4.1, the

**Algorithm 4.1** Calculating the surrogate marginals

1. Calculate thresholded sample moments for each time  $t$ , as per (4.20), (4.22):

$$\begin{aligned}\widehat{Y}_{t,s} &= \max(\overline{Y}_{t,s}, \delta) \\ \widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2} &= \max(\overline{Y}_{t,s_1} \overline{Y}_{t,s_2}, \delta) \\ \widehat{Y}_{t,s}^2 &= \max(\overline{Y}_{t,s}^2, \widehat{Y}_{t,s} + \widehat{Y}_{t,s} + \delta) \quad \delta > 0\end{aligned}\quad (4.29)$$

2. Construct mean-thresholded parameters as per (4.23):

$$\begin{aligned}\widehat{\mu}_{t,s} &= 2\widehat{z}_1(t,s) - \frac{1}{2}\widehat{z}_2(t,s) \\ \widehat{\sigma}_{t,s}^2 &= \widehat{z}_2(t,s) - 2\widehat{z}_1(t,s) \\ \widehat{\sigma}_{(t,s_1),(t,s_2)} &= \log\left(\frac{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}}{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}}\right) \\ \widehat{S}_t = \{\widehat{\sigma}_{t,s_1,s_2}\} &= \begin{cases} \widehat{\sigma}_{t,s}^2 & s_1 = s_2 = s \\ \widehat{\sigma}_{(t,s_1),(t,s_2)} & s_1 \neq s_2 \end{cases}\end{aligned}\quad (4.30)$$

3. Threshold the eigenvalues of  $\widehat{S}_t$ , and rescale resulting positive definite matrix such that its diagonal equals that of  $\widehat{S}_t$ , as per (4.27):

$$\begin{aligned}\widehat{\Sigma}_t &= \text{diag}(R) \cdot V \cdot \text{diag}(\widehat{D}) \cdot V' \cdot \text{diag}(R) \\ R_s &= \sqrt{\frac{\widehat{\sigma}_{t,s}^2}{\widehat{S}_{t,s,s}}}\end{aligned}\quad (4.31)$$

4. Use  $\widehat{\mu}_t = \widehat{\mu}_{t,:}$ ,  $\widehat{\Sigma}_t$  as parameters for surrogate marginals:

$$\tilde{P}(X_{t,:},d) = \mathcal{N}\left(X_{t,:},d \mid \widehat{\mu}_t, \widehat{\Sigma}_t\right)\quad (4.32)$$

closest Gaussians to the latent posteriors from

$$\begin{aligned} \mathcal{Q}_{\mathcal{N}} &= \{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) : \dim \boldsymbol{\mu} = \dim \mathcal{X}^n\} \\ \mathcal{Q}_{\Pi\mathcal{N}} &= \{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) : \dim \boldsymbol{\mu} = \dim \mathcal{X}^n, \boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_{\dim \mathcal{X}^n}^2)\} \end{aligned} \quad (4.33)$$

with  $n = 2$  kept fixed, will be used as both variational approximations for variational EM estimators, and importance distributions for stochastic EM estimators. For clarity it should be noted that the independence between the different realisations of data  $Y_{:,:,d}$  implies that the posterior distribution of all latent variables factorises over realisations, and as such there are  $D(T - 1)$  independent component posteriors of dimension  $2S$  being independently approximated by Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$ .

As described in sec 4.2.1, the surrogate marginals  $\tilde{\mathcal{P}}(X_{t, :, d})$  are exploited in both the expectation steps and the maximisation steps of the approximate EM algorithms described above. Expectations are made by first finding the closest Gaussians to the approximate posteriors

$$\begin{aligned} \tilde{\mathcal{P}}(X_{t:t+1, :, d} | Y_{t:t+1, :, d}, \hat{\boldsymbol{\theta}}^{(k)}) &\stackrel{\text{approx.}}{\propto} \tilde{\mathcal{P}}(X_{t, :, d}) \mathcal{N}\left(X_{t+1, :, d} | \hat{A}_t^{(k)} X_{t, :, d} + \hat{\mathbf{b}}_t^{(k)}, \hat{V}_t^{(k)}\right) \\ &\times \prod_{\tau=t}^{t+1} \prod_{s=1}^S \mathcal{PO}(Y_{\tau, s, d} | \exp(X_{\tau, s, d})) \end{aligned} \quad (4.34)$$

The maximisation step exploits the break of functional dependence of each component in the log composite likelihood. Each component depends only on the parameters associated to the same time step, as per (4.10).

#### 4.4.1 Expectation step

The expectation of each component

$$\sum_{d=1}^D \log \mathcal{P}(X_{t, t+1, :, d}, Y_{t:t+1, d} | \boldsymbol{\theta}) \quad (4.35)$$

in the full log composite likelihood is taken with respect to a different approximation to the latent posterior for each approximation method. Underlying each of these

approximate distributions is the closest Gaussian to the latent posterior:

$$q^*(X_{t:t+1, :, d}) = \arg \min_{q \in \mathcal{Q}} \text{KL} [q(X_{t:t+1, :, d}) \parallel \mathcal{P}(X_{t:t+1, :, d} \mid Y_{t:t+1, :, d}, \theta)]$$

$$(\mathcal{Q} = \mathcal{Q}_{\mathcal{N}}) \vee (\mathcal{Q} = \mathcal{Q}_{\Pi \mathcal{N}}) \quad (4.36)$$

which is in turn approximated by substituting the true latent posterior with an approximation  $\tilde{P}(X_{t:t+1, d} \mid \theta)$  based on the surrogate marginal  $\tilde{P}(X_{t, :, d})$ :

$$P(X_{t:t+1, :, d} \mid Y_{t:t+1, :, d}, \hat{\theta}^{(k)}) \stackrel{\text{approx.}}{\infty} \tilde{P}(X_{t, :, d}) \mathcal{N}(X_{t+1, :, d} \mid \hat{A}_t^{(k)} X_{t, :, d} + \hat{\mathbf{b}}_t^{(k)}, \hat{\mathbf{V}}_t^{(k)})$$

$$\times \prod_{\tau=t}^{t+1} \prod_{s=1}^S \mathcal{PO}(Y_{\tau, s, d} \mid \exp(X_{\tau, s, d}))$$

$$= \tilde{P}(X_{t:t+1, d} \mid \theta) \quad (4.37)$$

Variational approximations use these closest Gaussians directly as approximating distributions. Stochastic approximations use them as importance distributions.

#### 4.4.1.1 Closest Gaussians

The procedure for finding the closest Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi \mathcal{N}}$  to the latent posterior will now be described. It should be emphasised that by using  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi \mathcal{N}}$  as classes of tractable distributions, rather than product distributions as in chapter 3, the procedure for finding the optimal variational distributions changes substantially.

When product distributions are used as classes  $\mathcal{Q}$ , the optimal  $q \in \mathcal{Q}$ :

$$q^* = \arg \min_{q \in \mathcal{Q}} \text{KL} [q \parallel p] \quad (4.38)$$

is found by iteratively taking expectations of the total log composite likelihood with respect to each factor. When  $\mathcal{Q}$  is a parametric class of distributions, the KL divergence is minimised with respect to the parameters directly.

In the current context, the parameters corresponding to the optimal Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi \mathcal{N}}$  cannot be found analytically. A gradient based numerical minimiser is used, which takes the KL divergence (as a function of the variational Gaussian

parameters) and its first and second partial derivatives as inputs. The parameters that numerically minimise the KL divergence are returned.

The following derivation is therefore not analytical but rather numerical. The procedure description develops a re-parametrisation of the Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$  that accepts unbounded parameter values. This re-parametrisation allows derivatives to be taken analytically. The derivatives themselves are listed in appendices 4.A - 4.D.

Taking the minimisation over  $\mathcal{Q}_{\mathcal{N}}$  first, the KL divergence from  $q \in \mathcal{Q}_{\mathcal{N}}$  to the latent posterior takes the form:

$$\text{KL}[q||p] \approx \mathbb{E}_{\mathcal{N}(\mu_q, \Sigma_q)} \left[ \log \left( \frac{\mathcal{N}(X_{t:t+1, :, d} | \mu_q, \Sigma_q)}{\tilde{\mathcal{P}}(X_{t:t+1, :, d} | \theta)} \right) \right] + \text{constant} \quad (4.39)$$

where  $p = \mathcal{P}(X_{t:t+1, :, d} | Y_{t:t+1, :, d}, \theta)$  is shorthand for purposes of brevity, and

$$\theta = \left( \hat{\mu}_t', \text{vec}(\hat{\Sigma}_t)', \text{vec}(A_t)', \mathbf{b}_t', \text{vec}(V_t)' \right)' \quad (4.40)$$

is the parameter vector containing the surrogate marginal derived in sec 4.3 and the auto-regression parameters for time  $t$ . To ease exposition, these parameters are combined using standard results for Gaussian distributions into a joint mean vector and joint covariance matrix:

$$\begin{aligned} X_{t:t+1, :, d} &\sim \mathcal{N}(X_{t:t+1, :, d} | \hat{\mu}_p, \hat{\Sigma}_p) \\ \hat{\mu}_p &= \begin{pmatrix} \hat{\mu}_t \\ A_t \hat{\mu}_t + \mathbf{b}_t \end{pmatrix} \\ \hat{\Sigma}_p &= \begin{pmatrix} \hat{\Sigma}_t & \hat{\Sigma}_t \cdot A_t' \\ A_t \cdot \hat{\Sigma}_t & V_t + A_t \cdot \hat{\Sigma}_t \cdot A_t' \end{pmatrix} \end{aligned} \quad (4.41)$$

The definition in (4.39) can then be expanded further:

$$\begin{aligned} \text{KL}[q||p] = & -\frac{1}{2} \log(|\Sigma_q|) + \frac{1}{2} \left( \text{trace}(\Sigma_q \cdot \widehat{\Sigma}_p^{-1}) + \mu_q' \widehat{\Sigma}_p^{-1} \mu_q - 2\mu_q' \widehat{\Sigma}_p^{-1} \widehat{\mu}_p \right) \\ & - \mu_q' \text{vec}(Y_{t:t+1, :, d}) + \sum_{\tau=t}^{t+1} \sum_{s=1}^S \exp \left( \mu_{q,(\tau,s)} + \frac{1}{2} \sigma_{q,(\tau,s)}^2 \right) + \text{constant} \end{aligned} \quad (4.42)$$

where the index  $(\tau, s)$  in the exponential terms denotes the elements of  $\mu_q$  and the diagonal of  $\Sigma_q$  corresponding to the dimension of the latent variable associated with  $Y_{\tau,s,d}$ . The notation  $\mu_{q,i}$  where  $i = (\tau - 1)S + s$  is also used in this section when the values of  $\tau, s$  are clear.

At this stage a particular re-parametrisation of  $\Sigma_q$  is employed, which can make the numerical minimisation of (4.42) less cumbersome. The positive definite constraint on  $\Sigma_q$  is generally non-trivial to enforce in a numerical minimiser, but enforcement becomes implicit when expressing  $\Sigma_q$  as a function of the elements of its Cholesky factorisation:

$$\Sigma_q = L_q \cdot L_q^T \quad (4.43)$$

where  $L_q$  is a lower triangular matrix with strictly positive numbers on the diagonal, and the transpose of  $L_q$  is denoted  $L_q^T$  to avoid confusion with the matrix of derivatives  $L'_q = \{f'_{a,b}(l_{a,b})\}$  in appendices 4.A - 4.D.

The positive definite constraint on  $\Sigma_q$  can be enforced simply by substituting  $\Sigma_q$  with  $L_q \cdot L_q^T$  in (4.42). Care must be taken though, to ensure the diagonal of  $L_q$  contains only positive numbers. This can be achieved with the following parametrisation:

$$L_{q,i,j} = \begin{cases} f_{i,j}(l_{i,j}) & j \leq i \\ 0 & j > i \end{cases} \quad (4.44)$$

where  $L_{q,i,j}$  is the  $(i, j)$ <sup>th</sup> element of  $L_q$ , and

$$f_{i,j}(l_{i,j}) = \begin{cases} \exp(l_{i,j}) & i = j \\ l_{i,j} & j < i \end{cases} \quad (4.45)$$

Formulating  $L_q$  as in (4.44) allows numerical minimisation routines to search the space of unbounded parameters  $l_{i,j} \in \mathbb{R}$ . This re-parametrisation gives the alternate formulation of (4.42):

$$\begin{aligned} \text{KL}[q||p] = & c - \sum_{i=1}^{2S} l_{i,i} + \frac{1}{2} \left( \text{trace}(L_q \cdot L_q^T \cdot \widehat{\Sigma}_p^{-1}) + \mu_q' \widehat{\Sigma}_p^{-1} \mu_q - 2\mu_q' \widehat{\Sigma}_p^{-1} \widehat{\mu}_p \right) \\ & - \mu_q' \text{vec}(Y_{t:t+1, :, d}) + \sum_{\tau=t}^{t+1} \sum_{s=1}^S \exp \left( \mu_{q,i} + \frac{1}{2} \sum_{j=1}^i f_{i,j}(l_{i,j})^2 \right) \end{aligned} \quad (4.46)$$

where  $i = (\tau - 1)S + s$  is the index of  $\mu_q$  corresponding to time  $\tau$  and dimension  $s$ .

The numerical minimisation routines used here to minimise (4.46) are gradient based methods that require as inputs the grad vector of first order partial derivatives and the Hessian matrix of second order partial derivatives, along with the function to be minimised. These partial derivatives are listed in appendices 4.A - 4.B.

Plugging the function (4.46) into a gradient based minimiser along with the partial derivatives (4.72), (4.73), (4.76), (4.77), and (4.78) returns the numerical estimate of the closest Gaussian in  $Q_N$  to the latent posterior.

Finding the closest Gaussian in  $Q_{\Pi N}$  to the posterior is an essentially identical process, only with the off-diagonal elements of  $L_q$  set to 0:

$$L_{q,i,j} = \begin{cases} f_{i,j}(l_{i,j}) & i = j \\ 0 & i \neq j \end{cases} \quad (4.47)$$

with non-zero elements  $l_{i,i}$  now only needing to be referenced with one index, i.e.  $l_{i,i} \equiv l_i$ , and

$$f_i(l_i) = \exp(l_i) \quad (4.48)$$

resulting in the simpler equation:

$$\begin{aligned} \text{KL}[q||p] = c + \sum_{i=1}^{2S} l_i + \frac{1}{2} \left( \sum_{i=1}^{2S} \exp(2l_i) \widehat{\Sigma}_{p,i,i}^{-1} + \mu_q' \widehat{\Sigma}_p^{-1} \mu_q - 2\mu_q' \widehat{\Sigma}_p^{-1} \widehat{\mu}_p \right) \\ - \mu_q' \text{vec}(Y_{t:t+1, :, d}) + \sum_{\tau=t}^{t+1} \sum_{s=1}^S \exp \left( \mu_{q,i} + \frac{1}{2} \exp(2l_i) \right) \end{aligned} \quad (4.49)$$

The partial derivatives of (4.49) are listed in appendices 4.C - 4.D. Plugging the function (4.49) into a gradient based minimiser along with the partial derivatives (4.81), (4.82), (4.83), (4.84), and (4.85) returns the numerical estimate of the closest Gaussian in  $\mathcal{Q}_{\Pi\mathcal{N}}$  to the latent posterior. When the dimension of the Gaussians in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$  is small then both the first and second order partial derivatives can be used in Newton's method of optimisation. Computation of the Hessian is relatively slow though. As the latent dimensionality increases, therefore, its calculation will become impractical. In this case, L-BFGS can be used instead of Newton's method, which requires only first order partial derivatives. The procedure for finding the closest Gaussian in  $\mathcal{Q}_{\mathcal{N}}, \mathcal{Q}_{\Pi\mathcal{N}}$  is summarised in algorithm 4.2.

