VARIATIONAL ALGORITHMS FOR APPROXIMATE BAYESIAN INFERENCE

by

Matthew J. Beal

M.A., M.Sci., Physics, University of Cambridge, UK (1998)

The Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR

A Thesis submitted for the degree of
Doctor of Philosophy of the University of London

May 2003

Abstract

The Bayesian framework for machine learning allows for the incorporation of prior knowledge in a coherent way, avoids overfitting problems, and provides a principled basis for selecting between alternative models. Unfortunately the computations required are usually intractable. This thesis presents a unified variational Bayesian (VB) framework which approximates these computations in models with latent variables using a lower bound on the marginal likelihood.

Chapter 1 presents background material on Bayesian inference, graphical models, and propagation algorithms. Chapter 2 forms the theoretical core of the thesis, generalising the expectation-maximisation (EM) algorithm for learning maximum likelihood parameters to the VB EM algorithm which integrates over model parameters. The algorithm is then specialised to the large family of conjugate-exponential (CE) graphical models, and several theorems are presented to pave the road for automated VB derivation procedures in both directed and undirected graphs (Bayesian and Markov networks, respectively).

Chapters 3-5 derive and apply the VB EM algorithm to three commonly-used and important models: mixtures of factor analysers, linear dynamical systems, and hidden Markov models. It is shown how model selection tasks such as determining the dimensionality, cardinality, or number of variables are possible using VB approximations. Also explored are methods for combining sampling procedures with variational approximations, to estimate the tightness of VB bounds and to obtain more effective sampling algorithms. Chapter 6 applies VB learning to a long-standing problem of scoring discrete-variable directed acyclic graphs, and compares the performance to annealed importance sampling amongst other methods. Throughout, the VB approximation is compared to other methods including sampling, Cheeseman-Stutz, and asymptotic approximations such as BIC. The thesis concludes with a discussion of evolving directions for model selection including infinite models and alternative approximations to the marginal likelihood.

Acknowledgements

I am very grateful to my advisor Zoubin Ghahramani for his guidance in this work, bringing energy and thoughtful insight into every one of our discussions. I would also like to thank other senior Gatsby Unit members including Hagai Attias, Phil Dawid, Peter Dayan, Geoff Hinton, Carl Rasmussen and Sam Roweis, for numerous discussions and inspirational comments.

My research has been punctuated by two internships at Microsoft Research in Cambridge and in Redmond. Whilst this thesis does not contain research carried out in these labs, I would like to thank colleagues there for interesting and often seductive discussion, including Christopher Bishop, Andrew Blake, David Heckerman, Nebojsa Jojic and Neil Lawrence.

Amongst many others I would like to thank especially the following people for their support and useful comments: Andrew Brown, Nando de Freitas, Oliver Downs, Alex Gray, Yoel Haitovsky, Sham Kakade, Alex Korenberg, David MacKay, James Miskin, Quaid Morris, Iain Murray, Radford Neal, Simon Osindero, Lawrence Saul, Matthias Seeger, Amos Storkey, Yee-Whye Teh, Eric Tuttle, Naonori Ueda, John Winn, Chris Williams, and Angela Yu.

I should thank my friends, in particular Paola Atkinson, Tania Lillywhite, Amanda Parmar, James Tinworth and Mark West for providing me with various combinations of shelter, companionship and retreat during my time in London. Last, but by no means least I would like to thank my family for their love and nurture in all my years, and especially my dear fiancée Cassandre Creswell for her love, encouragement and endless patience with me.

The work in this thesis was carried out at the Gatsby Computational Neuroscience Unit which is funded by the Gatsby Charitable Foundation. I am grateful to the Institute of Physics, the NIPS foundation, the UCL graduate school and Microsoft Research for generous travel grants.

Contents

Abstract **2**

Acknowledgements **3**

Contents **4**

List of figures **8**

List of tables **11**

List of algorithms **12**

1 Introduction **13**
1.1 Probabilistic inference 16
1.1.1 Probabilistic graphical models: directed and undirected networks . . . 17
1.1.2 Propagation algorithms 19
1.2 Bayesian model selection 24
1.2.1 Marginal likelihood and Occam's razor 25
1.2.2 Choice of priors 27
1.3 Practical Bayesian approaches 32
1.3.1 Maximum a posteriori (MAP) parameter estimates 33
1.3.2 Laplace's method 34
1.3.3 Identifiability: aliasing and degeneracy 35
1.3.4 BIC and MDL 36
1.3.5 Cheeseman & Stutz's method 37
1.3.6 Monte Carlo methods 38
1.4 Summary of the remaining chapters 42

2 Variational Bayesian Theory **44**
2.1 Introduction 44
2.2 Variational methods for ML / MAP learning 46
2.2.1 The scenario for parameter learning 46
2.2.2 EM for unconstrained (exact) optimisation 48

2.2.3 EM with constrained (approximate) optimisation . . . 49
2.3 Variational methods for Bayesian learning . . . 53
2.3.1 Deriving the learning rules . . . 53
2.3.2 Discussion . . . 58
2.4 Conjugate-Exponential models . . . 64
2.4.1 Definition . . . 64
2.4.2 Variational Bayesian EM for CE models . . . 66
2.4.3 Implications . . . 69
2.5 Directed and undirected graphs . . . 73
2.5.1 Implications for directed networks . . . 73
2.5.2 Implications for undirected networks . . . 74
2.6 Comparisons of VB to other criteria . . . 75
2.6.1 BIC is recovered from VB in the limit of large data . . . 75
2.6.2 Comparison to Cheeseman-Stutz (CS) approximation . . . 76
2.7 Summary . . . 80

3 Variational Bayesian Hidden Markov Models **82**
3.1 Introduction . . . 82
3.2 Inference and learning for maximum likelihood HMMs . . . 83
3.3 Bayesian HMMs . . . 88
3.4 Variational Bayesian formulation . . . 91
3.4.1 Derivation of the VBEM optimisation procedure . . . 92
3.4.2 Predictive probability of the VB model . . . 97
3.5 Experiments . . . 98
3.5.1 Synthetic: discovering model structure . . . 98
3.5.2 Forwards-backwards English discrimination . . . 99
3.6 Discussion . . . 104

4 Variational Bayesian Mixtures of Factor Analysers **106**
4.1 Introduction . . . 106
4.1.1 Dimensionality reduction using factor analysis . . . 107
4.1.2 Mixture models for manifold learning . . . 109
4.2 Bayesian Mixture of Factor Analysers . . . 110
4.2.1 Parameter priors for MFA . . . 111
4.2.2 Inferring dimensionality using ARD . . . 114
4.2.3 Variational Bayesian derivation . . . 115
4.2.4 Optimising the lower bound . . . 119
4.2.5 Optimising the hyperparameters . . . 122
4.3 Model exploration: birth and death . . . 124
4.3.1 Heuristics for component death . . . 126
4.3.2 Heuristics for component birth . . . 127

4.3.3 Heuristics for the optimisation endgame . . . 130
4.4 Handling the predictive density . . . 130
4.5 Synthetic experiments . . . 132
4.5.1 Determining the number of components . . . 133
4.5.2 Embedded Gaussian clusters . . . 133
4.5.3 Spiral dataset . . . 135
4.6 Digit experiments . . . 138
4.6.1 Fully-unsupervised learning . . . 138
4.6.2 Classification performance of BIC and VB models . . . 141
4.7 Combining VB approximations with Monte Carlo . . . 144
4.7.1 Importance sampling with the variational approximation . . . 144
4.7.2 Example: Tightness of the lower bound for MFAs . . . 148
4.7.3 Extending simple importance sampling . . . 151
4.8 Summary . . . 157

5 Variational Bayesian Linear Dynamical Systems 159
5.1 Introduction . . . 159
5.2 The Linear Dynamical System model . . . 160
5.2.1 Variables and topology . . . 160
5.2.2 Specification of parameter and hidden state priors . . . 163
5.3 The variational treatment . . . 168
5.3.1 VBM step: Parameter distributions . . . 170
5.3.2 VBE step: The Variational Kalman Smoother . . . 173
5.3.3 Filter (forward recursion) . . . 174
5.3.4 Backward recursion: sequential and parallel . . . 177
5.3.5 Computing the single and joint marginals . . . 181
5.3.6 Hyperparameter learning . . . 184
5.3.7 Calculation of $\mathcal{F}$ . . . 185
5.3.8 Modifications when learning from multiple sequences . . . 186
5.3.9 Modifications for a fully hierarchical model . . . 189
5.4 Synthetic Experiments . . . 189
5.4.1 Hidden state space dimensionality determination (no inputs) . . . 189
5.4.2 Hidden state space dimensionality determination (input-driven) . . . 191
5.5 Elucidating gene expression mechanisms . . . 195
5.5.1 Generalisation errors . . . 198
5.5.2 Recovering gene-gene interactions . . . 200
5.6 Possible extensions and future research . . . 201
5.7 Summary . . . 204

6 Learning the structure of discrete-variable graphical models with hidden variables 206

6.1 Introduction . . . 206
6.2 Calculating marginal likelihoods of DAGs . . . 207
6.3 Estimating the marginal likelihood . . . 210
6.3.1 ML and MAP parameter estimation . . . 210
6.3.2 BIC . . . 212
6.3.3 Cheeseman-Stutz . . . 213
6.3.4 The VB lower bound . . . 215
6.3.5 Annealed Importance Sampling (AIS) . . . 218
6.3.6 Upper bounds on the marginal likelihood . . . 222
6.4 Experiments . . . 223
6.4.1 Comparison of scores to AIS . . . 226
6.4.2 Performance averaged over the parameter prior . . . 232
6.5 Open questions and directions . . . 236
6.5.1 AIS analysis, limitations, and extensions . . . 236
6.5.2 Estimating dimensionalities of the incomplete and complete-data models 245
6.6 Summary . . . 247

7 Conclusion **250**
7.1 Discussion . . . 250
7.2 Summary of contributions . . . 254

Appendix A Conjugate Exponential family examples **259**

Appendix B Useful results from matrix theory **262**
B.1 Schur complements and inverting partitioned matrices . . . 262
B.2 The matrix inversion lemma . . . 263

Appendix C Miscellaneous results **265**
C.1 Computing the digamma function . . . 265
C.2 Multivariate gamma hyperparameter optimisation . . . 266
C.3 Marginal KL divergence of gamma-Gaussian variables . . . 267

Bibliography **270**

List of figures

1.1 The elimination algorithm on a simple Markov network . . . 20
1.2 Forming the junction tree for a simple Markov network . . . 22
1.3 The marginal likelihood embodies Occam's razor . . . 27

2.1 Variational interpretation of EM for ML learning . . . 50
2.2 Variational interpretation of constrained EM for ML learning . . . 51
2.3 Variational Bayesian EM . . . 56
2.4 Hidden-variable / parameter factorisation steps . . . 59
2.5 Hyperparameter learning for VB EM . . . 62

3.1 Graphical model representation of a hidden Markov model . . . 83
3.2 Evolution of the likelihood for ML hidden Markov models, and the subsequent VB lower bound. . . . 100
3.3 Results of ML and VB HMM models trained on synthetic sequences. . . . 101
3.4 Test data log predictive probabilities and discrimination rates for ML, MAP, and VB HMMs . . . 103

4.1 ML Mixtures of Factor Analysers . . . 110
4.2 Bayesian Mixtures of Factor Analysers . . . 114
4.3 Determination of number of components in synthetic data . . . 134
4.4 Factor loading matrices for dimensionality determination . . . 135
4.5 The Spiral data set of Ueda et. al . . . 136
4.6 Birth and death processes with VBMFA on the Spiral data set . . . 137
4.7 Evolution of the lower bound $\mathcal{F}$ for the Spiral data set . . . 137
4.8 Training examples of digits from the CEDAR database . . . 138
4.9 A typical model of the digits learnt by VBMFA . . . 139
4.10 Confusion tables for the training and test digit classifications . . . 140
4.11 Distribution of components to digits in BIC and VB models . . . 143
4.12 Logarithm of the marginal likelihood estimate and the VB lower bound during learning of the digits $\{0, 1, 2\}$ . . . 150
4.13 Discrepancies between marginal likelihood and lower bounds during VBMFA model search . . . 152

4.14 Importance sampling estimates of marginal likelihoods for learnt models of data of differently spaced clusters 156

5.1 Graphical model representation of a state-space model 161
5.2 Graphical model for a state-space model with inputs 162
5.3 Graphical model representation of a Bayesian state-space model 164
5.4 Recovered LDS models for increasing data size 190
5.5 Hyperparameter trajectories showing extinction of state-space dimensions . . . 191
5.6 Data for the input-driven LDS synthetic experiment 193
5.7 Evolution of the lower bound and its gradient 194
5.8 Evolution of precision hyperparameters, recovering true model structure 196
5.9 Gene expression data for input-driven experiments on real data 197
5.10 Graphical model of an LDS with feedback of observations into inputs 199
5.11 Reconstruction errors of LDS models trained using MAP and VB algorithms as a function of state-space dimensionality 200
5.12 Gene-gene interaction matrices learnt by MAP and VB algorithms, showing significant entries 202
5.13 Illustration of the gene-gene interactions learnt by the feedback model on expression data 203

6.1 The chosen structure for generating data for the experiments 225
6.2 Illustration of the trends in marginal likelihood estimates as reported by MAP, BIC, BICp, CS, VB and AIS methods, as a function of data set size and number of parameters 228
6.3 Graph of rankings given to the true structure by BIC, BICp, CS, VB and AIS methods 230
6.4 Differences in marginal likelihood estimate of the top-ranked and true structures, by BIC, BICp, CS, VB and AIS 232
6.5 The median ranking given to the true structure over repeated settings of its parameters drawn from the prior, by BIC, BICp, CS and VB methods 233
6.6 Median score difference between the true and top-ranked structures, under BIC, BICp, CS and VB methods. 234
6.7 The best ranking given to the true structure by BIC, BICp, CS and VB methods 235
6.8 The smallest score difference between true and top-ranked structures, by BIC, BICp, CS and VB methods 236
6.9 Overall success rates of BIC, BICp, CS and VB scores, in terms of ranking the true structure top 237
6.10 Example of the variance of the AIS sampler estimates with annealing schedule granularity, using various random initialisations, shown against the BIC and VB estimates for comparison 238

6.11 Acceptance rates of the Metropolis-Hastings proposals as a function of size of data set 240
6.12 Acceptance rates of the Metropolis-Hastings sampler in each of four quarters of the annealing schedule 242
6.13 Non-linear AIS annealing schedules 244

List of tables

2.1 Comparison of EM for ML/MAP estimation against VB EM with CE models . 70

4.1 Simultaneous determination of number of components and their dimensionalities 135
4.2 Test classification performance of BIC and VB models 142
4.3 Specifications of six importance sampling distributions 155

6.1 Rankings of the true structure amongst the alternative candidates, by MAP, BIC, BICp, VB and AIS estimates, both corrected and uncorrected for posterior aliasing 230
6.2 Comparison of performance of VB to BIC, BICp and CS methods, as measured by the ranking given to the true model 233
6.3 Improving the AIS estimate by pooling the results of several separate sampling runs 241
6.4 Rate of AIS violations of the VB lower bound, alongside Metropolis-Hastings rejection rates 241
6.5 Number of times the true structure is given the highest ranking by the BIC, BICp, CS, CS†, and VB scores 247

List of Algorithms

5.1 Forward recursion for variational Bayesian state-space models 178
5.2 Backward parallel recursion for variational Bayesian state-space models 181
5.3 Pseudocode for variational Bayesian state-space models 187
6.1 AIS algorithm for computing all ratios to estimate the marginal likelihood . . . 221
6.2 Algorithm to estimate the complete- and incomplete-data dimensionalities of a model 246

Chapter 1

Introduction

Our everyday experiences can be summarised as a series of decisions to take actions which manipulate our environment in some way or other. We base our decisions on the results of predictions or inferences of quantities that have some bearing on our quality of life, and we come to arrive at these inferences based on *models* of what we expect to observe. Models are designed to capture salient trends or regularities in the observed data with a view to predicting future events. Sometimes the models can be constructed with existing expertise, but for the majority of real applications the data are far too complex or the underlying processes not nearly well enough understood for the modeller to design a perfectly accurate model. If this is the case, we can hope only to design models that are simplifying approximations of the true processes that generated the data.

For example, the data might be a time series of the price of stock recorded every day for the last six months, and we would like to know whether to buy or sell stock today. This decision, and its particulars, depend on what the price of the stock is likely to be a week from now. There are obviously a very large number of factors that influence the price and these do so to varying degrees and in convoluted and complex ways. Even in the unlikely scenario that we knew exactly how all these factors affected the price, we would still have to gather every piece of data for each one and process it all in a short enough time to decide our course of action. Another example is trying to predict the best location to drill for oil, knowing the positions of existing drill sites in the region and their yields. Since we are unable to probe deep beneath the Earth's surface, we need to rely on a model of the geological processes that gave rise to the yields in those sites for which we have data, in order to be able to predict the best location.

The *machine learning* approach to modelling data constructs models by beginning with a flexible model specified by a set of *parameters* and then finds the setting of these model parameters that explains or fits the data best. The idea is that if we can explain our observations well, then we should also be confident that we can predict future observations well. We might also hope

that the particular setting of the best-fit parameters provides us with some understanding of the underlying processes. The procedure of fitting model parameters to observed data is termed *learning* a model.

Since our models are simplifications of reality there will inevitably be aspects of the data which cannot be modelled exactly, and these are considered noise. Unfortunately it is often difficult to know which aspects of the data are relevant for our inference or prediction tasks, and which aspects should be regarded as noise. With a sufficiently complex model, parameters can be found to fit the observed data exactly, but any predictions using this best-fit model will be sub-optimal as it has erroneously fitted the noise instead of the trends. Conversely, too simple a model will fail to capture the underlying regularities in the data and so will also produce sub-optimal inferences and predictions. This trade-off between the complexity of the model and its generalisation performance is well studied, and we return to it in section 1.2.

The above ideas can be formalised using the concept of probability and the rules of Bayesian inference. Let us denote the data set by $\mathbf{y}$, which may be made up of several variables indexed by j: $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_j, \ldots, \mathbf{y}_J\}$. For example, $\mathbf{y}$ could be the data from an oil well for which the variables might be measurements of the type of oil found, the geographical location of the well, its average monthly yield, its operational age, and a host of other measurable quantities regarding its local geological characteristics. Generally each variable can be real-valued or discrete. Machine learning approaches define a *generative model* of the data through a set of parameters $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$ which define a probability distribution over data, $p(\mathbf{y} \,|\, \boldsymbol{\theta})$. One approach to learning the model then involves finding the parameters $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \; p(\mathbf{y} \,|\, \boldsymbol{\theta}) \; . \tag{1.1}$$

This process is often called *maximum likelihood* learning as the parameters $\boldsymbol{\theta}^*$ are set to maximise the likelihood of $\boldsymbol{\theta}$, which is probability of the observed data under the model. The generative model may also include *latent* or *hidden* variables, which are unobserved yet interact through the parameters to generate the data. We denote the hidden variables by $\mathbf{x}$, and the probability of the data can then be written by summing over the possible settings of the hidden states:

$$p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \sum_{\mathbf{x}} p(\mathbf{x} \,|\, \boldsymbol{\theta}) p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta}) \; , \tag{1.2}$$

where the summation is replaced by an integral for those hidden variables that are real-valued. The quantity (1.2) is often called the *incomplete-data likelihood*, and the summand in (1.2) correspondingly called the *complete-data likelihood*. The interpretation is that with hidden variables in the model, the observed data is an incomplete account of all the players in the model.

For a particular parameter setting, it is possible to infer the states of the hidden variables of the model, having observed data, using Bayes' rule:

$$p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{y} \mid \boldsymbol{\theta})} \,. \tag{1.3}$$

This quantity is known as the *posterior* distribution over the hidden variables. In the oil well example we might have a hidden variable for the amount of oil remaining in the reserve, and this can be inferred based on observed measurements such as the operational age, monthly yield and geological characteristics, through the generative model with parameters $\boldsymbol{\theta}$. The term $p(\mathbf{x} \mid \boldsymbol{\theta})$ is a *prior* probability of the hidden variables, which could be set by the modeller to reflect the distribution of amounts of oil in wells that he or she would expect. Note that the probability of the data in (1.2) appears in the denominator of (1.3). Since the hidden variables are by definition unknown, finding $\boldsymbol{\theta}^*$ becomes more difficult, and the model is learnt by alternating between estimating the posterior distribution over hidden variables for a particular setting of the parameters and then re-estimating the best-fit parameters given that distribution over the hidden variables. This procedure is the well-known expectation-maximisation (EM) algorithm and is discussed in more detail in section 2.2.

Given that the parameters themselves are unknown quantities we can treat them as random variables. This is the *Bayesian* approach to uncertainty, which treats all uncertain quantities as random variables and uses the laws of probability to manipulate those uncertain quantities. The proper Bayesian approach attempts to integrate over the possible settings of all uncertain quantities rather than optimise them as in (1.1). The quantity that results from integrating out both the hidden variables and the parameters is termed the *marginal likelihood*:

$$p(\mathbf{y}) = \int d\boldsymbol{\theta} \; p(\boldsymbol{\theta}) \sum_{\mathbf{x}} p(\mathbf{x} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) \,, \tag{1.4}$$

where $p(\boldsymbol{\theta})$ is a prior over the parameters of the model. We will see in section 1.2 that the marginal likelihood is a key quantity used to choose between different models in a Bayesian model selection task. Model selection is a necessary step in understanding and representing the data that we observe. The diversity of the data available to machine learners is ever increasing thanks to the advent of large computational power, networking capabilities and the technologies available to the scientific research communities. Furthermore, expertise and techniques of analysis are always improving, giving rise to ever more diverse and complicated models for representing this data. In order to 'understand' the data with a view to making predictions based on it, we need to whittle down our models to one (or a few) to which we can devote our limited computational and conceptual resources. We can use the rules of Bayesian probability theory to entertain several models and choose between them in the light of data. These steps necessarily involve managing the marginal likelihood.

Unfortunately the marginal likelihood, $p(\mathbf{y})$, is an intractable quantity to compute for almost all models of interest (we will discuss why this is so in section 1.2.1, and see several examples in the course of this thesis). Traditionally, the marginal likelihood has been approximated either using analytical methods, for example the Laplace approximation, or via sampling-based approaches such as Markov chain Monte Carlo. These methods are reviewed in section 1.3. This thesis is devoted to one particular method of approximation, *variational Bayes*, sometimes referred to as *ensemble learning*. The variational Bayesian method constructs a lower bound on the marginal likelihood, and attempts to optimise this bound using an iterative scheme that has intriguing similarities to the standard expectation-maximisation algorithm. There are other variational methods, for example those based on Bethe and Kikuchi free energies, which for the most part are approximations rather than bounds; these are briefly discussed in the final chapter.

Throughout this thesis we assume that the reader is familiar with the basic concepts of probability and integral and differential calculus. Included in the appendix are reference tables for some of the more commonly used probability distributions.

The rest of this chapter reviews some key methods relevant to Bayesian model inference and learning. Section 1.1 reviews the use of graphical models as a tool for visualising the probabilistic relationships between the variables in a model and explains how efficient algorithms for computing the posterior distributions of hidden variables as in (1.3) can be designed which exploit independence relationships amongst the variables. In section 1.2, we address the issue of model selection in a Bayesian framework, and explain why the marginal likelihood is the key quantity for this task, and how it is intractable to compute. Since all Bayesian reasoning needs to begin with some prior beliefs, we examine different schools of thought for expressing these priors in section 1.2.2, including *conjugate*, *reference*, and *hierarchical* priors. In section 1.3 we review several practical methods for approximating the marginal likelihood, which we shall be comparing to variational Bayes in the following chapters. Finally, section 1.4 briefly summarises the remaining chapters of this thesis.

1.1 Probabilistic inference

Bayesian probability theory provides a language for representing beliefs and a calculus for manipulating these beliefs in a coherent manner. It is an extension of the formal theory of logic which is based on axioms that involve propositions that are true or false. The rules of probability theory involve propositions which have *plausibilities* of being true or false, and can be arrived at on the basis of just three *desiderata*: (1) degrees of plausibility should be represented by real numbers; (2) plausibilities should have qualitative correspondence with common sense; (3) different routes to a conclusion should yield the same result. It is quite astonishing that from just these desiderata, the product and sum rules of probability can be mathematically derived

(Cox, 1946). Cox showed that plausibilities can be measured on any scale and it is possible to transform them onto the canonical scale of probabilities that sum to one. For good introductions to probability theory the reader is referred to Pearl (1988) and Jaynes (2003).

Statistical modelling problems often involve large numbers of interacting random variables and it is often convenient to express the dependencies between these variables graphically. In particular such graphical models are an intuitive tool for visualising *conditional independency* relationships between variables. A variable a is said to be conditionally independent of b, given c if and only if $p(a, b \,|\, c)$ can be written $p(a \,|\, c)p(b \,|\, c)$. By exploiting conditional independence relationships, graphical models provide a backbone upon which it has been possible to derive efficient message-propagating algorithms for conditioning and marginalising variables in the model given observation data (Pearl, 1988; Lauritzen and Spiegelhalter, 1988; Jensen, 1996; Heckerman, 1996; Cowell et al., 1999; Jordan, 1999). Many standard statistical models, especially Bayesian models with hierarchical priors (see section 1.2.2), can be expressed naturally using probabilistic graphical models. This representation can be helpful in developing both sampling methods (section 1.3.6) and exact inference methods such as the junction tree algorithm (section 1.1.2) for these models. All of the models used in this thesis have very simple graphical model descriptions, and the theoretical results derived in chapter 2 for variational Bayesian approximate inference are phrased to be readily applicable to general graphical models.

1.1.1 Probabilistic graphical models: directed and undirected networks

A graphical model expresses a family of probability distributions on sets of variables in a model. Here and for the rest of the thesis we use the variable $\mathbf{z}$ to denote all the variables in the model, be they observed or unobserved (hidden). To differentiate between observed and unobserved variables we partition $\mathbf{z}$ into $\mathbf{z} = \{\mathbf{x}, \mathbf{y}\}$ where $\mathbf{x}$ and $\mathbf{y}$ are the sets of unobserved and observed variables, respectively. Alternatively, the variables are indexed by the subscript j, with $j \in \mathcal{H}$ the set of indices for unobserved (hidden) variables and $j \in \mathcal{V}$ the set of indices for observed variables. We will later introduce a further subscript, i, which will denote which data point out of a data set of size n is being referred to, but for the purposes of the present exposition we consider just a single data point and omit this further subscript.

Each arc between two nodes in the graphical model represents a probabilistic connection between two variables. We use the terms 'node' and 'variable' interchangeably. Depending on the pattern of arcs in the graph and their type, different independence relations can be represented between variables. The pattern of arcs is commonly referred to as the *structure* of the model.

The arcs between variables can be all *directed* or all *undirected*. There is a class of graphs in which some arcs are directed and some are undirected, commonly called *chain graphs*, but these are not reviewed here. Undirected graphical models, also called *Markov networks* or *Markov*

random fields, express the probability distribution over variables as a product over *clique potentials*:

$$p(\mathbf{z}) = \frac{1}{\mathcal{Z}} \prod_{j=1}^{J} \psi_j(C_j(\mathbf{z})) \,, \tag{1.5}$$

where $\mathbf{z}$ is the set of variables in the model, $\{C_j\}_{j=1}^{J}$ are *cliques* of the graph, and $\{\psi_j\}_{j=1}^{J}$ are a set of clique potential functions each of which returns a non-negative real value for every possible configuration of settings of the variables in the clique. Each clique is defined to be a fully connected subgraph (that is to say each clique C_j selects a subset of the variables in $\mathbf{z}$), and is usually *maximal* in the sense that there are no other variables whose inclusion preserves its fully connected property. The cliques can be overlapping, and between them cover all variables such that $\{C_1(\mathbf{z}) \cup \cdots \cup C_J(\mathbf{z})\} = \mathbf{z}$. Here we have written a normalisation constant, $\mathcal{Z}$, into the expression (1.5) to ensure that the total probability of all possible configurations sums to one. Alternatively, this normalisation can be absorbed into the definition of one or more of the potential functions. Markov networks can express a very simple form of independence relationship: two sets of nodes A and B are conditionally independent from each other given a third set of nodes C, if all paths connecting any node in A to any node in B via a sequence of arcs are separated by any node (or group of nodes) in C. Then C is said to *separate* A from B. The *Markov blanket* for the node (or set of nodes) A is defined as the smallest set of nodes C, such that A is conditionally independent of all other variables not in C, given C.

Directed graphical models, also called *Directed Acyclic Graphs* (DAGs), or *Bayesian networks*, express the probability distribution over J variables, $\mathbf{z} = \{\mathbf{z}_j\}_{j=1}^{J}$, as a product of conditional probability distributions on each variable:

$$p(\mathbf{z}) = \prod_{j=1}^{J} p(\mathbf{z}_j \,|\, \mathbf{z}_{\mathbf{pa}(j)}) \,, \tag{1.6}$$

where $\mathbf{z}_{\mathbf{pa}(j)}$ is the set of variables that are *parents* of the node j in the graph. A node a is said to be a parent of a node b if there is a directed arc from a to b, and in which case b is said to be a *child* of a. In necessarily recursive definitions: the *descendents* of a node are defined to include its children and its childrens' descendents; and the *ancestors* of a node are its parents and those parents' ancestors. Note that there is no need for a normalisation constant in (1.6) because by the definition of the conditional probabilities it is equal to one. A *directed path* between two nodes a and b is a sequence of variables such that every node is a parent of the following node in the sequence. An *undirected path* from a to b is any sequence of nodes such that every node is a parent or child of the following node. An *acyclic* graph is a graphical model in which there exist no directed paths including the same variable more than once. The semantics of a Bayesian network can be summarised as: each node is conditionally independent from its non-descendents given its parents.

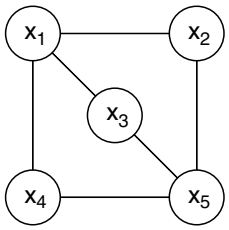
More generally, we have the following representation of independence in Bayesian networks: two sets of nodes A and B are conditionally independent given the set of nodes C if they are *d-separated* by C (here the *d-* prefix stands for *directed*). The nodes A and B are d-separated by C if, along every undirected path from A to B, there exists a node d which satisfies *either* of the following conditions: either (i) d has converging arrows (i.e. d is the child of the previous node and the parent of the following node in the path) *and* neither d nor its descendents are in C; or (ii) d does not have converging arrows and is in C. From the above definition of the Markov blanket, we find that for Bayesian networks the minimal Markov blanket for a node is given by the union of its parents, its children, *and* the parents of its children. A more simple rule for d-separation can be obtained using the idea of the 'Bayes ball' (Shachter, 1998). Two sets of nodes A and B are conditionally dependent given C if there exists a path by which the Bayes ball can reach a node in B from a node in A (or vice-versa), where the ball can move according to the following rules: it can pass through a node in the conditioning set C provided the entry and exit arcs are a pair of arrows converging on that node; similarly, it can only pass through every node in the remainder of the graph provided it does so on non-converging arrows. If there exist no such linking paths, then the sets of nodes A and B are conditionally independent given C.

Undirected models tend to be used in the physics and vision communities, where the systems under study can often be simply expressed in terms of many localised potential functions. The nature of the interactions often lack causal or direct probabilistic interpretations, and instead express degrees of agreement, compatibility, constraint or frustration between nodes. In the artificial intelligence and statistics communities directed graphs are more popular as they can more easily express underlying causal generative processes that give rise to our observations. For more detailed examinations of directed and undirected graphs see Pearl (1988).

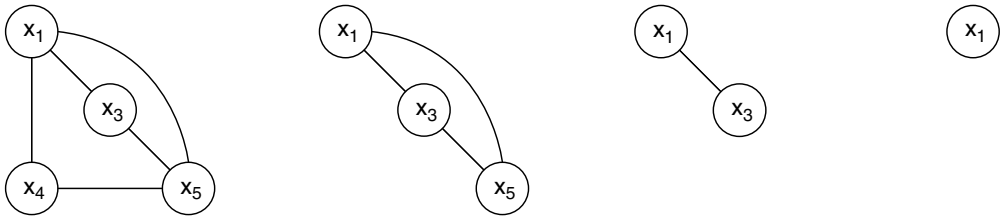
1.1.2 Propagation algorithms

The conditional independence relationships discussed in the previous subsection can be exploited to design efficient message-passing algorithms for obtaining the posterior distributions over hidden variables given the observations of some other variables, which is called inference. In this section we briefly present an inference algorithm for Markov networks, called the *junction tree* algorithm. We will explain at the end of this subsection why it suffices to present the inference algorithm for the undirected network case, since the inference algorithm for a directed network is just a special case.

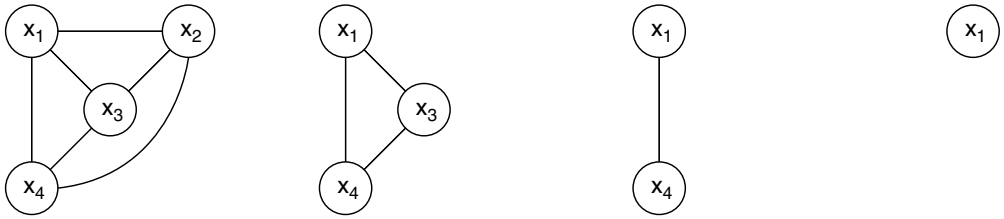
For data in which every variable is observed there is no inference problem for hidden variables, and learning for example the maximum likelihood (ML) parameters for the model using (1.1) often consists of a straightforward optimisation procedure. However, as we will see in chapter 2, if some of the variables are hidden this complicates finding the ML parameters. The common



(a) Original Markov network.



(b) One possible elimination ordering: (x_2, x_4, x_5, x_3)



(c) Another possible elimination ordering: (x_5, x_2, x_3, x_4).

Figure 1.1: **(a)** The original Markov network; **(b)** The sequence of intermediate graphs resulting from eliminating (integrating out) nodes to obtain the marginal on x_1 — see equations (1.9–1.14); **(c)** Another sequence of graphs resulting from a different elimination ordering, which results in a suboptimal inference algorithm.

practice in these cases is to utilise expectation-maximisation (EM) algorithms, which in their E step require the computation of at least certain properties of the posterior distribution over the hidden variables.

We illustrate the basics of inference using a simple example adapted from Jordan and Weiss (2002). Figure 1.1(a) shows a Markov network for five variables $\mathbf{x} = \{x_1, \ldots, x_5\}$, each of which is discrete and takes on k possible states. Using the Markov network factorisation given by (1.5), the probability distribution over the variables can be written as a product of potentials defined over five cliques:

$$p(\mathbf{x}) = p(x_1, \ldots, x_5) = \frac{1}{\mathcal{Z}} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1, x_4)\psi(x_2, x_5)\psi(x_3, x_5)\psi(x_4, x_5) \ , \quad (1.7)$$

where we have included a normalisation constant $\mathcal{Z}$ to allow for arbitrary clique potentials. Note that in this graph 1.1(a) the maximal cliques are all pairs of nodes connected by an arc, and therefore the potential functions are defined over these same pairs of nodes. Suppose we wanted to obtain the marginal distribution $p(x_1)$, given by

$$p(x_1) = \frac{1}{\mathcal{Z}} \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1, x_4)\psi(x_2, x_5)\psi(x_3, x_5)\psi(x_4, x_5) \; . \tag{1.8}$$

At first glance this requires k^5 computations, since there are k^4 summands to be computed for each of the k settings of the variable x_1. However this complexity can be reduced by exploiting the conditional independence structure in the graph. For example, we can rewrite (1.8) as

$$p(x_1) = \frac{1}{\mathcal{Z}} \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \psi(x_1, x_3)\psi(x_3, x_5)\psi(x_1, x_4)\psi(x_4, x_5)\psi(x_1, x_2)\psi(x_2, x_5) \tag{1.9}$$

$$= \frac{1}{\mathcal{Z}} \sum_{x_3} \psi(x_1, x_3) \sum_{x_5} \psi(x_3, x_5) \sum_{x_4} \psi(x_1, x_4)\psi(x_4, x_5) \sum_{x_2} \psi(x_1, x_2)\psi(x_2, x_5) \tag{1.10}$$

$$= \frac{1}{\mathcal{Z}} \sum_{x_3} \psi(x_1, x_3) \sum_{x_5} \psi(x_3, x_5) \sum_{x_4} \psi(x_1, x_4)\psi(x_4, x_5) m_2(x_1, x_5) \tag{1.11}$$

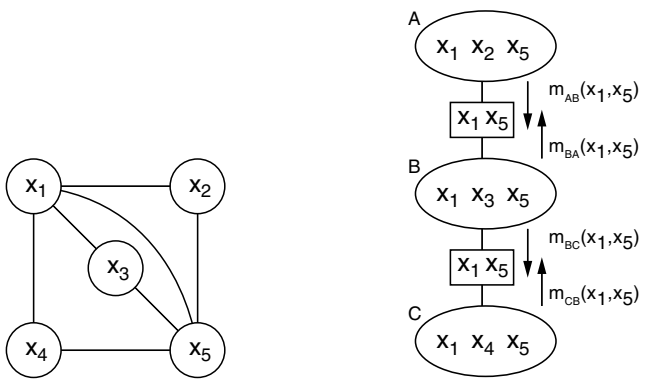
$$= \frac{1}{\mathcal{Z}} \sum_{x_3} \psi(x_1, x_3) \sum_{x_5} \psi(x_3, x_5) m_4(x_1, x_5) m_2(x_1, x_5) \tag{1.12}$$

$$= \frac{1}{\mathcal{Z}} \sum_{x_3} \psi(x_1, x_3) m_5(x_1, x_3) \tag{1.13}$$

$$= \frac{1}{\mathcal{Z}} m_1(x_1) \tag{1.14}$$

where each 'message' $m_j(x_\cdot, \dots)$ is a new potential obtained by *eliminating* the jth variable, and is a function of all the variables linked to that variable. By choosing this ordering (x_2, x_4, x_5, x_3) for summing over the variables, the most number of variables in any summand is three, meaning that the complexity has been reduced to $\mathcal{O}(k^3)$ for each possible setting of x_1, which results in an overall complexity of $\mathcal{O}(k^4)$.

This process can be described by the sequence of graphs resulting from the repeated application of a *triangulation* algorithm (see figure 1.1(b)) following these four steps: (i) choose a node x_j to eliminate; (ii) find all potentials ψ and any messages m that may reference this node; (iii) define a new potential m_j that is the sum with respect to x_j of the product of these potentials; (iv) remove the node x_j and *replace it with edges* connecting each of its neighbours — these represent the dependencies from the new potentials. This process is repeated until only the variables of interest remain, as shown in the above example. In this way marginal probabilities of single variables or joint probabilities over several variables can be obtained. Note that the second elimination step in figure 1.1(b)), that of marginalising out x_4, introduces a new message $m_4(x_1, x_5)$ but since there is already an arc connecting x_1 and x_5 we need not add a further one.



(a) (b)

Figure 1.2: **(a)** The triangulated graph corresponding to the elimination ordering in figure 1.1(b); **(b)** the corresponding junction tree including maximal cliques (ovals), separators (rectangles), and the messages produced in belief propagation.

The ordering chosen for this example is optimal; different orderings of elimination may result in suboptimal complexity. For example, figure 1.1(c) shows the process of an elimination ordering (x_5, x_2, x_3, x_4) which results in a complexity $\mathcal{O}(k^5)$. In general though, it is an NP-hard problem to find the optimal ordering of elimination that minimises the complexity. If all the nodes have the same cardinality, the optimal elimination ordering is independent of the functional forms on the nodes and is purely a graph-theoretic property.

We could use the above elimination algorithm repeatedly to find marginal probabilities for each and every node, but we would find that we had needlessly computed certain messages several times over. We can use the junction tree algorithm to compute all the messages we might need just once. Consider the graph shown in figure 1.2(a) which results from retaining all edges that were either initially present or added during the elimination algorithm (using the ordering in our worked example). Alongside in figure 1.2(b) is the junction tree for this graph, formed by linking the maximal cliques of the graph, of which there are three, labelled A, B and C. In between the clique nodes are *separators* for the junction tree, which contain nodes that are common to both the cliques attached to the separator, that is to say $\mathcal{S}_{AB} = \mathcal{C}_A \cap \mathcal{C}_B$. Here we use calligraphic $\mathcal{C}$ to distinguish these cliques from the original maximal cliques in the network 1.1(a). For a triangulated graph it is always possible to obtain such a singly-connected graph, or tree (to be more specific, it is always then possible to obtain a tree that satisfies the *running intersection property*, which states that if a variable appears in two different cliques, then it should also appear in every clique in the path between the two cliques). The so-called 'messages' in the elimination algorithm can now be considered as messages sent from one clique to another in the junction tree. For example, the message $m_2(x_1, x_5)$ produced in equation (1.11) as a result of summing over x_2 can be identified with the message $m_{AB}(x_1, x_5)$ that clique A sends to clique B. Similarly, the message $m_4(x_1, x_5)$ in (1.12) resulting from summing over x_4 is identified

with the message $m_{CB}(x_1, x_5)$ that C passes on to B. To complete the marginalisation to obtain $p(x_1)$, the clique B *absorbs* the incoming messages to obtain a joint distribution over its variables (x_1, x_3, x_5), and then marginalises out x_3 and x_5 in either order. Included in figure 1.2(b) are two other messages, $m_{BA}(x_1, x_5)$ and $m_{BC}(x_1, x_5)$, which would be needed if we wanted the marginal over x_2 or x_4, respectively.

For general junction trees it can be shown that the message that clique r sends to clique s is a function of the variables in their separator, $\mathcal{S}_{rs}(\mathbf{x})$, and is given by

$$m_{rs}(\mathcal{S}_{rs}(\mathbf{x})) = \sum_{\mathcal{C}_r(\mathbf{x})\setminus\mathcal{S}_{rs}(\mathbf{x})} \psi_r(\mathcal{C}_r(\mathbf{x})) \prod_{t\in\mathcal{N}(r)\setminus s} m_{tr}(\mathcal{S}_{tr}(\mathbf{x})) \,, \tag{1.15}$$

where $\mathcal{N}(r)$ are the set of neighbouring cliques of clique r. In words, the message from r to s is formed by: taking the product of all messages r has received from elsewhere other than s, multiplying in the potential ψ_r, and then summing out all those variables in r which are not in s.

The joint probability of the variables within clique r is obtained by combining messages into clique r with its potential:

$$p(\mathcal{C}_r(\mathbf{x})) \propto \psi_r(\mathcal{C}_r(\mathbf{x})) \prod_{t\in\mathcal{N}(r)} m_{tr}(\mathcal{S}_{tr}(\mathbf{x})) \,. \tag{1.16}$$

Note that from definition (1.15) a clique is unable to send a message until it has received messages from all other cliques except the receiving one. This means that the message-passing protocol must begin at the leaves of the junction tree and move inwards, and then naturally the message-passing moves back outwards to the leaves. In our example problem the junction tree has a very trivial structure and happens to have both separators containing the same variables (x_1, x_5).

Here we have explained how inference in a Markov network is possible: (i) through a process of triangulation the junction tree is formed; (ii) messages (1.15) are then propagated between junction tree cliques until all cliques have received and sent all their messages; (iii) clique marginals (1.16) can then be computed; (iv) individual variable marginals can be obtained by summing out other variables in the clique. The algorithm used for inference in a Bayesian network (which is directed) depends on whether it is singly- or multiply-connected (a graph is said to be singly-connected if it includes no pairs of nodes with more than one path between them, and multiply-connected otherwise). For singly-connected networks, an exactly analogous algorithm can be used, and is called *belief propagation*. For multiply-connected networks, we first require a process to convert the Bayesian network into a Markov network, called *moralisation*. We can then form the junction tree after a triangulation process and perform the same message-passing algorithm. The process of moralisation involves adding an arc between any variables sharing the same child (i.e. co-parents), and then dropping the directionality of all arcs.

Moralisation does not introduce any further conditional independence relationships into the graph, and in this sense the resulting Markov network is able to represent a superset of the probability distributions representable by the Bayesian network. Therefore, having derived the inference procedure for the more general Markov network, we already have the result for the Bayesian network as a special case.

1.2 Bayesian model selection

In this thesis we are primarily concerned with the task of model selection, or structure discovery. We use the term 'model' and 'model structure' to denote a variety of things, some already mentioned in the previous sections. A few particular examples of model selection tasks are given below:

Structure learning In probabilistic graphical models, each graph implies a set of conditional independence statements between the variables in the graph. The model structure learning problem is inferring the conditional independence relationships that hold given a set of (complete or incomplete) observations of the variables. Another related problem is learning the direction of the dependencies, i.e. the causal relationships between variables ($A \to B$, or $B \to A$).

Input dependence A special case of this problem is input variable selection in regression. Selecting which input (i.e. explanatory) variables are needed to predict the output (i.e. response) variable in the regression can be equivalently cast as deciding whether each input variable is a parent (or, more accurately, an ancestor) of the output variable in the corresponding directed graph.

Cardinality Many statistical models contain discrete nominal latent variables. A model structure learning problem of interest is then choosing the cardinality of each discrete latent variable. Examples of this problem include deciding how many mixture components are required in a finite mixture model, or how many hidden states are needed in a hidden Markov model.

Dimensionality Other statistical models contain real-valued vectors of latent variables. The dimensionality of this latent vector is usually unknown and needs to be inferred. Examples include choosing the intrinsic dimensionality in a probabilistic principal components analysis (PCA), or factor analysis (FA) model, or in a linear-Gaussian state-space model.

In the course of this thesis we tackle several of the above model selection problems using Bayesian learning. The machinery and tools for Bayesian model selection are presented in the following subsection.

1.2.1 Marginal likelihood and Occam's razor

An obvious problem with using maximum likelihood methods (1.1) to learn the parameters of models such as those described above is that the probability of the data will generally be greater for more complex model structures, leading to overfitting. Such methods fail to take into account model complexity. For example, inserting an arc between two variables in a graphical model can only help the model give higher probability to the data. Common ways for avoiding overfitting have included early stopping, regularisation, and cross-validation. Whilst it is possible to use cross-validation for simple searches over model size and structures — for example, if the search is limited to a single parameter that controls the model complexity — for more general searches over many parameters cross-validation is computationally prohibitive.

A Bayesian approach to learning starts with some prior knowledge or assumptions about the model structure — for example the set of arcs in the Bayesian network. This initial knowledge is represented in the form of a prior probability distribution over model structures. Each model structure has a set of parameters which have prior probability distributions. In the light of observed data, these are updated to obtain a posterior distribution over models and parameters. More formally, assuming a prior distribution over models structures $p(m)$ and a prior distribution over the parameters for each model structure $p(\boldsymbol{\theta} \,|\, m)$, observing the data set $\mathbf{y}$ induces a posterior distribution over models given by Bayes' rule:

$$p(m \,|\, \mathbf{y}) = \frac{p(m)p(\mathbf{y} \,|\, m)}{p(\mathbf{y})} \,. \tag{1.17}$$

The most probable model or model structure is the one that maximises $p(m \,|\, \mathbf{y})$. For a given model structure, we can also compute the posterior distribution over the parameters:

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}, m) = \frac{p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)p(\boldsymbol{\theta} \,|\, m)}{p(\mathbf{y} \,|\, m)} \,, \tag{1.18}$$

which allows us to quantify our uncertainty about parameter values after observing the data. We can also compute the density at a new data point $\mathbf{y}'$, obtained by averaging over both the uncertainty in the model structure and in the parameters,

$$p(\mathbf{y}' \,|\, \mathbf{y}) = \sum_m \int d\boldsymbol{\theta} \; p(\mathbf{y}' \,|\, \boldsymbol{\theta}, m, \mathbf{y}) p(\boldsymbol{\theta} \,|\, m, \mathbf{y}) p(m \,|\, \mathbf{y}) \,, \tag{1.19}$$

which is known as the *predictive distribution*.

The second term in the numerator of (1.17) is called the *marginal likelihood*, and results from integrating the likelihood of the data over all possible parameter settings under the prior:

$$p(\mathbf{y} \,|\, m) = \int d\boldsymbol{\theta} \; p(\mathbf{y} \,|\, \boldsymbol{\theta}, m) p(\boldsymbol{\theta} \,|\, m) \,. \tag{1.20}$$

In the machine learning community this quantity is sometimes referred to as the *evidence* for model m, as it constitutes the data-dependent factor in the posterior distribution over models (1.17). In the absence of an informative prior $p(m)$ over possible model structures, this term alone will drive our model inference process. Note that this term also appears as the normalisation constant in the denominator of (1.18). We can think of the marginal likelihood as the average probability of the data, where the average is taken with respect to the model parameters drawn from the prior $p(\boldsymbol{\theta})$.

Integrating out the parameters penalises models with more degrees of freedom since these models can *a priori* model a larger range of data sets. This property of Bayesian integration has been called *Occam's razor*, since it favours simpler explanations (models) for the data over complex ones (Jefferys and Berger, 1992; MacKay, 1995). Having more parameters may impart an advantage in terms of the ability to model the data, but this is offset by the cost of having to code those extra parameters under the prior (Hinton and van Camp, 1993). The overfitting problem is avoided simply because no parameter in the pure Bayesian approach is actually *fit* to the data. A caricature of Occam's razor is given in figure 1.3, where the horizontal axis denotes all possible data sets to be modelled, and the vertical axis is the marginal probability $p(\mathbf{y} \mid m)$ under each of three models of increasing complexity. We can relate the complexity of a model to the range of data sets it can capture. Thus for a simple model the probability is concentrated over a small range of data sets, and conversely a complex model has the ability to model a wide range of data sets.

Since the marginal likelihood as a function of the data $\mathbf{y}$ should integrate to one, the simple model can give a higher marginal likelihood to those data sets it can model, whilst the complex model gives only small marginal likelihoods to a wide range of data sets. Therefore, given a data set, $\mathbf{y}$, on the basis of the marginal likelihood it is possible to discard both models that are too complex and those that are too simple. In these arguments it is tempting, but not correct, to associate the complexity of a model with the number of parameters it has: it is easy to come up with a model with many parameters that can model only a limited range of data sets, and also to design a model capable of capturing a huge range of data sets with just a single parameter (specified to high precision).

We have seen how the marginal likelihood is an important quantity in Bayesian learning, for computing quantities such as Bayes factors (the ratio of two marginal likelihoods, Kass and Raftery, 1995), or the normalising constant of a posterior distribution (known in statistical physics as the 'partition function' and in machine learning as the 'evidence'). Unfortunately the marginal likelihood is a very difficult quantity to compute because it involves integrating over all parameters and latent variables, which is usually such a high dimensional and complicated integral that most simple approximations fail catastrophically. We will see in section 1.3 some of the approximations to the marginal likelihood and will investigate variational Bayesian approximations in the following chapter.

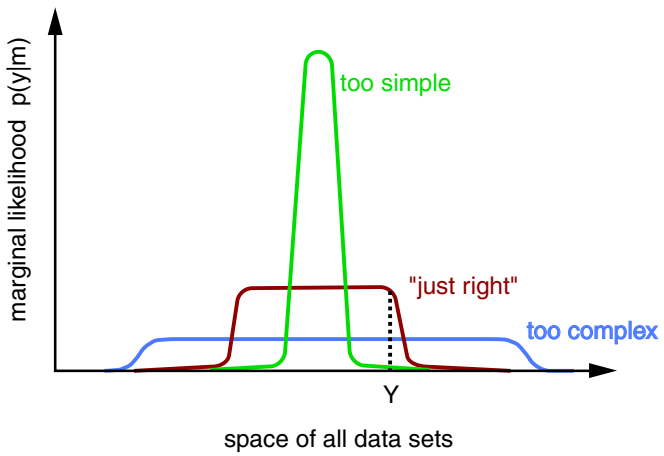


Figure 1.3: Caricature depicting Occam's razor (adapted from MacKay, 1995). The horizontal axis denotes all possible data sets of a particular size and the vertical axis is the marginal likelihood for three different model structures of differing complexity. Simple model structures can model certain data sets well but cannot model a wide range of data sets; complex model structures can model many different data sets but, since the marginal likelihood has to integrate to one, will necessarily not be able to model all simple data sets as well as the simple model structure. Given a particular data set (labelled Y), model selection is possible because model structures that are too simple are unlikely to generate the data set in question, while model structures that are too complex can generate many possible data sets, but again, are unlikely to generate that particular data set at random.

It is important to keep in mind that a realistic model of the data might need to be complex. It is therefore often advisable to use the most 'complex' model for which it is possible to do inference, ideally setting up priors that allow the limit of infinitely many parameters to be taken, rather than to artificially limit the number of parameters in the model (Neal, 1996; Rasmussen and Ghahramani, 2001). Although we do not examine any such infinite models in this thesis, we do return to them in the concluding comments of chapter 7.

Bayes' theorem provides us with the posterior over different models (1.17), and we can combine predictions by weighting them according to the posterior probabilities (1.19). Although in theory we should average over all possible model structures, in practice computational or representational constraints may make it necessary to select a single most probable structure by maximising $p(m \,|\, \mathbf{y})$. In most problems we may also have good reason to believe that the marginal likelihood is strongly peaked, and so the task of model selection is then justified.

1.2.2 Choice of priors

Bayesian model inference relies on the marginal likelihood, which has at its core a set of prior distributions over the parameters of each possible structure, $p(\boldsymbol{\theta} \,|\, m)$. Specification of parameter priors is obviously a key element of the Bayesian machinery, and there are several diverse

schools of thought when it comes to assigning priors; these can be loosely categorised into *subjective*, *objective*, and *empirical* approaches. We should point out that all Bayesian approaches are necessarily subjective in the sense that any Bayesian inference first requires some expression of prior knowledge $p(\boldsymbol{\theta})$. Here the emphasis is not on whether we use a prior or not, but rather *what* knowledge (if any) is conveyed in $p(\boldsymbol{\theta})$. We expand on these three types of prior design in the following paragraphs.

Subjective priors

The subjective Bayesian attempts to encapsulate prior knowledge as fully as possible, be it in the form of previous experimental data or expert knowledge. It is often difficult to articulate qualitative experience or beliefs in mathematical form, but one very convenient and analytically favourable class of subjective priors are *conjugate* priors in the *exponential family*. Generally speaking, a prior is conjugate if the posterior distribution resulting from multiplying the likelihood and prior terms is of the same form as the prior. Expressed mathematically:

$$f(\boldsymbol{\theta} \,|\, \tilde{\boldsymbol{\mu}}) = p(\boldsymbol{\theta} \,|\, \mathbf{y}) \propto f(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \,, \tag{1.21}$$

where $f(\boldsymbol{\theta} \,|\, \boldsymbol{\mu})$ is some probability distribution specified by a parameter (or set of parameters) $\boldsymbol{\mu}$. Conjugate priors have at least three advantages: first, they often lead to analytically tractable Bayesian integrals; second, if computing the posterior in (1.21) is tractable, then the modeller can be assured that subsequent inferences, based on using the posterior as prior, will also be tractable; third, conjugate priors have an intuitive interpretation as expressing the results of previous (or indeed imaginary) observations under the model. The latter two advantages are somewhat related, and can be understood by observing that the only likelihood functions $p(\mathbf{y} \,|\, \boldsymbol{\theta})$ for which conjugate prior families exist are those belonging to general *exponential family* models. The definition of an exponential family model is one that has a likelihood function of the form

$$p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = g(\boldsymbol{\theta})\, f(\mathbf{y}_i)\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{y}_i)} \,, \tag{1.22}$$

where $g(\boldsymbol{\theta})$ is a normalisation constant:

$$g(\boldsymbol{\theta})^{-1} = \int d\mathbf{y}_i \; f(\mathbf{y}_i)\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{y}_i)} \,, \tag{1.23}$$

and we have used the subscript notation $\mathbf{y}_i$ to denote each data point (not each variable!). We assume that n data points arrive independent and identically distributed (i.i.d.) such that the probability of the data $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ under this model is given by $p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^n p(\mathbf{y}_i \,|\, \boldsymbol{\theta})$.