#### 4.4.1.2 Approximating distributions

Parameter estimates made using variational EM follow algorithm 4.3 using the closest Gaussians

$$q_{t,d}^*(X_{t:t+1, :, d}) = \arg \min_{q \in \mathcal{Q}} \text{KL} \left[ q || \mathcal{P}(X_{t:t+1, :, d} | Y_{t:t+1, :, d}, \hat{\theta}^{(k)}) \right] \quad (4.53)$$

to directly calculate the expected sufficient statistics in (4.58). Parameter estimates made using stochastic EM follow algorithm 4.3 using importance distributions to approximate sampling from the posterior:

$$\mathbb{E}_{q_{t,d}^*} [f(X_{t:t+1, :, d})] = \sum_{i=1}^N w_i f \left( X_{t:t+1, :, d}^{(i)} \right), \quad X_{t:t+1, :, d}^{(i)} \stackrel{\text{i.i.d}}{\sim} q_{t,d}^* \quad (4.54)$$



**Algorithm 4.2** Finding the closest Gaussian in  $\mathcal{Q}_{\mathcal{N}}$ ,  $\mathcal{Q}_{\Pi\mathcal{N}}$  to the latent posterior

1. For current parameter estimates  $\hat{\theta}^{(k)}$  and surrogate marginals  $\hat{\mathcal{P}}(X_{t,:},d)$ , construct the parameters of the surrogate pairwise marginals  $\tilde{\mathcal{P}}(X_{t,:},d)\mathcal{N}(X_{t+1,:},d | X_{t,:},d, \hat{\theta}^{(k)})$  as per (4.41):

$$\begin{aligned}\hat{\mu}_p &= \begin{pmatrix} \hat{\mu}_t \\ \hat{A}_t^{(k)} \hat{\mu}_t + \hat{\mathbf{b}}_t^{(k)} \end{pmatrix} \\ \hat{\Sigma}_p &= \begin{pmatrix} \hat{\Sigma}_t & \hat{\Sigma}_t \cdot \hat{A}_t^{(k)'} \\ \hat{A}_t^{(k)} \cdot \hat{\Sigma}_t & \hat{V}_t^{(k)} + \hat{A}_t^{(k)} \cdot \hat{\Sigma}_t \cdot \hat{A}_t^{(k)'} \end{pmatrix}\end{aligned}\quad (4.50)$$

2. Minimise the KL divergence

$$\begin{aligned}\text{KL}[q||p] &= c - \sum_{i=1}^{2S} l_{i,i} + \frac{1}{2} \left( \text{trace}(L_q \cdot L_q' \cdot \hat{\Sigma}_p^{-1}) + \mu_q' \hat{\Sigma}_p^{-1} \mu_q - 2\mu_q' \hat{\Sigma}_p^{-1} \hat{\mu}_p \right) \\ &\quad - \mu_q' \text{vec}(Y_{t:t+1,:},d) + \sum_{\tau=t}^{t+1} \sum_{s=1}^S \exp \left( \mu_{q,i} + \frac{1}{2} \sum_{j=1}^i f_{i,j}(l_{i,j})^2 \right)\end{aligned}\quad (4.51)$$

for  $q \in \mathcal{Q}_{\mathcal{N}}$  and

$$\begin{aligned}\text{KL}[q||p] &= c + \sum_{i=1}^{2S} l_i + \frac{1}{2} \left( \sum_{i=1}^{2S} \exp(2l_i) \hat{\Sigma}_{p,i,i}^{-1} + \mu_q' \hat{\Sigma}_p^{-1} \mu_q - 2\mu_q' \hat{\Sigma}_p^{-1} \hat{\mu}_p \right) \\ &\quad - \mu_q' \text{vec}(Y_{t:t+1,:},d) + \sum_{\tau=t}^{t+1} \sum_{s=1}^S \exp \left( \mu_{q,i} + \frac{1}{2} \exp(2l_i) \right)\end{aligned}\quad (4.52)$$

for  $q \in \mathcal{Q}_{\Pi\mathcal{N}}$  using a gradient based minimiser, for example Newton's method or L-BFGS. For  $q \in \mathcal{Q}_{\mathcal{N}}$  the grad vector is determined by (4.72), (4.73) and the Hessian matrix is determined by (4.76), (4.77), (4.78). For  $q \in \mathcal{Q}_{\Pi\mathcal{N}}$ , the grad vector is determined by (4.81), (4.82), and the Hessian matrix is determined by (4.83), (4.84), and (4.85).

where  $\mathbb{E}_{q_{t,d}^*} [f(X_{t:t+1, :, d})]$  represents one of the expected sufficient statistics in (4.58), and

$$w_i \propto \frac{\tilde{\mathcal{P}}(X_{t:t+1, :, d}^{(i)}) \mathcal{P}\mathcal{O}(Y_{t:t+1, :, d} | X_{t:t+1, :, d}^{(i)}, \hat{\boldsymbol{\theta}}^{(k)})}{\tilde{q}_{t,d}^*(X_{t:t+1, :, d}^{(i)})}, \quad \sum_{i=1}^N w_i = 1 \quad (4.55)$$

and the importance distributions  $\tilde{q}_{t,d}^*$  are the closest Gaussians from  $\mathcal{Q}_N, \mathcal{Q}_{\Pi N}$  to the posterior.

#### 4.4.2 Maximisation step

For each approximation method, once the approximation to the latent posterior has been made, it can be used to take the expectation of the total log composite likelihood and obtain a lower bound to the data log composite likelihood. As noted in sec 4.2.1, using the surrogate marginals  $\tilde{P}(X_{t, :, d})$  breaks the functional dependence of each component  $\log \mathcal{P}(Y_{t:t+1, :, d} | \boldsymbol{\theta})$  on parameters associated to time steps prior to  $t$ . The maximisation step, therefore, depends only on expected sufficient statistics relating to the log transitions. The parameter estimates that maximise the lower bound are:

$$\begin{aligned} \hat{A}_t^+ &= \left( \sum_{d=1}^D \mathbb{E}_{q_{t,d}^*} [X_{t+1, :, d} X_{t, :, d}^+ \prime] \right) \left( \sum_{d=1}^D \mathbb{E}_{q_{t,d}^*} [X_{t, :, d}^+ X_{t, :, d}^+ \prime] \right)^{-1} \\ \hat{V}_t &= \frac{1}{D} \sum_{d=1}^D \left( \mathbb{E}_{q_{t,d}^*} [X_{t+1, :, d} X_{t+1, :, d}^+ \prime] - \hat{A}_t^+ \mathbb{E}_{q_{t,d}^*} [X_{t, :, d}^+ X_{t+1, :, d}^+ \prime] \right. \\ &\quad \left. - \mathbb{E}_{q_{t,d}^*} [X_{t+1, :, d} X_{t, :, d}^+ \prime] \hat{A}_t^{+\prime} + \hat{A}_t^+ \mathbb{E}_{q_{t,d}^*} [X_{t, :, d}^+ X_{t, :, d}^+ \prime] \hat{A}_t^{+\prime} \right) \end{aligned} \quad (4.56)$$

where

$$\begin{aligned} X_{t, :, d}^+ &= \begin{pmatrix} X_{t, :, d} \\ 1 \end{pmatrix} \\ A_t^+ &= (A_t \mathbf{b}_t) \end{aligned} \quad (4.57)$$

are augmented so as to include the intercept vector  $\mathbf{b}_t$  into  $A_t$ . The output required from the expectation step is therefore calculation of the expected sufficient statistics

$$\mathbb{E}_{q_{t,d}^*} \left[ \text{vec} (X_{t:t+1, :, d}) \text{vec} (X_{t:t+1, :, d})' \right] \quad (4.58)$$

The approximate EM procedures are summarised together in algorithm 4.3.

## 4.5 Method of moments parameter estimates

Parameter estimates  $\hat{\theta}^{(0)}$  can also be made via the method of moments. These estimates can be used directly in, for example, prediction or smoothing tasks, or they can be used to initialise approximate EM algorithms as per step 2 of algorithm 4.3.

The method of sec 4.3 is trivially extended to estimate pairwise surrogate marginals  $\tilde{P}(X_{t:t+1, :, d})$ . Parameter estimates can be calculated in closed form from these pairwise marginals. The pairwise surrogates are estimated simply by augmenting the sample moments of sec 4.3 with the cross-time moments

$$\bar{\sigma}_{(t,s_1),(t+1,s_2)}, \quad s_1, s_2 \in 1, \dots, S \quad (4.61)$$

and constructing estimated pairwise covariance matrices

$$\hat{\mathcal{S}}_{t:t+1} = \begin{pmatrix} \hat{\mathcal{S}}_t & \hat{\mathcal{S}}_{t,t+1} \\ \hat{\mathcal{S}}_{t+1,t} & \hat{\mathcal{S}}_{t+1:t+1} \end{pmatrix} \quad (4.62)$$

via the thresholding and rescaling procedure described in sec 4.3.1. The method of

**Algorithm 4.3** Approximate EM

1. Determine the parameters for the surrogate marginals  $\tilde{\mathcal{P}}(X_{t,:},d)$  as per algorithm 4.1.
2. Initialise parameter estimates  $\hat{\theta}^{(0)}$ .
3. Repeat until convergence of  $\hat{\theta}^{(k)}$ :
  - (a) Use the surrogate marginals and current parameter estimates  $\hat{\theta}^{(k)}$  to find the closest Gaussians  $q^* \in \mathcal{Q}$  to the latent posteriors for either  $\mathcal{Q} = \mathcal{Q}_N$  or  $\mathcal{Q} = \mathcal{Q}_{\Pi N}$  as per algorithm 4.2.
  - (b) Evaluate the expected sufficient statistics (4.58):

$$\mathbb{E}_{q_{t,d}^*} \left[ \begin{array}{c} \mathbb{E}_{q_{t,d}^*} [X_{t:t+1,:},d] \\ \text{vec} (X_{t:t+1,:},d) \text{vec} (X_{t:t+1,:},d)' \end{array} \right] \quad (4.59)$$

for the lower bound on the data log composite likelihood. Expectations are taken with respect to the particular approximate distribution being employed:

- Variational EM: Closest Gaussians found using algorithm 4.2 are used directly.
  - Stochastic EM: Closest Gaussians found using algorithm 4.2 are used as importance distributions to approximate the posterior.
- (c) Maximise the lower bound with respect to the parameters as per (4.56):

$$\begin{aligned} \hat{A}_t^+ &= \left( \sum_{d=1}^D \mathbb{E}_{q_{t,d}^*} [X_{t+1,:},d X_{t,:},d^+]' \right) \left( \sum_{d=1}^D \mathbb{E}_{q_{t,d}^*} [X_{t,:},d^+ X_{t+1,:},d]' \right)^{-1} \\ \hat{V}_t &= \frac{1}{D} \sum_{d=1}^D \left( \mathbb{E}_{q_{t,d}^*} [X_{t+1,:},d X_{t+1,:},d]' - \hat{A}_t^+ \mathbb{E}_{q_{t,d}^*} [X_{t,:},d^+ X_{t+1,:},d] \right. \\ &\quad \left. - \mathbb{E}_{q_{t,d}^*} [X_{t+1,:},d X_{t,:},d^+]' \hat{A}_t^+ + \hat{A}_t^+ \mathbb{E}_{q_{t,d}^*} [X_{t,:},d^+ X_{t,:},d^+]' \hat{A}_t^+ \right) \end{aligned} \quad (4.60)$$

moments parameter estimates are made as:

$$\begin{aligned}
\mathcal{P}(X_{t:t+1},:,d) &= \mathcal{N}(X_{t:t+1},:,d \mid \widehat{\boldsymbol{\mu}}_{t:t+1}, \widehat{\boldsymbol{\Sigma}}_{t:t+1}) \\
\Rightarrow \widehat{A}_t &= \widehat{\boldsymbol{\Sigma}}_{t+1,t} \cdot \widehat{\boldsymbol{\Sigma}}_t^{-1} \\
\widehat{\mathbf{b}}_t &= \widehat{\boldsymbol{\mu}}_{t+1} - \widehat{A}_t \widehat{\boldsymbol{\mu}}_t \\
\widehat{V}_t &= \widehat{\boldsymbol{\Sigma}}_{t+1} - \widehat{\boldsymbol{\Sigma}}_{t+1,t} \cdot \widehat{\boldsymbol{\Sigma}}_t^{-1} \cdot \widehat{\boldsymbol{\Sigma}}_{t,t+1}
\end{aligned} \tag{4.63}$$

## 4.6 Experiments

Synthetic data drawn as per sec 4.6.1 is used to investigate the results of fitting parameters via each approximation method described in algorithm 4.3. The trade-offs between compute time and the statistical qualities of the estimators are examined. The dimension  $S$  of the data varies across synthetic data sets to observe its effect on the trade-offs.

For each dataset, parameters fitted by each method are compared and the differences

$$\begin{aligned}
\delta_{i,j} &= \|\widehat{\boldsymbol{\theta}}^i - \widehat{\boldsymbol{\theta}}^j\| \\
i, j &\in \{\text{VEM}(Q_{\mathcal{N}}), \text{VEM}(Q_{\Pi\mathcal{N}}), \text{SEM}(Q_{\mathcal{N}}), \text{SEM}(Q_{\Pi\mathcal{N}}), \text{MM}\}
\end{aligned} \tag{4.64}$$

are computed, where VEM, SEM indicate approximate EM method,  $Q_{\mathcal{N}}, Q_{\Pi\mathcal{N}}$  indicate the class of Gaussians underlying the approximations, and MM indicates parameters estimated directly via the method of moments estimates of the pairwise marginals  $\mathcal{P}(X_{t:t+1})$  as per sec 4.5. The time taken to compute each estimate is also recorded.

After all parameters have been fitted they are used in smoothing and prediction tests to compare their utility. Comparable experiments to the smoothing and predictions experiments in chapter 4 are performed. Particle methods are used to produce approximate smoothed distributions and predictions for each set of parameter estimates. In-sample estimates are made for 5 different realisations of data that were part of the model fitting dataset. Out-of-sample estimates are made for 5 dif-

ferent realisations of data that were not part of the model fitting dataset. The closest Gaussians  $q^* \in Q_N$  are used as proposal distributions for the particle algorithms.

Once particle approximations to smoothed distributions have been made, the loss functions

$$L^{(1),j} = \sqrt{\frac{1}{TSD} \sum_{t=1}^T \sum_{s=1}^S \sum_{d=1}^D \left( X_{t,s,d} - \bar{X}_{t,s,d}^j \right)^2}$$

$$j \in \{\text{VEM}(Q_N), \text{VEM}(Q_{\Pi N}), \text{SEM}(Q_N), \text{SEM}(Q_{\Pi N}), \text{MM}\} \quad (4.65)$$

where  $\bar{X}_{t,s,d}^j$  are the means of the smoothing estimates for each dimension  $s$  of the latent space at each time  $t$  and replicate  $d$ :

$$\bar{X}_{t,s,d}^j = \frac{1}{N} \sum_{i=1}^N \hat{X}_{t,s,d}^{(i),j} \quad (4.66)$$

and

$$L^{(2),j} = \sqrt{\frac{1}{NTSD} \sum_{i=1}^N \sum_{t=1}^T \sum_{s=1}^S \sum_{d=1}^D \left( X_{t,s,d} - \hat{X}_{t,s,d}^{(i),j} \right)^2} \quad (4.67)$$

with  $j$  as in (4.65) will be computed. As with the corresponding loss functions in chapter 3,  $L^{(1),j}$  measures the accuracy of particle means and  $L^{(2),j}$  measures the accuracy of individual particles. The difference  $L^{(2),j} - L^{(1),j}$  measures the spread of particles in the smoothing estimates.

One step ahead prediction using particle filters is performed, with the corresponding loss functions to (4.65):

$$L^{(1),j} = \sqrt{\frac{1}{(T-1)SD} \sum_{t=2}^T \sum_{s=1}^S \sum_{d=1}^D \left( Y_{t,s,d} - \bar{Y}_{t,s,d}^j \right)^2} \quad (4.68)$$

where  $\bar{Y}_{t,s,d}^j$  are the means of the predictions for each dimension  $s$  and replicate  $d$  at time  $t$ , with the mean taken with respect to importance weights  $w_i^{t-1}$  for the filtered

particles at time  $t - 1$ :

$$\bar{Y}_{t,s,d}^j = \sum_{i=1}^N w_i^{t-1} \hat{Y}_{t,s,d}^j \quad (4.69)$$

and to (4.67):

$$L^{(2),j} = \sqrt{\frac{1}{(T-1)SD} \sum_{i=1}^N \sum_{t=2}^T \sum_{s=1}^S \sum_{d=1}^D w_i^{t-1} \left( Y_{t,s,d} - \hat{Y}_{t,s,d}^{(i),j} \right)^2} \quad (4.70)$$

being computed, both with  $j$  as in (4.65).

All experiments are written and performed in MATLAB.

### 4.6.1 Synthetic data

Synthetic data are drawn using randomly generated parameters and generative models (4.4) with  $T = 10$ ,  $D = 200$ , and  $\dim \mathcal{X} = \dim \mathcal{Y} = S \in 5, 10, 15, 20$ . One draw of data is sampled for each value of  $S$ . For each value of  $S$ , each transition matrix  $A_t$  is generated by first randomly selecting  $S$  eigenvalues  $\lambda_i \sim \text{i.i.d.} \mathcal{U}(-1.5, 1.5)$ . These eigenvalues then replace the eigenvalues of a random positive definite matrix:

$$\begin{aligned} \tilde{A}_t &\sim \mathcal{W}(I_S, S) \\ \tilde{A}_t &= V \cdot \text{diag}(\{\varepsilon_i\}_{i=1}^S) \cdot V' \\ A_t &= V \cdot \text{diag}(\{\lambda_i\}_{i=1}^S) \cdot V' \end{aligned} \quad (4.71)$$

where  $\mathcal{W}(V, \nu)$  is the Wishart distribution with scale matrix  $V$  and  $\nu$  degrees of freedom,  $I_S$  is the  $S \times S$  identity matrix, and  $V \cdot \text{diag}(\{\varepsilon_i\}_{i=1}^S) \cdot V'$  is the eigen-decomposition of  $\tilde{A}_t$ . By selecting  $\lambda_i \sim \text{i.i.d.} \mathcal{U}(-1.5, 1.5)$ , the non-stationarity of  $Y$  is emphasised; fitting a model to  $Y$  that assumed stationarity would not produce a good fit. The benefits of approximating the latent marginals  $\mathcal{P}(X_{t,:}, d)$  with their surrogates are thus brought into focus.

The latent intercept vectors  $\mathbf{m}_t$  are randomly drawn  $\mathbf{m}_t \sim \text{i.i.d.} \mathcal{N}(0, 0.5I_S)$ . The conditional covariance matrices are randomly drawn  $V_t \sim \text{i.i.d.} \mathcal{W}(0.2I_S, S)$ .

## 4.7 Results

The times taken to calculate each set of approximate estimators is listed in table 4.1. Finding the closest general Gaussians in  $Q_N$  has quadratic cost in  $S$ , which is borne out in the rapidly increasing compute times for the  $VEM(Q_N)$  and  $SEM(Q_N)$  estimators. For the dataset where  $S = 5$ , the cost of drawing the samples and calculating importance weights can be seen to be a significant portion of the cost of computing the SEM estimators, as they take a notable amount more time to run than their variational equivalents.

As  $S$  increases though, the cost of finding the closest Gaussians becomes the dominant cost in computation. For the general Gaussian this effect is almost immediate; compute times for  $VEM(Q_N)$  and  $SEM(Q_N)$  estimators are very similar for the datasets with  $S = 10, 15, 20$ . The  $SEM(Q_N)$  estimator actually takes less time to compute for  $S = 20$  than the  $VEM(Q_N)$  estimator does. This could possibly be explained by more accurate parameter updates at each EM iteration. The method of moments estimators take a negligible amount of time to compute.