Here $\boldsymbol{\phi}(\boldsymbol{\theta})$ is a vector of so-called *natural parameters*, and $\mathbf{u}(\mathbf{y}_i)$ and $f(\mathbf{y}_i)$ are functions defining the exponential family. Now consider the conjugate prior:

$$p(\boldsymbol{\theta} \,|\, \eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})\, g(\boldsymbol{\theta})^{\eta}\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top}\boldsymbol{\nu}} \,, \tag{1.24}$$

where η and $\boldsymbol{\nu}$ are parameters of the prior, and $h(\eta, \boldsymbol{\nu})$ is an appropriate normalisation constant. The conjugate prior contains the same functions $g(\boldsymbol{\theta})$ and $\boldsymbol{\phi}(\boldsymbol{\theta})$ as in (1.22), and the result of using a conjugate prior can then be seen by substituting (1.22) and (1.24) into (1.21), resulting in:

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}) \propto p(\boldsymbol{\theta} \,|\, \eta, \boldsymbol{\nu}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \propto p(\boldsymbol{\theta} \,|\, \tilde{\eta}, \tilde{\boldsymbol{\nu}}) \,, \tag{1.25}$$

where $\tilde{\eta} = \eta + n$ and $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + \sum_{i=1}^{n} \mathbf{u}(\mathbf{y}_i)$ are the new parameters for the posterior distribution which has the *same functional form* as the prior. We have omitted some of the details, as a more general approach will be described in the following chapter (section 2.4). The important point to note is that the parameters of the prior can be viewed as the number (or amount), η, and the 'value', $\boldsymbol{\nu}$, of imaginary data observed prior to the experiment (by 'value' we in fact refer to the vector of sufficient statistics of the data). This correspondence is often apparent in the expressions for predictive densities and other quantities which result from integrating over the posterior distribution, where statistics gathered from the data are simply augmented with prior quantities. Therefore the knowledge conveyed by the conjugate prior is specific and clearly interpretable. On a more mathematical note, the attraction of the conjugate exponential family of models is that they can represent probability densities with a finite number of sufficient statistics, and are closed under the operation of Bayesian inference. Unfortunately, a conjugate analysis becomes difficult, and for the majority of interesting problems impossible, for models containing hidden variables $\mathbf{x}_i$.

Objective priors

The objective Bayesian's goal is in stark contrast to a subjectivist's approach. Instead of attempting to encapsulate rich knowledge into the prior, the objective Bayesian tries to impart as *little* information as possible in an attempt to allow the data to carry as much weight as possible in the posterior distribution. This is often called 'letting the data speak for themselves' or 'prior ignorance'. There are several reasons why a modeller may want to resort to the use of objective priors (sometimes called non-informative priors): often the modeller has little expertise and does not want to sway the inference process in any particular direction unknowingly; it may be difficult or impossible to elicit expert advice or translate expert opinions into a mathematical form for the prior; also, the modeller may want the inference to be robust to misspecifications of the prior. It turns out that expressing such vagueness or ignorance is in fact quite difficult, partly because the very concept of 'vagueness' is itself vague. Any prior expressed on the parameters

has to follow through and be manifest in the posterior distribution in some way or other, so this quest for uninformativeness needs to be more precisely defined.

One such class of noninformative priors are *reference priors*. These originate from an information theoretic argument which asks the question: "which prior should I use such that I maximise the expected amount of information about a parameter that is provided by observing the data?". This expected information can be written as a function of $p(\theta)$ (we assume θ is one-dimensional):

$$I(p(\theta), n) = \int d\mathbf{y}^{(n)}\; p(\mathbf{y}^{(n)}) \int d\theta\; p(\theta \,|\, \mathbf{y}^{(n)}) \ln \frac{p(\theta \,|\, \mathbf{y}^{(n)})}{p(\theta)} \,, \tag{1.26}$$

where we use $\mathbf{y}^{(n)}$ to make it obvious that the data set is of size n. This quantity is strictly positive as it is an expected Kullback-Leibler (KL) divergence between the parameter posterior and parameter prior, where the expectation is taken with respect to the underlying distribution of the data $\mathbf{y}^{(n)}$. Here we assume, as before, that the data arrive i.i.d. such that $\mathbf{y}^{(n)} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ and $p(\mathbf{y}^{(n)} \,|\, \theta) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \theta)$. Then the n-reference prior is defined as the prior that maximises this expected information from n data points:

$$p_n(\theta) = \arg\max_{p(\theta)}\; I(p(\theta), n) \,. \tag{1.27}$$

Equation (1.26) can be rewritten directly as a KL divergence:

$$I(p(\theta), \mathbf{y}^{(n)}) = \int d\theta\; p(\theta) \ln \frac{f_n(\theta)}{p(\theta)} \,, \tag{1.28}$$

where the function $f_n(\theta)$ is given by

$$f_n(\theta) = \exp\left[\int d\mathbf{y}^{(n)}\; p(\mathbf{y}^{(n)} \,|\, \theta) \ln p(\theta \,|\, \mathbf{y}^{(n)}) \right] \,, \tag{1.29}$$

and n is the size of the data set $\mathbf{y}$. A naive solution that maximises (1.28) is

$$p_n(\theta) \propto f_n(\theta) \,, \tag{1.30}$$

but unfortunately this is only an implicit solution for the n-reference prior as $f_n(\theta)$ (1.29) is a function of the prior through the term $p(\theta \,|\, \mathbf{y}^{(n)})$. Instead, we make the approximation for large n that the posterior distribution $p(\theta \,|\, \mathbf{y}^{(n)}) \propto p(\theta) \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \theta)$ is given by $p^*(\theta \,|\, \mathbf{y}^{(n)}) \propto \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \theta)$, and write the reference prior as:

$$p(\theta) \propto \lim_{n\to\infty} \frac{f_n^*(\theta)}{f_n^*(\theta_0)} \,, \tag{1.31}$$

where $f_n^*(\theta)$ is the expression (1.29) using the approximation to the posterior $p^*(\theta \,|\, \mathbf{y}^{(n)})$ in place of $p(\theta \,|\, \mathbf{y}^{(n)})$, and θ_0 is a fixed parameter (or subset of parameters) used to normalise the

limiting expression. For discrete parameter spaces, it can be shown that the reference prior is uniform. More interesting is the case of real-valued parameters that exhibit asymptotic normality in their posterior (see section 1.3.2), where it can be shown that the reference prior coincides with Jeffreys' prior (see Jeffreys, 1946),

$$p(\theta) \propto h(\theta)^{1/2} \,, \tag{1.32}$$

where $h(\boldsymbol{\theta})$ is the Fisher information

$$h(\theta) = \int d\mathbf{y}_i \; p(\mathbf{y}_i \,|\, \theta) \left[-\frac{\partial^2}{\partial \theta^2} \ln p(\mathbf{y}_i \,|\, \theta) \right] \,. \tag{1.33}$$

Jeffreys' priors are motivated by requiring that the prior is invariant to one-to-one reparameterisations, so this equivalence is intriguing. Unfortunately, the multivariate extensions of reference and Jeffreys' priors are fraught with complications. For example, the form of the reference prior for one parameter can be different depending on the order in which the remaining parameters' reference priors are calculated. Also multivariate Jeffreys' priors are not consistent with their univariate equivalents. As an example, consider the mean and standard deviation parameters of a Gaussian, (μ, σ). If μ is known, both Jeffreys' and reference priors are given by $p(\sigma) \propto \sigma^{-1}$. If the standard deviation is known, again both Jeffreys' and reference priors over the mean are given by $p(\mu) \propto 1$. However, if neither the mean nor the standard deviation are known, the Jeffreys' prior is given by $p(\mu, \sigma) \propto \sigma^{-2}$, which does not agree with the reference prior $p(\mu, \sigma) \propto \sigma^{-1}$ (here the reference prior happens not to depend on the ordering of the parameters in the derivation). This type of ambiguity is often a problem in defining priors over multiple parameters, and it is often easier to consider other ways of specifying priors, such as hierarchically. A more in depth analysis of reference and Jeffreys' priors can be found in Bernardo and Smith (1994, section 5.4).

Empirical Bayes and hierarchical priors

When there are many common parameters in the vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$, it often makes sense to consider each parameter as being drawn from the same prior distribution. An example of this would be the prior specification of the means of each of the Gaussian components in a mixture model — there is generally no a priori reason to expect any particular component to be different from another. The parameter prior is then formed from integrating with respect to a hyperprior with hyperparameter γ:

$$p(\boldsymbol{\theta} \,|\, \gamma) = \int d\gamma \; p(\gamma) \prod_{k=1}^{K} p(\theta_k \,|\, \gamma) \,. \tag{1.34}$$

Therefore, each parameter is independent *given* the hyperparameter, although they are dependent marginally. Hierarchical priors are useful even when applied only to a single parameter,

often offering a more intuitive interpretation for the parameter's role. For example, the precision parameter ν for a Gaussian variable is often given a (conjugate) gamma prior, which itself has two hyperparameters (a_γ, b_γ) corresponding to the shape and scale of the prior. Interpreting the marginal distribution of the variable in this generative sense is often more intuitively appealing than simply enforcing a Student-t prior. Hierarchical priors are often designed using conjugate forms (described above), both for analytical ease and also because previous knowledge can be readily expressed.

Hierarchical priors can be easily visualised using directed graphical models, and there will be many examples in the following chapters. The phrase *empirical Bayes* refers to the practice of optimising the hyperparameters (e.g. γ) of the priors, so as to maximise the marginal likelihood of a data set $p(\mathbf{y} \mid \gamma)$. In this way Bayesian learning can be seen as maximum marginal likelihood learning, where there are always distributions over the parameters, but the hyperparameters are optimised just as in maximum likelihood learning. This practice is somewhat suboptimal as it ignores the uncertainty in the hyperparameter γ. Alternatively, a more coherent approach is to define priors over the hyperparameters and priors on the parameters of those priors, etc., to the point where at the top level the modeller is content to leave those parameters unoptimised. With sufficiently vague priors at the top level, the posterior distributions over intermediate parameters should be determined principally by the data. In this fashion, no parameters are actually ever fit to the data, and all predictions and inferences are based on the posterior distributions over the parameters.

1.3 Practical Bayesian approaches

Bayes' rule provides a means of updating the distribution over parameters from the prior to the posterior distribution in light of observed data. In theory, the posterior distribution captures all information inferred from the data about the parameters. This posterior is then used to make optimal decisions or predictions, or to select between models. For almost all interesting applications these integrals are analytically intractable, and are inaccessible to numerical integration techniques — not only do the computations involve very high dimensional integrals, but for models with parameter symmetries (such as mixture models) the integrand can have exponentially many modes.

There are various ways we can tackle this problem. At one extreme we can restrict ourselves only to models and prior distributions that lead to tractable posterior distributions and integrals for the marginal likelihoods and predictive densities. This is highly undesirable since it inevitably leads us to lose prior knowledge and modelling power. More realistically, we can approximate the exact answer.

1.3.1 Maximum a posteriori (MAP) parameter estimates

The simplest approximation to the posterior distribution is to use a point estimate, such as the maximum a posteriori (MAP) parameter estimate,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta}) \;, \tag{1.35}$$

which chooses the model with highest posterior probability density (the mode). Whilst this estimate does contain information from the prior, it is by no means completely Bayesian (although it is often erroneously claimed to be so) since the mode of the posterior may not be representative of the posterior distribution at all. In particular, we are likely (in typical models) to be over-confident of predictions made with the MAP model, since by definition *all* the posterior probability mass is contained in models which give poorer likelihood to the data (modulo the prior influence). In some cases it might be argued that instead of the MAP estimate it is sufficient to specify instead a set of *credible regions* or *ranges* in which most of the probability mass for the parameter lies (connected credible regions are called credible ranges). However, both point estimates and credible regions (which are simply a collection of point estimates) have the drawback that they are not unique: it is always possible to find a one-to-one monotonic mapping of the parameters such that any particular parameter setting is at the mode of the posterior probability density in that mapped space (provided of course that that value has non-zero probability density under the prior). This means that two modellers with identical priors and likelihood functions will in general find different MAP estimates if their parameterisations of the model differ.

The key ingredient in the Bayesian approach is then not just the use of a prior but the fact that all variables that are unknown are averaged over, i.e. that uncertainty is handled in a coherent way. In this way is it not important which parameterisation we adopt because the parameters are integrated out.

In the rest of this section we review some of the existing methods for approximating marginal likelihoods. The first three methods are analytical approximations: the Laplace method (Kass and Raftery, 1995), the Bayesian Information Criterion (BIC; Schwarz, 1978), and the criterion due to Cheeseman and Stutz (1996). All these methods make use of the MAP estimate (1.35), and in some way or other try to account for the probability mass about the mode of the posterior density. These methods are attractive because finding the MAP estimate is usually a straightforward procedure. To almost complete the toolbox of practical methods for Bayesian learning, there follows a brief survey of sampling-based approximations, such as importance sampling and Markov chain Monte Carlo methods. We leave the topic of variational Bayesian learning until the next chapter, where we will look back to these approximations for comparison.

1.3.2 Laplace's method

By Bayes' rule, the posterior over parameters $\boldsymbol{\theta}$ of a model m is

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}, m) = \frac{p(\boldsymbol{\theta} \,|\, m)\, p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)}{p(\mathbf{y} \,|\, m)} \,. \tag{1.36}$$

Defining the logarithm of the numerator as

$$t(\boldsymbol{\theta}) \equiv \ln \left[p(\boldsymbol{\theta} \,|\, m)\, p(\mathbf{y} \,|\, \boldsymbol{\theta}, m) \right] = \ln p(\boldsymbol{\theta} \,|\, m) + \sum_{i=1}^{n} \ln p(\mathbf{y}_i \,|\, \boldsymbol{\theta}, m) \,, \tag{1.37}$$

the *Laplace approximation* (Kass and Raftery, 1995; MacKay, 1995) makes a local Gaussian approximation around a MAP parameter estimate $\hat{\boldsymbol{\theta}}$ (1.35). The validity of this approximation is based on the large data limit and some regularity conditions which are discussed below. We expand $t(\boldsymbol{\theta})$ to second order as a Taylor series about this point:

$$t(\boldsymbol{\theta}) = t(\hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \left. \frac{\partial t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} + \frac{1}{2!} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \left. \frac{\partial^2 t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + \ldots \tag{1.38}$$

$$\approx t(\hat{\boldsymbol{\theta}}) + \frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top H(\hat{\boldsymbol{\theta}}) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \,, \tag{1.39}$$

where $H(\hat{\boldsymbol{\theta}})$ is the Hessian of the log posterior (matrix of the second derivatives of (1.37)), evaluated at $\hat{\boldsymbol{\theta}}$,

$$H(\hat{\boldsymbol{\theta}}) = \left. \frac{\partial^2 \ln p(\boldsymbol{\theta} \,|\, \mathbf{y}, m)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \left. \frac{\partial^2 t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \,, \tag{1.40}$$

and the linear term has vanished as the gradient of the posterior $\frac{\partial t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ at $\hat{\boldsymbol{\theta}}$ is zero as this is the MAP setting (or a local maximum). Substituting (1.39) into the log marginal likelihood and integrating yields

$$\ln p(\mathbf{y} \,|\, m) = \ln \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m)\, p(\mathbf{y} \,|\, \boldsymbol{\theta}, m) \tag{1.41}$$

$$= \ln \int d\boldsymbol{\theta}\; \exp\left[t(\boldsymbol{\theta}) \right] \,, \tag{1.42}$$

$$\approx t(\hat{\boldsymbol{\theta}}) + \frac{1}{2} \ln \left| 2\pi H^{-1} \right| \tag{1.43}$$

$$= \ln p(\hat{\boldsymbol{\theta}} \,|\, m) + \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m) + \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |H| \,, \tag{1.44}$$

where d is the dimensionality of the parameter space. Equation (1.44) can be written

$$p(\mathbf{y} \,|\, m)_{\text{Laplace}} = p(\hat{\boldsymbol{\theta}} \,|\, m)\, p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m) \left| 2\pi H^{-1} \right|^{1/2} \,. \tag{1.45}$$

Thus the Laplace approximation to the marginal likelihood consists of a term for the data likelihood at the MAP setting, a penalty term from the prior, and a volume term calculated from the local curvature.

Approximation (1.45) has several shortcomings. The Gaussian assumption is based on the large data limit, and will represent the posterior poorly for small data sets for which, in principle, the advantages of Bayesian integration over ML or MAP are largest. The Gaussian approximation is also poorly suited to bounded, constrained, or positive parameters, such as mixing proportions or precisions, since it assigns non-zero probability mass outside of the parameter domain. Of course, this can often be alleviated by a change of parameter basis (see for example, MacKay, 1998); however there remains the undesirable fact that in the non-asymptotic regime the approximation is still not invariant to reparameterisation. Moreover, the posterior may not be log quadratic for likelihoods with hidden variables, due to problems of identifiability discussed in the next subsection. In these cases the regularity conditions required for convergence do not hold. Even if the exact posterior is unimodal the resulting approximation may well be a poor representation of the nearby probability *mass*, as the approximation is made about a locally maximum probability *density*. The volume term requires the calculation of $|H|$: this takes $\mathcal{O}(nd^2)$ operations to compute the derivatives in the Hessian, and then a further $\mathcal{O}(d^3)$ operations to calculate the determinant; this becomes burdensome for high dimensions, so approximations to this calculation usually ignore off-diagonal elements or assume a block-diagonal structure for the Hessian, which correspond to neglecting dependencies between parameters. Finally, the second derivatives themselves may be intractable to compute.

1.3.3 Identifiability: aliasing and degeneracy

The convergence to Gaussian of the posterior holds only if the model is *identifiable*. Therefore the Laplace approximation may be inaccurate if this is not the case. A model is not identifiable if there is *aliasing* or *degeneracy* in the parameter posterior.

Aliasing arises in models with symmetries, where the assumption that there exists a single mode in the posterior becomes incorrect. As an example of symmetry, take the model containing a discrete hidden variable $\mathbf{x}_i$ with k possible settings (e.g. the indicator variable in a mixture model). Since the variable is hidden these settings can be arbitrarily labelled $k!$ ways. If the likelihood is invariant to these permutations, and if the prior over parameters is also invariant to these permutations, then the landscape for the posterior parameter distribution will be made up of $k!$ identical aliases. For example the posterior for HMMs converges to a mixture of Gaussians, not a single mode, corresponding to the possible permutations of the hidden states. If the aliases are sufficiently distinct, corresponding to well defined peaks in the posterior as a result of large amounts of data, the error in the Laplace method can be corrected by multiplying the marginal likelihood by a factor of $k!$. In practice it is difficult to ascertain the degree of separation of the aliases, and so a simple modification of this sort is not possible. Although corrections have been devised to account for this problem, for example estimating the *permanent* of the model, they are complicated and computationally burdensome. The interested reader is referred

to Barvinok (1999) for a description of a polynomial randomised approximation scheme for estimating permanents, and to Jerrum et al. (2001) for a review of permanent calculations.

Parameter degeneracy arises when there is some redundancy in the choice of parameterisation for the model. For example, consider a model that has two parameters $\boldsymbol{\theta} = (\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)$, whose difference specifies the noise precision of an observed Gaussian variable $\mathbf{y}_i$ with mean $\mathbf{0}$, say, $\mathbf{y}_i \sim \mathrm{N}(\mathbf{y}_i \,|\, \mathbf{0},\, \boldsymbol{\nu}_1 - \boldsymbol{\nu}_2)$. If the prior over parameters does not disambiguate $\boldsymbol{\nu}_1$ from $\boldsymbol{\nu}_2$, the posterior over $\boldsymbol{\theta}$ will contain an infinity of distinct configurations of $(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)$, all of which give the same likelihood to the data; this degeneracy causes the volume element $\propto \left|H^{-1}\right|$ to be infinite and renders the marginal likelihood estimate (1.45) useless. Parameter degeneracy can be thought of as a continuous form of aliasing in parameter space, in which there are infinitely many aliases.

1.3.4 BIC and MDL

The Bayesian Information Criterion (BIC) (Schwarz, 1978) can be obtained from the Laplace approximation by retaining only those terms that grow with n. From (1.45), we have

$$\ln p(\mathbf{y} \,|\, m)_{\text{Laplace}} = \underbrace{\ln p(\hat{\boldsymbol{\theta}} \,|\, m)}_{\mathcal{O}(1)} + \underbrace{\ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m)}_{\mathcal{O}(n)} + \underbrace{\frac{d}{2} \ln 2\pi}_{\mathcal{O}(1)} - \underbrace{\frac{1}{2} \ln |H|}_{\mathcal{O}(d \ln n)} , \tag{1.46}$$

where each term's dependence on n has been annotated. Retaining $\mathcal{O}(n)$ and $\mathcal{O}(\ln n)$ terms yields

$$\ln p(\mathbf{y} \,|\, m)_{\text{Laplace}} = \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m) - \frac{1}{2} \ln |H| + \mathcal{O}(1) \,. \tag{1.47}$$

Using the fact that the entries of the Hessian scale linearly with n (see (1.37) and (1.40)), we can write

$$\lim_{n \to \infty} \frac{1}{2} \ln |H| = \frac{1}{2} \ln |n H_0| = \frac{d}{2} \ln n + \underbrace{\frac{1}{2} \ln |H_0|}_{\mathcal{O}(1)} , \tag{1.48}$$

and then assuming that the prior is non-zero at $\hat{\boldsymbol{\theta}}$, in the limit of large n equation (1.47) becomes the BIC score:

$$\ln p(\mathbf{y} \,|\, m)_{\text{BIC}} = \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m) - \frac{d}{2} \ln n \,. \tag{1.49}$$

The BIC approximation is interesting for two reasons: first, it does not depend on the prior $p(\boldsymbol{\theta} \,|\, m)$; second, it does not take into account the local geometry of the parameter space and hence is invariant to reparameterisations of the model. A Bayesian would obviously baulk at the first of these features, but the second feature of reparameterisation invariance is appealing because this should fall out of an exact Bayesian treatment in any case. In practice the dimension of the model d that is used is equal to the number of *well-determined* parameters, or the number

of *effective* parameters, after any potential parameter degeneracies have been removed. In the example mentioned above the reparameterisation $\boldsymbol{\nu}^* = \boldsymbol{\nu}_1 - \boldsymbol{\nu}_2$ is sufficient, yielding $d = |\boldsymbol{\nu}|$. The BIC is in fact exactly minus the minimum description length (MDL) penalty used in Rissanen (1987). However, the minimum message length (MML) framework of Wallace and Freeman (1987) is closer in spirit to Bayesian integration over parameters. We will be revisiting the BIC in the following chapters as a comparison to our variational Bayesian method for approximating the marginal likelihood.

1.3.5 Cheeseman & Stutz's method

If the *complete-data* marginal likelihood defined as

$$p(\mathbf{x}, \mathbf{y} \,|\, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m) \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m) \tag{1.50}$$

can be computed efficiently then the method proposed in Cheeseman and Stutz (1996) can be used to approximate the marginal likelihood of incomplete data. For any completion of the data $\hat{\mathbf{x}}$, the following identity holds

$$p(\mathbf{y} \,|\, m) = p(\hat{\mathbf{x}}, \mathbf{y} \,|\, m) \frac{p(\mathbf{y} \,|\, m)}{p(\hat{\mathbf{x}}, \mathbf{y} \,|\, m)} \tag{1.51}$$

$$= p(\hat{\mathbf{x}}, \mathbf{y} \,|\, m) \frac{\int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)}{\int d\boldsymbol{\theta}'\; p(\boldsymbol{\theta}' \,|\, m) p(\hat{\mathbf{x}}, \mathbf{y} \,|\, \boldsymbol{\theta}', m)} \,. \tag{1.52}$$

If we now apply Laplace approximations (1.45) to both numerator and denominator we obtain

$$p(\mathbf{y} \,|\, m) \approx p(\hat{\mathbf{x}}, \mathbf{y} \,|\, m) \frac{p(\hat{\boldsymbol{\theta}} \,|\, m) p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m) \left|2\pi H^{-1}\right|^{1/2}}{p(\hat{\boldsymbol{\theta}}' \,|\, m) p(\hat{\mathbf{x}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}', m) \left|2\pi H'^{-1}\right|^{1/2}} \,. \tag{1.53}$$

If the approximations are made about the same point $\hat{\boldsymbol{\theta}}' = \hat{\boldsymbol{\theta}}$, then the hope is that errors in each Laplace approximation will tend to cancel one another out. If the completion $\hat{\mathbf{x}}$ is set to be the expected sufficient statistics calculated from an E step of the EM algorithm (discussed in more detail in chapter 2), then the ML/MAP setting $\hat{\boldsymbol{\theta}}'$ will be at the same point as $\hat{\boldsymbol{\theta}}$. The final part of the Cheeseman-Stutz approximation is to form the BIC asymptotic limit of each of the Laplace approximations (1.49). In the original *Autoclass* application (Cheeseman and Stutz, 1996) the dimensionalities of the parameter spaces for the incomplete and complete-data integrals were assumed equal so the terms scaling as $\ln n$ cancel. Since $\hat{\boldsymbol{\theta}}' = \hat{\boldsymbol{\theta}}$, the terms relating to the prior probability of $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}'$ also cancel (although these are $\mathcal{O}(1)$ in any case), and we obtain:

$$p(\mathbf{y} \,|\, m)_{\mathrm{CS}} = p(\hat{\mathbf{x}}, \mathbf{y} \,|\, m) \frac{p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m)}{p(\hat{\mathbf{x}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}, m)} \,. \tag{1.54}$$

where $\hat{\boldsymbol{\theta}}$ is the MAP estimate. In chapter 2 we see how the Cheeseman-Stutz approximation is related to the variational Bayesian lower bound. In chapter 6 we compare its performance empirically to variational Bayesian methods on a hard problem, and discuss the situation in which the dimensionalities of the complete and incomplete-data parameters are different.

1.3.6 Monte Carlo methods

Unfortunately the large data limit approximations discussed in the previous section are limited in their ability to trade-off computation time to improve their accuracy. For example, even if the Hessian determinant were calculated exactly (costing $\mathcal{O}(nd^2)$ operations to find the Hessian and then $\mathcal{O}(d^3)$ to find its determinant), the Laplace approximation may still be very inaccurate. Numerical integration methods hold the answer to more accurate, but computationally intensive solutions.

The Monte Carlo integration method estimates the expectation of a function $\phi(\mathbf{x})$ under a probability distribution $f(\mathbf{x})$, by taking samples $\{\mathbf{x}^{(i)}\}_{i=1}^{N} \;:\; \mathbf{x}^{(i)} \sim f(\mathbf{x})$. An unbiased estimate, $\hat{\Phi}$, of the expectation of $\phi(\mathbf{x})$ under $f(\mathbf{x})$, using N samples is given by:

$$\Phi = \int d\mathbf{x}\; f(\mathbf{x})\phi(\mathbf{x}) \simeq \hat{\Phi} = \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x}^{(i)}) \;. \tag{1.55}$$

Expectations such as the predictive density, the marginal likelihood, posterior distributions over hidden variables etc. can be obtained using such estimates. Most importantly, the Monte Carlo method returns more accurate and reliable estimates the more samples are taken, and scales well with the dimensionality of $\mathbf{x}$.

In situations where $f(x)$ is hard to sample from, one can use samples from a different auxiliary distribution $g(x)$ and then correct for this by weighting the samples accordingly. This method is called *importance sampling* and it constructs the following estimator using N samples, $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$, generated such that each $\mathbf{x}^{(i)} \sim g(\mathbf{x})$:

$$\Phi = \int d\mathbf{x}\; g(\mathbf{x})\frac{f(\mathbf{x})}{g(\mathbf{x})}\phi(\mathbf{x}) \simeq \hat{\Phi} = \frac{1}{N}\sum_{i=1}^{N} w^{(i)}\phi(\mathbf{x}^{(i)}) \;, \tag{1.56}$$

$$\text{where} \quad w^{(i)} = \frac{f(\mathbf{x}^{(i)})}{g(\mathbf{x}^{(i)})} \tag{1.57}$$

are known as the *importance weights*. Note that the estimator in (1.56) is unbiased just as that in (1.55). It is also possible to estimate Φ even if $p(\mathbf{x})$ and $g(\mathbf{x})$ can be computed only up to

multiplicative constant factors, that is to say: $f(\mathbf{x}) = f^*(\mathbf{x})/\mathcal{Z}_f$ and $g(\mathbf{x}) = g^*(\mathbf{x})/\mathcal{Z}_g$. In such cases it is straightforward to show that an estimator for Φ is given by:

$$\Phi = \int d\mathbf{x}\; g(\mathbf{x}) \frac{f(\mathbf{x})}{g(\mathbf{x})} \phi(\mathbf{x}) \simeq \hat{\Phi} = \frac{\sum_{i=1}^{N} w^{(i)} \phi(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} w^{(i)}} \;, \tag{1.58}$$

$$\text{where} \quad w^{(i)} = \frac{f^*(\mathbf{x}^{(i)})}{g^*(\mathbf{x}^{(i)})} \tag{1.59}$$

are a slightly different set of importance weights. Unfortunately this estimate is now biased as it is really the ratio of two estimates, and the ratio of two unbiased estimates is in general not an unbiased estimate of the ratio. Although importance sampling is simple, $\hat{\Phi}$ can often have very high variance. Indeed, even in some simple models it can be shown that the variance of the weights $w^{(i)}$, and therefore of $\hat{\Phi}$ also, are unbounded. These and related problems are discussed in section 4.7 of chapter 4 where importance sampling is used to estimate the exact marginal likelihood of a mixture of factor analysers model trained with variational Bayesian EM. We use this analysis to provide an assessment of the tightness of the variational lower bound, which indicates how much we are conceding when using such an approximation (see section 4.7.2).

A method related to importance sampling is *rejection sampling*. It avoids the use of a set of weights $\{w^{(i)}\}_{i=1}^{N}$ by stochastically deciding whether or not to include each sample from $g(\mathbf{x})$. The procedure requires the existence of a constant c such that $c\,g(\mathbf{x}) > f(\mathbf{x})$ for all $\mathbf{x}$, that is to say $c\,g(\mathbf{x})$ envelopes the probability density $f(\mathbf{x})$. Samples are obtained from $f(\mathbf{x})$ by drawing samples from $g(\mathbf{x})$, and then accepting or rejecting each stochastically based on the ratio of its densities under $f(\mathbf{x})$ and $g(\mathbf{x})$. That is to say, for each sample an auxiliary variable $u^{(i)} \sim U(0,1)$ is drawn, and the sample under $g(\mathbf{x})$ accepted only if

$$f(\mathbf{x}^{(i)}) > u^{(i)} c\, g(\mathbf{x}^{(i)}) \;. \tag{1.60}$$

Unfortunately, this becomes impractical in high dimensions and with complex functions since it is hard to find a simple choice of $g(\mathbf{x})$ such that c is small enough to allow the rejection rate to remain reasonable across the whole space. Even in simple examples the acceptance rate falls exponentially with the dimensionality of $\mathbf{x}$.

To overcome the limitations of rejection sampling it is possible to adapt the density $c\,g(\mathbf{x})$ so that it envelopes $f(\mathbf{x})$ more tightly, but only in cases where $f(\mathbf{x})$ is log-concave. This method is called *adaptive rejection sampling* (Gilks and Wild, 1992): the envelope function $c\,g(\mathbf{x})$ is piecewise exponential and is updated to more tightly fit the density $f(\mathbf{x})$ after each sample is drawn. The result is that the probability of rejection monotonically decreases with each sample evaluation. However it is only designed for log-concave $f(\mathbf{x})$ and relies on gradient information to construct tangents which upper bound the density $f(\mathbf{x})$. An interesting extension (Gilks, 1992) to this constructs a *lower* bound $b\,l(\mathbf{x})$ as well (where b is a constant) which is updated in a similar fashion using chords between evaluations of $f(\mathbf{x})$. The advantage of also

using a piecewise exponential lower bound is that the method can become very computationally efficient by not having to evaluate densities under $f(\mathbf{x})$ (which we presume is costly) for some samples. To see how this is possible, consider drawing a sample $\mathbf{x}^{(i)}$ which satisfies

$$b\, l(\mathbf{x}^{(i)}) > u^{(i)} c\, g(\mathbf{x}^{(i)}) \,. \tag{1.61}$$

This sample can be automatically accepted *without* evaluation of $f(\mathbf{x}^{(i)})$, since if inequality (1.61) is satisfied then automatically inequality (1.60) is also satisfied. If the sample does not satisfy (1.61), then of course $f(\mathbf{x}^{(i)})$ needs to be computed, but this can then be used to tighten the bound further. Gilks and Wild (1992) report that the number of density evaluations required to sample N points from $f(\mathbf{x})$ increases as $\sqrt[3]{N}$, even for quite non-Gaussian densities. Their example obtains 100 samples from the standard univariate Gaussian with approximately 15 evaluations, and a further 900 samples with only 15 further evaluations. Moreover, in cases where the log density is close to but not log concave, the adaptive rejection sampling algorithm can still be used with Metropolis methods (see below) to correct for this (Gilks et al., 1995).

Markov chain Monte Carlo (MCMC) methods (as reviewed in Neal, 1992) can be used to generate a chain of samples, starting from $\mathbf{x}^{(1)}$, such that the next sample is a non-deterministic function of the previous sample: $\mathbf{x}^{(i)} \overset{\mathcal{P}}{\leftarrow} \mathbf{x}^{(i-1)}$, where we define $\mathcal{P}(\mathbf{x}', \mathbf{x})$ as the probability of transition from $\mathbf{x}'$ to $\mathbf{x}$. If $\mathcal{P}$ has $f(\mathbf{x})$ as its stationary (equilibrium) distribution, i.e. $f(\mathbf{x}) = \int d\mathbf{x}'\ f(\mathbf{x}')\mathcal{P}(\mathbf{x}', \mathbf{x})$, then the set $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ can be used to obtain an unbiased estimate of Φ as in (1.55) in the limit of a large number of samples. The set of samples have to drawn from the equilibrium distribution, so it is advisable to discard all samples visited at the beginning of the chain. In general $\mathcal{P}$ is implemented using a proposal density $\mathbf{x}^{(i)} \sim g(\mathbf{x}, \mathbf{x}^{(i-1)})$ about the previous sample. In order to ensure *reversibility* of the Markov chain, the probability of accepting the proposal needs to take into account the probability of a reverse transition. This gives rise to the the Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970) acceptance function $a(\cdot, \cdot)$:

$$a(\mathbf{x}^{(i)}, \mathbf{x}^{(i-1)}) = \frac{f^*(\mathbf{x}^{(i)}) g(\mathbf{x}^{(i-1)}, \mathbf{x}^{(i)})}{f^*(\mathbf{x}^{(i-1)}) g(\mathbf{x}^{(i)}, \mathbf{x}^{(i-1)})} \,. \tag{1.62}$$

If $a(\mathbf{x}^{(i)}, \mathbf{x}^{(i-1)}) \geq 1$ the sample is accepted, otherwise it is accepted according to the probability $a(\mathbf{x}^{(i)}, \mathbf{x}^{(i-1)})$. Several extensions to the MCMC method have been proposed including over-relaxation (Adler, 1981), hybrid MCMC (Neal, 1993), and reversible-jump MCMC (Green, 1995). These and many others can be found at the MCMC preprint service (Brooks).

Whilst MCMC sampling methods are guaranteed to yield exact estimates in the limit of a large number of samples, even for well-designed procedures the number of samples required for accurate estimates can be infeasibly large. There is a large amount of active research dedicated to constructing measures to ascertain whether the Markov chain has reached equilibrium, whether the samples it generates are independent, and analysing the reliability of the estimates. This

thesis is concerned with fast, reliable, deterministic alternatives to MCMC. Long MCMC runs can then be used to check the accuracy of these deterministic methods.

In contrast to MCMC methods, a new class of sampling methods has been recently devised in which samples from exactly the equilibrium distribution are generated in a finite number of steps of a Markov chain. These are termed *exact sampling* methods, and make use of trajectory *coupling* and *coalescence* via pseudorandom transitions, and is sometimes referred to as *coupling from the past* (Propp and Wilson, 1996). Variations on exact sampling include interruptible algorithms (Fill, 1998) and continuous state-space versions (Murdoch and Green, 1998). Such methods have been applied to graphical models for machine learning problems in the contexts of mixture modelling (Casella et al., 2000), and noisy-or belief networks (Harvey and Neal, 2000).

Finally, one important role of MCMC methods is to compute partition functions. One such powerful method for computing normalisation constants, such as $\mathcal{Z}_f$ used above, is called *annealed importance sampling* (Neal, 2001). It is based on methods such as thermodynamic integration for estimating the free energy of systems at different temperatures, and work on tempered transitions (Neal, 1996). It estimates the ratio of two normalisation constants $\mathcal{Z}_t$ and $\mathcal{Z}_0$, which we can think of for our purposes as the ratio of marginal likelihoods of two models, by collating the results of a chain of intermediate likelihood ratios of 'close' models,

$$\frac{\mathcal{Z}_t}{\mathcal{Z}_0} = \frac{\mathcal{Z}_1}{\mathcal{Z}_0} \cdots \frac{\mathcal{Z}_{t-2}}{\mathcal{Z}_{t-3}} \frac{\mathcal{Z}_{t-1}}{\mathcal{Z}_{t-2}} \frac{\mathcal{Z}_t}{\mathcal{Z}_{t-1}} \,. \tag{1.63}$$

Each of the ratios is estimated using samples from a Markov chain Monte Carlo method. We will look at this method in much more detail in Chapter 6, where it will be used as a gold standard against which we test the ability of the variational Bayesian EM algorithm to approximate the marginal likelihoods of a large set of models.

To conclude this section we note that Monte Carlo is a purely frequentist procedure and in the words of O'Hagan (1987) is 'fundamentally unsound'. The objections raised therein can be summarised as follows. First, the estimate $\hat{\Phi}$ depends on the sampling density $g(\mathbf{x})$, even though $g(\mathbf{x})$ itself is ancillary to the estimation. Put another way, the same set of samples $\{\mathbf{x}^{(i)}\}_{i=1}^n$, conveying exactly the same information about $p(\mathbf{x})$, but generated under a different $g(\mathbf{x})$ would produce a different estimate $\hat{\Phi}$. Of course, the density $g(\mathbf{x})$ is often tailored to the problem at hand and so we would expect it to contain some of the essence of the estimate. Second, the estimate does not depend on the location of the $\mathbf{x}^{(i)}$s, but only on function evaluations at those points, e.g. $f(\mathbf{x}^{(i)})$. This is surely suboptimal, as the spatial distribution of the function evaluations provides information on the integrand $f(\mathbf{x})\phi(\mathbf{x})$ as a whole. To summarise, classical Monte Carlo bases its estimate on irrelevant information, $g(\mathbf{x})$, and also discards relevant information from the location of the samples. Bayesian variants of Monte Carlo integration procedures have been devised to address these objections using Gaussian process models (O'Hagan, 1991; Rasmussen and Ghahramani, 2003), and there is much future work to do in this direction.

1.4 Summary of the remaining chapters

Chapter 2 Forms the theoretical core of the thesis, and examines the use of variational methods for obtaining lower bounds on the likelihood (for point-parameter learning) and the marginal likelihood (in the case of Bayesian learning). The implications of VB applied to the large family of *conjugate-exponential* graphical models are investigated, for both directed and undirected representations. In particular, a general algorithm for conjugate-exponential models is derived and it is shown that existing propagation algorithms can be employed for inference, with approximately the same complexity as for point-parameters.

In addition, the relations of VB to a number of other commonly used approximations are covered. In particular, it is shown that the Cheeseman-Stutz (CS) score is in fact a looser lower bound on the marginal likelihood than the VB score.

Chapter 3 Applies the results of chapter 2 to hidden Markov models (HMMs). It is shown that it is possible to recover the number of hidden states required to model a synthetic data set, and that the variational Bayesian algorithm can outperform maximum likelihood and maximum a posteriori parameter learning algorithms on real data in terms of generalisation.

Chapter 4 Applies the variational Bayesian method to a mixtures of factor analysers (MFA) problem, where it is shown that the procedure can automatically determine the optimal number of components and the local dimensionality of each component (i.e. the number of factors in each analyser). Through a stochastic procedure for adding components to the model, it is possible to perform the variational optimisation incrementally and avoid local maxima. The algorithm is shown to perform well on a variety of synthetic data sets, and is compared to a BIC-penalised maximum likelihood algorithm on a real-world data set of hand-written digits.

This chapter also investigates the generally applicable method of drawing importance samples from the variational approximation to estimate the marginal likelihood and the KL divergence between the approximate and exact posterior. Specific results applying variants of this procedure to the MFA model are analysed.

Chapter 5 Presents an application of the theorems presented in chapter 2 to linear dynamical systems (LDSs). The result is the derivation of a variational Bayesian input-dependent Rauch-Tung-Striebel smoother, such that it is possible to infer the posterior hidden state trajectory whilst integrating over all model parameters. Experiments on synthetic data show that it is possible to infer the dimensionality of the hidden state space and determine which dimensions of the inputs and the data are relevant. Also presented are preliminary experiments for elucidating gene-gene interactions in a well-studied human immune response mechanism.

Chapter 6 Investigates a novel application of the VB framework to approximating the marginal likelihood of discrete-variable directed acyclic graphs (DAGs) that contain hidden variables. The VB lower bound is compared to MAP, BIC, CS, and annealed importance sampling (AIS), on a simple (yet non-trivial) model selection task of determining which of all possible structures within a class generated a data set.

The chapter also discusses extensions and improvements to the particular form of AIS used, and suggests related approximations which may be of interest.

Chapter 7 Concludes the thesis with a discussion on some topics closely related to the ideas already investigated. These include: Bethe and Kikuchi approximations, infinite models, inferring causality using the marginal likelihood, and automated algorithm derivation. The chapter then concludes with a summary of the main contributions of the thesis.

Chapter 2

Variational Bayesian Theory

2.1 Introduction

This chapter covers the majority of the theory for variational Bayesian learning that will be used in rest of this thesis. It is intended to give the reader a context for the use of variational methods as well as a insight into their general applicability and usefulness.

In a model selection task the role of a Bayesian is to calculate the posterior distribution over a set of models given some a priori knowledge and some new observations (data). The knowledge is represented in the form of a prior over model structures $p(m)$, and their parameters $p(\boldsymbol{\theta} \,|\, m)$ which define the probabilistic dependencies between the variables in the model. By Bayes' rule, the posterior over models m having seen data $\mathbf{y}$ is given by:

$$p(m \,|\, \mathbf{y}) = \frac{p(m)p(\mathbf{y} \,|\, m)}{p(\mathbf{y})} \,. \tag{2.1}$$

The second term in the numerator is the *marginal likelihood* or *evidence* for a model m, and is the key quantity for Bayesian model selection:

$$p(\mathbf{y} \,|\, m) = \int d\boldsymbol{\theta} \; p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m) \,. \tag{2.2}$$

For each model structure we can compute the posterior distribution over parameters:

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}, m) = \frac{p(\boldsymbol{\theta} \,|\, m)p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)}{p(\mathbf{y} \,|\, m)} \,. \tag{2.3}$$

We might also be interested in calculating other related quantities, such as the *predictive density* of a new datum $\mathbf{y}'$ given a data set $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$:

$$p(\mathbf{y}' \,|\, \mathbf{y}, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y}, m)\, p(\mathbf{y}' \,|\, \boldsymbol{\theta}, \mathbf{y}, m)\;, \tag{2.4}$$

which can be simplified into

$$p(\mathbf{y}' \,|\, \mathbf{y}, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y}, m)\, p(\mathbf{y}' \,|\, \boldsymbol{\theta}, m) \tag{2.5}$$

if $\mathbf{y}'$ is conditionally independent of $\mathbf{y}$ given $\boldsymbol{\theta}$. We also may be interested in calculating the posterior distribution of a hidden variable, $\mathbf{x}'$, associated with the new observation $\mathbf{y}'$

$$p(\mathbf{x}' \,|\, \mathbf{y}', \mathbf{y}, m) \propto \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y}, m)\, p(\mathbf{x}', \mathbf{y}' \,|\, \boldsymbol{\theta}, m)\;. \tag{2.6}$$

The simplest way to approximate the above integrals is to estimate the value of the integrand at a single point estimate of $\boldsymbol{\theta}$, such as the maximum likelihood (ML) or the maximum a posteriori (MAP) estimates, which aim to maximise respectively the second and both terms of the integrand in (2.2),

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}}\; p(\mathbf{y} \,|\, \boldsymbol{\theta}, m) \tag{2.7}$$

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}}\; p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)\;. \tag{2.8}$$

ML and MAP examine only probability *density*, rather than *mass*, and so can neglect potentially large contributions to the integral. A more principled approach is to estimate the integral numerically by evaluating the integrand at many different $\boldsymbol{\theta}$ via Monte Carlo methods. In the limit of an infinite number of samples of $\boldsymbol{\theta}$ this produces an accurate result, but despite ingenious attempts to curb the curse of dimensionality in $\boldsymbol{\theta}$ using methods such as Markov chain Monte Carlo, these methods remain prohibitively computationally intensive in interesting models. These methods were reviewed in the last chapter, and the bulk of this chapter concentrates on a third way of approximating the integral, using *variational* methods. The key to the variational method is to approximate the integral with a simpler form that is tractable, forming a lower or upper *bound*. The integration then translates into the implementationally simpler problem of bound *optimisation*: making the bound as tight as possible to the true value.

We begin in section 2.2 by describing how variational methods can be used to derive the well-known expectation-maximisation (EM) algorithm for learning the maximum likelihood (ML) parameters of a model. In section 2.3 we concentrate on the Bayesian methodology, in which priors are placed on the parameters of the model, and their uncertainty integrated over to give the *marginal likelihood* (2.2). We then generalise the variational procedure to yield the *variational Bayesian EM* (VBEM) algorithm, which iteratively optimises a lower bound on this marginal

likelihood. In analogy to the EM algorithm, the iterations consist of a variational Bayesian E (VBE) step in which the hidden variables are inferred using an *ensemble* of models according to their posterior probability, and a variational Bayesian M (VBM) step in which a posterior *distribution* over model parameters is inferred. In section 2.4 we specialise this algorithm to a large class of models which we call *conjugate-exponential* (CE): we present the variational Bayesian EM algorithm for CE models and discuss the implications for both directed graphs (Bayesian networks) and undirected graphs (Markov networks) in section 2.5. In particular we show that we can incorporate existing propagation algorithms into the variational Bayesian framework and that the complexity of inference for the variational Bayesian treatment is approximately the same as for the ML scenario. In section 2.6 we compare VB to the BIC and Cheeseman-Stutz criteria, and finally summarise in section 2.7.

2.2 Variational methods for ML / MAP learning

In this section we review the derivation of the EM algorithm for probabilistic models with hidden variables. The algorithm is derived using a variational approach, and has exact and approximate versions. We investigate themes on convexity, computational tractability, and the Kullback-Leibler divergence to give a deeper understanding of the EM algorithm. The majority of the section concentrates on maximum likelihood (ML) learning of the parameters; at the end we present the simple extension to maximum a posteriori (MAP) learning. The hope is that this section provides a good stepping-stone on to the variational Bayesian EM algorithm that is presented in the subsequent sections and used throughout the rest of this thesis.

2.2.1 The scenario for parameter learning

Consider a model with hidden variables $\mathbf{x}$ and observed variables $\mathbf{y}$. The parameters describing the (potentially) stochastic dependencies between variables are given by $\boldsymbol{\theta}$. In particular consider the generative model that produces a dataset $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ consisting of n independent and identically distributed (i.i.d.) items, generated using a set of hidden variables $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ such that the likelihood can be written as a function of $\boldsymbol{\theta}$ in the following way:

$$p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} \int d\mathbf{x}_i \; p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \; . \tag{2.9}$$

The integration over hidden variables $\mathbf{x}_i$ is required to form the likelihood of the parameters, as a function of just the observed data $\mathbf{y}_i$. We have assumed that the hidden variables are continuous as opposed to discrete (hence an integral rather than a summation), but we do so without loss of generality. As a point of nomenclature, note that we use $\mathbf{x}_i$ and $\mathbf{y}_i$ to denote collections of $|\mathbf{x}_i|$ hidden and $|\mathbf{y}_i|$ observed variables respectively: $\mathbf{x}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{i|\mathbf{x}_i|}\}$, and

$\mathbf{y}_i = \{\mathbf{y}_{i1}, \ldots, \mathbf{y}_{i|\mathbf{y}_i|}\}$. We use $|\cdot|$ notation to denote the size of the collection of variables. ML learning seeks to find the parameter setting $\boldsymbol{\theta}_{\mathrm{ML}}$ that maximises this likelihood, or equivalently the logarithm of this likelihood,

$$\mathcal{L}(\boldsymbol{\theta}) \equiv \ln p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \int d\mathbf{x}_i \; p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \tag{2.10}$$

so defining

$$\boldsymbol{\theta}_{\mathrm{ML}} \equiv \arg\max_{\boldsymbol{\theta}} \; \mathcal{L}(\boldsymbol{\theta}) \; . \tag{2.11}$$

To keep the derivations clear, we write $\mathcal{L}$ as a function of $\boldsymbol{\theta}$ only; the dependence on $\mathbf{y}$ is implicit. In Bayesian networks without hidden variables and with independent parameters, the log-likelihood decomposes into local terms on each $\mathbf{y}_{ij}$, and so finding the setting of each parameter of the model that maximises the likelihood is straightforward. Unfortunately, if some of the variables are hidden this will in general induce dependencies between all the parameters of the model and so make maximising (2.10) difficult. Moreover, for models with many hidden variables, the integral (or sum) over $\mathbf{x}$ can be intractable.

We simplify the problem of maximising $\mathcal{L}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ by introducing an auxiliary distribution over the hidden variables. *Any* probability distribution $q_{\mathbf{x}}(\mathbf{x})$ over the hidden variables gives rise to a *lower bound* on $\mathcal{L}$. In fact, for each data point $\mathbf{y}_i$ we use a distinct distribution $q_{\mathbf{x}_i}(\mathbf{x}_i)$ over the hidden variables to obtain the lower bound:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \sum_i \ln \int d\mathbf{x}_i \; p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) && (2.12)\\
&= \sum_i \ln \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \frac{p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta})}{q_{\mathbf{x}_i}(\mathbf{x}_i)} && (2.13)\\
&\geq \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta})}{q_{\mathbf{x}_i}(\mathbf{x}_i)} && (2.14)\\
&= \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) - \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln q_{\mathbf{x}_i}(\mathbf{x}_i) && (2.15)\\
&\equiv \mathcal{F}(q_{\mathbf{x}_1}(\mathbf{x}_1), \ldots, q_{\mathbf{x}_n}(\mathbf{x}_n), \boldsymbol{\theta}) && (2.16)
\end{aligned}$$

where we have made use of Jensen's inequality (Jensen, 1906) which follows from the fact that the $\log$ function is concave. $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ is a lower bound on $\mathcal{L}(\boldsymbol{\theta})$ and is a functional of the free distributions $q_{\mathbf{x}_i}(\mathbf{x}_i)$ and of $\boldsymbol{\theta}$ (the dependence on $\mathbf{y}$ is left implicit). Here we use $q_{\mathbf{x}}(\mathbf{x})$ to mean the set $\{q_{\mathbf{x}_i}(\mathbf{x}_i)\}_{i=1}^n$. Defining the *energy* of a global configuration $(\mathbf{x}, \mathbf{y})$ to be $-\ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta})$, the lower bound $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}) \leq \mathcal{L}(\boldsymbol{\theta})$ is the negative of a quantity known in statistical physics as the *free energy*: the expected energy under $q_{\mathbf{x}}(\mathbf{x})$ minus the entropy of $q_{\mathbf{x}}(\mathbf{x})$ (Feynman, 1972; Neal and Hinton, 1998).

2.2.2 EM for unconstrained (exact) optimisation

The Expectation-Maximization (EM) algorithm (Baum et al., 1970; Dempster et al., 1977) alternates between an E step, which infers posterior distributions over hidden variables given a current parameter setting, and an M step, which maximises $\mathcal{L}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ given the statistics gathered from the E step. Such a set of updates can be derived using the lower bound: at each iteration, the E step maximises $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ with respect to each of the $q_{\mathbf{x}_i}(\mathbf{x}_i)$, and the M step does so with respect to $\boldsymbol{\theta}$. Mathematically speaking, using a superscript (t) to denote iteration number, starting from some initial parameters $\boldsymbol{\theta}^{(0)}$, the update equations would be:

$$\textbf{E step:} \qquad q_{\mathbf{x}_i}^{(t+1)} \leftarrow \arg\max_{q_{\mathbf{x}_i}} \mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}^{(t)}) \;, \quad \forall\, i \in \{1, \ldots, n\} \;, \tag{2.17}$$

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{F}(q_{\mathbf{x}}^{(t+1)}(\mathbf{x}), \boldsymbol{\theta}) \;. \tag{2.18}$$

For the E step, it turns out that the maximum over $q_{\mathbf{x}_i}(\mathbf{x}_i)$ of the bound (2.14) is obtained by setting

$$q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) = p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \;, \quad \forall\, i \;, \tag{2.19}$$

at which point the bound becomes an equality. This can be proven by direct substitution of (2.19) into (2.14):

$$\mathcal{F}(q_{\mathbf{x}}^{(t+1)}(\mathbf{x}), \boldsymbol{\theta}^{(t)}) = \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)})}{q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i)} \tag{2.20}$$

$$= \sum_i \int d\mathbf{x}_i \; p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \ln \frac{p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)})}{p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)})} \tag{2.21}$$

$$= \sum_i \int d\mathbf{x}_i \; p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \ln \frac{p(\mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)})\, p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)})}{p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)})} \tag{2.22}$$

$$= \sum_i \int d\mathbf{x}_i \; p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)}) \tag{2.23}$$

$$= \sum_i \ln p(\mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)}) = \mathcal{L}(\boldsymbol{\theta}^{(t)}) \;, \tag{2.24}$$

where the last line follows as $\ln p(\mathbf{y}_i \,|\, \boldsymbol{\theta})$ is not a function of $\mathbf{x}_i$. After this E step the bound is tight. The same result can be obtained by functionally differentiating $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ with respect to $q_{\mathbf{x}_i}(\mathbf{x}_i)$, and setting to zero, subject to the normalisation constraints:

$$\int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) = 1 \;, \quad \forall\, i \;. \tag{2.25}$$

The constraints on each $q_{\mathbf{x}_i}(\mathbf{x}_i)$ can be implemented using Lagrange multipliers $\{\lambda_i\}_{i=1}^n$, forming the new functional:

$$\tilde{\mathcal{F}}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}) = \mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}) + \sum_i \lambda_i \left[\int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) - 1 \right] . \tag{2.26}$$

We then take the functional derivative of this expression with respect to each $q_{\mathbf{x}_i}(\mathbf{x}_i)$ and equate to zero, obtaining the following

$$\frac{\partial}{\partial q_{\mathbf{x}_i}(\mathbf{x}_i)} \tilde{\mathcal{F}}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}^{(t)}) = \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)}) - \ln q_{\mathbf{x}_i}(\mathbf{x}_i) - 1 + \lambda_i = 0 \tag{2.27}$$

$$\implies q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) = \exp(-1 + \lambda_i)\, p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)}) \tag{2.28}$$

$$= p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) , \quad \forall\, i , \tag{2.29}$$

where each λ_i is related to the normalisation constant:

$$\lambda_i = 1 - \ln \int d\mathbf{x}_i \; p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}^{(t)}) , \quad \forall\, i . \tag{2.30}$$

In the remaining derivations in this thesis we always enforce normalisation constraints using Lagrange multiplier terms, although they may not always be explicitly written.

The M step is achieved by simply setting derivatives of (2.14) with respect to $\boldsymbol{\theta}$ to zero, which is the same as optimising the expected energy term in (2.15) since the entropy of the hidden state distribution $q_{\mathbf{x}}(\mathbf{x})$ is not a function of $\boldsymbol{\theta}$:

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_i \int d\mathbf{x}_i \; p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) . \tag{2.31}$$

Note that the optimisation is over the second $\boldsymbol{\theta}$ in the integrand, whilst holding $p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)})$ fixed. Since $\mathcal{F}(q_{\mathbf{x}}^{(t+1)}(\mathbf{x}), \boldsymbol{\theta}^{(t)}) = \mathcal{L}(\boldsymbol{\theta}^{(t)})$ at the beginning of each M step, and since the E step does not change the parameters, the likelihood is guaranteed not to decrease after each combined EM step. This is the well known lower bound interpretation of EM: $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ is an auxiliary function which lower bounds $\mathcal{L}(\boldsymbol{\theta})$ for any $q_{\mathbf{x}}(\mathbf{x})$, attaining equality after each E step. These steps are shown schematically in figure 2.1. Here we have expressed the E step as obtaining the full distribution over the hidden variables for each data point. However we note that, in general, the M step may require only a few statistics of the hidden variables, so only these need be computed in the E step.