**Table 4.1:** Times taken to compute each set of parameter estimates, for datasets with latent dimension  $S = 5, 10, 15, 20$ . Times are in min:sec, except for the last column, for which times are in seconds.

| S  | VEM    |             | SEM    |             | MM     |
|----|--------|-------------|--------|-------------|--------|
|    | $Q_N$  | $Q_{\Pi N}$ | $Q_N$  | $Q_{\Pi N}$ |        |
| 5  | 30:51  | 16:40       | 65:42  | 28:00       | 0.002s |
| 10 | 90:01  | 29:21       | 92:39  | 38:16       | 0.004s |
| 15 | 104:19 | 33:40       | 107:46 | 36:18       | 0.006s |
| 20 | 244:06 | 46:57       | 201:30 | 45:11       | 0.008s |

The differences in estimators resulting from using different approximation methods are shown in table 4.2. When the latent dimensionality is low, at  $S = 5$ , using factorised Gaussians as variational approximations has little effect on parameter estimates. This does not hold for VEM estimators as the latent dimensionality increases though. For all other tested dimensions  $S = 10, 15, 20$ ,  $VEM(Q_{\Pi N})$  estimators differ significantly from the other likelihood estimators. Both  $VEM(Q_N)$  and  $SEM(Q_{\Pi N})$  estimators diverge from  $SEM(Q_N)$  estimators as  $S$  increases, though



at different rates. For  $VEM(Q_N)$  estimators the rate is significantly faster than for  $SEM(Q_{\Pi N})$  estimators. The method of moments estimators differ significantly from all likelihood estimators for all values of  $S$ .

The results of the smoothing experiment are shown in table 4.3. The most immediate observation to be made is the poor quality of smoothed distributions approximated using method of moments parameter estimates. In all cases, these smoothing estimates are of a lower quality than all other estimates. Secondly, smoothing estimates made using the true parameters perform worse than all likelihood based estimators at all sizes of the latent dimension  $S$ . Both the method of moments and the true parameters smoothing estimates consistently have either a negligible spread of particles, or no spread at all.

Smoothed distributions made using other parameter estimates are not significantly different in quality to each other. Some patterns can be observed, across both estimators and latent dimension sizes. In general, there is a pattern of decreasing quality with increasing latent dimension  $S$ . This observation holds for true parameter smoothing estimates too, but not those from the method of moments. For the case when  $S = 5$ , smoothing estimates made using the  $VEM(Q_{\Pi N})$  estimator perform slightly better than the other likelihood based estimators, but this pattern is reversed for all other dimension sizes.

The other three likelihood based estimators perform very similarly at all dimension sizes. The  $VEM(Q_N)$  estimates are generally of a slightly worse quality, but the difference is not significant. For dimension sizes  $S = 5$  and  $S = 10$ ,  $SEM(Q_N)$  estimates are slightly better than those for  $SEM(Q_{\Pi N})$ . For dimension sizes  $S = 15$  and  $S = 20$  though, they are slightly worse.

There is a fairly consistent difference in the quality of in-sample and out-of-sample smoothed distributions. For dimension sizes  $S = 5$ ,  $S = 10$ , and  $S = 15$ , out-of-sample smoothing estimates have a lower  $L1$  and  $L2$  loss for all likelihood based estimators. For  $S = 20$ , the difference in performance is negligible for each estimator.

The results of the prediction experiment are shown in table 4.4. As with the

smoothing experiment, predictions made using method of moments estimators are worse than all others. In this experiment though, they are not merely poor but pathologically bad. Also in line with the results of the smoothing experiment, the true parameter predictions are worse than the predictions of nearly all likelihood based estimators. The particle spreads of true parameter predictions are similar to those for the likelihood based estimators, but the  $L1$  losses are much worse.

There is a significant exception to the above observation:  $VEM(Q_{\Pi N})$  predictions. Predictions made using these estimators do not have a consistent trend across latent dimension sizes. For  $S = 5$ , they are comparable to the other likelihood based estimators;  $L1$  losses are better than for all other estimators, and  $L2$  losses are better than  $VEM(Q_N)$  for both in-sample and out-of-sample data. For  $S = 10$  and  $S = 15$ , both loss functions show pathologically bad performance. In these cases, the performance is comparable to the method of moments predictions. For  $S = 20$ , the picture is less clear. Performance is still much worse than the other likelihood based estimators, but only the  $L2$  loss for in-sample data is pathologically bad.

The predictions made using the other likelihood based estimators are generally similar to each other. Differences appear as the latent dimension increases, but there does not appear to be any observable pattern to them. For in-sample data with  $S = 15$ , both of the SEM estimators have higher  $L1$  and  $L2$  loss than  $VEM(Q_N)$ , and larger particle spreads too. The performances on out-of-sample data are all similar to each other though. For  $S = 20$ ,  $L1$  losses are similar for all three estimators, but  $L2$  losses are much higher for  $VEM(Q_N)$ . This holds for both in-sample and out-of-sample data.

Predictions for the two SEM estimators are almost indistinguishable from each other in performance. Both  $L1$  and  $L2$  losses get worse with increasing  $S$  for both estimators, but for  $SEM(Q_{\Pi N})$  it is at a slower rate than for  $SEM(Q_N)$ .

The true parameter predictions are much better for  $S = 20$  than for the other latent dimension sizes. In this case, in contrast to all other dimension sizes  $S$ , they are similar to the SEM estimators. The  $L1$  losses for the true parameters are a bit higher and the  $L2$  losses are lower, but both loss functions are broadly in line with

the SEM estimators.

## 4.8 Discussion

The conclusions that can be drawn from the experiments in the current chapter seem to be quite clear: *a)* the method of moments parameter estimates have practically no utility, *b)* the efficacy of using Gaussian proposals in particle filters is highly sensitive to parameter settings, *c)* VEM estimators do not perform well when the tractable class of approximating distributions is  $Q_{\Pi\mathcal{N}}$ , *d)* the performance of SEM estimators is not significantly affected by the choice of  $Q$ , and *e)* using surrogate marginals to approximate latent marginals at each time does introduce significant errors.

The effect of increasing the latent dimension  $S$  can be seen in all experiments. The accuracy of smoothed distributions and predictions goes down, but not uniformly across estimators. It is seen much more dramatically in the VEM estimators, particularly the VEM  $Q_{\Pi\mathcal{N}}$ . The performance of SEM estimators are similar to each other in all experiments, and are increasingly better than both VEM estimators as  $S$  increases. The benefits of stochastically approximating expectations become increasingly clear to see as  $S$  increases.

The poor performance of the VEM( $Q_{\Pi\mathcal{N}}$ ) estimators in the smoothing and prediction experiments for all latent dimensions other than  $S = 5$  is quite informative. VEM( $Q_{\mathcal{N}}$ ) estimators have quite similar performance to the SEM estimators, so variational approximations *per se* are not necessarily a poor choice for approximating expectations. Additionally restricting the class of tractable distributions to be factorised Gaussians seems to have critically negative consequences.

When approximate expectations are stochastic though, the factorised Gaussian restriction does not seem to be significant. The performance of SEM( $Q_{\mathcal{N}}$ ) and SEM( $Q_{\Pi\mathcal{N}}$ ) estimators are similar in both experiments. The use of importance sampling with either class of Gaussians as proposal distributions must mitigate the effect of which class the proposals belong to.

Another clear result of increasing the latent dimension was the increasing dis-

parity between compute times for the approximations. Finding the closest Gaussian in  $Q_N$  to the posteriors has a cost that is quadratic in  $S$ . As  $SEM(Q_N)$  do not perform better than  $SEM(Q_{\Pi N})$  estimators the experiments, there is no evidence that justifies such a computational cost.

A more surprising result also became apparent through the experiments. The performance of both the method of moments parameters and the true parameters show how sensitive the particle filter is to parameter settings. In the smoothing experiment, both these parameters had negligible spreads to their particles and incurred losses greater than for the likelihood based estimators. In the prediction experiment, their performances were significantly worse than for all other estimators excluding  $VEM(Q_{\Pi N})$ . These results are particularly surprising for the true parameters, as it is reasonable to expect their performance to at least equal that of any estimator.

The pathologically bad performance of the method of moments and  $VEM(Q_{\Pi N})$  estimators in the prediction experiment also deserves mention. Such large losses as shown by these estimators in table 4.4 suggest that the particle filter can only make reasonable predictions if parameters lie in some sensible range. As shown in the bias experiment, the  $VEM(Q_{\Pi N})$  estimators were generally at least as far from the other likelihood based estimators as the method of moments estimators were. This distance from the other estimators is very likely to be the cause of the pathologically bad predictions.

Such pathological predictions, and the under-performance of the true parameters, suggest that the particle filter is not producing high quality approximations to posteriors. Using Gaussians as proposal distributions is the likely cause, which possibly leads to an interesting consequence. The likelihood based estimators (with the exception of  $VEM(Q_{\Pi N})$ ) consistently outperformed the true parameters at both smoothing and prediction, and also did not produce any pathological output. When plugged into the particle filter these estimators are more effective at reproducing features of the data, both in-sample and out-of-sample, than the true parameters. Perhaps the parameters are implicitly learning to generate effective Gaussian ap-

proximations to posteriors, rather than learning the data generating mechanism. This conclusion appears to hold for the SEM parameters as well as  $VEM(Q_N)$ . Even though importance sampling approximates the true posterior, if the Gaussian proposals are inadequate then there seems to be a limit to the benefit of approximating expectations stochastically rather than variationally.

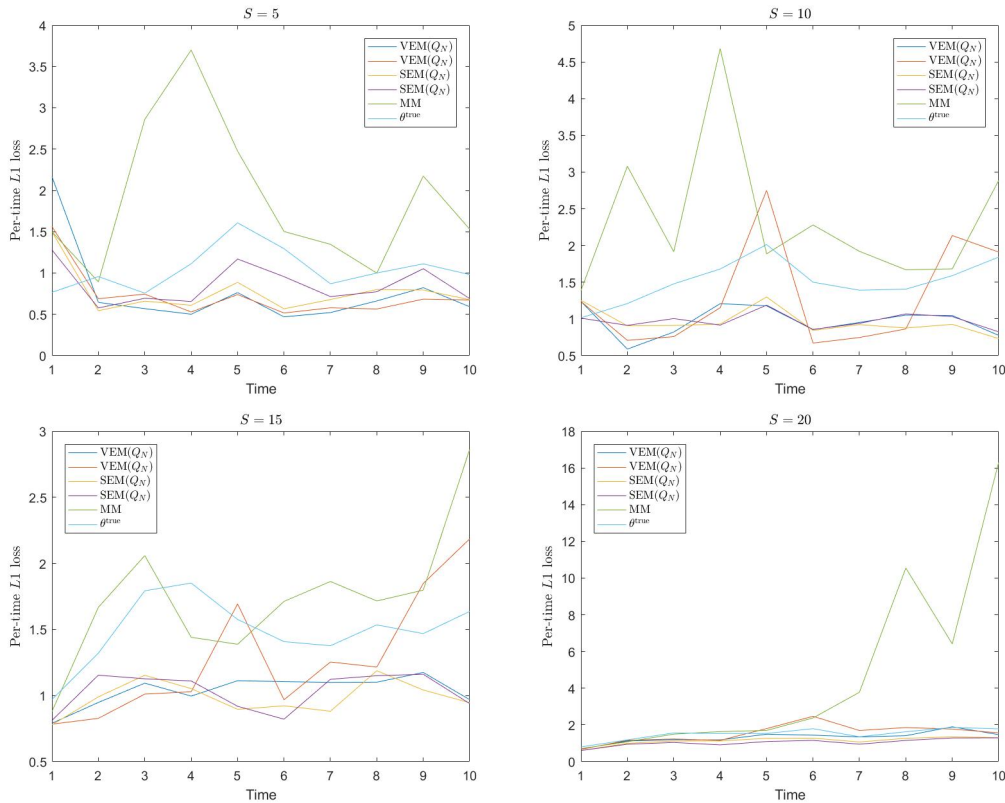
If this speculation is true, then using  $Q_N$  and  $Q_{\Pi N}$  as tractable classes cannot be considered to be an effective choice. Having made the choice, however, using importance sampling to approximate expectations stochastically appears to be more effective than using the variational approximations directly. As this holds for  $SEM(Q_{\Pi N})$  when compared to  $VEM(Q_N)$ , and considering the relative costs of finding closest Gaussians in  $Q_N$  and  $Q_{\Pi N}$ , a clear recommendation can be made. If Gaussians are to be used as variational approximations to posteriors then using factorised Gaussians, and then using importance sampling to estimate expectations, optimises the trade-offs between computational cost and statistical efficacy.

In addition to the analysis above, the utility of using the surrogate marginals in the parameter estimation algorithm can be indirectly assessed. If the use of surrogate marginals introduced a significant bias to estimators, then it should be expected that the effect of the biases would accumulate over time. Propagating particles in a particle filter, for example, should compound the bias effects from the parameters at each time point. The losses in the smoothing and prediction experiments would consequently be expected to increase over time. By examining the accuracy of smoothing estimates and/or predictions over time, any cumulative effects from the use of surrogate marginals would be observable.

Fig 4.1 shows the per-time  $L1$  loss for each of the smoothing estimates at each latent dimension size  $S$ . As described above, these plots should reveal any significant biases introduced via the use of surrogate marginals. If the biases were significant, it would be expected that propagating particles according to these parameters would introduce errors that accumulated, i.e. that the accuracy of smoothing estimates would decrease with time.

The plots in fig 4.1 show that this is not the case. There is only one instance

**Figure 4.1:** Plots showing the per-time  $L_1$  loss for in-sample smoothing estimates for each set of parameter estimates at each latent dimension size  $S = 5$  (top left),  $S = 10$  (top right),  $S = 15$  (bottom left), and  $S = 20$  (bottom right).



where losses increase over time, and that is for the method of moments estimator at  $S = 20$ . These estimators use the (pairwise) marginals implied by the method of moments to estimate parameters, so this result suggests larger datasets could improve the efficacy of surrogate marginals derived in this manner. None of the likelihood based estimators show any evidence of decreasing accuracy over time though. Similar plots for the prediction experiment are not presented here as the pathological results for some estimators meant that not all plots could show all results. There is a similarly positive result in regards to the use of surrogate marginals though. These experiments therefore support the recommendation of their use in the contexts of smoothing and prediction.

**Table 4.2:** Differences in estimators across approximation methods, for each latent dimension  $\dim \mathcal{X} = \dim \mathcal{Y} = S$ .

| $S$ | Method | VEM         |             | SEM     |             | MM     |         |
|-----|--------|-------------|-------------|---------|-------------|--------|---------|
|     |        | $Q_N$       | $Q_{\Pi N}$ | $Q_N$   | $Q_{\Pi N}$ |        |         |
| 5   | VEM    | $Q_N$       | 0           | 0.1206  | 2.1261      | 2.2837 | 22.3368 |
|     |        | $Q_{\Pi N}$ | 0.1206      | 0       | 2.0943      | 2.2472 | 22.3053 |
|     | SEM    | $Q_N$       | 2.1261      | 2.0943  | 0           | 1.0650 | 22.1644 |
|     |        | $Q_{\Pi N}$ | 2.2837      | 2.2472  | 1.0650      | 0      | 21.9792 |
|     | MM     | 22.3368     | 22.3053     | 22.1644 | 21.9792     | 0      |         |
| 10  | VEM    | $Q_N$       | 0           | 57.166  | 6.577       | 6.763  | 32.497  |
|     |        | $Q_{\Pi N}$ | 57.166      | 0       | 58.169      | 58.115 | 67.104  |
|     | SEM    | $Q_N$       | 6.577       | 58.169  | 0           | 3.097  | 31.927  |
|     |        | $Q_{\Pi N}$ | 6.763       | 58.115  | 3.097       | 0      | 31.887  |
|     | MM     | 32.497      | 67.104      | 31.927  | 31.887      | 0      |         |
| 15  | VEM    | $Q_N$       | 0           | 41.569  | 10.922      | 11.959 | 32.469  |
|     |        | $Q_{\Pi N}$ | 41.569      | 0       | 41.070      | 41.282 | 52.277  |
|     | SEM    | $Q_N$       | 10.922      | 41.070  | 0           | 6.593  | 30.529  |
|     |        | $Q_{\Pi N}$ | 11.959      | 41.282  | 6.593       | 0      | 29.987  |
|     | MM     | 32.469      | 52.277      | 30.529  | 29.987      | 0      |         |
| 20  | VEM    | $Q_N$       | 0           | 34.904  | 24.397      | 24.331 | 45.922  |
|     |        | $Q_{\Pi N}$ | 34.904      | 0       | 37.176      | 35.509 | 54.136  |
|     | SEM    | $Q_N$       | 24.397      | 37.176  | 0           | 15.197 | 40.145  |
|     |        | $Q_{\Pi N}$ | 24.331      | 35.509  | 15.197      | 0      | 40.204  |
|     | MM     | 45.922      | 54.136      | 40.145  | 40.204      | 0      |         |

**Table 4.3:** Root mean squared error (RMSE) of smoothing estimates made using estimated parameters and true parameters. Rows labelled *L1* use the first loss function  $L^{(1),j}$  described in sec 4.6, equation (4.65). Rows labelled *L2* use the second loss function described in sec 4.6, equation (4.67). Rows labelled Diff show the difference  $L^{(2),j} - L^{(1),j}$  between the two loss functions for each estimator  $\hat{\theta}^j$  and dataset (In/Out). The RMSE of smoothing estimates made using the true parameters are shown in the right-most column for reference.

| <i>S</i> | <i>Y</i> | <i>L</i>  | VEM   |             | SEM   |             | MM    | $\theta^{\text{true}}$ |
|----------|----------|-----------|-------|-------------|-------|-------------|-------|------------------------|
|          |          |           | $Q_N$ | $Q_{\Pi N}$ | $Q_N$ | $Q_{\Pi N}$ |       |                        |
| 5        | In       | <i>L1</i> | 0.907 | 0.784       | 0.816 | 0.887       | 2.078 | 1.074                  |
|          |          | <i>L2</i> | 1.057 | 0.952       | 0.979 | 1.010       | 2.078 | 1.078                  |
|          |          | Diff      | 0.150 | 0.168       | 0.164 | 0.123       | 0.000 | 0.004                  |
|          | Out      | <i>L1</i> | 0.632 | 0.625       | 0.726 | 0.865       | 2.605 | 1.177                  |
|          |          | <i>L2</i> | 0.841 | 0.852       | 0.903 | 1.012       | 2.607 | 1.177                  |
|          |          | Diff      | 0.209 | 0.226       | 0.178 | 0.145       | 0.001 | 0.000                  |
| 10       | In       | <i>L1</i> | 0.992 | 1.465       | 0.974 | 0.981       | 2.518 | 1.537                  |
|          |          | <i>L2</i> | 1.090 | 1.535       | 1.079 | 1.071       | 2.518 | 1.537                  |
|          |          | Diff      | 0.098 | 0.070       | 0.105 | 0.090       | 0.000 | 0.000                  |
|          | Out      | <i>L1</i> | 0.967 | 1.335       | 0.843 | 0.886       | 2.548 | 1.428                  |
|          |          | <i>L2</i> | 1.064 | 1.408       | 0.945 | 0.998       | 2.548 | 1.428                  |
|          |          | Diff      | 0.096 | 0.073       | 0.102 | 0.112       | 0.000 | 0.000                  |
| 15       | In       | <i>L1</i> | 1.045 | 1.357       | 0.992 | 1.041       | 1.805 | 1.513                  |
|          |          | <i>L2</i> | 1.086 | 1.401       | 1.049 | 1.102       | 1.805 | 1.513                  |
|          |          | Diff      | 0.041 | 0.044       | 0.057 | 0.062       | 0.001 | 0.000                  |
|          | Out      | <i>L1</i> | 0.882 | 1.151       | 0.964 | 0.884       | 2.194 | 1.472                  |
|          |          | <i>L2</i> | 0.923 | 1.215       | 1.028 | 0.963       | 2.194 | 1.472                  |
|          |          | Diff      | 0.042 | 0.063       | 0.064 | 0.079       | 0.000 | 0.000                  |
| 20       | In       | <i>L1</i> | 1.359 | 1.611       | 1.159 | 1.064       | 6.692 | 1.535                  |
|          |          | <i>L2</i> | 1.376 | 1.623       | 1.180 | 1.120       | 6.692 | 1.535                  |
|          |          | Diff      | 0.017 | 0.013       | 0.021 | 0.057       | 0.000 | 0.000                  |
|          | Out      | <i>L1</i> | 1.484 | 1.715       | 1.244 | 1.104       | 2.014 | 1.450                  |
|          |          | <i>L2</i> | 1.503 | 1.728       | 1.255 | 1.144       | 2.015 | 1.450                  |
|          |          | Diff      | 0.019 | 0.014       | 0.011 | 0.039       | 0.002 | 0.000                  |



**Table 4.4:** Root mean squared error (RMSE) of 1-step-ahead predictions made using estimated parameters and true parameters. Rows labelled  $L1$  use the first loss function  $L^{(1),j}$  described in sec 4.6, equation (4.68). Rows labelled  $L2$  use the second loss function described in sec 4.6, equation (4.70). Rows labelled Diff show the difference  $L^{(2),j} - L^{(1),j}$  between the two loss functions for each estimator  $\hat{\theta}^j$  and dataset (In/Out). The RMSE of predictions made using the true parameters are shown in the right-most column for reference.

| $S$ | $Y$ | $L$  | VEM     |             | SEM    |             | MM   | $\theta^{\text{true}}$ |
|-----|-----|------|---------|-------------|--------|-------------|------|------------------------|
|     |     |      | $Q_N$   | $Q_{\Pi N}$ | $Q_N$  | $Q_{\Pi N}$ |      |                        |
| 5   | In  | L1   | 2.220   | 1.760       | 1.916  | 1.935       | 1e18 | 66.532                 |
|     |     | L2   | 7.247   | 6.108       | 5.391  | 5.553       | 1e18 | 71.500                 |
|     |     | Diff | 5.026   | 4.348       | 3.475  | 3.618       | 8e16 | 4.968                  |
|     | Out | L1   | 2.780   | 2.729       | 3.096  | 2.823       | 6e5  | 61.529                 |
|     |     | L2   | 7.067   | 7.004       | 7.097  | 6.534       | 1e6  | 66.243                 |
|     |     | Diff | 4.287   | 4.274       | 4.001  | 3.711       | 6e5  | 4.714                  |
| 10  | In  | L1   | 3.200   | 2e3         | 2.805  | 3.101       | 9e22 | 22.506                 |
|     |     | L2   | 8.295   | 3e5         | 8.438  | 7.897       | 1e23 | 25.383                 |
|     |     | Diff | 5.096   | 3e5         | 5.633  | 4.796       | 1e22 | 2.877                  |
|     | Out | L1   | 2.809   | 3e4         | 2.790  | 2.730       | 3e13 | 18.513                 |
|     |     | L2   | 8.188   | 9e6         | 7.900  | 6.972       | 4e13 | 21.426                 |
|     |     | Diff | 5.380   | 9e6         | 5.110  | 4.242       | 4e12 | 2.913                  |
| 15  | In  | L1   | 5.707   | 7e4         | 7.668  | 6.631       | 3e6  | 51.007                 |
|     |     | L2   | 13.175  | 4e7         | 17.074 | 14.927      | 3e6  | 60.736                 |
|     |     | Diff | 7.469   | 4e7         | 9.406  | 8.296       | 3e5  | 9.729                  |
|     | Out | L1   | 2.661   | 3e6         | 2.790  | 3.092       | 3e7  | 202.185                |
|     |     | L2   | 10.188  | 1e9         | 9.901  | 10.895      | 3e7  | 242.841                |
|     |     | Diff | 7.527   | 1e9         | 7.112  | 7.803       | 2e6  | 40.656                 |
| 20  | In  | L1   | 21.920  | 58.896      | 20.091 | 18.839      | 1e48 | 24.869                 |
|     |     | L2   | 384.037 | 4e3         | 36.574 | 33.200      | 1e48 | 28.217                 |
|     |     | Diff | 362.118 | 4e3         | 16.483 | 14.361      | 2e47 | 3.348                  |
|     | Out | L1   | 39.962  | 47.227      | 35.100 | 31.085      | 3e13 | 39.420                 |
|     |     | L2   | 144.999 | 879.308     | 49.956 | 43.115      | 4e13 | 42.213                 |
|     |     | Diff | 105.037 | 832.081     | 14.856 | 12.030      | 5e12 | 2.793                  |



# Appendix

## 4.A Grad vector for KL divergence from $q \in Q_{\mathcal{N}}$

The entries of the grad vector of first order partial derivatives of the KL divergence (4.46) has entries are listed according to the parameters within  $\theta$  to which they are associated. First, the case  $\partial/\partial\theta_i$  when  $\theta_i = \mu_{q,a}$ :

$$\frac{\partial \text{KL}[q||p]}{\partial \mu_{q,a}} = (\widehat{\Sigma}_p^{-1}(\mu_q - \widehat{\mu}_p))_a - \text{vec}(Y_{t:t+1, :, d})_a + \exp\left(\mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2\right) \quad (4.72)$$

where  $(\widehat{\Sigma}_p^{-1}(\mu_q - \widehat{\mu}_p))_a, \text{vec}(Y_{t:t+1, :, d})_a$  denote the  $a^{\text{th}}$  element of the vectorised forms of  $(\widehat{\Sigma}_p^{-1}(\mu_q - \widehat{\mu}_p))_a, Y_{t:t+1, :, d}$  respectively. Next, the case when  $\theta_i = l_{a,b}$ :

$$\begin{aligned} \frac{\partial \text{KL}[q||p]}{\partial l_{a,b}} &= -\delta_{a,b} + \left( (\widehat{\Sigma}_p^{-1} \cdot L_q) \circ L'_q \right)_{a,b} \\ &\quad + f_{a,b}(l_{a,b}) f'_{a,b}(l_{a,b}) \exp\left(\mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2\right) \end{aligned} \quad (4.73)$$

where  $\delta_{a,b} = 1$  if  $a = b$  and 0 otherwise is the Kronecker delta, and  $\circ$  denotes the element-wise product, and

$$L'_{q,i,j} = \begin{cases} f'_{i,j}(l_{i,j}) & j \leq i \\ 0 & j > i \end{cases} \quad (4.74)$$

with

$$f'_{i,j}(l_{i,j}) = \begin{cases} \exp(l_{i,j}) & i = j \\ 1 & j < i \end{cases} \quad (4.75)$$

## 4.B Hessian matrix for KL divergence from $q \in \mathcal{Q}_{\mathcal{N}}$

The entries of the Hessian matrix  $H$  of second order partial derivatives of the KL divergence (4.46) has entries are listed according to the parameters within  $\theta$  to which they are associated. First the case  $\partial^2 / \partial \theta_i \partial \theta_j$  when  $\theta_i = \mu_{q,a}$ ,  $\theta_j = \mu_{q,b}$ :

$$\frac{\partial^2 \text{KL}[q||p]}{\partial \mu_{q,a} \partial \mu_{q,b}} = \widehat{\Sigma}_{p,a,b}^{-1} + \delta_{a,b} \left\{ \exp \left( \mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2 \right) \right\} \quad (4.76)$$

where  $\delta_{a,b} = 1$  if  $a = b$  and 0 otherwise is the Kronecker delta. Next, the case when  $\theta_i = \mu_{q,a}$ ,  $\theta_j = l_{b,c}$ :

$$\frac{\partial^2 \text{KL}[q||p]}{\partial \mu_{q,a} \partial l_{b,c}} = \delta_{a,b} \left\{ f_{a,c}(l_{a,c}) f'_{a,c}(l_{a,c}) \exp \left( \mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2 \right) \right\} \quad (4.77)$$

Finally, the case when  $\theta_i = l_{a,b}$ ,  $\theta_j = l_{c,d}$ :

$$\begin{aligned} \frac{\partial^2 \text{KL}[q||p]}{\partial l_{a,b} \partial l_{c,d}} &= \delta_{b,d} \left\{ \widehat{\Sigma}_{p,a,c}^{-1} f'_{c,b}(l_{c,b}) f'_{a,b}(l_{a,b}) \right\} \\ &+ \delta_{a,c} \left\{ f_{a,d}(l_{a,d}) f'_{a,d}(l_{a,d}) f_{a,b}(l_{a,b}) f'_{a,b}(l_{a,b}) \right. \\ &\times \left. \exp \left( \mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2 \right) \right\} \\ &\delta_{a,c} \delta_{b,d} \left\{ \left( (\widehat{\Sigma}_p^{-1} \cdot L) \circ L'' \right)_{a,b} + \left( f'_{a,b}(l_{a,b})^2 + f_{a,b}(l_{a,b}) f''_{a,b}(l_{a,b}) \right) \right. \\ &\times \left. \exp \left( \mu_{q,a} + \frac{1}{2} \sum_{j=1}^a f_{a,j}(l_{a,j})^2 \right) \right\} \end{aligned} \quad (4.78)$$

where

$$L''_{q,i,j} = \begin{cases} f''_{i,j}(l_{i,j}) & j \leq i \\ 0 & j > i \end{cases} \quad (4.79)$$

with

$$f''_{i,j}(l_{i,j}) = \begin{cases} \exp(l_{i,j}) & i = j \\ 0 & j < i \end{cases} \quad (4.80)$$

## 4.C Grad vector for KL divergence from $q \in Q_{\Pi\mathcal{N}}$

The entries of the grad vector of first order partial derivatives of the KL divergence (4.49) has entries are listed according to the parameters within  $\theta$  to which they are associated. First, the case  $\partial/\partial\theta_i$  when  $\theta_i = \mu_{q,a}$ :

$$\frac{\partial \text{KL}[q||p]}{\partial \mu_{q,a}} = (\widehat{\Sigma}_p^{-1}(\mu_q - \widehat{\mu}_p))_a - \text{vec}(Y_{t:t+1, :, d})_a + \exp\left(\mu_{q,a} + \frac{1}{2}\exp(2l_a)\right) \quad (4.81)$$

Next, the case when  $\theta_i = l_a$ :

$$\frac{\partial \text{KL}[q||p]}{\partial l_a} = -1 + \left((\widehat{\Sigma}_p^{-1} \cdot L_q) \circ L_q\right)_a + \exp\left(\mu_{q,a} + 2l_a + \frac{1}{2}\exp(2l_a)\right) \quad (4.82)$$

## 4.D Hessian matrix for KL divergence from $q \in Q_{\Pi\mathcal{N}}$

The entries of the Hessian matrix  $H$  of second order partial derivatives of the KL divergence (4.49) has entries are listed according to the parameters within  $\theta$  to which they are associated. First the case  $\partial^2/\partial\theta_i\partial\theta_j$  when  $\theta_i = \mu_{q,a}$ ,  $\theta_j = \mu_{q,b}$ :

$$\frac{\partial^2 \text{KL}[q||p]}{\partial \mu_{q,a} \partial \mu_{q,b}} = \widehat{\Sigma}_{p,a,b}^{-1} + \delta_{a,b} \left\{ \exp\left(\mu_{q,a} + \frac{1}{2}\exp(2l_a)\right) \right\} \quad (4.83)$$

Next, the case when  $\theta_i = \mu_{q,a}$ ,  $\theta_j = l_b$ :

$$\frac{\partial^2 \text{KL}[q||p]}{\partial \mu_{q,a} \partial l_b} = \delta_{a,b} \left( \exp \left( \mu_{q,a} + 2l_a + \frac{1}{2} \exp(2l_a) \right) \right) \quad (4.84)$$

Finally, the case where  $\theta_i = l_a$ ,  $\theta_j = l_b$ :

$$\begin{aligned} \frac{\partial^2 \text{KL}[q||p]}{\partial l_a \partial l_b} = & \delta_{a,b} \left( 2\widehat{\Sigma}_{p,a,a}^{-1} \exp(2l_a) + \exp \left( \mu_{q,a} + 4l_a + \frac{1}{2} \exp(2l_a) \right) \right. \\ & \left. + 2 \exp \left( \mu_{q,a} + 2l_a + \frac{1}{2} \exp(2l_a) \right) \right) \end{aligned} \quad (4.85)$$

## Chapter 5

# Applying the method of moments in hierarchical clustering

The current chapter is an investigation into an efficient method of model based hierarchical clustering. Finding an optimal clustering can be computationally expensive, and hierarchical clustering is a popular method that aims to find a sequence of ‘good’ clusterings. In its traditional form (Duda et al., 1973), cluster dissimilarities are computed from pairwise distances between elements, without regard to any probabilistic considerations. This can be a considerable shortcoming in a statistical context; model based hierarchical clustering methods (Heller and Ghahramani, 2005; Stolcke and Omohundro, 1993) have consequently been developed that use probabilistic dissimilarities instead.

A hierarchical clustering algorithm can be quickly summarised as a sequence of optimal cluster mergers. Starting from the trivial clustering  $k(s) = s$ , at each iteration of the procedure a pair of clusters are merged to produce the next clustering in the sequence. All possible pairwise cluster mergers are considered, and optimality is based on a pre-specified measure of dissimilarity  $d(A, B)$  between each pair. Traditional algorithms compute cluster dissimilarities as some function of a distance metric between elements of the clusters. To incorporate statistical considerations, model based cluster dissimilarities can be used instead, and are often likelihood based.

Such likelihood based dissimilarities can be expensive to compute, and often

employ approximations to probabilistic quantities, e.g. Harmeling and Williams (2011). A model based measure of cluster dissimilarity is proposed here that is cheaper to compute than likelihood based quantities. The state space model of chapter 4 is extended to incorporate clusterings within the observed data, and the method of moments estimators developed previously are adapted to produce estimates of cluster dissimilarity.

The proposed method is appropriate for high dimensional functional data that consists of multiple i.i.d replicates. Many such datasets can be modelled with non-stationary state space models, such as the Gaussian-Poisson model used in chapter 4. The examples detailed in chapter 4 are equally relevant in this context.

In particular, the example mentioned previously of exit counts at subway stations is to be used in the algorithm developed in the current chapter. The dataset used is available from Transport for London (TfL), the organisation that operates the London Underground Tube network. A sample of 10 days of entrance and exit counts at each of the 374 stations across London, aggregated over 30 minute intervals, is available through a link on the TfL web site at <http://blog.tfl.gov.uk/2015/12/09/is-customer-flow-data-useful-to-developers/>.

**Figure 5.1:** Graph showing the average exit counts, averaged over 10 days, at Tube stations throughout the day, aggregated over 30 minute intervals. Each line represents a Tube station.

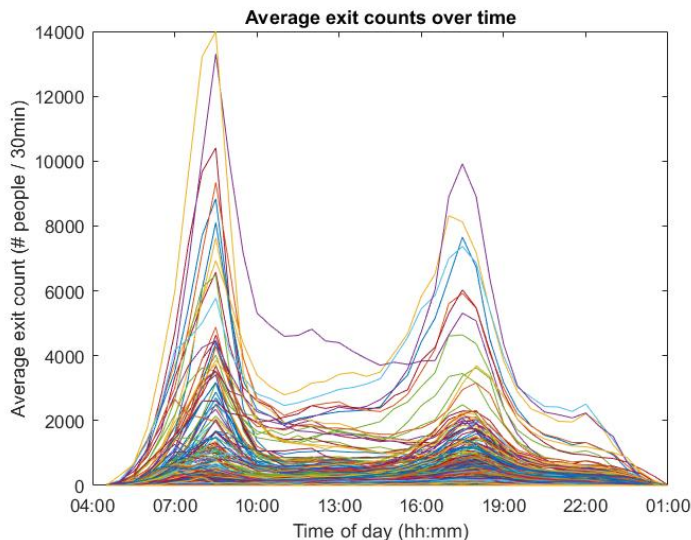




Fig 5.1 shows the average exit count over time for each of the 374 stations in the dataset, averaged over the 10 days. It is clear from the figure that there is structure among stations regarding exit count behaviour. The question of clustering stations on the basis of this shared behaviour is a natural line of enquiry. Such a clustering could provide insight into passenger flows around London and facilitate the prediction of various events, including for example the impact of station closures.