2.2.3 EM with constrained (approximate) optimisation

Unfortunately, in many interesting models the data are explained by multiple interacting hidden variables which can result in intractable posterior distributions (Williams and Hinton, 1991;

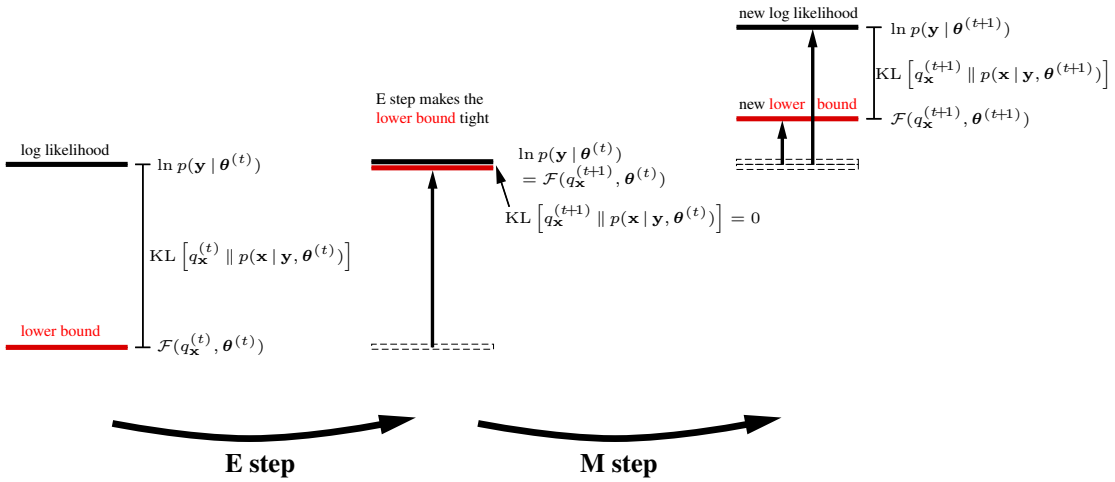


Figure 2.1: The variational interpretation of EM for maximum likelihood learning. In the E step the hidden variable variational posterior is set to the exact posterior $p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)})$, making the bound tight. In the M step the parameters are set to maximise the lower bound $\mathcal{F}(q_{\mathbf{x}}^{(t+1)}, \boldsymbol{\theta})$ while holding the distribution over hidden variables $q_{\mathbf{x}}^{(t+1)}(\mathbf{x})$ fixed.

Neal, 1992; Hinton and Zemel, 1994; Ghahramani and Jordan, 1997; Ghahramani and Hinton, 2000). In the variational approach we can constrain the posterior distributions to be of a particular tractable form, for example factorised over the variable $\mathbf{x}_i = \{\mathbf{x}_{ij}\}_{j=1}^{|\mathbf{x}_i|}$. Using calculus of variations we can still optimise $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ as a functional of constrained distributions $q_{\mathbf{x}_i}(\mathbf{x}_i)$. The M step, which optimises $\boldsymbol{\theta}$, is conceptually identical to that described in the previous subsection, except that it is based on sufficient statistics calculated with respect to the constrained posterior $q_{\mathbf{x}_i}(\mathbf{x}_i)$ instead of the exact posterior.

We can write the lower bound $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ as

$$\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}) = \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{p(\mathbf{x}_i, \mathbf{y}_i \mid \boldsymbol{\theta})}{q_{\mathbf{x}_i}(\mathbf{x}_i)} \tag{2.32}$$

$$= \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln p(\mathbf{y}_i \mid \boldsymbol{\theta}) + \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{p(\mathbf{x}_i \mid \mathbf{y}_i, \boldsymbol{\theta})}{q_{\mathbf{x}_i}(\mathbf{x}_i)} \tag{2.33}$$

$$= \sum_i \ln p(\mathbf{y}_i \mid \boldsymbol{\theta}) - \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{q_{\mathbf{x}_i}(\mathbf{x}_i)}{p(\mathbf{x}_i \mid \mathbf{y}_i, \boldsymbol{\theta})} \; . \tag{2.34}$$

Thus in the E step, maximising $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ with respect to $q_{\mathbf{x}_i}(\mathbf{x}_i)$ is equivalent to minimising the following quantity

$$\int d\mathbf{x}_i \; q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{q_{\mathbf{x}_i}(\mathbf{x}_i)}{p(\mathbf{x}_i \mid \mathbf{y}_i, \boldsymbol{\theta})} \equiv \mathrm{KL}\left[q_{\mathbf{x}_i}(\mathbf{x}_i) \,\|\, p(\mathbf{x}_i \mid \mathbf{y}_i, \boldsymbol{\theta})\right] \tag{2.35}$$

$$\geq 0 \; , \tag{2.36}$$

which is the Kullback-Leibler divergence between the variational distribution $q_{\mathbf{x}_i}(\mathbf{x}_i)$ and the exact hidden variable posterior $p(\mathbf{x}_i \mid \mathbf{y}_i, \boldsymbol{\theta})$. As is shown in figure 2.2, the E step does not

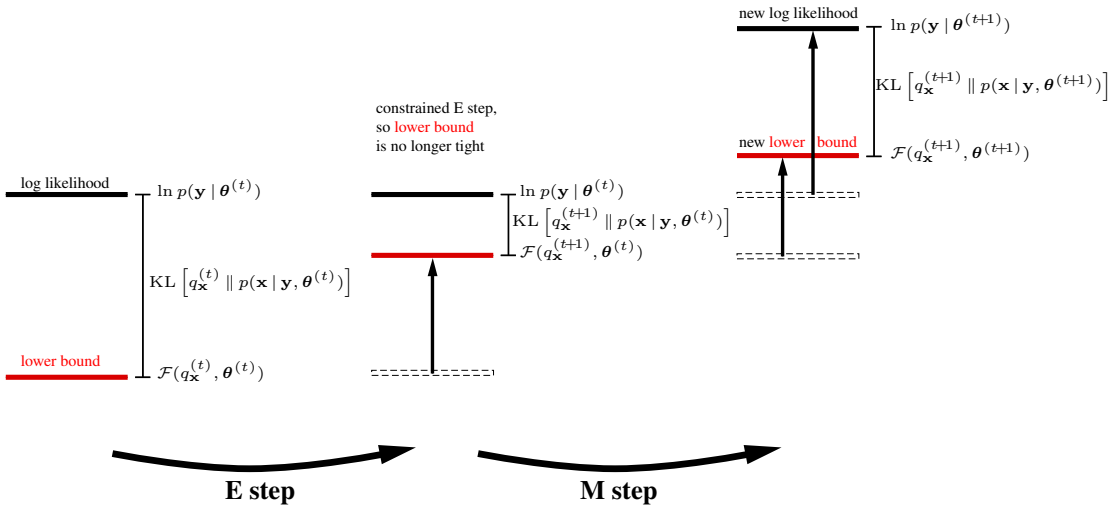


Figure 2.2: The variational interpretation of constrained EM for maximum likelihood learning. In the E step the hidden variable variational posterior is set to that which minimises $\mathrm{KL}\left[q_{\mathbf{x}}(\mathbf{x}) \,\|\, p(\mathbf{x} \,|\, \mathbf{y}, \boldsymbol{\theta}^{(t)})\right]$, subject to $q_{\mathbf{x}}(\mathbf{x})$ lying in the family of constrained distributions. In the M step the parameters are set to maximise the lower bound $\mathcal{F}(q_{\mathbf{x}}^{(t+1)}, \boldsymbol{\theta})$ given the current distribution over hidden variables.

generally result in the bound becoming an equality, unless of course the exact posterior lies in the family of constrained posteriors $q_{\mathbf{x}}(\mathbf{x})$.

The M step looks very similar to (2.31), but is based on the current variational posterior over hidden variables:

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \; . \tag{2.37}$$

One can choose $q_{\mathbf{x}_i}(\mathbf{x}_i)$ to be in a particular parameterised family:

$$q_{\mathbf{x}_i}(\mathbf{x}_i) = q_{\mathbf{x}_i}(\mathbf{x}_i \,|\, \boldsymbol{\lambda}_i) \tag{2.38}$$

where $\boldsymbol{\lambda}_i = \{\boldsymbol{\lambda}_{i1}, \ldots, \boldsymbol{\lambda}_{ir}\}$ are r *variational parameters* for each datum. If we constrain each $q_{\mathbf{x}_i}(\mathbf{x}_i \,|\, \boldsymbol{\lambda}_i)$ to have easily computable moments (e.g. a Gaussian), and especially if $\ln p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta})$ is polynomial in $\mathbf{x}_i$, then we can compute the KL divergence up to a constant and, more importantly, can take its derivatives with respect to the set of variational parameters $\boldsymbol{\lambda}_i$ of each $q_{\mathbf{x}_i}(\mathbf{x}_i)$ distribution to perform the constrained E step. The E step of the *variational EM* algorithm therefore consists of a sub-loop in which each of the $q_{\mathbf{x}_i}(\mathbf{x}_i \,|\, \boldsymbol{\lambda}_i)$ is optimised by taking derivatives with respect to each $\boldsymbol{\lambda}_{is}$, for $s = 1, \ldots, r$.

The mean field approximation

The *mean field* approximation is the case in which each $q_{\mathbf{x}_i}(\mathbf{x}_i)$ is fully factorised over the hidden variables:

$$q_{\mathbf{x}_i}(\mathbf{x}_i) = \prod_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \; . \tag{2.39}$$

In this case the expression for $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta})$ given by (2.32) becomes:

$$\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \boldsymbol{\theta}) = \sum_i \int d\mathbf{x}_i \; \left[\prod_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) - \prod_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \ln \prod_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \right] \tag{2.40}$$

$$= \sum_i \int d\mathbf{x}_i \; \left[\prod_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) - \sum_{j=1}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \ln q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) \right] \; . \tag{2.41}$$

Using a Lagrange multiplier to enforce normalisation of the each of the approximate posteriors, we take the functional derivative of this form with respect to each $q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij})$ and equate to zero, obtaining:

$$q_{\mathbf{x}_{ij}}(\mathbf{x}_{ij}) = \frac{1}{\mathcal{Z}_{ij}} \exp \left[\int d\mathbf{x}_{i/j} \prod_{j'/j}^{|\mathbf{x}_i|} q_{\mathbf{x}_{ij'}}(\mathbf{x}_{ij'}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right] \; , \tag{2.42}$$

for each data point $i \in \{1, \ldots, n\}$, and each variational factorised component $j \in \{1, \ldots, |\mathbf{x}_i|\}$. We use the notation $d\mathbf{x}_{i/j}$ to denote the element of integration for all items in $\mathbf{x}_i$ except $\mathbf{x}_{ij}$, and the notation $\prod_{j'/j}$ to denote a product of all terms excluding j. For the ith datum, it is clear that the update equation (2.42) applied to each hidden variable j in turn represents a set of coupled equations for the approximate posterior over each hidden variable. These fixed point equations are called *mean-field equations* by analogy to such methods in statistical physics. Examples of these variational approximations can be found in the following: Ghahramani (1995); Saul et al. (1996); Jaakkola (1997); Ghahramani and Jordan (1997).

EM for maximum a posteriori learning

In MAP learning the parameter optimisation includes prior information about the parameters $p(\boldsymbol{\theta})$, and the M step seeks to find

$$\boldsymbol{\theta}_{\text{MAP}} \equiv \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \; . \tag{2.43}$$

In the case of an exact E step, the M step is simply augmented to:

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \left[\ln p(\boldsymbol{\theta}) + \sum_i \int d\mathbf{x}_i \; p(\mathbf{x}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right] . \tag{2.44}$$

In the case of a constrained approximate E step, the M step is given by

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \left[\ln p(\boldsymbol{\theta}) + \sum_i \int d\mathbf{x}_i \; q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right] . \tag{2.45}$$

However, as mentioned in section 1.3.1, we reiterate that an undesirable feature of MAP estimation is that it is inherently basis-dependent: it is always possible to find a basis in which any particular $\boldsymbol{\theta}^*$ is the MAP solution, provided $\boldsymbol{\theta}^*$ has non-zero prior probability.

2.3 Variational methods for Bayesian learning

In this section we show how to extend the above treatment to use variational methods to approximate the integrals required for Bayesian learning. By treating the parameters as unknown quantities as well as the hidden variables, there are now correlations between the parameters and hidden variables in the posterior. The basic idea in the VB framework is to approximate the distribution over both hidden variables and parameters with a simpler distribution, usually one which assumes that the hidden states and parameters are independent given the data.

There are two main goals in Bayesian learning. The first is approximating the marginal likelihood $p(\mathbf{y} \,|\, m)$ in order to perform model comparison. The second is approximating the posterior distribution over the parameters of a model $p(\boldsymbol{\theta} \,|\, \mathbf{y}, m)$, which can then be used for prediction.

2.3.1 Deriving the learning rules

As before, let $\mathbf{y}$ denote the observed variables, $\mathbf{x}$ denote the hidden variables, and $\boldsymbol{\theta}$ denote the parameters. We assume a prior distribution over parameters $p(\boldsymbol{\theta} \,|\, m)$ conditional on the model m. The marginal likelihood of a model, $p(\mathbf{y} \,|\, m)$, can be lower bounded by introducing any

distribution over both latent variables and parameters which has support where $p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)$ does, by appealing to Jensen's inequality once more:

$$\ln p(\mathbf{y} \,|\, m) = \ln \int d\boldsymbol{\theta}\, d\mathbf{x}\; p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta} \,|\, m) \tag{2.46}$$

$$= \ln \int d\boldsymbol{\theta}\, d\mathbf{x}\; q(\mathbf{x}, \boldsymbol{\theta}) \frac{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta} \,|\, m)}{q(\mathbf{x}, \boldsymbol{\theta})} \tag{2.47}$$

$$\geq \int d\boldsymbol{\theta}\, d\mathbf{x}\; q(\mathbf{x}, \boldsymbol{\theta}) \ln \frac{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta} \,|\, m)}{q(\mathbf{x}, \boldsymbol{\theta})} \,. \tag{2.48}$$

Maximising this lower bound with respect to the free distribution $q(\mathbf{x}, \boldsymbol{\theta})$ results in $q(\mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)$ which when substituted above turns the inequality into an equality (in exact analogy with (2.19)). This does not simplify the problem since evaluating the exact posterior distribution $p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)$ requires knowing its normalising constant, the marginal likelihood. Instead we constrain the posterior to be a simpler, factorised (separable) approximation to $q(\mathbf{x}, \boldsymbol{\theta}) \approx q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$:

$$\ln p(\mathbf{y} \,|\, m) \geq \int d\boldsymbol{\theta}\, d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta} \,|\, m)}{q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \tag{2.49}$$

$$= \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \left[\int d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x}) \ln \frac{p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)}{q_{\mathbf{x}}(\mathbf{x})} + \ln \frac{p(\boldsymbol{\theta} \,|\, m)}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \right] \tag{2.50}$$

$$= \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \tag{2.51}$$

$$= \mathcal{F}_m(q_{\mathbf{x}_1}(\mathbf{x}_1), \ldots, q_{\mathbf{x}_n}(\mathbf{x}_n), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \,, \tag{2.52}$$

where the last equality is a consequence of the data $\mathbf{y}$ arriving i.i.d. (this is shown in theorem 2.1 below). The quantity $\mathcal{F}_m$ is a functional of the free distributions, $q_{\mathbf{x}}(\mathbf{x})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$.

The variational Bayesian algorithm iteratively maximises $\mathcal{F}_m$ in (2.51) with respect to the free distributions, $q_{\mathbf{x}}(\mathbf{x})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, which is essentially coordinate ascent in the function space of variational distributions. The following very general theorem provides the update equations for variational Bayesian learning.

Theorem 2.1: Variational Bayesian EM (VBEM).
Let m be a model with parameters $\boldsymbol{\theta}$ giving rise to an i.i.d. data set $\mathbf{y} = \{\mathbf{y}_1, \ldots \mathbf{y}_n\}$ with corresponding hidden variables $\mathbf{x} = \{\mathbf{x}_1, \ldots \mathbf{x}_n\}$. A lower bound on the model log marginal likelihood is

$$\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \int d\boldsymbol{\theta}\, d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta} \,|\, m)}{q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \tag{2.53}$$

and this can be iteratively optimised by performing the following updates, using superscript (t) to denote iteration number:

$$\textbf{VBE step:} \quad q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) = \frac{1}{\mathcal{Z}_{\mathbf{x}_i}} \exp \left[\int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta})\; \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m) \right] \quad \forall\, i \tag{2.54}$$

where

$$q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = \prod_{i=1}^{n} q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) \;, \tag{2.55}$$

and

$$\textbf{VBM step:} \qquad q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}}\, p(\boldsymbol{\theta} \,|\, m)\, \exp\left[\int d\mathbf{x}\; q_{\mathbf{x}}^{(t+1)}(\mathbf{x})\; \ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)\right] \;. \tag{2.56}$$

Moreover, the update rules converge to a local maximum of $\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}))$.

Proof of $q_{\mathbf{x}_i}(\mathbf{x}_i)$ update: using variational calculus.

Take functional derivatives of $\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}))$ with respect to $q_{\mathbf{x}}(\mathbf{x})$, and equate to zero:

$$\frac{\partial}{\partial q_{\mathbf{x}}(\mathbf{x})} \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \left[\frac{\partial}{\partial q_{\mathbf{x}}(\mathbf{x})} \int d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x}) \ln \frac{p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)}{q_{\mathbf{x}}(\mathbf{x})}\right] \tag{2.57}$$

$$= \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \left[\ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m) - \ln q_{\mathbf{x}}(\mathbf{x}) - 1\right] \tag{2.58}$$

$$= 0 \tag{2.59}$$

which implies

$$\ln q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta})\; \ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m) - \ln \mathcal{Z}_{\mathbf{x}}^{(t+1)} \;, \tag{2.60}$$

where $\mathcal{Z}_{\mathbf{x}}$ is a normalisation constant (from a Lagrange multiplier term enforcing normalisation of $q_{\mathbf{x}}(\mathbf{x})$, omitted for brevity). As a consequence of the i.i.d. assumption, this update can be broken down across the n data points

$$\ln q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta}) \sum_{i=1}^{n} \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m) - \ln \mathcal{Z}_{\mathbf{x}}^{(t+1)} \;, \tag{2.61}$$

which implies that the optimal $q_{\mathbf{x}}^{(t+1)}(\mathbf{x})$ is factorised in the form $q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = \prod_{i=1}^{n} q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i)$, with

$$\ln q_{\mathbf{x}_i}^{(t+1)}(\mathbf{x}_i) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta}) \ln p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m) - \ln \mathcal{Z}_{\mathbf{x}_i}^{(t+1)} \quad \forall\, i \;, \tag{2.62}$$

$$\text{with} \qquad \mathcal{Z}_{\mathbf{x}} = \prod_{i=1}^{n} \mathcal{Z}_{\mathbf{x}_i} \;. \tag{2.63}$$

Thus for a given $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, there is a unique stationary point for each $q_{\mathbf{x}_i}(\mathbf{x}_i)$. □

Proof of $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ update: using variational calculus.

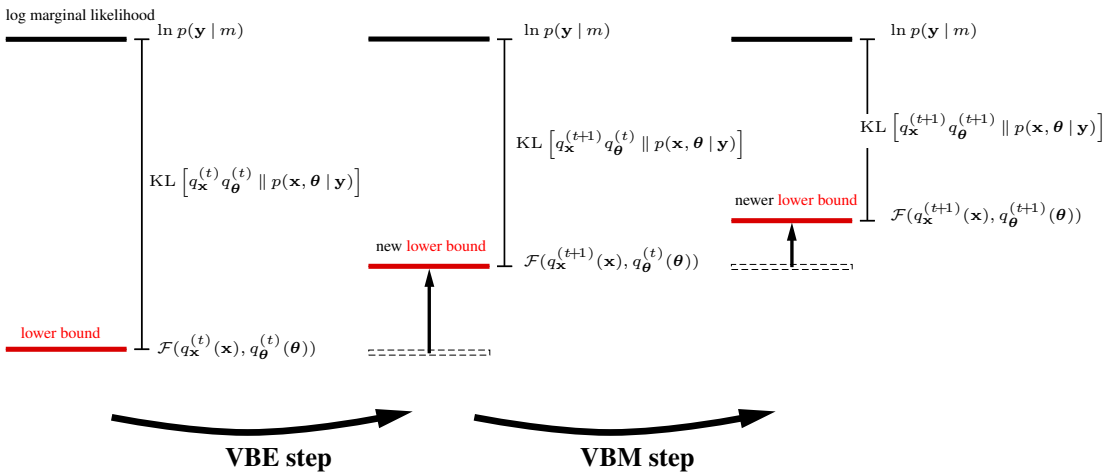


Figure 2.3: The variational Bayesian EM (VBEM) algorithm. In the VBE step, the variational posterior over hidden variables $q_{\mathbf{x}}(\mathbf{x})$ is set according to (2.60). In the VBM step, the variational posterior over parameters is set according to (2.56). Each step is guaranteed to increase (or leave unchanged) the lower bound on the marginal likelihood. (Note that the exact log marginal likelihood is a *fixed* quantity, and does not change with VBE or VBM steps — it is only the lower bound which increases.)

Proceeding as above, take functional derivatives of $\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}))$ with respect to $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and equate to zero yielding:

$$\frac{\partial}{\partial q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \frac{\partial}{\partial q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \left[\int d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x})\; \ln p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}, m) \right. \tag{2.64}$$

$$\left. + \ln \frac{p(\boldsymbol{\theta} \mid m)}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \right] \tag{2.65}$$

$$= \int d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x})\; \ln p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta} \mid m) - \ln q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + c' \tag{2.66}$$

$$= 0\;, \tag{2.67}$$

which upon rearrangement produces

$$\ln q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) = \ln p(\boldsymbol{\theta} \mid m) + \int d\mathbf{x}\; q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) - \ln \mathcal{Z}_{\boldsymbol{\theta}}^{(t+1)}\;, \tag{2.68}$$

where $\mathcal{Z}_{\boldsymbol{\theta}}$ is the normalisation constant (related to the Lagrange multiplier which has again been omitted for succinctness). Thus for a given $q_{\mathbf{x}}(\mathbf{x})$, there is a unique stationary point for $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$.

□

At this point it is well worth noting the symmetry between the hidden variables and the parameters. The individual VBE steps can be written as one batch VBE step:

$$q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathbf{x}}} \exp\left[\int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta})\; \ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)\right] \tag{2.69}$$

$$\text{with} \qquad \mathcal{Z}_{\mathbf{x}} = \prod_{i=1}^{n} \mathcal{Z}_{\mathbf{x}_i} \; . \tag{2.70}$$

On the surface, it seems that the variational update rules (2.60) and (2.56) differ only in the prior term $p(\boldsymbol{\theta} \,|\, m)$ over the parameters. There actually also exists a prior term over the hidden variables as part of $p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)$, so this does not resolve the two. The distinguishing feature between hidden variables and parameters is that the number of hidden variables increases with data set size, whereas the number of parameters is assumed fixed.

Re-writing (2.53), it is easy to see that maximising $\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is simply equivalent to minimising the KL divergence between $q_{\mathbf{x}}(\mathbf{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and the joint posterior over hidden states and parameters $p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)$:

$$\ln p(\mathbf{y} \,|\, m) - \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \int d\boldsymbol{\theta}\, d\mathbf{x}\; q_{\mathbf{x}}(\mathbf{x})\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\; \ln \frac{q_{\mathbf{x}}(\mathbf{x})\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)} \tag{2.71}$$

$$= \mathrm{KL}\left[q_{\mathbf{x}}(\mathbf{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \,\|\, p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, m)\right] \tag{2.72}$$

$$\geq 0 \; . \tag{2.73}$$

Note the similarity between expressions (2.35) and (2.72): while we minimise the former with respect to hidden variable distributions and the parameters, the latter we minimise with respect to the hidden variable distribution and a *distribution* over parameters.

The variational Bayesian EM algorithm reduces to the ordinary EM algorithm for ML estimation if we restrict the parameter distribution to a point estimate, i.e. a Dirac delta function, $q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, in which case the M step simply involves re-estimating $\boldsymbol{\theta}^*$. Note that the same cannot be said in the case of MAP estimation, which is inherently basis dependent, unlike both VB and ML algorithms. By construction, the VBEM algorithm is guaranteed to monotonically increase an objective function $\mathcal{F}$, as a function of a distribution over parameters and hidden variables. Since we integrate over model parameters there is a naturally incorporated model complexity penalty. It turns out that for a large class of models (see section 2.4) the VBE step has approximately the same computational complexity as the standard E step in the ML framework, which makes it viable as a Bayesian replacement for the EM algorithm.

2.3.2 Discussion

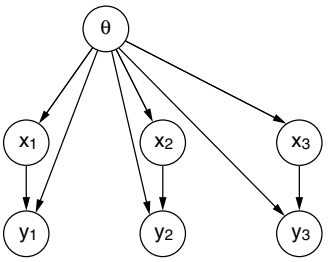
The impact of the $q(\mathbf{x}, \boldsymbol{\theta}) \approx q_{\mathbf{x}}(\mathbf{x})q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ factorisation

Unless we make the assumption that the posterior over parameters and hidden variables factorises, we will not generally obtain the further hidden variable factorisation over n that we have in equation (2.55). In that case, the distributions of $\mathbf{x}_i$ and $\mathbf{x}_j$ will be coupled for all cases $\{i, j\}$ in the data set, greatly increasing the overall computational complexity of inference. This further factorisation is depicted in figure 2.4 for the case of $n = 3$, where we see: (a) the original directed graphical model, where $\boldsymbol{\theta}$ is the collection of parameters governing prior distributions over the hidden variables $\mathbf{x}_i$ and the conditional probability $p(\mathbf{y}_i \,|\, \mathbf{x}_i, \boldsymbol{\theta})$; (b) the moralised graph given the data $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$, which shows that the hidden variables are now dependent in the posterior through the uncertain parameters; (c) the effective graph after the factorisation assumption, which not only removes arcs between the parameters and hidden variables, but also removes the dependencies between the hidden variables. This latter independence falls out from the optimisation as a result of the i.i.d. nature of the data, and is not a further approximation.

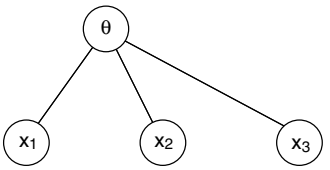
Whilst this factorisation of the posterior distribution over hidden variables and parameters may seem drastic, one can think of it as replacing *stochastic* dependencies between $\mathbf{x}$ and $\boldsymbol{\theta}$ with *deterministic* dependencies between relevant moments of the two sets of variables. The advantage of ignoring how fluctuations in $\mathbf{x}$ induce fluctuations in $\boldsymbol{\theta}$ (and vice-versa) is that we can obtain analytical approximations to the log marginal likelihood. It is these same ideas that underlie mean-field approximations from statistical physics, from where these lower-bounding variational approximations were inspired (Feynman, 1972; Parisi, 1988). In later chapters the consequences of the factorisation for particular models are studied in some detail; in particular we will use sampling methods to estimate by how much the variational bound falls short of the marginal likelihood.

What forms for $q_{\mathbf{x}}(\mathbf{x})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$?

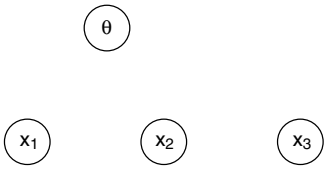
One might need to approximate the posterior further than simply the hidden-variable / parameter factorisation. A common reason for this is that the parameter posterior may still be intractable despite the hidden-variable / parameter factorisation. The free-form extremisation of $\mathcal{F}$ normally provides us with a functional form for $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, but this may be unwieldy; we therefore need to assume some simpler space of parameter posteriors. The most commonly used distributions are those with just a few sufficient statistics, such as the Gaussian or Dirichlet distributions. Taking a Gaussian example, $\mathcal{F}$ is then explicitly extremised with respect to a set of variational parameters $\boldsymbol{\zeta}_{\boldsymbol{\theta}} = (\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\nu}_{\boldsymbol{\theta}})$ which parameterise the Gaussian $q_{\boldsymbol{\theta}}(\boldsymbol{\theta} \,|\, \boldsymbol{\zeta}_{\boldsymbol{\theta}})$. We will see examples of this approach in later chapters. There may also exist intractabilities in the hidden variable



(a) The generative graphical model.



(b) Graph representing the exact posterior.



(c) Posterior graph after the variational approximation.

Figure 2.4: Graphical depiction of the hidden-variable / parameter factorisation. **(a)** The original generative model for $n = 3$. **(b)** The exact posterior graph given the data. Note that for all case pairs $\{i, j\}$, $\mathbf{x}_i$ and $\mathbf{x}_j$ are not directly coupled, but interact through $\boldsymbol{\theta}$. That is to say all the hidden variables are conditionally independent of one another, but only given the parameters. **(c)** the posterior graph after the variational approximation between parameters and hidden variables, which removes arcs between parameters and hidden variables. Note that, on assuming this factorisation, as a consequence of the i.i.d. assumption the hidden variables become independent.

posterior, for which further approximations need be made (some examples are mentioned below).

There is something of a dark art in discovering a factorisation amongst the hidden variables and parameters such that the approximation remains faithful at an 'acceptable' level. Of course it does not make sense to use a posterior form which holds fewer conditional independencies than those implied by the *moral* graph (see section 1.1). The key to a good variational approximation is then to remove as few arcs as possible from the moral graph such that inference becomes tractable. In many cases the goal is to find tractable substructures (*structured* approximations) such as trees or mixtures of trees, which capture as many of the arcs as possible. Some arcs may capture crucial dependencies between nodes and so need be kept, whereas other arcs might induce a weak local correlation at the expense of a long-range correlation which to first order can be ignored; removing such an arc can have dramatic effects on the tractability.

The advantage of the variational Bayesian procedure is that *any* factorisation of the posterior yields a lower bound on the marginal likelihood. Thus in practice it may pay to approximately evaluate the computational cost of several candidate factorisations, and implement those which can return a completed optimisation of $\mathcal{F}$ within a certain amount of computer time. One would expect the more complex factorisations to take more computer time but also yield progressively tighter lower bounds on average, the consequence being that the marginal likelihood estimate improves over time. An interesting avenue of research in this vein would be to use the variational posterior resulting from a simpler factorisation as the initialisation for a slightly more complicated factorisation, and move in a chain from simple to complicated factorisations to help avoid local free energy minima in the optimisation. Having proposed this, it remains to be seen if it is possible to form a coherent closely-spaced chain of distributions that are of any use, as compared to starting from the fullest posterior approximation from the start.

Using the lower bound for model selection and averaging

The log ratio of posterior probabilities of two competing models m and m' is given by

$$\ln \frac{p(m \,|\, \mathbf{y})}{p(m' \,|\, \mathbf{y})} = +\ln p(m) + p(\mathbf{y} \,|\, m) - \ln p(m') - \ln p(\mathbf{y} \,|\, m') \tag{2.74}$$

$$\begin{aligned} &= +\ln p(m) \;+\mathcal{F}(q_{\mathbf{x},\boldsymbol{\theta}}) \;\; + \mathrm{KL}\left[q(\mathbf{x},\boldsymbol{\theta}) \,\|\, p(\mathbf{x},\boldsymbol{\theta} \,|\, \mathbf{y}, m)\right] \\ &\quad - \ln p(m') - \mathcal{F}'(q'_{\mathbf{x},\boldsymbol{\theta}}) - \mathrm{KL}\left[q'(\mathbf{x},\boldsymbol{\theta}) \,\|\, p(\mathbf{x},\boldsymbol{\theta} \,|\, \mathbf{y}, m')\right] \end{aligned} \tag{2.75}$$

where we have used the form in (2.72), which is exact regardless of the quality of the bound used, or how tightly that bound has been optimised. The lower bounds for the two models, $\mathcal{F}$ and $\mathcal{F}'$, are calculated from VBEM optimisations, providing us for each model with an approximation to the posterior over the hidden variables and parameters of that model, $q_{\mathbf{x},\boldsymbol{\theta}}$ and $q'_{\mathbf{x},\boldsymbol{\theta}}$; these may in general be functionally very different (we leave aside for the moment local maxima problems

in the optimisation process which can be overcome to an extent by using several differently initialised optimisations or in some models by employing heuristics tailored to exploit the model structure). When we perform model selection by comparing the lower bounds, $\mathcal{F}$ and $\mathcal{F}'$, we are assuming that the KL divergences in the two approximations are the same, so that we can use just these lower bounds as guide. Unfortunately it is non-trivial to predict how tight in theory any particular bound can be — if this were possible we could more accurately estimate the marginal likelihood from the start.

Taking an example, we would like to know whether the bound for a model with S mixture components is similar to that for $S+1$ components, and if not then how badly this inconsistency affects the posterior over this set of models. Roughly speaking, let us assume that every component in our model contributes a (constant) KL divergence penalty of KL_s. For clarity we use the notation $\mathcal{L}(S)$ and $\mathcal{F}(S)$ to denote the exact log marginal likelihood and lower bounds, respectively, for a model with S components. The difference in log marginal likelihoods, $\mathcal{L}(S+1) - \mathcal{L}(S)$, is the quantity we wish to estimate, but if we base this on the lower bounds the difference becomes

$$\mathcal{L}(S+1) - \mathcal{L}(S) = [\mathcal{F}(S+1) + (S+1)\,\mathrm{KL}_s] - [\mathcal{F}(S) + S\,\mathrm{KL}_s] \tag{2.76}$$

$$= \mathcal{F}(S+1) - \mathcal{F}(S) + \mathrm{KL}_s \tag{2.77}$$

$$\neq \mathcal{F}(S+1) - \mathcal{F}(S)\ , \tag{2.78}$$

where the last line is the result we would have basing the difference on lower bounds. Therefore there exists a systematic error when comparing models if each component contributes independently to the KL divergence term. Since the KL divergence is strictly positive, and we are basing our model selection on (2.78) rather than (2.77), this analysis suggests that there is a systematic bias towards simpler models. We will in fact see this in chapter 4, where we find an importance sampling estimate of the KL divergence showing this behaviour.

Optimising the prior distributions

Usually the parameter priors are functions of hyperparameters, $\mathbf{a}$, so we can write $p(\boldsymbol{\theta} \,|\, \mathbf{a}, m)$. In the variational Bayesian framework the lower bound can be made higher by maximising $\mathcal{F}_m$ with respect to these hyperparameters:

$$\mathbf{a}^{(t+1)} = \arg\max_{\mathbf{a}}\ \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \mathbf{y}, \mathbf{a})\ . \tag{2.79}$$

A simple depiction of this optimisation is given in figure 2.5. Unlike earlier in section 2.3.1, the marginal likelihood of model m can now be increased with hyperparameter optimisation. As we will see in later chapters, there are examples where these hyperparameters themselves have governing hyperpriors, such that they can be integrated over as well. The result being that

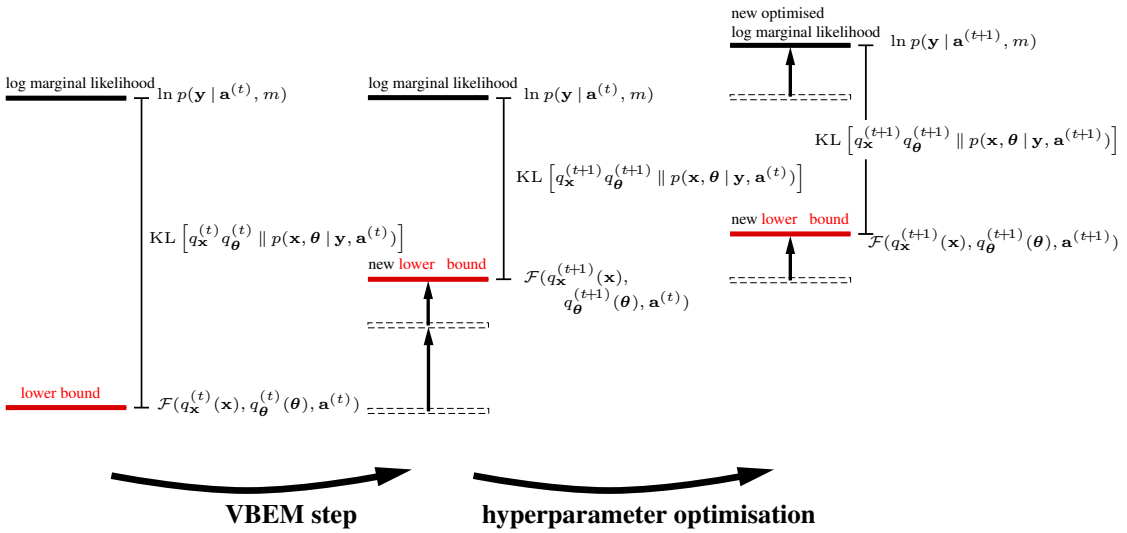


Figure 2.5: The variational Bayesian EM algorithm with hyperparameter optimisation. The VBEM step consists of VBE and VBM steps, as shown in figure 2.3. The hyperparameter optimisation increases the lower bound and also improves the marginal likelihood.

we can infer distributions over these as well, just as for parameters. The reason for abstracting from the parameters this far is that we would like to integrate out all variables whose cardinality increases with model complexity; this standpoint will be made clearer in the following chapters.

Previous work, and general applicability of VBEM

The variational approach for lower bounding the marginal likelihood (and similar quantities) has been explored by several researchers in the past decade, and has received a lot of attention recently in the machine learning community. It was first proposed for one-hidden layer neural networks (which have no hidden variables) by Hinton and van Camp (1993) where $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ was restricted to be Gaussian with diagonal covariance. This work was later extended to show that tractable approximations were also possible with a full covariance Gaussian (Barber and Bishop, 1998) (which in general will have the mode of the posterior at a different location than in the diagonal case). Neal and Hinton (1998) presented a generalisation of EM which made use of Jensen's inequality to allow partial E-steps; in this paper the term *ensemble learning* was used to describe the method since it fits an ensemble of models, each with its own parameters. Jaakkola (1997) and Jordan et al. (1999) review variational methods in a general context (i.e. non-Bayesian). Variational Bayesian methods have been applied to various models with hidden variables and no restrictions on $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $q_{\mathbf{x}_i}(\mathbf{x}_i)$ other than the assumption that they factorise in some way (Waterhouse et al., 1996; Bishop, 1999; Ghahramani and Beal, 2000; Attias, 2000). Of particular note is the variational Bayesian HMM of MacKay (1997), in which free-form optimisations are explicitly undertaken (see chapter 3); this work was the inspiration for the examination of Conjugate-Exponential (CE) models, discussed in the next section. An example

of a constrained optimisation for a logistic regression model can be found in Jaakkola and Jordan (2000).

Several researchers have investigated using mixture distributions for the approximate posterior, which allows for more flexibility whilst maintaining a degree of tractability (Lawrence et al., 1998; Bishop et al., 1998; Lawrence and Azzouzi, 1999). The lower bound in these models is a sum of a two terms: a first term which is a convex combination of bounds from each mixture component, and a second term which is the mutual information between the mixture labels and the hidden variables of the model. The first term offers no improvement over a naive combination of bounds, but the second (which is non-negative) has to improve on the simple bounds. Unfortunately this term contains an expectation over all configurations of the hidden states and so has to be itself bounded with a further use of Jensen's inequality in the form of a convex bound on the log function ($\ln(x) \leq \lambda x - \ln(\lambda) - 1$) (Jaakkola and Jordan, 1998). Despite this approximation drawback, empirical results in a handful of models have shown that the approximation does improve the simple mean field bound and improves monotonically with the number of mixture components.

A related method for approximating the integrand for Bayesian learning is based on an idea known as *assumed density filtering* (ADF) (Bernardo and Giron, 1988; Stephens, 1997; Boyen and Koller, 1998; Barber and Sollich, 2000; Frey et al., 2001), and is called the Expectation Propagation (EP) algorithm (Minka, 2001a). This algorithm approximates the integrand of interest with a set of *terms*, and through a process of repeated deletion-inclusion of term expressions, the integrand is iteratively refined to resemble the true integrand as closely as possible. Therefore the key to the method is to use terms which can be tractably integrated. This has the same flavour as the variational Bayesian method described here, where we iteratively update the approximate posterior over a hidden state $q_{\mathbf{x}_i}(\mathbf{x}_i)$ or over the parameters $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$. The key difference between EP and VB is that in the update process (i.e. deletion-inclusion) EP seeks to minimise the KL divergence which averages according to the true distribution, $\mathrm{KL}\left[p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}) \,\|\, q(\mathbf{x}, \boldsymbol{\theta})\right]$ (which is simply a moment-matching operation for exponential family models), whereas VB seeks to minimise the KL divergence according to the approximate distribution, $\mathrm{KL}\left[q(\mathbf{x}, \boldsymbol{\theta}) \,\|\, p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y})\right]$. Therefore, EP is at least attempting to average according to the correct distribution, whereas VB has the wrong cost function at heart. However, in general the KL divergence in EP can only be minimised separately one term at a time, while the KL divergence in VB is minimised globally over all terms in the approximation. The result is that EP may still not result in representative posterior distributions (for example, see Minka, 2001a, figure 3.6, p. 6). Having said that, it may be that more generalised deletion-inclusion steps can be derived for EP, for example removing two or more terms at a time from the integrand, and this may alleviate some of the 'local' restrictions of the EP algorithm. As in VB, EP is constrained to use particular parametric families with a small number of moments for tractability. An example of EP used with an assumed Dirichlet density for the term expressions can be found in Minka and Lafferty (2002).

In the next section we take a closer look at the variational Bayesian EM equations, (2.54) and (2.56), and ask the following questions:

- To which models can we apply VBEM? i.e. which forms of data distributions $p(\mathbf{y}, \mathbf{x} \,|\, \boldsymbol{\theta})$ and priors $p(\boldsymbol{\theta} \,|\, m)$ result in tractable VBEM updates?
- How does this relate formally to conventional EM?
- When can we utilise existing belief propagation algorithms in the VB framework?

2.4 Conjugate-Exponential models

2.4.1 Definition

We consider a particular class of graphical models with latent variables, which we call *conjugate-exponential* (CE) models. In this section we explicitly apply the variational Bayesian method to these parametric families, deriving a simple general form of VBEM for the class.

Conjugate-exponential models satisfy two conditions:

Condition (1). *The complete-data likelihood is in the exponential family:*

$$p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) = g(\boldsymbol{\theta})\, f(\mathbf{x}_i, \mathbf{y}_i)\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)}\,, \tag{2.80}$$

where $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the vector of natural parameters, $\mathbf{u}$ and f are the functions that define the exponential family, and g is a normalisation constant:

$$g(\boldsymbol{\theta})^{-1} = \int d\mathbf{x}_i\, d\mathbf{y}_i\; f(\mathbf{x}_i, \mathbf{y}_i)\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)}\,. \tag{2.81}$$

The natural parameters for an exponential family model $\boldsymbol{\phi}$ are those that interact linearly with the sufficient statistics of the data $\mathbf{u}$. For example, for a univariate Gaussian in x with mean μ and standard deviation σ, the necessary quantities are obtained from:

$$p(x \,|\, \mu, \sigma) = \exp\left\{ -\frac{x^2}{2\sigma^2} + \frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2) \right\} \tag{2.82}$$

$$\boldsymbol{\theta} = \left(\sigma^2, \mu\right) \tag{2.83}$$

and are:

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = \left(\frac{1}{\sigma^2}, \frac{\mu}{\sigma^2}\right) \tag{2.84}$$

$$\mathbf{u}(x) = \left(-\frac{x^2}{2}, x\right) \tag{2.85}$$

$$f(x) = 1 \tag{2.86}$$

$$g(\boldsymbol{\theta}) = \exp\left\{-\frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right\} . \tag{2.87}$$

Note that whilst the parameterisation for $\boldsymbol{\theta}$ is arbitrary, e.g. we could have let $\boldsymbol{\theta} = (\sigma, \mu)$, the natural parameters $\boldsymbol{\phi}$ are unique up to a multiplicative constant.

Condition (2). *The parameter prior is conjugate to the complete-data likelihood:*

$$p(\boldsymbol{\theta} \,|\, \eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})\, g(\boldsymbol{\theta})^{\eta}\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top}\boldsymbol{\nu}} , \tag{2.88}$$

where η and $\boldsymbol{\nu}$ are hyperparameters of the prior, and h is a normalisation constant:

$$h(\eta, \boldsymbol{\nu})^{-1} = \int d\boldsymbol{\theta}\; g(\boldsymbol{\theta})^{\eta}\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top}\boldsymbol{\nu}} . \tag{2.89}$$

Condition 1 (2.80) in fact usually implies the existence of a conjugate prior which satisfies condition 2 (2.88). The prior $p(\boldsymbol{\theta} \,|\, \eta, \boldsymbol{\nu})$ is said to be conjugate to the likelihood $p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta})$ if and only if the posterior

$$p(\boldsymbol{\theta} \,|\, \eta', \boldsymbol{\nu}') \propto p(\boldsymbol{\theta} \,|\, \eta, \boldsymbol{\nu}) p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}) \tag{2.90}$$

is of the same parametric form as the prior. In general the exponential families are the only classes of distributions that have natural conjugate prior distributions because they are the only distributions with a fixed number of sufficient statistics apart from some irregular cases (see Gelman et al., 1995, p. 38). From the definition of conjugacy, we see that the hyperparameters of a conjugate prior can be interpreted as the number (η) and values ($\boldsymbol{\nu}$) of pseudo-observations under the corresponding likelihood.

We call models that satisfy conditions 1 (2.80) and 2 (2.88) *conjugate-exponential*.

The list of latent-variable models of practical interest with complete-data likelihoods in the exponential family is very long, for example: Gaussian mixtures, factor analysis, principal components analysis, hidden Markov models and extensions, switching state-space models, discrete-variable belief networks. Of course there are also many as yet undreamt-of models combining Gaussian, gamma, Poisson, Dirichlet, Wishart, multinomial, and other distributions in the exponential family.

However there are some notable outcasts which do not satisfy the conditions for membership of the CE family, namely: Boltzmann machines (Ackley et al., 1985), logistic regression and sigmoid belief networks (Bishop, 1995), and independent components analysis (ICA) (as presented in Comon, 1994; Bell and Sejnowski, 1995), all of which are widely used in the machine learning community. As an example let us see why logistic regression is not in the conjugate-exponential family: for $y_i \in \{-1, 1\}$, the likelihood under a logistic regression model is

$$p(y_i \,|\, \mathbf{x}_i, \boldsymbol{\theta}) = \frac{e^{y_i \boldsymbol{\theta}^\top \mathbf{x}_i}}{e^{\boldsymbol{\theta}^\top \mathbf{x}_i} + e^{-\boldsymbol{\theta}^\top \mathbf{x}_i}} \,, \tag{2.91}$$

where $\mathbf{x}_i$ is the regressor for data point i and $\boldsymbol{\theta}$ is a vector of weights, potentially including a bias. This can be rewritten as

$$p(y_i \,|\, \mathbf{x}_i, \boldsymbol{\theta}) = e^{y_i \boldsymbol{\theta}^\top \mathbf{x}_i - f(\boldsymbol{\theta}, \mathbf{x}_i)} \,, \tag{2.92}$$

where $f(\boldsymbol{\theta}, \mathbf{x}_i)$ is a normalisation constant. To belong in the exponential family the normalising constant must split into functions of only $\boldsymbol{\theta}$ and only $(\mathbf{x}_i, \mathbf{y}_i)$. Expanding $f(\boldsymbol{\theta}, \mathbf{x}_i)$ yields a series of powers of $\boldsymbol{\theta}^\top \mathbf{x}_i$, which *could* be assimilated into the $\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)$ term by augmenting the natural parameter and sufficient statistics vectors, if it were not for the fact that the series is infinite meaning that there would need to be an infinity of natural parameters. This means we cannot represent the likelihood with a finite number of sufficient statistics.

Models whose complete-data likelihood is not in the exponential family can often be approximated by models which are in the exponential family and have been given additional hidden variables. A very good example is the Independent Factor Analysis (IFA) model of Attias (1999a). In conventional ICA, one can think of the model as using non-Gaussian sources, or using Gaussian sources passed through a non-linearity to make them non-Gaussian. For most non-linearities commonly used (such as the logistic), the complete-data likelihood becomes non-CE. Attias recasts the model as a mixture of Gaussian sources being fed into a linear mixing matrix. This model is in the CE family and so can be tackled with the VB treatment. It is an open area of research to investigate how best to bring models into the CE family, such that inferences in the modified model resemble the original as closely as possible.

2.4.2 Variational Bayesian EM for CE models

In Bayesian inference we want to determine the posterior over parameters and hidden variables $p(\mathbf{x}, \boldsymbol{\theta} \,|\, \mathbf{y}, \eta, \boldsymbol{\nu})$. In general this posterior is *neither* conjugate nor in the exponential family. In this subsection we see how the properties of the CE family make it especially amenable to the VB approximation, and derive the VBEM algorithm for CE models.

Theorem 2.2: Variational Bayesian EM for Conjugate-Exponential Models.
Given an i.i.d. data set $\mathbf{y} = \{\mathbf{y}_1, \ldots \mathbf{y}_n\}$*, if the model satisfies conditions (1) and (2), then the following (a), (b) and (c) hold:*

(a) the VBE step yields:

$$q_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{n} q_{\mathbf{x}_i}(\mathbf{x}_i) \,, \tag{2.93}$$

and $q_{\mathbf{x}_i}(\mathbf{x}_i)$ *is in the exponential family:*

$$q_{\mathbf{x}_i}(\mathbf{x}_i) \propto f(\mathbf{x}_i, \mathbf{y}_i)\, e^{\overline{\boldsymbol{\phi}}^{\top} \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)} = p(\mathbf{x}_i \,|\, \mathbf{y}_i, \overline{\boldsymbol{\phi}}) \,, \tag{2.94}$$

with a natural parameter vector

$$\overline{\boldsymbol{\phi}} = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \boldsymbol{\phi}(\boldsymbol{\theta}) \equiv \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \tag{2.95}$$

obtained by taking the expectation of $\boldsymbol{\phi}(\boldsymbol{\theta})$ *under* $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ *(denoted using angle-brackets* $\langle \cdot \rangle$*). For invertible* $\boldsymbol{\phi}$*, defining* $\tilde{\boldsymbol{\theta}}$ *such that* $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}$*, we can rewrite the approximate posterior as*

$$q_{\mathbf{x}_i}(\mathbf{x}_i) = p(\mathbf{x}_i \,|\, \mathbf{y}_i, \tilde{\boldsymbol{\theta}}) \,. \tag{2.96}$$

(b) the VBM step yields that $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ *is conjugate and of the form:*

$$q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = h(\tilde{\eta}, \tilde{\boldsymbol{\nu}})\, g(\boldsymbol{\theta})^{\tilde{\eta}}\, e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \tilde{\boldsymbol{\nu}}} \,, \tag{2.97}$$

where

$$\tilde{\eta} = \eta + n \,, \tag{2.98}$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + \sum_{i=1}^{n} \overline{\mathbf{u}}(\mathbf{y}_i) \,, \tag{2.99}$$

and

$$\overline{\mathbf{u}}(\mathbf{y}_i) = \langle \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i) \rangle_{q_{\mathbf{x}_i}(\mathbf{x}_i)} \tag{2.100}$$

is the expectation of the sufficient statistic $\mathbf{u}$*. We have used* $\langle \cdot \rangle_{q_{\mathbf{x}_i}(\mathbf{x}_i)}$ *to denote expectation under the variational posterior over the latent variable(s) associated with the* i*th datum.*

(c) parts (a) and (b) hold for every iteration of variational Bayesian EM.

Proof of (a): by direct substitution.

Starting from the variational extrema solution (2.60) for the VBE step:

$$q_{\mathbf{x}}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathbf{x}}} e^{\langle \ln p(\mathbf{x},\mathbf{y} \,|\, \boldsymbol{\theta}, m) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}} , \tag{2.101}$$

substitute the parametric form for $p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m)$ in condition 1 (2.80), which yields (omitting iteration superscripts):

$$q_{\mathbf{x}}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathbf{x}}} e^{\sum_{i=1}^{n} \left\langle \ln g(\boldsymbol{\theta}) + \ln f(\mathbf{x}_i,\mathbf{y}_i) + \boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i,\mathbf{y}_i) \right\rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}} \tag{2.102}$$

$$= \frac{1}{\mathcal{Z}_{\mathbf{x}}} \left[\prod_{i=1}^{n} f(\mathbf{x}_i, \mathbf{y}_i) \right] e^{\sum_{i=1}^{n} \overline{\boldsymbol{\phi}}^\top u(\mathbf{x}_i,\mathbf{y}_i)} , \tag{2.103}$$

where $\mathcal{Z}_{\mathbf{x}}$ has absorbed constants independent of $\mathbf{x}$, and we have defined without loss of generality:

$$\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} . \tag{2.104}$$

If $\boldsymbol{\phi}$ is invertible, then there exists a $\tilde{\boldsymbol{\theta}}$ such that $\overline{\boldsymbol{\phi}} = \boldsymbol{\phi}(\tilde{\boldsymbol{\theta}})$, and we can rewrite (2.103) as:

$$q_{\mathbf{x}}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathbf{x}}} \left[\prod_{i=1}^{n} f(\mathbf{x}_i, \mathbf{y}_i) e^{\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}})^\top \mathbf{u}(\mathbf{x}_i,\mathbf{y}_i)} \right] \tag{2.105}$$

$$\propto \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{y}_i \,|\, \tilde{\boldsymbol{\theta}}, m) \tag{2.106}$$

$$= \prod_{i=1}^{n} q_{\mathbf{x}_i}(\mathbf{x}_i) \tag{2.107}$$

$$= p(\mathbf{x}, \mathbf{y} \,|\, \tilde{\boldsymbol{\theta}}, m) . \tag{2.108}$$

Thus the result of the approximate VBE step, which averages over the ensemble of models $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, is exactly the same as an exact E step, calculated at the *variational Bayes point* estimate $\tilde{\boldsymbol{\theta}}$. □

Proof of (b): by direct substitution.

Starting from the variational extrema solution (2.56) for the VBM step:

$$q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}} p(\boldsymbol{\theta} \,|\, m) \, e^{\langle \ln p(\mathbf{x},\mathbf{y} \,|\, \boldsymbol{\theta}, m) \rangle_{q_{\mathbf{x}}(\mathbf{x})}} , \tag{2.109}$$

substitute the parametric forms for $p(\boldsymbol{\theta} \,|\, m)$ and $p(\mathbf{x}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m)$ as specified in conditions 2 (2.88) and 1 (2.80) respectively, which yields (omitting iteration superscripts):

$$q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}} h(\eta, \boldsymbol{\nu}) g(\boldsymbol{\theta})^{\eta} e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \boldsymbol{\nu}} \, e^{\left\langle \sum_{i=1}^{n} \ln g(\boldsymbol{\theta}) + \ln f(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i) \right\rangle_{q_{\mathbf{x}}(\mathbf{x})}} \tag{2.110}$$

$$= \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}} h(\eta, \boldsymbol{\nu}) g(\boldsymbol{\theta})^{\eta+n} e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \left[\boldsymbol{\nu} + \sum_{i=1}^{n} \overline{\mathbf{u}}(\mathbf{y}_i) \right]} \underbrace{e^{\sum_{i=1}^{n} \langle \ln f(\mathbf{x}_i, \mathbf{y}_i) \rangle_{q_{\mathbf{x}}(\mathbf{x})}}}_{\text{has no } \boldsymbol{\theta} \text{ dependence}} \tag{2.111}$$

$$= h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{\theta})^{\tilde{\eta}} e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \tilde{\boldsymbol{\nu}}} \,, \tag{2.112}$$

where

$$h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\theta}}} e^{\sum_{i=1}^{n} \langle \ln f(\mathbf{x}_i, \mathbf{y}_i) \rangle_{q_{\mathbf{x}}(\mathbf{x})}} \,. \tag{2.113}$$

Therefore the variational posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ in (2.112) is of conjugate form, according to condition 2 (2.88). □

Proof of (c): by induction.