It is likely that the shared properties of stations sharing a cluster would be statistical in nature, therefore justifying a probabilistic hierarchical clustering approach. A plausible model for the data is an adapted version of the Gaussian-Poisson state space model of chapter 4, described in sec 5.1. As the results of chapter 4 demonstrated, a likelihood based measure of cluster dissimilarity using such a model would be expensive to compute. Even if parameters were fitted by some other means, just evaluating the likelihood is a non-trivial computation.

Particularly in the early stages of the hierarchical clustering procedure, the dimension of the latent space will be very high. As seen in chapter 4, using Gaussian proposals with importance sampling for EM style likelihood evaluation had limited efficacy even in much smaller latent spaces. In addition, the total number of likelihood evaluations throughout the clustering procedure is quadratic in the number of dimensions. The cost of finding closest Gaussians will be prohibitive when the number of likelihood evaluations required is considered.

By using a moment based dissimilarity measure, the cost of evaluating likelihoods for each potential cluster merger is avoided. This reduces the cost of the procedure significantly. The resulting sequence of clusterings is still derived from modelling criteria, and thus retains theoretical integrity, but it can be computed more quickly than likelihood based model measures.

A short-listing procedure is also introduced here that limits the merger search at each iteration to a subset of all possibilities. Searching all possible cluster pairs for the optimal merger has quadratic cost in the number of clusters at each iteration. The short-lists are linear in the number of clusters, and as such they overcome the

quadratic computation costs that are otherwise unavoidable.

The proposed dissimilarity measure is based on the reconstruction error of modelled moments in a dataset. Parameters are fitted using estimating equations derived from moment equations. The parameter estimates imply specific values for the moments of the data, and errors in these reconstructed moments are used in the construction of the dissimilarity measure.

The following exposition first derives the parameter estimates, and then constructs the dissimilarity measure. Short-lists are introduced subsequently, followed by implementations of the algorithm on both real and synthetic datasets.

## 5.1 Parameter estimation

As discussed above, the state space model for which parameters will be estimated using the method of moments is a generalisation of the Gaussian-Poisson model of chapter 4:

$$\begin{aligned} \mathcal{P}(X_{1,:;d} \mid \boldsymbol{\theta}) &= \mathcal{N}(X_{1,:;d} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \quad d \in 1, \dots, D \\ \mathcal{P}(X_{t+1,:;d} \mid X_{t,:;d}, \boldsymbol{\theta}) &= \mathcal{N}(X_{t+1,:;d} \mid A_t X_{t,:;d} + \mathbf{b}_t, V_t), \quad t \in 1, \dots, T-1 \\ \mathcal{P}(Y_{t,:;d} \mid x_{t,:;d}, \boldsymbol{\theta}) &= \prod_{s=1}^S \mathcal{P}(Y_{t,s,d} \mid X_{t,k(s),d}) \\ &= \mathcal{P}\mathcal{O}(Y_{t,s,d} \mid \exp(\boldsymbol{\beta}_{0,s} + \boldsymbol{\beta}_{1,s} X_{t,k(s),d})), \quad s \in 1, \dots, S \end{aligned} \quad (5.1)$$

where  $\dim \mathcal{X} = M$ ,  $\dim \mathcal{Y} = S$ , and  $X_{t,i,d}, Y_{t,i,d}$  denote the  $i^{\text{th}}$  dimension of the latent and observed variable  $X, Y$  at time  $t$  in realisation  $d$ . At each time  $t$ , and for each realisation  $d$ ,  $X_{t,:;d}$  is the  $M$  dimensioned latent variable for that time point, and  $Y_{t,:;d}$  is the  $S$  dimensioned observed variable. The dimensionality of the latent space  $\mathcal{X}$  is reduced through a clustering  $k: s \mapsto k(s)$  of the dimensions  $s \in 1, \dots, S$  of the observed data:

$$\begin{aligned} k(s) &\in 1, \dots, M, \quad M \leq S \\ Y_{t,s,d} \mid X_{t,:;d} &\sim Y_{t,s,d} \mid X_{t,k(s),d} \end{aligned} \quad (5.2)$$

The fully dimensioned model with  $\dim \mathcal{X} = \dim \mathcal{Y} = S$  can be recovered from (5.1) by using the trivial clustering  $k(s) = s$  and setting  $\beta_{0,s} = 0, \beta_{1,s} = 1$  for all  $s \in 1, \dots, S$ . Parameter estimates for the model (5.1) are derived from the moment equations

$$\begin{aligned}\mathbb{E}[Y_{t,s,d}] &= \exp\left(\beta_{0,s} + \beta_{1,s}\mu_{t,k(s)} + \frac{1}{2}\beta_{1,s}^2\sigma_{t,k(s)}^2\right) \\ \mathbb{E}[Y_{t,s,d}^2 - Y_{t,s,d}] &= \exp(2\beta_{0,s} + 2\beta_{1,s}\mu_{t,k(s)} + 2\beta_{1,s}^2\sigma_{t,k(s)}^2) \\ \mathbb{E}[Y_{t_1,s_1,d}Y_{t_2,s_2,d}] &= \mathbb{E}[Y_{t_1,s_1,d}]\mathbb{E}[Y_{t_2,s_2,d}]\exp(\beta_{1,s_1}\beta_{1,s_2}\sigma_{(t_1,k(s_1)),(t_2,k(s_2))}) \\ &\quad (s_1 \neq s_2) \vee (t_1 \neq t_2)\end{aligned}\tag{5.3}$$

As the latent process has linear Gaussian transitions, its transition parameters  $A_t, \mathbf{b}_t, V_t$  can be estimated from the approximate prior marginals for adjacent time point pairs:

$$\begin{aligned}X_{t:t+1, :, d} &\sim \mathcal{N}(X_{t:t+1, :, d} \mid \mu_{t:t+1}, \Sigma_{t:t+1}) \\ \Rightarrow A_t &= \Sigma_{t+1,t} \cdot \Sigma_t^{-1} \\ \mathbf{b}_t &= \mu_{t+1} - A_t \mu_t \\ V_t &= \Sigma_{t+1} - \Sigma_{t+1,t} \cdot \Sigma_t^{-1} \cdot \Sigma_{t,t+1}\end{aligned}\tag{5.4}$$

where

$$\Sigma_{t:t+1} = \begin{pmatrix} \Sigma_t & \Sigma_{t,t+1} \\ \Sigma_{t+1,t} & \Sigma_{t+1} \end{pmatrix}\tag{5.5}$$

is the covariance matrix for the latent process over two adjacent time points. All transition parameters will be estimated from such time-adjacent pairwise marginals, i.e. only the within-time and time-adjacent latent covariances

$$\begin{aligned}\sigma_{(t,s'_1),(t',s'_2)} \\ s'_1, s'_2 \in 1, \dots, M \quad t' \in t, t+1 \quad (t \neq t') \vee (s'_1 \neq s'_2)\end{aligned}\tag{5.6}$$

will be estimated from data. As with calculating the surrogate marginals in chapter 4, to derive parameter estimates from the moment equations (5.3) it is convenient to first define the quantities

$$\begin{aligned} z_1(t, s) &= \log(\mathbb{E}[Y_{t,s,d}]) \\ z_2(t, s) &= \log(\mathbb{E}[Y_{t,s,d}^2 - Y_{t,s,d}]) \end{aligned} \quad (5.7)$$

Using these quantities, the moment equations (5.3) imply the following parameter estimating equations:

$$\begin{aligned} \beta_{0,s} + \beta_{1,s} \mu_{t,k(s)} &= 2z_1(t, s) - \frac{1}{2} z_2(t, s) \\ \beta_{1,s}^2 \sigma_{t,k(s)}^2 &= z_2(t, s) - 2z_1(t, s) \\ \beta_{1,s_1} \beta_{1,s_2} \sigma_{(t,k(s_1)), (t',k(s_2))} &= \log \left( \frac{\mathbb{E}[Y_{t,s_1,d} Y_{t',s_2,d}]}{\mathbb{E}[Y_{t,s_1,d}] \mathbb{E}[Y_{t',s_2,d}]} \right) \\ & \quad t' \in t, t+1, (t \neq t') \vee (s_1 \neq s_2) \end{aligned} \quad (5.8)$$

Unfortunately, when estimating all parameters simultaneously these estimating equations are under-determined. One way of overcoming this is to first estimate the parameters of the latent process separately, and then plug those estimates into (5.8). The estimating equations will then be over-determined for the unknown  $\beta$  parameters, and can be solved by a least squares minimisation.

### 5.1.1 Fitting the latent process

It is clear that the under-determination of the estimating equations (5.8) is a direct result of reducing the dimensionality of  $\mathcal{X}$  via the clustering function  $k$ . By projecting the data  $Y_{:,k^{-1}(s'),:}$  for each cluster  $s'$  onto a single dimension of an auxiliary data object  $\tilde{Y}_{:,s',:} \in \mathbb{N}^{T \times 1 \times D}$ , the latent process could be fitted to the auxiliary data

without encountering this problem:

$$\begin{aligned}
\tilde{Y}_{t,s',d} &= f_{s'}(Y_{t,k^{-1}(s'),d}), \quad s' \in 1, \dots, M \\
\Rightarrow \tilde{\mu}_{t,s'} &= 2\tilde{z}_1(t, s') - \frac{1}{2}\tilde{z}_2(t, s') \\
\tilde{\sigma}_{t,s'}^2 &= \tilde{z}_2(t, s') - 2\tilde{z}_1(t, s') \\
\tilde{\sigma}_{(t,s'_1),(t',s'_2)} &= \log \left( \frac{\mathbb{E}[\tilde{Y}_{t,s'_1,d}\tilde{Y}_{t',s'_2,d}]}{\mathbb{E}[\tilde{Y}_{t,s'_1,d}]\mathbb{E}[\tilde{Y}_{t',s'_2,d}]} \right) \\
& \quad t' \in t, t+1, \quad (t \neq t') \vee (s'_1 \neq s'_2)
\end{aligned} \tag{5.9}$$

where the  $f_{s'}$  are some functions and

$$\begin{aligned}
\tilde{z}_1(t, s') &= \log(\mathbb{E}[\tilde{Y}_{t,s',d}]) \\
\tilde{z}_2(t, s') &= \log(\mathbb{E}[\tilde{Y}_{t,s',d}^2 - \tilde{Y}_{t,s',d}])
\end{aligned} \tag{5.10}$$

are analogous to  $z_1(t, s), z_2(t, s)$ .

In terms of both interpretability and computational convenience, restricting  $f_{s'}$  to be a linear function of its arguments can be easily justified, and this approach is taken here. Several choices of function are available in this regard, including cluster averages:

$$f_{s'}(Y_{t,k^{-1}(s'),d}) = \frac{1}{|k^{-1}(s')|} \sum_{s \in k^{-1}(s')} Y_{t,s,d} \tag{5.11}$$

principal component projections:

$$\begin{aligned}
f_{s'}(Y_{t,k^{-1}(s'),d}) &= P_{Y_{:,k^{-1}(s'),d}}^{\text{PCA}} Y_{t,k^{-1}(s'),d} \quad \text{or} \\
f_{s'}(Y_{t,k^{-1}(s'),d}) &= P_{Y_{t,k^{-1}(s'),:}}^{\text{PCA}} Y_{t,k^{-1}(s'),d}
\end{aligned} \tag{5.12}$$

and a ‘representative dimension’ projection:

$$f_{s'}(Y_{t,k^{-1}(s'),d}) = Y_{t,s'^*,d}, \quad s'^* \in k^{-1}(s') \tag{5.13}$$

After a preliminary investigation into these choices it was found that choosing between them did not have a significant effect on subsequent inference. For reasons of interpretability it was decided to use the representative dimension approach of (5.13). For this approach, one dimension in each cluster is chosen to represent the cluster. The representative dimension  $s'^*$  was chosen to be the dimension  $s \in k^{-1}(s')$  whose average data over replicates  $\frac{1}{D} \sum_{d=1}^D Y_{:,s,d}$  was closest in norm to the average data over replicates and over cluster dimensions  $\frac{1}{D|k^{-1}(s')|} \sum_{d=1}^D \sum_{s'' \in k^{-1}(s')} Y_{:,s'',d}$ :

$$s'^* = \arg \min_{s \in k^{-1}(s')} \left\| \frac{1}{D} \sum_{d=1}^D Y_{:,s,d} - \frac{1}{D|k^{-1}(s')|} \sum_{d=1}^D \sum_{s'' \in k^{-1}(s')} Y_{:,s'',d} \right\| \quad (5.14)$$

and if more than one dimension in  $k^{-1}(s')$  minimises the norm in (5.14) then one of them can be chosen arbitrarily. Once a representative dimension for each cluster  $s' \in 1, \dots, M$  has been chosen, sample moments from the data can be used to approximate the expected values in (5.9) and estimate the parameters of the latent process. As with the surrogate marginals derived in chapter 4, sample means are thresholded to be above zero, and estimated covariance matrices are eigenvalue thresholded and rescaled:

$$\begin{aligned} \widehat{\mu}_{t,s'} &= 2\widehat{z}_1(t, s') - \frac{1}{2}\widehat{z}_2(t, s') \\ \widehat{\sigma}_{t,s'}^2 &= \widehat{z}_2(t, s') - 2\widehat{z}_1(t, s') \\ \widehat{\sigma}_{(t,s'_1),(t',s'_2)} &= \log \left( \frac{\widehat{\widetilde{Y}}_{t,s'_1^*,d} \widehat{\widetilde{Y}}_{t',s'_2^*,d}}{\widehat{\widetilde{Y}}_{t,s'_1^*,d} \widehat{\widetilde{Y}}_{t',s'_2^*,d}} \right), \quad t' \in t, t+1, \quad (t \neq t') \vee (s_1 \neq s_2) \\ \widehat{\Sigma}_{t:t+1} &= \text{diag}(R) \cdot V \cdot \text{diag}(\widetilde{D}) \cdot V' \cdot R \end{aligned} \quad (5.15)$$

where

$$\begin{aligned}
\widehat{Y}_{t,s'^*} &= \max(\overline{Y}_{t,s'^*}, \delta) \\
\widehat{Y^2}_{t,s'^*} &= \max\left(\widehat{Y}_{t,s'^*} + \widehat{Y}_{t,s'^*}^2, \delta\right) \\
\widehat{\overline{Y}_{t,s_1} \overline{Y}_{t',s_2}} &= \max(\overline{Y}_{t,s_1} \overline{Y}_{t',s_2}, \delta), \quad t' \in t, t+1, \quad (t \neq t') \vee (s_1 \neq s_2) \\
\widehat{z}_1(t, s') &= \log\left(\widehat{Y}_{t,s'^*}\right) \\
\widehat{z}_2(t, s') &= \log\left(\widehat{Y^2}_{t,s'^*}, -\widehat{Y}_{t,s'^*}\right)
\end{aligned} \tag{5.16}$$

for some small  $\delta > 0$ , and  $R, V, \text{diag}(\tilde{D})$  are the rescaling, eigenvector, and eigenvalue matrices for  $\widehat{\Sigma}_{t:t+1}$ . See chapter 4 sec 4.3.1 for details of the eigenvalue thresholding and rescaling of estimated covariance matrices.

### 5.1.2 Fitting the $\beta$ parameters

Once the latent process has been estimated, its parameters can be plugged into the estimating equations (5.8). The equations can then be used to form a squared error objective function to estimate the  $\beta$  parameters:

$$\begin{aligned}
\widehat{\beta} &= \arg \min_{\beta \in \mathbb{R}^{2S}} \sum_{t=1}^T \left( \sum_{s=1}^S (e_1(t, s)^2 + e_2(t, s)^2) + \sum_{s_1=1}^{S-1} \sum_{s_2=s_1+1}^S e_3(t, s_1, t, s_2)^2 \right) \\
&\quad + \sum_{t=1}^{T-1} \sum_{s_1, s_2=1}^S e_3(t, s_1, t+1, s_2)^2 \\
e_1(t, s) &= \beta_{0,s} + \beta_{1,s} \widehat{\mu}_{t,k(s)} - \left( 2\widehat{z}_1(t, s) - \frac{1}{2}\widehat{z}_2(t, s) \right) \\
e_2(t, s) &= \beta_{1,s} \widehat{\sigma}_{t,k(s)} - \sqrt{\widehat{z}_2(t, s) - 2\widehat{z}_1(t, s)} \\
e_3(t_1, s_1, t_2, s_2) &= \beta_{1,s_1} \beta_{1,s_2} \widehat{\sigma}_{(t_1, k(s_1)), (t_2, k(s_2))} - \log \left( \frac{\widehat{\overline{Y}_{t_1, s_1, d} \overline{Y}_{t_2, s_2, d}}}{\widehat{Y}_{t_1, s_1, d} \widehat{Y}_{t_2, s_2, d}} \right)
\end{aligned} \tag{5.17}$$

where

$$\begin{aligned}
\widehat{z}_1(t, s) &= \log\left(\widehat{Y}_{t,s}\right) \\
\widehat{z}_2(t, s) &= \log\left(\widehat{Y^2}_{t,s} - \widehat{Y}_{t,s}\right)
\end{aligned} \tag{5.18}$$

and hatted over-lines indicate thresholded sample means as per (5.16). Though estimates  $\hat{\beta}$  made by minimising the objective function in (5.17) are valid method of moments parameter estimates, they can be improved upon. In the real world case of finite data sizes, parameters estimated via (5.17) do not necessarily minimise the errors of the reconstructed data moments

$$\begin{aligned}
 \widehat{Y}_{t,s}(\hat{\theta}) &= \exp\left(\hat{\beta}_{0,s} + \hat{\beta}_{1,s}\widehat{\mu}_{t,k(s)} + \frac{1}{2}\hat{\beta}_{1,s}^2\widehat{\sigma}_{t,k(s)}^2\right) \\
 \widehat{Y}_{t,s}^2(\hat{\theta}) &= \exp\left(\hat{\beta}_{0,s} + \hat{\beta}_{1,s}\widehat{\mu}_{t,k(s)} + \frac{1}{2}\hat{\beta}_{1,s}^2\widehat{\sigma}_{t,k(s)}^2\right) \\
 &\quad + \exp\left(2\hat{\beta}_{0,s} + 2\hat{\beta}_{1,s}\widehat{\mu}_{t,k(s)} + 2\hat{\beta}_{1,s}^2\widehat{\sigma}_{t,k(s)}^2\right) \\
 \widehat{Y}_{t,s_1} \widehat{Y}_{t',s_2}(\hat{\theta}) &= \widehat{Y}_{t,s_1} \widehat{Y}_{t',s_2} \exp\left(\hat{\beta}_{1,s_1}\hat{\beta}_{1,s_2}\widehat{\sigma}_{(t,k(s_1)),(t',k(s_2))}\right) \\
 &\quad t' \in t, t+1, \quad (t \neq t') \vee (s_1 \neq s_2)
 \end{aligned} \tag{5.19}$$

To minimise the errors of the reconstructed moments simultaneously, they need to be summed in an analogous fashion to the sum of squared errors in (5.17). As the second and third reconstructed moments in (5.19) are in squared units, they need to be square-rooted before they can all be added together. This provides the alternative objective function to minimise:

$$\begin{aligned}
 \hat{\beta} &= \arg \min_{\beta \in \mathbb{R}^{2S}} \sum_{t=1}^T \left\{ \sum_{s=1}^S \left( \overline{Y}_{t,s} - \widehat{Y}_{t,s}(\hat{\theta}) \right)^2 + \left( \sqrt{\overline{Y}_{t,s}^2} - \sqrt{\widehat{Y}_{t,s}^2(\hat{\theta})} \right)^2 \right. \\
 &\quad \left. + \sum_{s_1=1}^{S-1} \sum_{s_2=s_1+1}^S \left( \sqrt{\overline{Y}_{t,s_1} \overline{Y}_{t,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}(\hat{\theta})} \right)^2 \right\} \\
 &\quad + \sum_{t=1}^{T-1} \sum_{s_1, s_2=1}^S \left( \sqrt{\overline{Y}_{t,s_1} \overline{Y}_{t+1,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t+1,s_2}(\hat{\theta})} \right)^2
 \end{aligned} \tag{5.20}$$

Minimising the objective function in (5.20) using a gradient based minimiser can be challenging, as the gradients when starting from a poorly chosen initial estimate can be so steep as to cause numerical issues. This problem is due to the exponential functions in the reconstructed moments, and is not present when estimating the  $\beta$  parameters using the objective function in (5.17). If provisional estimates are made using (5.17) and then plugged in as initial estimates into (5.20), they were found



in practice to be close enough to the final estimates not to suffer from excessively steep gradients. The complete parameter estimation procedure is summarised in algorithm 5.1.

**Algorithm 5.1** Parameter estimation

1. Identify which dimension  $s'^*$  in each cluster  $s'$  is to be the representative dimension, according to (5.14):

$$s'^* = \arg \min_{s \in k^{-1}(s')} \left\| \frac{1}{D} \sum_{d=1}^D Y_{:,s,d} - \frac{1}{D|k^{-1}(s')|} \sum_{d=1}^D \sum_{s'' \in k^{-1}(s')} Y_{:,s'',d} \right\| \quad (5.21)$$

2. Fit the latent process to the auxiliary data object  $\tilde{Y}_{:,s',:} = Y_{:,s'^*,:}$ ,  $s' \in 1, \dots, M$  according to (5.15):

$$\begin{aligned} \hat{\mu}_{t,s'} &= 2\hat{z}_1(t, s') - \frac{1}{2}\hat{z}_2(t, s') \\ \hat{\sigma}_{t,s'}^2 &= \hat{z}_2(t, s') - 2\hat{z}_1(t, s') \\ \hat{\sigma}_{(t,s'_1),(t',s'_2)} &= \log \left( \frac{\widehat{\tilde{Y}_{t,s'_1^*,d} \tilde{Y}_{t',s'_2^*,d}}}{\widehat{\tilde{Y}_{t,s'_1^*,d} \tilde{Y}_{t',s'_2^*,d}}} \right), \quad t' \in t, t+1, \quad (t \neq t') \vee (s_1 \neq s_2) \\ \hat{\Sigma}_{t:t+1} &= \text{diag}(R) \cdot V \cdot \text{diag}(\tilde{D}) \cdot V' \cdot R \end{aligned} \quad (5.22)$$

3. Fit the  $\beta$  parameters to the data and fitted latent process according to (5.20):

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta \in \mathbb{R}^{2S}} \sum_{t=1}^T \left\{ \sum_{s=1}^S \left( \bar{Y}_{t,s} - \widehat{Y}_{t,s}(\hat{\theta}) \right)^2 + \left( \sqrt{Y_{t,s}^2} - \sqrt{\widehat{Y}_{t,s}^2(\hat{\theta})} \right)^2 \right. \\ &\quad \left. + \sum_{s_1=1}^{S-1} \sum_{s_2=s_1+1}^S \left( \sqrt{Y_{t,s_1} Y_{t,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}(\hat{\theta})} \right)^2 \right\} \\ &\quad + \sum_{t=1}^{T-1} \sum_{s_1, s_2=1}^S \left( \sqrt{Y_{t,s_1} Y_{t+1,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t+1,s_2}(\hat{\theta})} \right)^2 \end{aligned} \quad (5.23)$$

## 5.2 Hierarchical clustering

For a given clustering, estimating the model parameters via algorithm 5.1 provides a method for performing further inference such as prediction and outlier detection.

Unless an exogenous clustering function is provided though, a specific clustering will have to be chosen upon before any further inference can be performed. In general this requires choosing the number of clusters as well as the assignment of each dimension to a cluster. When the dimensionality  $\dim \mathcal{Y} = S$  is large, searching the space of clusterings naively is practically impossible.

A common method of circumventing this problem, and the method used in the current thesis, is known as *hierarchical clustering*. This method produces a sequence of clusterings  $\{k_i\}_{i=1}^S$  starting at the trivial clustering  $k_1(s) = s \forall s$ , and each subsequent member of the sequence is formed by merging two clusters  $(A^*, B^*) \mapsto A^* \cup B^*$  from the previous member:

$$k_{i+1}(s) = \begin{cases} A^* \cup B^* & s: (k_i(s) = A^*) \vee (k_i(s) = B^*) \\ k_i(s) & s: (k_i(s) \neq A^*) \wedge (k_i(s) \neq B^*) \end{cases} \quad (5.24)$$

where  $A^*, B^* \in \mathcal{C}_i$  and  $A^* \cup B^* \in \mathcal{C}_{i+1}$  are cluster labels in the mappings  $k_i: 1, \dots, S \rightarrow \mathcal{C}_i$  and  $k_{i+1}: 1, \dots, S \rightarrow \mathcal{C}_{i+1}$  respectively. The clusters to be merged at each iteration are chosen to be the minimisers of a given dissimilarity function  $d$ :

$$(A^*, B^*) = \arg \min_{(A, B) \in \mathcal{C}_i^2} d(A, B) \quad (5.25)$$

where  $d$  is a measure of how dissimilar clusters are to each other. This method ensures that each merger is between the two most similar clusters under  $d$ . The sequence  $\{k_i\}_{i=1}^S$  of clusterings has the hierarchical quality

$$\begin{aligned} \exists i, s_1, s_2 \in 1, \dots, S: k_i(s_1) = k_i(s_2) \\ \Rightarrow k_j(s_1) = k_j(s_2) \quad \forall j \in i+1, \dots, S \end{aligned} \quad (5.26)$$

In the current context of parameter estimation via the method of moments, the following probabilistic measure of cluster dissimilarity is used:

$$d(A, B) = R_{k_{i+1}}(A \cup B, \hat{\theta}_{k_{i+1}}) - (R_{k_i}(A, \hat{\theta}_{k_i}) + R_{k_i}(B, \hat{\theta}_{k_i})) \quad (5.27)$$

where  $k_{i+1}$  is the proposed clustering such that  $(A, B) \mapsto A \cup B$  and  $\hat{\theta}_k$  are the parameters fitted to the data under clustering  $k$ , and

$$\begin{aligned}
R_k(C, \hat{\theta}) = \frac{1}{Z} & \left[ \sum_{t=1}^T \left\{ \sum_{s \in k^{-1}(C)} \left| \bar{Y}_{t,s} - \widehat{Y}_{t,s}(\hat{\theta}) \right| + \left| \sqrt{\bar{Y}_{t,s}^2} - \sqrt{\widehat{Y}_{t,s}^2(\hat{\theta})} \right| \right. \right. \\
& + \sum_{s_1 \in k^{-1}(C)} \sum_{s_2 \in k^{-1}(C) \setminus s_1} \left. \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}(\hat{\theta})} \right| \right\} \\
& + \sum_{t=1}^{T-1} \sum_{s_1, s_2 \in k^{-1}(C)} \left. \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t+1,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t+1,s_2}(\hat{\theta})} \right| \right] \quad (5.28)
\end{aligned}$$

is the mean absolute error of all reconstructed moments internal to cluster  $C$  under clustering  $k$  using fitted parameters  $\hat{\theta}$ , and

$$Z = T \left( 2|k^{-1}(C)| + \frac{1}{2}|k^{-1}(C)|(|k^{-1}(C)| - 1) \right) + (T - 1)|k^{-1}(C)|^2 \quad (5.29)$$

is the total number of summands in (5.28).

This dissimilarity measure conforms to the intuition that the optimal cluster merger has the least cost (amongst potential mergers) in terms of representing the data. Here, ‘representing data’ is implicitly defined to be the ability to reproduce the low order moments of the data using the fitted parameters. Using this condition instead of a likelihood based one has a benefit in terms of computing costs; evaluating the moments of a fitted model is significantly cheaper than evaluating the likelihood.

As chapter 4 showed, using an SEM style approach to approximate log composite likelihoods is not cheap, even when the latent dimensionality is relatively low. Furthermore, the results of the experiments in chapter 4 suggest that accuracy of such approximations cannot be relied upon with great confidence either. And using Monte Carlo samples drawn from the fitted prior to integrate out the latent variables will be computationally expensive when the latent dimension is large. Avoiding such computational costs would be a very desirable property for any alternative model based dissimilarity measure.

The focus of the current chapter is, therefore, an exploration of the utility of using (5.27) as a cluster dissimilarity measure. As it is a model based measure, it is reasonable to assume it will capture some of the statistical information in each potential cluster merger. Whether it captures enough information to be useful will be explored by implementing a hierarchical clustering algorithm and analysing the resulting sequence of clusterings.

Before the algorithm is implemented though, a further issue requires consideration. Searching the space of possible cluster mergers naively has quadratic cost in the number of clusters  $|\mathcal{C}_i|$ . It should be noted that each evaluation of (5.27) requires a refitting of parameters for the potential cluster pair to be merged. As such, the unit cost of each evaluation is not insignificant. A method for reducing this to a linear cost is proposed in sec 5.2.1. To ease the exposition of the proposed method, the hierarchical clustering procedure that naively searches the whole merger space is summarised in algorithm 5.2.

### 5.2.1 Short-lists

Searching for the optimal cluster pair to merge in step 2a of algorithm 5.2 has a computational cost that is quadratic in the number of clusters  $|\mathcal{C}_i|$  for each iteration  $i$ . As noted above, the unit cost of each evaluation of (5.27) is non-negligible. As the context of the current chapter has  $S = \dim \mathcal{Y}$  large, algorithm 5.2 can be expensive to run, particularly in its early iterations.

The following proposed method mitigates this expense by reducing the cost to be linear in  $|\mathcal{C}_i|$ . The underlying principle is the use of short-lists  $k_i^* \subset \mathcal{C}_i^2$  generated by an alternative, cheap to evaluate dissimilarity measure  $d_2$ . The alternative measure could simply be a cheap proxy for (5.27), or it could be used to impose exogenous constraints on clusters. The current exposition focusses on the case where exogenous constraints are being imposed. Suppose, for example, that the dimensions  $s \in 1, \dots, S$  had an associated geographical location in the real world. If  $d_2(A, B)$  was the mean of the geographical distances  $d^{\text{geo}}(s_1, s_2)$  between locations

**Algorithm 5.2** Naive hierarchical clustering

1. Set  $k_1$  to be the trivial clustering  $k_1(s) = s \quad \forall s \in 1, \dots, S$ .
2. for  $i \in 1, \dots, S-1$ :
  - (a) Find the least dissimilar cluster pair  $(A^*, B^*) \in \mathcal{C}_i^2$  according to dissimilarity measure (5.27):

$$(A^*, B^*) = \arg \min_{(A, B) \in \mathcal{C}_i^2} R_{k_{i+1}}(A \cup B, \hat{\theta}_{k_{i+1}}) - (R_{k_i}(A, \hat{\theta}_{k_i}) + R_{k_i}(B, \hat{\theta}_{k_i})) \quad (5.30)$$

where

$$\begin{aligned} R_k(C, \hat{\theta}) = \frac{1}{Z} & \left[ \sum_{t=1}^T \left\{ \sum_{s \in k^{-1}(C)} \left| \bar{Y}_{t,s} - \widehat{Y}_{t,s}(\hat{\theta}) \right| + \left| \sqrt{Y_{t,s}^2} - \sqrt{\widehat{Y}_{t,s}^2(\hat{\theta})} \right| \right. \right. \\ & + \left. \sum_{s_1 \in k^{-1}(C)} \sum_{s_2 \in k^{-1}(C) \setminus s_1} \left| \sqrt{Y_{t,s_1} Y_{t,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}(\hat{\theta})} \right| \right\} \\ & + \left. \sum_{t=1}^{T-1} \sum_{s_1, s_2 \in k^{-1}(C)} \left| \sqrt{Y_{t,s_1} Y_{t+1,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t+1,s_2}(\hat{\theta})} \right| \right] \quad (5.31) \end{aligned}$$

and

$$Z = T \left( 2|k^{-1}(C)| + \frac{1}{2}|k^{-1}(C)|(|k^{-1}(C)| - 1) \right) + (T-1)|k^{-1}(C)|^2 \quad (5.32)$$

- (b) Construct the next clustering in the sequence by merging cluster pair  $(A^*, B^*) \mapsto A^* \cup B^*$ :

$$k_{i+1}(s) = \begin{cases} A^* \cup B^* & s: (k_i(s) = A^*) \vee (k_i(s) = B^*) \\ k_i(s) & s: (k_i(s) \neq A^*) \wedge (k_i(s) \neq B^*) \end{cases} \quad (5.33)$$

associated to  $s_1 \in k^{-1}(A), s_2 \in k^{-1}(B)$ :

$$d_2(A, B) = \frac{1}{|k^{-1}(A)| \cdot |k^{-1}(B)|} \sum_{s_1 \in k^{-1}(A)} \sum_{s_2 \in k^{-1}(B)} d^{\text{geo}}(s_1, s_2) \quad (5.34)$$

then constructing short-lists using (5.34) could represent some constraint regarding the geographical interpretation of each dimension in a cluster.

The short-lists proposed here are constructed such that, for each cluster  $A \in \mathcal{C}_i$ , the cluster pairs containing itself and its  $n$  least dissimilar clusters are included in the short-list:

$$(A, B) \in k_i^* \Leftrightarrow |\{(A, C) : d_2(A, C) < d_2(A, B)\}| < n \quad A, B, C \in \mathcal{C}_i \quad (5.35)$$

which clearly has a maximum size of  $|\mathcal{C}_i|n$ . Such a short-list could represent the heuristic constraint that geographically distant clusters should not be considered for merging. By restricting the search for optimal cluster pairs for merging to only include pairs in the short-list:

$$(A^*, B^*) = \arg \min_{(A, B) \in k_i^*} R_{k_{i+1}}(A \cup B, \hat{\theta}_{k_{i+1}}) - (R_{k_i}(A, \hat{\theta}_{k_i}) + R_{k_i}(B, \hat{\theta}_{k_i})) \quad (5.36)$$

the cost of finding the cluster pair to merge at each iteration will be linear in  $|\mathcal{C}_i|$ . The procedure for hierarchical clustering using short-lists is summarised in algorithm 5.3.