Assume conditions 1 (2.80) and 2 (2.88) are met (i.e. the model is in the CE family). From part (a), the VBE step produces a posterior distribution $q_{\mathbf{x}}(\mathbf{x})$ in the exponential family, preserving condition 1 (2.80); the parameter distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ remains unaltered, preserving condition 2 (2.88). From part (b), the VBM step produces a parameter posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ that is of conjugate form, preserving condition 2 (2.88); $q_{\mathbf{x}}(\mathbf{x})$ remains unaltered from the VBE step, preserving condition 1 (2.80). Thus under both the VBE and VBM steps, conjugate-exponentiality is preserved, which makes the theorem applicable at every iteration of VBEM. □

As before, since $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $q_{\mathbf{x}_i}(\mathbf{x}_i)$ are coupled, (2.97) and (2.94) do not provide an analytic solution to the minimisation problem, so the optimisation problem is solved numerically by iterating between the fixed point equations given by these equations. To summarise briefly:

> **VBE Step**: *Compute the expected sufficient statistics* $\{\overline{\mathbf{u}}(\mathbf{y}_i)\}_{i=1}^{n}$ *under the hidden variable distributions* $q_{\mathbf{x}_i}(\mathbf{x}_i)$*, for all* i*.*
>
> **VBM Step**: *Compute the expected natural parameters* $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle$ *under the parameter distribution given by* $\tilde{\eta}$ *and* $\tilde{\boldsymbol{\nu}}$*.*

2.4.3 Implications

In order to really understand what the conjugate-exponential formalism buys us, let us reiterate the main points of theorem 2.2 above. The first result is that in the VBM step the analytical form of the variational posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ does not change during iterations of VBEM — e.g. if the posterior is Gaussian at iteration $t = 1$, then only a Gaussian need be represented at future iterations. If it were able to change, which is the case in general (theorem 2.1), the

EM for MAP estimation	**Variational Bayesian EM**
Goal: maximise $p(\boldsymbol{\theta} \mid \mathbf{y}, m)$ w.r.t. $\boldsymbol{\theta}$	**Goal:** lower bound $p(\mathbf{y} \mid m)$
E Step: compute $q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)})$	**VBE Step:** compute $q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y}, \overline{\boldsymbol{\phi}}^{(t)})$
M Step: $\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \int d\mathbf{x}\, q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$	**VBM Step:** $q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) \propto \exp \int d\mathbf{x}\, q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$

Table 2.1: Comparison of EM for ML/MAP estimation against variational Bayesian EM for CE models.

posterior could quickly become unmanageable, and (further) approximations would be required to prevent the algorithm becoming too complicated. The second result is that the posterior over hidden variables calculated in the VBE step is exactly the posterior that would be calculated had we been performing an ML/MAP E step. That is, the inferences using an ensemble of models $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ can be represented by the effect of a point parameter, $\tilde{\boldsymbol{\theta}}$. The task of performing many inferences, each of which corresponds to a different parameter setting, can be replaced with a single inference step — it is possible to infer the hidden states in a conjugate exponential model tractably while integrating over an ensemble of model parameters.

Comparison to EM for ML/MAP parameter estimation

We can draw a tight parallel between the EM algorithm for ML/MAP estimation, and our VBEM algorithm applied specifically to conjugate-exponential models. These are summarised in table 2.1. This general result of VBEM for CE models was reported in Ghahramani and Beal (2001), and generalises the well known EM algorithm for ML estimation (Dempster et al., 1977). It is a special case of the variational Bayesian algorithm (theorem 2.1) used in Ghahramani and Beal (2000) and in Attias (2000), yet encompasses many of the models that have been so far subjected to the variational treatment. Its particular usefulness is as a guide for the design of models, to make them amenable to efficient approximate Bayesian inference.

The VBE step has about the same time complexity as the E step, and is in all ways identical except that it is re-written in terms of the expected natural parameters. In particular, we can make use of all relevant propagation algorithms such as junction tree, Kalman smoothing, or belief propagation. The VBM step computes a *distribution* over parameters (in the conjugate family) rather than a point estimate. Both ML/MAP EM and VBEM algorithms monotonically increase an objective function, but the latter also incorporates a model complexity penalty by

integrating over parameters so embodying an Occam's razor effect. Several examples will be presented in the following chapters of this thesis.

Natural parameter inversions

Unfortunately, even though the algorithmic complexity is the same, the implementations may be hampered since the propagation algorithms need to be re-derived in terms of the natural parameters (this is essentially the difference between the forms in (2.94) and (2.96)). For some models, such as HMMs (see chapter 3, and MacKay, 1997), this is very straightforward, whereas the LDS model (see chapter 5) quickly becomes quite involved. Automated algorithm derivation programs are currently being written to alleviate this complication, specifically for the case of variational Bayesian EM operations (Bishop et al., 2003), and also for generic algorithm derivation (Buntine, 2002; Gray et al., 2003); both these projects build on results in Ghahramani and Beal (2001).

The difficulty is quite subtle and lies in the natural parameter inversion problem, which we now briefly explain. In theorem 2.2 we *conjectured* the existence of a $\tilde{\boldsymbol{\theta}}$ such that $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \stackrel{?}{=} \boldsymbol{\phi}(\tilde{\boldsymbol{\theta}})$, which was a point of convenience. But, the operation $\boldsymbol{\phi}^{-1}\left[\langle \boldsymbol{\phi} \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}\right]$ may not be well defined if the dimensionality of $\boldsymbol{\phi}$ is greater than that of $\boldsymbol{\theta}$. Whilst not undermining the theorem's result, this does mean that representationally speaking the resulting algorithm may look different having had to be cast in terms of the natural parameters.

Online and continuous variants

The VBEM algorithm for CE models very readily lends itself to online learning scenarios in which data arrives incrementally. I briefly present here an online version of the VBEM algorithm above (but see also Ghahramani and Attias, 2000; Sato, 2001). In the standard VBM step (2.97) the variational posterior hyperparameter $\tilde{\eta}$ is updated according to the size of the dataset n (2.98), and $\tilde{\boldsymbol{\nu}}$ is updated with a simple sum of contributions from each datum $\overline{\mathbf{u}}(\mathbf{y}_i)$, (2.99).

For the online scenario, we can take the posterior over parameters described by $\tilde{\eta}$ and $\tilde{\boldsymbol{\nu}}$ to be the *prior* for subsequent inferences. Let the data be split in to batches indexed by k, each of size $n^{(k)}$, which are presented one by one to the model. Thus if the kth batch of data consists of the

$n^{(k)}$ i.i.d. points $\{\mathbf{y}_i\}_{i=j^{(k)}}^{j^{(k)}+n^{(k)}-1}$, then the online VBM step replaces equations (2.98) and (2.99) with

$$\tilde{\eta} = \eta^{(k-1)} + n^{(k)} \,, \tag{2.114}$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}^{(k-1)} + \sum_{i=j^{(k)}}^{j^{(k)}+n^{(k)}-1} \overline{\mathbf{u}}(\mathbf{y}_i) \,. \tag{2.115}$$

In the online VBE step only the hidden variables $\{\mathbf{x}_i\}_{i=j^{(k)}}^{j^{(k)}+n^{(k)}-1}$ need be inferred to calculate the required $\overline{\mathbf{u}}$ statistics. The online VBM and VBE steps are then iterated until convergence, which may be fast if the size of the batch $n^{(k)}$ is small compared to the amount of data previously seen $\sum_{k'=1}^{k-1} n^{(k')}$. After convergence, the prior for the next batch is set to the current posterior, according to

$$\eta^{(k)} \leftarrow \tilde{\eta} \,, \tag{2.116}$$

$$\boldsymbol{\nu}^{(k)} \leftarrow \tilde{\boldsymbol{\nu}} \,. \tag{2.117}$$

The online VBEM algorithm has several benefits. First and foremost, the update equations give us a very transparent picture of how the algorithm incorporates evidence from a new batch of data (or single data point). The way in which it does this makes it possible to discard data from earlier batches: the hyperparameters $\tilde{\eta}$ and $\tilde{\boldsymbol{\nu}}$ represent *all* information gathered from previous batches, and the process of incorporating new information is not a function of the previous batches' statistics $\{\overline{\mathbf{u}}(\mathbf{y}_i)\}_{i=j^{(1)}}^{j^{(k-1)}+n^{(k-1)}-1}$, nor previous hyperparameter settings $\{\eta^{(l)}, \boldsymbol{\nu}^{(l)}\}_{l=1}^{k-2}$, nor the previous batch sizes $\{n^{(l)}\}_{l=1}^{k-1}$, nor the previous data $\{\mathbf{y}_i\}_{i=j^{(1)}}^{j^{(k-1)}+n^{(k-1)}-1}$. Implementationally this offers a large memory saving. Since we hold a distribution over the parameters of the model, which is updated in a consistent way using Bayesian inference, we should hope that the online model makes a flexible and measured response to data as it arrives. However it has been observed (personal communication, Z. Ghahramani) that serious underfitting occurs in this type of online algorithm; this is due to excessive self-pruning of the parameters by the VB algorithm.

From the VBM step (2.97) we can straightforwardly propose an annealing variant of the VBEM algorithm. This would make use of an inverse temperature parameter $\beta \in [0, 1]$ and adopt the following updates for the VBM step:

$$\tilde{\eta} = \eta + \beta n \,, \tag{2.118}$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + \beta \sum_{i=1}^{n} \overline{\mathbf{u}}(\mathbf{y}_i) \,, \tag{2.119}$$

which is similar to the online algorithm but "introduces" the data continuously with a schedule of β from $0 \rightarrow 1$. Whilst this is a tempting avenue for research, it is not clear that in this

setting we should expect any better results than if we were to present the algorithm with all the data (i.e. $\beta = 1$) from the start — after all, the procedure of Bayesian inference should produce the same inferences whether presented with the data incrementally, continuously or all at once. The advantage of an annealed model, however, is that we are giving the algorithm a better chance of escaping the local minima in the free energy that plague EM-type algorithms, so that the Bayesian inference procedure can be given a better chance of reaching the proper conclusions, whilst at every iteration receiving information (albeit β-muted) about all the data at every iteration.

2.5 Directed and undirected graphs

In this section we present several important results which build on theorems 2.1 and 2.2 by specifying the *form* of the joint density $p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$. A convenient way to do this is to use the formalism and expressive power of graphical models. We derive variational Bayesian learning algorithms for two important classes of these models: directed graphs (Bayesian networks) and undirected graphs (Markov networks), and also give results pertaining to CE families for these classes. The corollaries refer to propagation algorithms material which is covered in section 1.1.2; for a tutorial on belief networks and Markov networks the reader is referred to Pearl (1988). In the theorems and corollaries, VBEM and CE are abbreviations for *variational Bayesian Expectation-Maximisation* and *conjugate-exponential*.

2.5.1 Implications for directed networks

Corollary 2.1: (theorem 2.1) VBEM for Directed Graphs (Bayesian Networks).
Let m be a model with parameters $\boldsymbol{\theta}$ and hidden and visible variables $\mathbf{z} = \{\mathbf{z}_i\}_{i=1}^n = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ that satisfy a belief network factorisation. That is, each variable $\mathbf{z}_{ij}$ has parents $\mathbf{z}_{i\mathbf{pa}(j)}$ such that the complete-data joint density can be written as a product of conditional distributions,

$$p(\mathbf{z} \,|\, \boldsymbol{\theta}) = \prod_i \prod_j p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \,. \tag{2.120}$$

Then the approximating joint distribution for m satisfies the same belief network factorisation:

$$q_{\mathbf{z}}(\mathbf{z}) = \prod_i q_{\mathbf{z}_i}(\mathbf{z}_i) \,, \qquad q_{\mathbf{z}_i}(\mathbf{z}_i) = \prod_j \overline{q}_j(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}) \,, \tag{2.121}$$

where

$$\overline{q}_j(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}) = \frac{1}{\mathcal{Z}_{\overline{q}_j}} e^{\left\langle \ln p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \right\rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}} \quad \forall \, \{i, j\} \tag{2.122}$$

are new conditional distributions obtained by averaging over $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, and $\mathcal{Z}_{\overline{q}_j}$ are normalising constants.

This corollary is interesting in that it states that a Bayesian network's posterior distribution can be factored into the same terms as the original belief network factorisation (2.120). This means that the inference for a particular variable depends only on those other variables in its *Markov blanket*; this result is trivial for the point parameter case, but definitely non-trivial in the Bayesian framework in which all the parameters and hidden variables are potentially coupled.

Corollary 2.2: (theorem 2.2) VBEM for CE Directed Graphs (CE Bayesian Networks). *Furthermore, if m is a conjugate-exponential model, then the conditional distributions of the approximate posterior joint have exactly the same form as those in the complete-data likelihood in the original model:*

$$\overline{q}_j(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}) = p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \tilde{\boldsymbol{\theta}}) \;, \tag{2.123}$$

but with natural parameters $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}$. Moreover, with the modified parameters $\tilde{\boldsymbol{\theta}}$, the expectations under the approximating posterior $q_{\mathbf{x}}(\mathbf{x}) \propto q_{\mathbf{z}}(\mathbf{z})$ required for the VBE step can be obtained by applying the belief propagation algorithm if the network is singly connected and the junction tree algorithm if the network is multiply-connected.

This result generalises the derivation of variational learning for HMMs (MacKay, 1997), which uses the forward-backward algorithm as a subroutine. We investigate the variational Bayesian HMM in more detail in chapter 3. Another example is *dynamic trees* (Williams and Adams, 1999; Storkey, 2000; Adams et al., 2000) in which belief propagation is executed on a single tree which represents an ensemble of singly-connected structures. Again there exists the natural parameter inversion issue, but this is merely an implementational inconvenience.

2.5.2 Implications for undirected networks

Corollary 2.3: (theorem 2.1) VBEM for Undirected Graphs (Markov Networks). *Let m be a model with hidden and visible variables $\mathbf{z} = \{\mathbf{z}_i\}_{i=1}^n = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ that satisfy a Markov network factorisation. That is, the joint density can be written as a product of clique-potentials $\{\psi_j\}_{j=1}^J$,*

$$p(\mathbf{z} \,|\, \boldsymbol{\theta}) = \frac{1}{\mathcal{Z}} \prod_i \prod_j \psi_j(C_j(\mathbf{z}_i), \boldsymbol{\theta}) \;, \tag{2.124}$$

where each clique C_j is a (fixed) subset of the variables in $\mathbf{z}_i$, such that $\{C_1(\mathbf{z}_i) \cup \cdots \cup C_J(\mathbf{z}_i)\} = \mathbf{z}_i$. Then the approximating joint distribution for m satisfies the same Markov network factorisation:

$$q_{\mathbf{z}}(\mathbf{z}) = \prod_i q_{\mathbf{z}_i}(\mathbf{z}_i) \;, \qquad q_{\mathbf{z}_i}(\mathbf{z}_i) = \frac{1}{\mathcal{Z}_q} \prod_j \overline{\psi}_j(C_j(\mathbf{z}_i)) \;, \tag{2.125}$$

where

$$\overline{\psi}_j(C_j(\mathbf{z}_i)) = e^{\langle \ln \psi_j(C_j(\mathbf{z}_i),\boldsymbol{\theta})\rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}} \quad \forall \{i,j\} \tag{2.126}$$

are new clique potentials obtained by averaging over $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, and $\mathcal{Z}_q$ is a normalisation constant.

Corollary 2.4: (theorem 2.2) VBEM for CE Undirected Graphs (CE Markov Networks). *Furthermore, if m is a conjugate-exponential model, then the approximating clique potentials have exactly the same form as those in the original model:*

$$\overline{\psi}_j(C_j(\mathbf{z}_i)) \propto \psi_j(C_j(\mathbf{z}_i), \tilde{\boldsymbol{\theta}}) \,, \tag{2.127}$$

but with natural parameters $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}$. Moreover, the expectations under the approximating posterior $q_{\mathbf{x}}(\mathbf{x}) \propto q_{\mathbf{z}}(\mathbf{z})$ required for the VBE Step can be obtained by applying the junction tree algorithm.

For conjugate-exponential models in which belief propagation and the junction tree algorithm over hidden variables are intractable, further applications of Jensen's inequality can yield tractable factorisations (Jaakkola, 1997; Jordan et al., 1999).

2.6 Comparisons of VB to other criteria

2.6.1 BIC is recovered from VB in the limit of large data

We show here informally how the Bayesian Information Criterion (BIC, see section 1.3.4) is recovered in the large data limit of the variational Bayesian lower bound (Attias, 1999b). $\mathcal{F}$ can be written as a sum of two terms:

$$\mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \underbrace{-\mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \,|\, m)\right]}_{\mathcal{F}_{m,pen}} + \underbrace{\left\langle \ln \frac{p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m)}{q_{\mathbf{x}}(\mathbf{x})} \right\rangle_{q_{\mathbf{x}}(\mathbf{x})\, q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}}_{\mathcal{D}_m} . \tag{2.128}$$

Let us consider separately the limiting forms of these two terms, constraining ourselves to the cases in which the model m is in the CE family. In such cases, theorem 2.2 states that $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is of conjugate form (2.97) with parameters given by (2.98) and (2.99). It can be shown that under mild conditions exponential family distributions of this form exhibit asymptotic normality (see, for example, the proof given in Bernardo and Smith, 1994, pp. 293–4). Therefore, the entropy

of $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ appearing in $\mathcal{F}_{m,pen}$ can be calculated assuming a Gaussian form (see appendix A), and the limit becomes

$$\lim_{n\to\infty} \mathcal{F}_{m,pen} = \lim_{n\to\infty} \left[\langle \ln p(\boldsymbol{\theta} \,|\, m) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} + \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |H| \right] \tag{2.129}$$

$$= -\frac{d}{2} \ln n + \mathcal{O}(1) \,, \tag{2.130}$$

where H is the Hessian (matrix of second derivatives of the parameter posterior evaluated at the mode), and we have used similar arguments to those taken in the derivation of BIC (section 1.3.4). The second term, $\mathcal{D}_m$, can be analysed by appealing to the fact that the term inside the expectation is equal to $\ln p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)$ if and only if $q_{\mathbf{x}}(\mathbf{x}) = p(\mathbf{x} \,|\, \mathbf{y}, \boldsymbol{\theta}, m)$. Theorem 2.1 states that the form of the variational posterior over hidden states $q_{\mathbf{x}}(\mathbf{x})$ is given by

$$\ln q_{\mathbf{x}}(\mathbf{x}) = \int d\boldsymbol{\theta} \; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln p(\mathbf{x}, \mathbf{y} \,|\, \boldsymbol{\theta}, m) - \ln \mathcal{Z}_{\mathbf{x}} \tag{2.131}$$

(which does not depend on CE family membership conditions). Therefore as $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ becomes concentrated about $\boldsymbol{\theta}_{\text{MAP}}$, this results in $q_{\mathbf{x}}(\mathbf{x}) = p(\mathbf{x} \,|\, \mathbf{y}, \boldsymbol{\theta}_{\text{MAP}}, m)$. Then $\mathcal{D}_m$ asymptotically becomes $\ln p(\mathbf{y} \,|\, \boldsymbol{\theta}_{\text{MAP}}, m)$. Combining this with the limiting form for $\mathcal{F}_{m,pen}$ given by (2.130) results in:

$$\lim_{n\to\infty} \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = -\frac{d}{2} \ln n + \ln p(\mathbf{y} \,|\, \boldsymbol{\theta}_{\text{MAP}}, m) + \mathcal{O}(1) \,, \tag{2.132}$$

which is the BIC approximation given by (1.49). For the case of a non-CE model, we would have to prove asymptotic normality for $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ outside of the exponential family, which may become complicated or indeed impossible. We note that this derivation of the limiting form of VB is heuristic in the sense that we have neglected concerns on precise regularity conditions and identifiability.

2.6.2 Comparison to Cheeseman-Stutz (CS) approximation

In this section we present results regarding the approximation of Cheeseman and Stutz (1996), covered in section 1.3.5. We briefly review the CS criterion, as used to approximate the marginal likelihood of finite mixture models, and then show that it is in fact a strict lower bound on the marginal likelihood. We conclude the section by presenting a construction that proves that VB can be used to obtain a bound that is *always* tighter than CS.

Let m be a directed acyclic graph with parameters $\boldsymbol{\theta}$ giving rise to an i.i.d. data set denoted by $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ with corresponding discrete hidden variables $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ each of cardinality k. Let $\hat{\boldsymbol{\theta}}$ be a result of an EM algorithm which has converged to a local maximum in the likelihood $p(\mathbf{y} \,|\, \boldsymbol{\theta})$, and let $\hat{\mathbf{s}} = \{\hat{\mathbf{s}}_i\}_{i=1}^n$ be a completion of the hidden variables, chosen

according to the posterior distribution over hidden variables given the data and $\hat{\boldsymbol{\theta}}$, such that $\hat{\mathbf{s}}_{ij} = p(\mathbf{s}_{ij} = j \,|\, \mathbf{y}, \hat{\boldsymbol{\theta}}) \; \forall \; i = 1, \ldots, n$.

Since we are completing the hidden variables with real, as opposed to discrete values, this complete data set does not in general correspond to a realisable data set under the generative model. This point raises the question of how its marginal probability $p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m)$ is defined. We will see in the following theorem and proof (theorem 2.3) that both the completion required of the hidden variables and the completed data marginal probability are well-defined, and follow from equations 2.141 and 2.142 below.

The CS approximation is given by

$$p(\mathbf{y} \,|\, m) \approx p(\mathbf{y} \,|\, m)_{\mathrm{CS}} = p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) \frac{p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}})}{p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}})} \; . \tag{2.133}$$

The CS approximation exploits the fact that, for many models of interest, the first term on the right-hand side, the complete-data marginal likelihood, is tractable to compute (this is the case for discrete-variable directed acyclic graphs with Dirichlet priors, see chapter 6 for details). The term in the numerator of the second term on the right-hand side is simply the likelihood of the data, which is an output of the EM algorithm (as is the parameter estimate $\hat{\boldsymbol{\theta}}$), and the denominator is a straightforward calculation that involves no summations over hidden variables or integrations over parameters.

Theorem 2.3: Cheeseman-Stutz approximation is a lower bound on the marginal likelihood.
Let $\hat{\boldsymbol{\theta}}$ be the result of the M step of EM, and let $\{p(\mathbf{s}_i \,|\, \mathbf{y}_i, \hat{\boldsymbol{\theta}})\}_{i=1}^n$ be the set of posterior distributions over the hidden variables obtained in the next E step of EM. Furthermore, let $\hat{\mathbf{s}} = \{\hat{\mathbf{s}}_i\}_{i=1}^n$ be a completion of the hidden variables, such that $\hat{\mathbf{s}}_{ij} = p(\mathbf{s}_{ij} = j \,|\, \mathbf{y}, \hat{\boldsymbol{\theta}}) \; \forall \; i = 1, \ldots, n$. Then the CS approximation is a lower bound on the marginal likelihood:

$$p(\mathbf{y} \,|\, m)_{CS} = p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) \frac{p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}})}{p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}})} \leq p(\mathbf{y} \,|\, m) \; . \tag{2.134}$$

This observation should be attributed to Minka (2001b), where it was noted that (in the context of mixture models with unknown mixing proportions and component parameters) whilst the CS approximation has been reported to obtain good performance in the literature (Cheeseman and Stutz, 1996; Chickering and Heckerman, 1997), it was not known to be a bound on the marginal likelihood. Here we provide a proof of this statement that is generally applicable to any model.

Proof of theorem 2.3: via marginal likelihood bounds using approximations over the posterior distribution of only the hidden variables. The marginal likelihood can be lower bounded by introducing a distribution over the settings of each data point's hidden variables $q_{\mathbf{s}_i}(s_i)$:

$$p(\mathbf{y} \,|\, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) \tag{2.135}$$

$$\geq \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta})}{q_{\mathbf{s}_i}(\mathbf{s}_i)} \right\} . \tag{2.136}$$

We return to this quantity shortly, but presently place a similar lower bound over the likelihood of the data:

$$p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}}) \geq \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{s}_i, \mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}})}{q_{\mathbf{s}_i}(\mathbf{s}_i)} \right\} \tag{2.137}$$

which can be made an equality if, for each data point, $q(\mathbf{s}_i)$ is set to the exact posterior distribution given the parameter setting $\boldsymbol{\theta}$ (for example see equation (2.19) and the proof following it),

$$p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}}) = \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{s}_i, \mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}})}{\hat{q}_{\mathbf{s}_i}(\mathbf{s}_i)} \right\} , \tag{2.138}$$

where

$$\hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \equiv p(\mathbf{s}_i \,|\, \mathbf{y}, \hat{\boldsymbol{\theta}}) , \tag{2.139}$$

which is the result obtained from an exact E step with the parameters set to $\hat{\boldsymbol{\theta}}$. Now rewrite the marginal likelihood bound (2.136), using this same choice of $\hat{q}_{\mathbf{s}_i}(\mathbf{s}_i)$, separate those terms that depend on $\boldsymbol{\theta}$ from those that do not, and substitute in the form from equation (2.138) to obtain:

$$p(\mathbf{y} \,|\, m) \geq \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{1}{\hat{q}_{\mathbf{s}_i}(\mathbf{s}_i)} \right\} \cdot \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right\} \tag{2.140}$$

$$= \frac{p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}})}{\prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}}) \right\}} \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right\} \tag{2.141}$$

$$= \frac{p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}})}{\prod_{i=1}^{n} p(\hat{\mathbf{s}}_i, \mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}})} \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\hat{\mathbf{s}}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) , \tag{2.142}$$

where $\hat{\mathbf{s}}_i$ are defined such that they satisfy:

$$\hat{\mathbf{s}}_i \quad \text{defined such that:} \qquad \ln p(\hat{\mathbf{s}}_i, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) = \sum_{\mathbf{s}_i} \hat{q}_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \tag{2.143}$$

$$= \sum_{\mathbf{s}_i} p(\mathbf{s}_i \,|\, \mathbf{y}, \hat{\boldsymbol{\theta}}) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \tag{2.144}$$

where the second line comes from the requirement of bound equality in (2.139). The existence of such a completion follows from the fact that, in discrete-variable directed acyclic graphs of the sort considered in Chickering and Heckerman (1997), the hidden variables appear only linearly in logarithm of the joint probability $p(\mathbf{s}, \mathbf{y} \,|\, \boldsymbol{\theta})$. Equation (2.142) is the Cheeseman-Stutz criterion, and is also a lower bound on the marginal likelihood. □

It is possible to derive CS-like approximations for types of graphical model other than discrete-variables DAGs. In the above proof no constraints were placed on the forms of the joint distributions over hidden and observed variables, other than in the simplifying step in equation (2.142). So, similar results to corollaries 2.2 and 2.4 can be derived straightforwardly to extend theorem 2.3 to incorporate CE models.

The following corollary shows that variational Bayes can always obtain a tighter bound than the Cheeseman-Stutz approximation.

Corollary 2.5: (theorem 2.3) VB is at least as tight as CS.
That is to say, it is always possible to find distributions $q_{\mathbf{s}}(\mathbf{s})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ such that

$$\ln p(\mathbf{y} \,|\, m)_{CS} \leq \mathcal{F}_m(q_{\mathbf{s}}(\mathbf{s}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \leq \ln p(\mathbf{y} \,|\, m) \,. \tag{2.145}$$

Proof of corollary 2.5. Consider the following forms for $q_{\mathbf{s}}(\mathbf{s})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$:

$$q_{\mathbf{s}}(\mathbf{s}) = \prod_{i=1}^{n} q_{\mathbf{s}_i}(\mathbf{s}_i) \,, \quad \text{with} \quad q_{\mathbf{s}_i}(\mathbf{s}_i) = p(\mathbf{s}_i \,|\, \mathbf{y}_i, \hat{\boldsymbol{\theta}}) \,, \tag{2.146}$$

$$q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \propto \langle \ln p(\boldsymbol{\theta}) p(\mathbf{s}, \mathbf{y} \,|\, \boldsymbol{\theta}) \rangle_{q_{\mathbf{s}}(\mathbf{s})} \,. \tag{2.147}$$

We write the form for $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ explicitly:

$$q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}) \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right\}}{\int d\boldsymbol{\theta}' \; p(\boldsymbol{\theta}') \prod_{i=1}^{n} \exp\left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}') \right\}} \,, \tag{2.148}$$

and note that this is exactly the result of a VBM step. We substitute this and the form for $q_{\mathbf{s}}(\mathbf{s})$ directly into the VB lower bound stated in equation (2.53) of theorem 2.1, obtaining:

$$\mathcal{F}(q_{\mathbf{s}}(\mathbf{s}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta})}{q_{\mathbf{s}_i}(\mathbf{s}_i)} + \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta})}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \tag{2.149}$$

$$= \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{1}{q_{\mathbf{s}_i}(\mathbf{s}_i)} + \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \int d\boldsymbol{\theta}'\; p(\boldsymbol{\theta}') \prod_{i=1}^{n} \exp \left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}') \right\} \tag{2.150}$$

$$= \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{1}{q_{\mathbf{s}_i}(\mathbf{s}_i)} + \ln \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} \exp \left\{ \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}) \right\}, \tag{2.151}$$

which is exactly the logarithm of equation (2.140). And so with this choice of $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $q_{\mathbf{s}}(\mathbf{s})$ we achieve equality between the CS and VB approximations in (2.145).