## 5.2.2 Analysis of clustering sequence

If the dimensions  $s$  of  $\mathcal{Y}$  represent or relate to objects in the real world, then each clustering  $k$  of them represents some equivalence relation on them. A sequence of clusterings  $\{k_i\}_{i=1}^S$  computed via either of algorithms 5.2 or 5.3 can be used as a basis for inference on such relations. Pairs  $(s_1, s_2)$  of dimensions that share a cluster from early in the sequence, for example, could be supposed to share an equivalence with more confidence than pairs whose only common cluster is the universal cluster

**Algorithm 5.3** Hierarchical clustering using short-lists

1. Set  $k_1$  to be the trivial clustering  $k_1(s) = s \quad \forall s \in 1, \dots, S$ .
2. for  $i \in 1, \dots, S-1$ :

(a) Construct the short-list  $k_i^*$  according to (5.35):

$$(A, B) \in k_i^* \Leftrightarrow |\{(A, C) : d_2(A, C) < d_2(A, B)\}| < n \quad A, B, C \in \mathcal{C}_i \quad (5.37)$$

(b) Find the least dissimilar cluster pair  $(A^*, B^*) \in k_i^*$  according to dissimilarity measure (5.27):

$$(A^*, B^*) = \arg \min_{(A, B) \in k_i^*} R_{k_{i+1}}(A \cup B, \hat{\theta}_{k_{i+1}}) - (R_{k_i}(A, \hat{\theta}_{k_i}) + R_{k_i}(B, \hat{\theta}_{k_i})) \quad (5.38)$$

where

$$\begin{aligned} R_k(C, \hat{\theta}) = & \frac{1}{Z} \left[ \sum_{t=1}^T \left\{ \sum_{s \in k^{-1}(C)} \left| \bar{Y}_{t,s} - \widehat{\bar{Y}}_{t,s}(\hat{\theta}) \right| + \left| \sqrt{\bar{Y}_{t,s}^2} - \sqrt{\widehat{\bar{Y}}_{t,s}^2(\hat{\theta})} \right| \right. \right. \\ & + \left. \sum_{s_1 \in k^{-1}(C)} \sum_{s_2 \in k^{-1}(C) \setminus s_1} \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t,s_2}} - \sqrt{\widehat{\bar{Y}}_{t,s_1} \widehat{\bar{Y}}_{t,s_2}(\hat{\theta})} \right| \right\} \\ & + \left. \sum_{t=1}^{T-1} \sum_{s_1, s_2 \in k^{-1}(C)} \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t+1,s_2}} - \sqrt{\widehat{\bar{Y}}_{t,s_1} \widehat{\bar{Y}}_{t+1,s_2}(\hat{\theta})} \right| \right] \quad (5.39) \end{aligned}$$

and

$$Z = T \left( 2|k^{-1}(C)| + \frac{1}{2}|k^{-1}(C)|(|k^{-1}(C)| - 1) \right) + (T-1)|k^{-1}(C)|^2 \quad (5.40)$$

(c) Construct the next clustering in the sequence by merging cluster pair  $(A^*, B^*) \mapsto A^* \cup B^*$ :

$$k_{i+1}(s) = \begin{cases} A^* \cup B^* & s : (k_i(s) = A^*) \vee (k_i(s) = B^*) \\ k_i(s) & s : (k_i(s) \neq A^*) \wedge (k_i(s) \neq B^*) \end{cases} \quad (5.41)$$

$k(s) = k \forall s$ . Defining

$$b(s_1, s_2) = S - \max_{i \in 1, \dots, S} k_i : k_i(s_1) \neq k_i(s_2) \quad (5.42)$$

as the strength of the implied relation between  $s_1, s_2$ , these strengths could be used to inform decisions regarding activity involving the real world objects. Similarly, the duration of the sequence for which each dimension remains in its initial singleton cluster

$$u(s) = \max_{i \in 1, \dots, S} k_i : |k_i(s)| = 1 \quad (5.43)$$

could be used as an ordinal measure of the extent to which the real world object associated to the dimension requires individual treatment.

## 5.3 Experimental data

Experiments are performed on synthetically generated data, and also on a real world dataset of exit counts at tube stations on the London Underground network. The synthetic data is not designed to be directly comparable to the real world data, but rather to have a known clustering structure and to have enough replicates to ensure the quality of the parameter estimates is reasonably high.

Hierarchical clustering sequences will be computed for both the synthetic and the real world data. Short-lists for finding each optimal cluster merger will use the inclusion criteria (5.35), with  $n = 5$ , and their cheap cluster dissimilarity measures will be as per (5.34). Elements  $k_i$  of each sequence with sufficiently low  $|\mathcal{C}_i|$  will be used to run prediction tests using particle filters. Additionally, the known clustering structure of the synthetic data will be compared to the clustering in its hierarchical sequence with the same number of clusters.

### 5.3.1 Synthetic data

Synthetic data will be generated with  $T = 10$ ,  $S = 200$ ,  $D = 100$ , and a random clustering  $k$  such that  $|\mathcal{C}| = 25$ . Parameters  $A_t, \mathbf{m}_t, V_t$  will be generated in the same manner as in chapter 4.  $\beta$  parameters will be drawn



$$\beta_{0,s} \sim \text{i.i.d. } \mathcal{N}\left(0, \frac{1}{4}\right), \beta_{1,s} \sim \text{i.i.d. } \mathcal{N}\left(1, \frac{1}{16}\right).$$

The clustering is generated by randomly sampling  $S$  points in  $(0, 1)^2$  and performing a k-means clustering for 25 clusters starting from a random initialisation. The Euclidean distances between the points will be used as the basis for the cheap cluster dissimilarity measure  $d_2$ .

### 5.3.2 TfL data

The real data is a set of exit counts for tube stations on the London Underground network, obtained from Transport for London (TfL). Counts are aggregated over disjoint 30 minute intervals throughout the opening hours of 4:00AM to 01:00AM every day. This gives 42 time points per day. Data for 10 different week days are available, which are assumed to meet the modelling assumption of being i.i.d. replicates of each other.

The dataset contains exit counts for each of 374 stations, but the data for many of these stations is mostly zeros. A large number of zeros in the dataset will strongly corrupt the results, so these pathological stations were removed. The removal criteria was a threshold of 90% of the data for any station being zero. All stations that exceeded the threshold were removed. This procedure resulted in the removal of 108 stations from the dataset, and leaving 266 stations in it. The final dataset therefore is of size  $T \times S \times D = 42 \times 266 \times 10$ .

Meta-data for the dataset includes the longitude  $\lambda_i$  and latitude  $\phi_i$  of each station  $s_i$ , allowing geographical distances between stations to be approximated via the haversine formula:

$$d^{\text{geo}}(s_1, s_2) = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (5.44)$$

where  $r = 6,371,000$  is the approximate radius of the Earth in metres, and a spherical approximation is made to its true shape.

## 5.4 Experiments

Hierarchical clustering sequences will be constructed for both the synthetic and TfL datasets. For each sequence, the behaviour of the mean absolute reconstruction errors will be recorded:

$$\begin{aligned} \bar{R}_k(\hat{\theta}) = \frac{1}{Z} & \left[ \sum_{t=1}^T \left\{ \sum_{s=1}^S \left| \bar{Y}_{t,s} - \widehat{Y}_{t,s}(\hat{\theta}) \right| + \left| \sqrt{\bar{Y}_{t,s}^2} - \sqrt{\widehat{Y}_{t,s}^2(\hat{\theta})} \right| \right. \right. \\ & + \sum_{s_1=1}^{S-1} \sum_{s_2=s_1+1}^S \left. \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t,s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t,s_2}(\hat{\theta})} \right| \right\} \\ & + \sum_{t=1}^{T-1} \sum_{s_1, s_2=1}^S \left. \left| \sqrt{\bar{Y}_{t,s_1} \bar{Y}_{t+1, s_2}} - \sqrt{\widehat{Y}_{t,s_1} \widehat{Y}_{t+1, s_2}(\hat{\theta})} \right| \right] \end{aligned} \quad (5.45)$$

where

$$Z = 2TS + \frac{1}{2}TS(S-1) + (T-1)S^2 \quad (5.46)$$

For the synthetic dataset, the clustering  $k_i$  in the sequence with  $|\mathcal{C}_i| = 25$  will be compared to the true clustering. The proportion of dimensions allocated to the correct cluster will be recorded:

$$\gamma = \frac{1}{S} \sum_{s=1}^S \mathbb{1}_{k_i(s)=k^{\text{true}}(s)} \quad i: |\mathcal{C}_i| = 25 \quad (5.47)$$

The short-list parameter  $n$ , which controls the maximum size of each short-list  $k_i^*$  via the inequality  $|k_i^*| \leq |\mathcal{C}_i|n$  is set to  $n = 5$ .

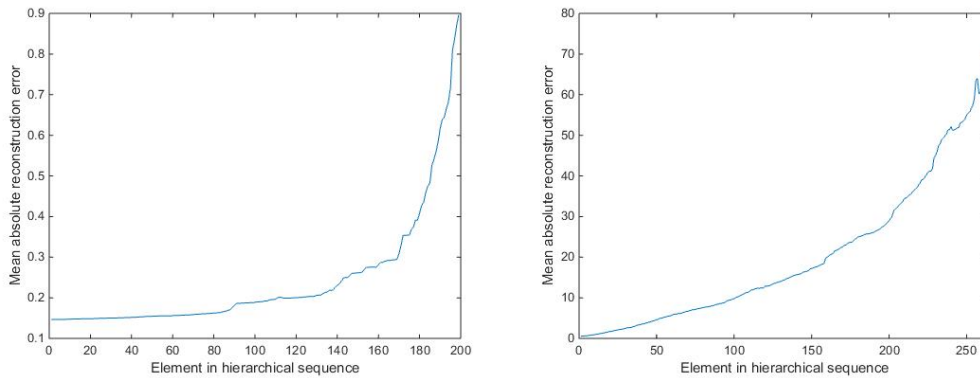
The modelled values of isolated duration  $u(s)$  and pairwise bond strength  $b(s_1, s_2)$  that result from the sequences are recorded. As the pairwise bonds are difficult to analyse, or even visualise, the number of pairwise bonds that exist at each iteration of the sequence is also recorded. When normalised by the total number of possible pairwise that exist, this gives the proportion of all station pairs that currently share a cluster at each iteration.

All experiments are written and performed in MATLAB.

## 5.5 Results

The mean absolute reconstruction errors (5.45) for the hierarchical clustering procedures are shown in fig 5.1. The procedure for the synthetic data took 59:27 (min:sec), and the procedure for the TfL data took 174:41 to run. For the synthetic data, the clustering in the hierarchical sequence with 25 clusters,  $k_i: \mathcal{C}_i = 25$ , was compared to the true clustering that generated the data. Only 5 elements were correctly clustered out of 200, giving the proportion  $\gamma = 0.025$ .

**Figure 5.1:** Mean absolute reconstruction errors for the hierarchical clustering sequences computed for synthetic data (left) and the TfL data (right).



The plots in fig 5.2 show for how much of the TfL clustering sequence did each station remain in a singleton cluster, and how quickly large clusters formed. Stations that remain isolated appear as small circles in the left plot. The majority of the smallest circles are centrally located. In the right plot, large jumps in the curve indicate mergers of large clusters. As the number of station pairs  $(s_1, s_2)$  that share each cluster is quadratic in the cluster size, mergers of two large clusters are easily identifiable. Equivalent plots for the synthetic data clustering are shown below them in the same figure.

## 5.6 Discussion

From the graphs in fig 5.1, the increase in reconstruction error as clusters are merged together is clearly seen. As every cluster merger restricts the freedom of the model to be fitted to each data dimension, this result is inevitable. The two plots are not identical in this respect though, which is possibly enlightening.

In the plot for the synthetic data, the increase is fairly shallow for most of the sequence, but undergoes a sharp acceleration towards the end. It is in fact just around the point in the sequence where modelled clusterings have approximately the true number of clusters in them. This indicates that, as the number of modelled clusters approaches the true number, the clustering sequence could perhaps be capturing some of the correlations induced by the true clustering. Once the sequence passes this point then it could be that the statistical cost of modelling fewer clusters is no longer balanced with any gain from representing the underlying truth.

Of course such a conclusion cannot be held with any strong confidence, though it does have circumstantial support. All that can be said with certainty is that the plots do not demonstrate any pathological behaviour. There are no indications that the hierarchical clustering algorithm is failing to capture any statistical information with its cluster mergers.

Knowing the names and locations of the TfL stations that are being clustered allows for a deeper analysis of the clusterings of the TfL data. The plots in fig 5.2 indicate some interesting possible conclusions. It appears from the left plot that the stations that remain in isolated clusters for the longest time are all in the city centre. When the role played by peripheral stations in transporting commuters is considered, this result is perhaps quite intuitive. Regular daily transport taken by the many workers who live in the outer city will clearly induce correlated behaviour among commuter stations. Central stations, however, have much more variety to their passengers and the routes they take. From this perspective it seems natural to cluster the commuter stations together more frequently than those in the city centre.

The upper right plot of figure 5.2 is also interesting. By showing the number of *cluster pairs* in shared clusters, which is quadratic in the size of each cluster, the plot visualises the development of large clusters throughout the sequence. From about halfway through the sequence, mergers of large clusters can be identified. Noting that these mergers must be taking place while the central stations are remaining isolated, these large merges must be of groups in the periphery of the city.

Heuristically speaking, it could be argued that the cluster dissimilarity measure

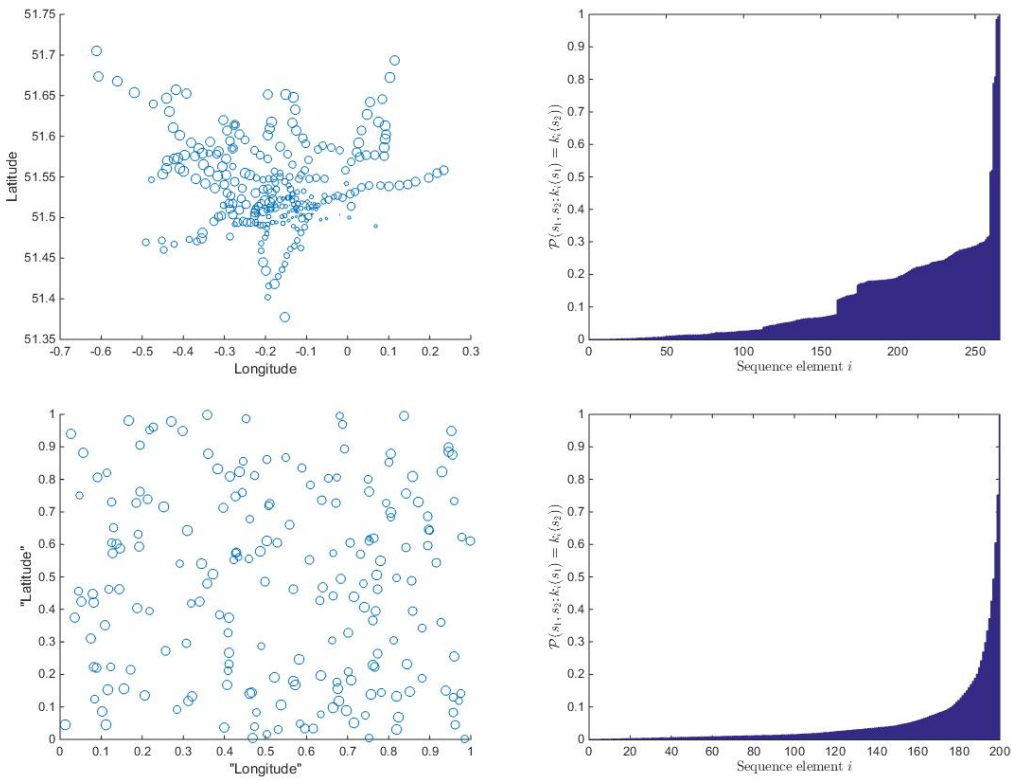
(5.27) should tend to favour smaller clusters over larger ones. Merging clusters will generally increase the reconstruction error made by the model for those stations, and this effect can be attenuated by only merging small clusters. This heuristic is supported by the lower right plot in fig 5.2, which shows the synthetic data clustering conforming to such an intuition. For large clusters of TfL stations to be forming outside of the city centre, instead of central stations merging together in smaller clusters, is somewhat surprising.

In contrast to the TfL data clustering, the synthetic data clustered in a much more regular fashion. Fig 5.2 shows equivalent plots to those just discussed for the TfL data. No pattern can be discerned from the plot of durations as singleton clusters. Such artefacts are almost certainly due to real datasets containing features not included in the model being used; they would not be expected in a synthetic dataset.

Additionally, the right plot of fig 5.2 shows a smooth curve of increasing cluster size as the sequence progresses. This closely fits the heuristic argument made above, that the dissimilarity measure should prefer the merging of small clusters over larger ones. For this pattern to be so notably absent in the TfL clustering, particularly when it is so strongly evident in the synthetic data, suggests there may be a structural difference between the behaviour of exit counts in central stations and in peripheral stations.

It should be noted that this implementation of the hierarchical clustering procedure is chiefly a proof of concept. The TfL dataset contains data for only 10 days, which is particularly insubstantial when the number of parameters in the fully dimensioned model is considered. Ideally, a much larger dataset would be used to produce the clustering sequence. In spite of this, the sequence produced had an interesting structure and was amenable to intuition. Furthermore, the running times of about 1 hour for the synthetic data, and 3 hours for the TfL data are very reasonable in comparison to likelihood based alternatives. As such, it seems likely that the algorithm proposed here could provide value as a cheap alternative to likelihood based hierarchical clustering algorithms.

**Figure 5.2:** Plots illustrating some characteristics of the Tfl hierarchical clustering (above) and the synthetic data clustering (below). The left plots shows the number of iterations of the hierarchical sequence for which each station / dimension remained in an isolated cluster. Smaller circles indicate a longer time until first merger. The right plots shows the proportion of station / dimension pairs in shared clusters as the sequences of clusterings progress. The rightmost bars corresponds to the trivial clusterings with all stations / dimensions in the same cluster.



## Chapter 6

# General Conclusions

The investigations of this thesis have proved informative and enlightening. By exploring and establishing the effect of making variational approximation to composite likelihoods, some clarity has been provided on when either of the frameworks are appropriate for performing inference. In addition, the use of non-likelihood based methods in both parameter estimation and hierarchical clustering has been shown to provide a positive trade-off. The computational cost of inference was significantly reduced within frameworks that remain effective in practice.

The findings of each chapter are now summarised in turn, before a general discussion of results.

The most significant result of the experiments conducted in chapter 3 concerns the initial hypothesis of the chapter. It is unclear *a priori* how variational parameter estimates would be affected by changing the block size of composite likelihoods. Larger block sizes would be conditioned on more data, which suggests they would become more accurate. Variational approximations would become less tight though, which could negatively affect their accuracy. In the chapter, it was hypothesised that increasing block size would result in less accurate estimators. The experiments of the chapter were able to shed some light on how the effects interacted with each other.

In regard to this question, there was a notable difference in outcomes across the difference experiments. For the bias experiment, the two effects appeared to be quite well balanced; VEM estimators stayed close to each other as block size

increased, while the corresponding SEM estimators moved increasingly towards the gold standard  $\hat{\theta}_{500}^{\text{SEM}}$ .

There was also evidence for a balance of effects from the smoothing and prediction experiments, but with differing net effects for the two experiments. The prediction experiment agreed with the bias experiment, in that both  $L1$  and  $L2$  loss improved with increasing block size. For the smoothing experiment though, the net effect was in the opposite direction. Both  $L1$  and  $L2$  losses for variational smoothing estimates increased slightly with increasing block size. It seems that the net effect of the interaction between positive and negative impacts depends on the application in question.

It is also apparent that the trade-off between cost and accuracy made when choosing between VEM and SEM estimators is also application dependent. For most of the contexts explored in the experiments, the SEM estimators proved more accurate than the VEM estimators. This accuracy comes at a large increase in computation cost though, so the question of choosing one method or the other will depend on the complete context of a given application. Only for point estimates in prediction tasks was a more certain recommendation available. As VEM estimators produced predictions with lower  $L1$  cost than SEM estimators, the trade-offs in this case point to their use in such tasks.

In chapter 4, the results of the experiments showed the expediency of using factorised Gaussians as variational approximations. The VEM estimators themselves were not shown to provide a positive trade-off between accuracy and computation cost, but when used in the computation of  $\text{SEM}(Q_{\Pi N})$ , the trade-off was strongly positive. This was increasingly true as the latent dimensionality increased. Finding the closest general Gaussians to posteriors has a quadratic cost in the latent dimensionality  $S$ , while for factorised Gaussians it is only linear. This explains the observation that the added cost of drawing importance samples was not a significant proportion of the total cost of parameter estimation.

$\text{SEM}(Q_{\Pi N})$  estimators, i.e. SEM estimators computed using Gaussians in  $Q_{\Pi N}$  as proposal distributions, performed as well as  $\text{SEM}(Q_N)$  estimators in



smoothing and prediction tasks. They were generally indistinguishable in terms of performance, and sometimes even outperformed the  $\text{SEM}(Q_N)$  estimators. Considering the significant cost reduction from using  $Q_{\Pi N}$  as the class of tractable distributions for variational approximations, the trade-off between cost and accuracy clearly favours  $\text{SEM}(Q_{\Pi N})$  estimators.

The results of the experiments also show that the use of Gaussians as variational approximations to the posteriors of the model (4.4) may not be well justified. In both the smoothing and the prediction experiments, the true parameters performed significantly worse than the likelihood based estimators (excluding  $\text{VEM}(Q_{\Pi N})$  in some instances). This suggests that the likelihood based estimators were learning parameter values under the constraint that they produced good Gaussian approximations to posteriors. Additionally, the scale at which the true parameters were outperformed suggests that Gaussian approximations are not necessarily appropriate as proposal distributions in particle filters. A further experiment, exploring the performance of the VEM and SEM estimators using a variety of proposal distributions in particle filters would shed further light on their utility.

Crucial to the inference conducted during both experiments was the introduction of the surrogate marginals  $\tilde{P}(X)$ . These approximations to the latent prior at each time point break the functional dependencies on parameters associated to previous time steps. Each component in a composite likelihood would otherwise have extremely complex dependencies on all previous parameters. Observing new data would require new derivations of parameter updates.

The per-time results of the smoothing experiment in chapter 4, as shown in fig 4.1, support the recommendation of using the surrogate marginals in parameter estimation. The plots in the figure show that smoothing estimates do not have decreasing accuracy with time, suggesting the biases introduced via the surrogates are not significant. By choosing method of moments estimates of the priors to be the surrogates, the law of large numbers ensures convergence to their true values with increasing data. As such, their integration into inference algorithms can be justified theoretically, and the empirical evidence of accurate predictions justifies their use

post-hoc.

In chapter 5, the proposed hierarchical clustering algorithm allowed model based cluster dissimilarities to be computed without having to perform expensive likelihood function evaluations. When implemented on both real and synthetic datasets, the resulting clustering sequences were computed quickly. Though not formally recorded in any experiments, the current author's personal experience with likelihood based hierarchical clustering algorithms has found them to take significantly longer to run.

The results of applying the clustering algorithm to both datasets were also encouraging. The synthetically generated data produced a clustering sequence that seemed to systematically avoid choosing large clusters to merge. If this was the case for all implementations then the algorithm would only have limited utility, but the TfL clustering showed that it is not a persistent feature of the algorithm itself. The underlying characteristics of the TfL data dominated the merger selection procedure, producing a clustering sequence that appears unique to the data. Furthermore, its idiosyncrasies can possibly be explained with context driven intuition regarding the nature of different TfL stations. The contrast between the two clustering sequences shows that the algorithm has a reasonable preference for small clusters, but this preference does not exclusively dominate the cluster merging procedure.

The algorithm itself still can be improved upon. The decision to construct the cluster dissimilarity (5.27) as the un-weighted sum of reconstructed error moments (5.28), for example, was somewhat arbitrary. Favouring some elements of the sum in (5.28) over others could produce superior results. Further research into improving the algorithm would provide insight into its utility, and also improve its performance.

In general, the use of non-likelihood based methods seems a promising avenue of research. They can be used to complement, facilitate, or substitute for likelihood based methods. Any data that is produced in i.i.d. replicates can potentially have method of moments approximations incorporated into inference frameworks for them. The positive results of the current thesis suggest such strategies could

provide beneficial trade-offs that significantly reduce the computational costs of inference.

A note of caution is also appropriate though. The varied quality of inference made using variational approximations illustrates that trade-offs can be complex. The choice of tractable class  $Q$  when making variational approximations can be significant, and what works for one model might not translate to others. Some good advice for any practitioners would be to verify the positive trade-offs in every novel application of an approximation method.

To summarise the results of the current thesis, it could be said that approximate inference in latent variable models has a lot of potential for introducing computational savings. Many approximations prove to be positive trade-offs with minimal statistical cost. Of course, any particular approximation in a given context needs its statistical efficacy to be established prior to its use. This point is exemplified by the poor performance of the true parameters in the smoothing and prediction tasks of chapter 4. When trade-offs do favour a cheap approximation though, the benefits can be significant.



# Bibliography

- S. Aihara, A. Bagchi, and S. Saha. On parameter estimation of stochastic volatility models from stock data using particle filter-application to aex index. *International journal of innovative computing, information and control*, 5(1):17–27, 2009.
- A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.
- C. Andrieu and A. Doucet. Online expectation-maximisation type algorithms for parameter estimation in general state space models. In *Proceedings (ICASSP 03)*, volume 6 of *International Conference on Accoustics, Speech, and Signal Processing*. IEEE, 2003.
- C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodolgy)*, 72(3):269–342, 2010.
- Y. F. Atchad . An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodology and Computing in applied Probability*, 8(2): 235–254, 2006.
- D. Barber and W. Wiergerinck. Tractable variational structures for approximating graphical models. *Advances in Neural Information Processing Systems*, pages 183–189, 1999.

- M.J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, UCL, The Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, London, WC1N 3AR, May 2003.
- J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.
- C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2007.
- M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. *Ann. Instit. Statist. Math.*, (62):61–89, 2010.
- L. Buesing, J. H. Macke, and M. Sahani. Spectral learning of linear dynamics from generalised-linear observations with application to neural population data. In *Advances in neural information processing systems*, pages 1682–1690, 2012.
- M. Y. Byron, A. Afshar, G. Santhanam, S. I. Ryu, and K. V. Shenoy. Extracting dynamical structure embedded in neural activity. In *Advances in neural information processing systems*, pages 1545–1552, 2005.
- M. Y. Byron, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.
- F. Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013, 2013.
- V. Chandrasekaran and M. I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13): E1181–E1190, 2013.
- S. Chib, F. Nardari, and N. Shephard. Analysis of high dimensional multivariate stochastic volatility models. *Journal of Econometrics*, 134(2):341–371, 2006.

- N. Chopin, P.E. Jacob, and O. Papaspiliopoulos. SMC<sup>2</sup>: an efficient algorithm for sequential analysis of state-space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2012.
- G. Claeskens and N.L. Hjort. *Model Selection and Model Averaging*. Cambridge University Press, Cambridge, 2008.
- D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, 2002.
- D. Crisan and T. Lyons. Nonlinear filtering and measure-valued processes. *Probability Theory and Related Fields*, 109(2):217–244, 1997.
- D. Crisan and T. Lyons. A particle approximation of the solution of the Kushner-Stratonovitch equation. *Probability Theory and Related Fields*, 115(4):549–578, 1999.
- P. Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.
- P. Del Moral. *Feynman-Kac Formulae: genealogical and interacting particle systems with applications*. New York: Springer, 2004.
- P. Del Moral and A. Guionnet. Central limit theorem for nonlinear filtering and interacting particle systems. *Annals of Applied Probability*, pages 275–297, 1999a.
- P. Del Moral and A. Guionnet. On the stability of measure valued processes with applications to filtering. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 329(5):429–434, 1999b.
- P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. In *Annales de l'IHP Probabilités et statistiques*, volume 37, pages 155–194, 2001.

- P. Del Moral, A. Doucet, and S. Singh. A backward particle interpretation of Feynman-Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(05):947–975, 2010a.
- P. Del Moral, A. Doucet, and S. Singh. Forward smoothing using sequential Monte Carlo. *arXiv preprint arXiv:1012.5390*, 2010b.
- P. Del Moral, A. Doucet, A. Jasra, et al. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012.
- 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*, 39:1–38, 1977.
- J. V. Dillon and G. Lebanon. Stochastic composite likelihood. *The Journal of Machine Learning Research*, 11:2597–2633, 2010.
- R. Douc, A. Garivier, E. Moulines, and J. Olsson. On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state spaces models. *arXiv preprint arXiv:0904.0316*, 2009.
- A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.
- A. Doucet, M. Pitt, and G. Deligiannidis. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *arXiv pre-print*, March 2014. arXiv:1210.1871v3.
- S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.
- R. O. Duda, P. E. Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.
- C. Faes, J.T. Ormerod, and M.P. Wand. Variational Bayesian inference for parametric and non-parameteric regression with missing data. Technical re-



- port, Centre for Statistical and Survey Methodology, University of Wollongong, <http://ro.uow.edu.au/cssmwp/58>, 2010. Working paper 07-10.
- P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, (97):447–464, 2010.
- C. Fernández and M. F. J. Steel. Multivariate Student-t regression models: Pitfalls and inference. *Biometrika*, 86(1):153–167, 1999.
- M. Fraccaro, U. Paquet, and O. Winther. Indexable probabilistic matrix factorization for maximum inner product search. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal on Computer Vision*, 40(1):25–47, 2000.
- B. J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. *Advances in Neural Information Processing Systems*, pages 737–743, 2001.
- X. Gao and P. X.-K. Song. Composite likelihood Bayesian information criteria for model selection in high dimensional data. *Journal of the American Statistical Association*, 105:1531–1540, 2010.
- X. Gao and P. X.-K. Song. Composite likelihood EM algorithm with applications to multivariate hidden Markov model. *Statistica Sinica*, pages 165–185, 2011.
- S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.
- C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, pages 473–483, 1992.
- H. Geys, G. Molenberghs, and L. M. Ryan. Pseudolikelihood modeling of multivariate outcomes in developmental toxicology. *Journal of the American Statistical Association*, 94:734–745, 1999.

- Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine Learning*, 29:245–273, 1997.
- M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.
- V. P. Godambe. An optimum property of regular maximum likelihood estimation. *The Annals of Mathematical Statistics*, 31(4):1208–1211, December 1960.
- S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465), 2004.
- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113, 1993.
- F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.
- Y. Halpern and D. Sontag. Unsupervised learning of noisy-OR Bayesian networks. *arXiv preprint arXiv:1309.6834*, 2013.
- S. Harmeling and C. K. I. Williams. Greedy learning of binary latent trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(6):1087–1097, 2011.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, February 2009.
- W. K. Hastings. Monte Carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- R. J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics & Probability Letters*, 4(2):53–56, 1986.

- K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.
- A. Ihler, J. Fisher, and A. S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, 2005.
- T. S. Jaakkola. Tutorial on variational approximation methods. *Advanced Mean Field Methods: Theory and Practice*, pages 129–160, 2001.
- T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer, 1998.
- A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64. Academic Press, 1970.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- K. G. Jöreskog and I. Moustaki. Factor analysis of ordinal variables: a comparison of three approaches. *Multivariate Behavioural Research*, 36(3):347–387, 2001.
- S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108, 1961.
- J. T. Kent. Robust properties of likelihood ratio tests. *Biometrika*, 69(1):19–27, 1982.

- B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.
- D. P. Kroese. *Monte Carlo Methods*. PhD thesis, The University of Queensland, 2011.
- J. E. Kulkarni and L. Paninski. Common-input models for multiple neural spike-train data. *Network: Computation in Neural Systems*, 18(4):375–407, 2007.
- S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.
- G. Liang and B. Yu. Maximum pseudo likelihood estimation in network tomography. *Signal Processing, IEEE Transactions on*, 51(8):2043–2053, 2003.
- C. Liu and D. B. Rubin. ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, pages 19–39, 1995.
- S. Livingstone, M. Betancourt, S. Byrne, and M. Girolami. On the geometric ergodicity of Hamiltonian Monte Carlo. *arXiv preprint arXiv:1601.08057*, 2016.
- S. N. MacEachern and L M. Berliner. Subsampling the Gibbs sampler. *The American Statistician*, 48(3):188–190, 1994.
- R. McEliece, D.J.C. MacKay, and J. Cheng. Turbo decoding as an instance of Pearl’s ”Belief Propagation” algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152, 1998.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- G. Molenberghs and G. Verbeke. *Models for discrete longitudinal data*. Springer, New York, 2005.
- R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993. Technical Report CRG-TR-93-1.

- R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.
- D.J. Nott, S.L. Tan, M. Villani, and R. Kohn. Regression density estimation with variational methods and stochastic approximation. *Journal of Computational And Graphical Statistics*, 21(3):797–820, 2012.
- J.T. Ormerod. Grid based variational approximations. *Computational Statistics and Data Analysis*, 55(1):45–56, 2011.
- J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- G. Poyiadjis, A. Doucet, and S.S. Singh. Particle methods for optimal filter derivative: application to parameter estimation. In *Proceedings (ICASSP 05)*, volume 5 of *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2005.
- G. Poyiadjis, A. Doucet, and S. Singh. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.
- H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.
- G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.
- T. G. Roosta, M. J. Wainwright, and S. S. Sastry. Convergence analysis of reweighted sum-product algorithms. *Signal Processing, IEEE Transactions on*, 56(9):4293–4305, 2008.
- A. Rotnitzky and N. P. Jewell. Hypothesis testing of regression parameters in semi-parametric generalized linear models for cluster correlated data. *Biometrika*, 77(3):485–497, 1990.

- R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.
- L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. *Advances in Neural Information Processing Systems*, pages 486–492, 1996.
- A. C. Smith and E. N. Brown. Estimating a state-space model from point process observations. *Neural Computation*, 15(5):965–991, 2003.
- A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. *Advances in neural information processing systems*, pages 11–11, 1993.
- M. Svensén and C. M. Bishop. Robust Bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2005.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In *Workshop on Inference and Estimation in Probabilistic Time-Series Models*, volume 2, 2008.
- L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- C. Varin. On composite marginal likelihoods. *Advances in Statistical Analysis*, 92(1):1–28, February 2008.
- C. Varin and P. Vidoni. A note on composite likelihood inference and model selection. *Biometrika*, 92(3):519–528, September 2005.
- C. Varin, G. Høst, and Ø. Skare. Pairwise likelihood inference in spatial generalized linear mixed models. *Computational statistics & data analysis*, 49(4):1173–1191, 2005.

- C. Varin, R. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21:5–42, 2011.
- V. G. S. Vasdeskis, S. Cagnone, and I. Moustaki. A composite likelihood inference in latent variable models for ordinal longitudinal responses. *Psychometrika*, 77(3):425–441, 2012.
- C. Vergé, C. Dubarry, P. Del Moral, and E. Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260, 2015.
- M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.
- M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation and approximate ML estimation by pseudo-moment matching. In *9th Workshop on Artificial Intelligence and Statistics*, Key West, Florida, January 2003.
- M.P. Wand, J.T. Ormerod, S.A. Padoan, and R. Fruhwirth. Variational Bayes for elaborate distributions. Technical report, Centre for Statistical and Survey Methodology, University of Wollongong, <http://ro.uow.edu.au/cssmwp/56>, 2011. Working paper 05-10.
- B. Wang and D. M. Titterton. Lack of consistency of mean field and variational Bayes approximations for state space models. *Neural Processing Letters*, 20(3):151–170, 2004.
- Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.
- W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the 16th Conference on Uncertainty in*

*Artificial Intelligence*, pages 626–633, San Francisco, 2000. Morgan Kaufman Publishers.

J. M. Winn and C. M. Bishop. Variational message passing. In *Journal of Machine Learning Research*, pages 661–694, 2005.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8: 236–239, 2003.

S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: review and experimental comparison. *Pattern Recognition*, 46(7):1772–1788, 2013.

S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, 13(11):1491–1506, 2004.