We complete the proof of corollary 2.5 by noting that any further VB optimisation is guaranteed to increase or leave unchanged the lower bound, and hence surpass the CS lower bound. We would expect the VB lower bound starting from the CS solution to improve upon the CS lower bound in *all* cases, except in the very special case when the MAP parameter $\hat{\boldsymbol{\theta}}$ is exactly the *variational Bayes point*, defined as $\boldsymbol{\theta}_{\mathrm{BP}} \equiv \boldsymbol{\phi}^{-1}(\langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})})$ (see proof of theorem 2.2(a)). Therefore, since VB is a lower bound on the marginal likelihood, the entire statement of (2.145) is proven. □

2.7 Summary

In this chapter we have shown how a variational bound can be used to derive the EM algorithm for ML/MAP parameter estimation, for both unconstrained and constrained representations of the hidden variable posterior. We then moved to the Bayesian framework, and presented the *variational Bayesian EM* algorithm which iteratively optimises a lower bound on the marginal likelihood of the model. The marginal likelihood, which integrates over model parameters, is the key component to Bayesian model selection. The VBE and VBM steps are obtained by taking functional derivatives with respect to variational distributions over hidden variables and parameters respectively.

We gained a deeper understanding of the VBEM algorithm by examining the specific case of *conjugate-exponential* models and showed that, for this large class of models, the posterior distributions $q_{\mathbf{x}}(\mathbf{x})$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ have intuitive and analytically stable forms. We have also presented

VB learning algorithms for both directed and undirected graphs (Bayesian networks and Markov networks).

We have explored the Cheeseman-Stutz model selection criterion as a lower bound of the marginal likelihood of the data, and have explained how it is a very specific case of variational Bayes. Moreover, using this intuition, we have shown that any CS approximation can be improved upon by building a VB approximation over it. It is tempting to derive conjugate-exponential versions of the CS criterion, but in my opinion this is not necessary since any implementations based on these results can be made only more accurate by using conjugate-exponential VB instead, which is at least as general in every case. In chapter 6 we present a comprehensive comparison of VB to a variety of approximation methods, including CS, for a model selection task involving discrete-variable DAGs.

The rest of this thesis applies the VB lower bound to several commonly used statistical models, with a view to performing model selection, learning from both real and synthetic data sets. Throughout we compare the variational Bayesian framework to competitor approximations, such as those reviewed in section 1.3, and also critically analyse the quality of the lower bound using advanced sampling methods.

Chapter 3

Variational Bayesian Hidden Markov Models

3.1 Introduction

Hidden Markov models (HMMs) are widely used in a variety of fields for modelling time series data, with applications including speech recognition, natural language processing, protein sequence modelling and genetic alignment, general data compression, information retrieval, motion video analysis and object/people tracking, and financial time series prediction. The core theory of HMMs was developed principally by Baum and colleagues (Baum and Petrie, 1966; Baum et al., 1970), with initial applications to elementary speech processing, integrating with linguistic models, and making use of insertion and deletion states for variable length sequences (Bahl and Jelinek, 1975). The popularity of HMMs soared the following decade, giving rise to a variety of elaborations, reviewed in Juang and Rabiner (1991). More recently, the realisation that HMMs can be expressed as Bayesian networks (Smyth et al., 1997) has given rise to more complex and interesting models, for example, factorial HMMs (Ghahramani and Jordan, 1997), tree-structured HMMs (Jordan et al., 1997), and switching state-space models (Ghahramani and Hinton, 2000). An introduction to HMM modelling in terms of graphical models can be found in Ghahramani (2001).

This chapter is arranged as follows. In section 3.2 we briefly review the learning and inference algorithms for the standard HMM, including ML and MAP estimation. In section 3.3 we show how an exact Bayesian treatment of HMMs is intractable, and then in section 3.4 follow MacKay (1997) and derive an approximation to a Bayesian implementation using a variational lower bound on the marginal likelihood of the observations. In section 3.5 we present the results of synthetic experiments in which VB is shown to avoid overfitting unlike ML. We also compare ML, MAP and VB algorithms' ability to learn HMMs on a simple benchmark problem of

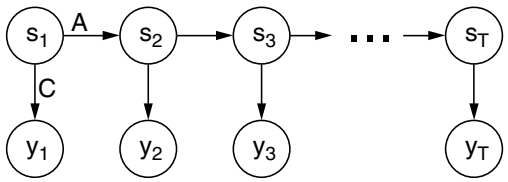


Figure 3.1: Graphical model representation of a hidden Markov model. The hidden variables $\mathbf{s}_t$ transition with probabilities specified in the rows of A, and at each time step emit an observation symbol $\mathbf{y}_t$ according to the probabilities in the rows of C.

discriminating between forwards and backwards English sentences. We present conclusions in section 3.6.

Whilst this chapter is not intended to be a novel contribution in terms of the variational Bayesian HMM, which was originally derived in the unpublished technical report of MacKay (1997), it has nevertheless been included for completeness to provide an immediate and straightforward example of the theory presented in chapter 2. Moreover, the wide applicability of HMMs makes the derivations and experiments in this chapter of potential general interest.

3.2 Inference and learning for maximum likelihood HMMs

We briefly review the learning and inference procedures for hidden Markov models (HMMs), adopting a similar notation to Rabiner and Juang (1986). An HMM models a sequence of p-valued discrete observations (symbols) $\mathbf{y}_{1:T} = \{y_1, \ldots, y_T\}$ by assuming that the observation at time t, y_t, was produced by a k-valued discrete hidden state s_t, and that the sequence of hidden states $\mathbf{s}_{1:T} = \{s_1, \ldots, s_T\}$ was generated by a first-order Markov process. That is to say the complete-data likelihood of a sequence of length T is given by:

$$p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T}) = p(s_1)p(y_1 \mid s_1) \prod_{t=2}^{T} p(s_t \mid s_{t-1})p(y_t \mid s_t) \,. \tag{3.1}$$

where $p(s_1)$ is the prior probability of the first hidden state, $p(s_t \mid s_{t-1})$ denotes the probability of *transitioning* from state s_{t-1} to state s_t (out of a possible k states), and $p(y_t \mid s_t)$ are the *emission* probabilities for each of p symbols at each state. In this simple HMM, all the parameters are assumed stationary, and we assume a fixed finite number of hidden states and number of observation symbols. The joint probability (3.1) is depicted as a graphical model in figure 3.1. For simplicity we first examine just a single sequence of observations, and derive learning and inference procedures for this case; it is straightforward to extend the results to multiple i.i.d. sequences.

The probability of the observations $\mathbf{y}_{1:T}$ results from summing over all possible hidden state sequences,

$$p(\mathbf{y}_{1:T}) = \sum_{\mathbf{s}_{1:T}} p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T}) \,. \tag{3.2}$$

The set of parameters for the initial state prior, transition, and emission probabilities are represented by the parameter $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = (A, C, \boldsymbol{\pi}) \tag{3.3}$$

$$A = \{a_{jj'}\} : a_{jj'} = p(s_t = j' \,|\, s_{t-1} = j) \qquad \text{state transition matrix } (k \times k) \tag{3.4}$$

$$C = \{c_{jm}\} : c_{jm} = p(y_t = m \,|\, s_t = j) \qquad \text{symbol emission matrix } (k \times p) \tag{3.5}$$

$$\boldsymbol{\pi} = \{\pi_j\} : \pi_j = p(s_1 = j) \qquad \text{initial hidden state prior } (k \times 1) \tag{3.6}$$

obeying the normalisation constraints:

$$A = \{a_{jj'}\} : \sum_{j'=1}^{k} a_{jj'} = 1 \quad \forall j \tag{3.7}$$

$$C = \{c_{jm}\} : \sum_{m=1}^{p} c_{jm} = 1 \quad \forall j \tag{3.8}$$

$$\boldsymbol{\pi} = \{\pi_j\} : \sum_{j=1}^{k} \pi_j = 1 \,. \tag{3.9}$$

For mathematical convenience we represent the state of the hidden variables using k-dimensional binary column vectors. For example, if s_t is in state j, then $\mathbf{s}_t$ is a vector of zeros with '1' in the jth entry. We use a similar notation for the observations y_t. The Kronecker-δ function is used to query the state, such that $\mathbf{s}_{t,j} = \delta(s_t, j)$ returns 1 if s_t is in state j, and zero otherwise.

Using the vectorial form of the hidden and observed variables, the initial hidden state, transition, and emission probabilities can be written as

$$p(s_1 \,|\, \boldsymbol{\pi}) = \prod_{j=1}^{k} \pi_j^{\mathbf{s}_{1,j}} \tag{3.10}$$

$$p(s_t \,|\, s_{t-1}, A) = \prod_{j=1}^{k} \prod_{j'=1}^{k} a_{jj'}^{\mathbf{s}_{t,j'}\mathbf{s}_{t-1,j}} \tag{3.11}$$

$$p(y_t \,|\, s_t, C) = \prod_{j=1}^{k} \prod_{m=1}^{p} c_{jm}^{\mathbf{s}_{t,j}\mathbf{y}_{t,m}} \tag{3.12}$$

and the log complete-data likelihood from (3.1) becomes:

$$\ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\theta}) = \sum_{j=1}^{k} \mathbf{s}_{1,j} \ln \pi_j + \sum_{t=2}^{T} \sum_{j=1}^{k} \sum_{j'=1}^{k} \mathbf{s}_{t-1,j} \ln a_{jj'} \mathbf{s}_{t,j'} + \sum_{t=1}^{T} \sum_{j=1}^{k} \sum_{m=1}^{p} \mathbf{s}_{t,j} \ln c_{jm} \mathbf{y}_{t,m} \tag{3.13}$$

$$= \mathbf{s}_1^\top \ln \boldsymbol{\pi} + \sum_{t=2}^{T} \mathbf{s}_{t-1}^\top \ln A \, \mathbf{s}_t + \sum_{t=1}^{T} \mathbf{s}_t^\top \ln C \, \mathbf{y}_t \,, \tag{3.14}$$

where the logarithms of the vector $\boldsymbol{\pi}$ and matrices A and C are taken element-wise. We are now in a position to derive the EM algorithm for ML parameter estimation for HMMs.

M step

Learning the maximum likelihood parameters of the model entails finding those settings of A, C and $\boldsymbol{\pi}$ which maximise the probability of the observed data (3.2). In chapter 2 we showed that the M step, as given by equation (2.31), is

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{s}_{1:T}} p(\mathbf{s}_{1:T} \,|\, \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\theta}) \,, \tag{3.15}$$

where the superscript notation $^{(t)}$ denotes iteration number. Note in particular that the log likelihood in equation (3.14) is a sum of separate contributions involving $\boldsymbol{\pi}$, A and C, and summing over the hidden state sequences does not couple the parameters. Therefore we can individually optimise each parameter of the HMM:

$$\boldsymbol{\pi} \;:\; \pi_j \leftarrow \langle \mathbf{s}_{1,j} \rangle \tag{3.16}$$

$$A \;:\; a_{jj'} \leftarrow \frac{\sum_{t=2}^{T} \langle \mathbf{s}_{t-1,j} \mathbf{s}_{t,j'} \rangle}{\sum_{t=2}^{T} \langle \mathbf{s}_{t-1,j} \rangle} \tag{3.17}$$

$$C \;:\; c_{jm} \leftarrow \frac{\sum_{t=1}^{T} \langle \mathbf{s}_{t,j} \mathbf{y}_{t,m} \rangle}{\sum_{t=1}^{T} \langle \mathbf{s}_{t,j} \rangle} \tag{3.18}$$

where the angled brackets $\langle \cdot \rangle$ denote expectation with respect to the posterior distribution over the hidden state sequence, $p(\mathbf{s}_{1:T} \,|\, \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(t)})$, as calculated from the E step.

E step: forward-backward algorithm

The E step is carried out using a dynamic programming trick which utilises the conditional independence of future hidden states from past hidden states given the setting of the current

hidden state. We define $\alpha_t(\mathbf{s}_t)$ to be the posterior over the hidden state s_t given the observed sequence up to and including time t:

$$\alpha_t(\mathbf{s}_t) \equiv p(\mathbf{s}_t \,|\, \mathbf{y}_{1:t}) \,, \tag{3.19}$$

and form the forward recursion from $t = 1, \ldots, T$:

$$\alpha_t(\mathbf{s}_t) = \frac{1}{p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1})} \sum_{\mathbf{s}_{t-1}} p(\mathbf{s}_{t-1} \,|\, \mathbf{y}_{1:t-1}) p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{s}_t) \tag{3.20}$$

$$= \frac{1}{\zeta_t(\mathbf{y}_t)} \left[\sum_{\mathbf{s}_{t-1}} \alpha_{t-1}(\mathbf{s}_{t-1}) p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}) \right] p(\mathbf{y}_t \,|\, \mathbf{s}_t) \,, \tag{3.21}$$

where in the first time step $p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1})$ is replaced with the prior $p(\mathbf{s}_1 \,|\, \boldsymbol{\pi})$, and for $t = 1$ we require the convention $\alpha_0(\mathbf{s}_0) = 1$. Here, $\zeta_t(\mathbf{y}_t)$ is a normalisation constant, a function of $\mathbf{y}_t$, given by

$$\zeta_t(\mathbf{y}_t) \equiv p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) \,. \tag{3.22}$$

Note that as a by-product of computing these normalisation constants we can compute the probability of the sequence:

$$p(\mathbf{y}_{1:T}) = p(y_1) p(y_2 \,|\, y_1) \ldots p(y_T \,|\, y_{1:T-1}) = \prod_{t=1}^{T} p(y_t \,|\, y_{1:t-1}) = \prod_{t=1}^{T} \zeta_t(\mathbf{y}_t) = \mathcal{Z}(\mathbf{y}_{1:T}) \,. \tag{3.23}$$

Obtaining these normalisation constants using a forward pass is simply equivalent to integrating out the hidden states one after the other in the forward ordering, as can be seen by writing the incomplete-data likelihood in the following way:

$$p(\mathbf{y}_{1:T}) = \sum_{\mathbf{s}_{1:T}} p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T}) \tag{3.24}$$

$$= \sum_{\mathbf{s}_1} \cdots \sum_{\mathbf{s}_T} p(\mathbf{s}_1) p(\mathbf{y}_1 \,|\, \mathbf{s}_1) \prod_{t=2}^{T} p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{s}_t) \tag{3.25}$$

$$= \sum_{\mathbf{s}_1} p(\mathbf{s}_1) p(\mathbf{y}_1 \,|\, \mathbf{s}_1) \cdots \sum_{\mathbf{s}_T} p(\mathbf{s}_T \,|\, \mathbf{s}_{T-1}) p(\mathbf{y}_T \,|\, \mathbf{s}_T) \,. \tag{3.26}$$

Similarly to the forward recursion, the backward recursion is carried out from $t = T, \ldots, 1$:

$$\beta_t(\mathbf{s}_t) \equiv p(\mathbf{y}_{(t+1):T} \,|\, \mathbf{s}_t) \tag{3.27}$$

$$= \sum_{\mathbf{s}_{t+1}} p(\mathbf{y}_{t+2:T} \,|\, \mathbf{s}_{t+1}) p(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{s}_{t+1}) \tag{3.28}$$

$$= \sum_{\mathbf{s}_{t+1}} \beta_{t+1}(\mathbf{s}_{t+1}) p(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{s}_{t+1}) \,, \tag{3.29}$$

with the end condition $\beta_T(\mathbf{s}_T) = 1$, as there is no future observed data beyond $t = T$.

The forward and backward recursions can be executed in parallel as neither depends on the results of the other. The quantities $\{\alpha_t\}_{t=1}^T$ and $\{\beta_t\}_{t=1}^T$ are now combined to obtain the single and pairwise state marginals:

$$p(\mathbf{s}_t \,|\, \mathbf{y}_{1:T}) \propto p(\mathbf{s}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{s}_t) \tag{3.30}$$

$$= \alpha_t(\mathbf{s}_t)\beta_t(\mathbf{s}_t)\,, \qquad t = 1, \ldots, T \tag{3.31}$$

and

$$p(\mathbf{s}_{t-1}, \mathbf{s}_t \,|\, \mathbf{y}_{1:T}) \propto p(\mathbf{s}_{t-1} \,|\, \mathbf{y}_{1:t-1}) p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{s}_t) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{s}_t) \tag{3.32}$$

$$= \alpha_{t-1}(\mathbf{s}_{t-1}) p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{s}_t) \beta_t(\mathbf{s}_t)\,, \qquad t = 2, \ldots, T \tag{3.33}$$

which give the expectations required for the M steps (3.16-3.18),

$$\langle \mathbf{s}_{t,j} \rangle = \frac{\alpha_{t,j}\beta_{t,j}}{\sum_{j'=1}^k \alpha_{t,j'}\beta_{t,j'}} \tag{3.34}$$

$$\langle \mathbf{s}_{t-1,j}\mathbf{s}_{t,j'} \rangle = \frac{\alpha_{t-1,j} a_{jj'} p(\mathbf{y}_t \,|\, \mathbf{s}_{t,j'}) \beta_{t,j'}}{\sum_{j=1}^k \sum_{j'=1}^k \alpha_{t-1,j} a_{jj'} p(\mathbf{y}_t \,|\, \mathbf{s}_{t,j'}) \beta_{t,j'}}\,. \tag{3.35}$$

The E and M steps described above form the iterations for the celebrated Baum-Welch algorithm (Baum et al., 1970). From the analysis in chapter 2, we can prove that each iteration of EM is guaranteed to increase, or leave unchanged, the log likelihood of the parameters, and converge to a local maximum.

When learning an HMM from multiple i.i.d. sequences $\{\mathbf{y}_{i,1:T_i}\}_{i=1}^n$ which are not necessarily constrained to have the same lengths $\{T_i\}_{i=1}^n$, the E and M steps remain largely the same. The E step is performed for each sequence separately using the forward-backward algorithm, and the M step then uses statistics pooled from all the sequences to estimate the mostly likely parameters.

HMMs as described above can be generalised in many ways. Often observed data are recorded as real-valued sequences and can be modelled by replacing the emission process $p(y_t \,|\, s_t)$ with a Gaussian or mixture of Gaussians distribution: each sequence of the HMM can now be thought of as defining a sequence of data drawn from a mixture model whose hidden state labels for the mixture components are no longer i.i.d., but evolve with Markov dynamics. Note that inference in such models remains possible using the forward and backward recursions, with only a change to the emission probabilities $p(y_t \,|\, s_t)$; furthermore, the M steps for learning the parameters $\boldsymbol{\pi}$ and A for the hidden state transitions remain identical.

Exactly analogous inference algorithms exist for the Linear Dynamical Systems (LDS) model, except that both the hidden state transition and emission processes are continuous (referred to

as dynamics and output processes, respectively). In the rest of this chapter we will see how a variational Bayesian treatment of HMMs results in a straightforwardly modified Baum-Welch algorithm, and as such it is a useful pedagogical example of the VB theorems given in chapter 2. On the other hand, for the LDS models the modified VB algorithms become substantially harder to derive and implement — these are the subject of chapter 5.

3.3 Bayesian HMMs

As has already been discussed in chapters 1 and 2, the maximum likelihood approach to learning models from data does not take into account model complexity, and so is susceptible to overfitting the data. More complex models can usually give ever-increasing likelihoods to the data. For a hidden Markov model, the complexity is related to several aspects: the number of hidden states k in the model, the degree of connectivity in the hidden state transition matrix A, and the distribution of probabilities to the symbols by each hidden state, as specified in the emission matrix, C. More generally the complexity is related to the richness of possible data sets that the model can produce. There are $k(k-1)$ parameters in the transition matrix, and $k(p-1)$ in the emission matrix, and so if there are many different observed symbols or if we expect to require more than a few hidden states then, aside from inference becoming very costly, the number of parameters to be fit may begin to overwhelm the amount of data available. Traditionally, in order to avoid overfitting, researchers have limited the complexity of their models in line with the amount of data they have available, and have also used sophisticated modifications to the basic HMM to reduce the number of free parameters. Such modifications include: parameter-tying, enforcing sparsity constraints (for example limiting the number of candidates a state can transition to or symbols it can emit), or constraining the form of the hidden state transitions (for example employing a strict left-to-right ordering of the hidden states).

A common technique for removing excessive parameters from a model is to regularise them using a prior, and then to maximise the a posteriori probability of the parameters (MAP). We will see below that it is possible to apply this type of regularisation to the multinomial parameters of the transition and emission probabilities using certain Dirichlet priors. However we would still expect the results of MAP optimisation to be susceptible to overfitting given that it searches for the maximum of the posterior density as opposed to integrating over the posterior distribution. Cross-validation is another method often employed to minimise the amount of overfitting, by repeatedly training subsets of the available data and evaluating the error on the remaining data. Whilst cross-validation is quite successful in practice, it has the drawback that it requires many sessions of training and so is computationally expensive, and often needs large amounts of data to obtain low-variance estimates of the expected test errors. Moreover, it is cumbersome to cross-validate over the many different ways in which model complexity could vary.

The Bayesian approach to learning treats the model parameters as unknown quantities and, prior to observing the data, assigns a set of beliefs over these quantities in the form of prior distributions. In the light of data, Bayes' rule can be used to infer the posterior distribution over the parameters. In this way the parameters of the model are treated as hidden variables and are integrated out to form the marginal likelihood:

$$p(\mathbf{y}_{1:T}) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) p(\mathbf{y}_{1:T} \,|\, \boldsymbol{\theta}) \qquad \text{where } \boldsymbol{\theta} = (\boldsymbol{\pi}, A, C)\;. \tag{3.36}$$

This Bayesian integration embodies the principle of Occam's razor since it automatically penalises those models with more parameters (see section 1.2.1; also see MacKay, 1992). A natural choice for parameter priors over $\boldsymbol{\pi}$, the rows of A, and the rows of C are Dirichlet distributions. Whilst there are many possible choices, Dirichlet distributions have the advantage that they are conjugate to the complete-data likelihood terms given in equations (3.1) (and with foresight we know that these forms will yield tractable variational Bayesian algorithms):

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\pi}) p(A) p(C) \tag{3.37}$$

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\{\pi_1, \ldots, \pi_k\} \,|\, \mathbf{u}^{(\pi)})) \tag{3.38}$$

$$p(A) = \prod_{j=1}^{k} \mathrm{Dir}(\{a_{j1}, \ldots, a_{jk}\} \,|\, \mathbf{u}^{(A)}) \tag{3.39}$$

$$p(C) = \prod_{j=1}^{k} \mathrm{Dir}(\{c_{j1}, \ldots, c_{jp}\} \,|\, \mathbf{u}^{(C)})\;. \tag{3.40}$$

Here, for each matrix the same single hyperparameter vector is used for every row. This hyperparameter sharing can be motivated because the hidden states are identical a priori. The form of the Dirichlet prior, using $p(\boldsymbol{\pi})$ as an example, is

$$p(\boldsymbol{\pi}) = \frac{\Gamma(u_0^{(\pi)})}{\prod_{j=1}^{k} \Gamma(u_j^{(\pi)})} \prod_{j=1}^{k} \pi_j^{u_j^{(\pi)} - 1}\;, \qquad u_j^{(\pi)} > 0,\; \forall\, j\;, \tag{3.41}$$

where $u_0^{(\pi)} = \sum_{j=1}^{k} u_j^{(\pi)}$ is the *strength* of the prior, and the positivity constraint on the hyperparameters is required for the prior to be proper. Conjugate priors have the intuitive interpretation of providing hypothetical observations to augment those provided by the data (see section 1.2.2). If these priors are used in a *maximum a posteriori* (MAP) estimation algorithm for HMMs, the priors add imaginary counts to the M steps. Taking the update for A as an example, equation (3.17) is modified to

$$A \;:\; a_{jj'} \leftarrow \frac{(u_{j'}^{(A)} - 1) + \sum_{t=2}^{T} \langle \mathbf{s}_{t-1,j} \mathbf{s}_{t,j'} \rangle}{\sum_{j'=1}^{k} (u_{j'}^{(A)} - 1) + \sum_{t=2}^{T} \langle \mathbf{s}_{t-1,j} \rangle}\;. \tag{3.42}$$

Researchers tend to limit themselves to hyperparameters $u_j \geq 1$ such that this MAP estimate is guaranteed to yield positive probabilities. However there are compelling reasons for having hy-

perparameters $u_j \leq 1$ (as discussed in MacKay and Peto, 1995; MacKay, 1998), and these arise naturally as described below. It should be emphasised that the MAP solution is not invariant to reparameterisations, and so (3.42) is just one possible result. For example, reparameterisation into the softmax basis yields a MAP estimate without the '-1' terms, which also coincides with the predictive distribution obtained from integrating over the posterior. The experiments carried out in this chapter for MAP learning do so in this basis.

We choose to use symmetric Dirichlet priors, with a fixed strength f, i.e.

$$\mathbf{u}^{(A)} = \left[\frac{f^{(A)}}{k}, \ldots, \frac{f^{(A)}}{k}\right]^\top , \qquad s.t. \sum_{j=1}^{k} u_j^{(A)} = f^{(A)} , \tag{3.43}$$

and similarly so for $\mathbf{u}^{(C)}$ and $\mathbf{u}^{(\pi)}$. A fixed strength is chosen because we do not want the amount of imaginary data to increase with the complexity of the model. This relates to a key issue in Bayesian prior specification regarding the *scaling* of model priors. Imagine an un-scaled prior over each row of A with hyperparameter $\left[f^{(A)}, \ldots, f^{(A)}\right]^\top$, where the division by k has been omitted. With a fixed strength prior, the contribution to the posterior distributions over the parameters from the prior diminishes with increasing data, whereas with the un-scaled prior the contribution increases linearly with the number of hidden states and can become greater than the amount of observed data for sufficiently large k. This means that for sufficiently complex models the modification terms in (3.42) would obfuscate the data entirely. This is clearly undesirable, and so the $\frac{1}{k}$ scaling of the hyperparameter entries is used. Note that this scaling will result in hyperparameters ≤ 1 for sufficiently large k.

The marginal probability of a sequence of observations is given by

$$p(\mathbf{y}_{1:T}) = \int d\boldsymbol{\pi}\; p(\boldsymbol{\pi}) \int dA\; p(A) \int dC\; p(C) \sum_{\mathbf{s}_{1:T}} p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\pi}, A, C) , \tag{3.44}$$

where the dependence on the hyperparameters is implicitly assumed as they are fixed beforehand. Unfortunately, we can no longer use the dynamic programming trick of the forward-backward algorithm, as the hidden states are now coupled by the integration over the parameters. Intuitively this means that, because the parameters of the model have become uncertain quantities, the future hidden states $\mathbf{s}_{(t+1):T}$ are no longer independent of past hidden states $\mathbf{s}_{1:(t-1)}$ given the current state $\mathbf{s}_t$. The summation and integration operations in (3.44) can be interchanged, but there are still an intractable number of possible sequences to sum over, a number exponential in the length of the sequence. This intractability becomes even worse with multiple sequences, as hidden states of different sequences also become dependent in the posterior.

It is true that for any *given* setting of the parameters, the likelihood calculation is possible, as is finding the distribution over possible hidden state sequences using the forward-backward algorithm; but since the parameters are continuous this insight is not useful for calculating

(3.44). It is also true that for any *given* trajectory representing a single hidden state sequence, we can treat the hidden variables as observed and analytically integrate out the parameters to obtain the marginal likelihood; but since the number of such trajectories is exponential in the sequence length (k^T), this approach is also ruled out.

These considerations form the basis of a very simple and elegant algorithm due to Stolcke and Omohundro (1993) for estimating the marginal likelihood of an HMM. In that work, the posterior distribution over hidden state trajectories is approximated with the most likely sequence, obtained using a Viterbi algorithm for discrete HMMs (Viterbi, 1967). This single sequence (let us assume it is unique) is then treated as observed data, which causes the parameter posteriors to be Dirichlet, which are then easily integrated over to form an estimate of the marginal likelihood. The MAP parameter setting (the mode of the Dirichlet posterior) is then used to infer the most probable hidden state trajectory to iterate the process. Whilst the reported results are impressive, substituting MAP estimates for both parameters and hidden states seems safe only if: there is plenty of data to determine the parameters (i.e. many long sequences); and the individual sequences are long enough to reduce any ambiguity amongst the hidden state trajectories.

Markov chain Monte Carlo (MCMC) methods can be used to approximate the posterior distribution over parameters (Robert et al., 1993), but in general it is hard to assess the convergence and reliability of the estimates required for learning. An analytically-based approach is to approximate the posterior distribution over the parameters with a Gaussian, which usually allows the integral to become tractable. Unfortunately the Laplace approximation is not well-suited to bounded or constrained parameters (e.g. sum-to-one constraints), and computation of the likelihood Hessian can be computationally expensive. In MacKay (1998) an argument for transforming the Dirichlet prior into the softmax basis is presented, although to the best of our knowledge this approach is not widely used for HMMs.

3.4 Variational Bayesian formulation

In this section we derive the variational Bayesian implementation of HMMs, first presented in MacKay (1997). We show that by making only the approximation that the posterior over hidden variables and parameters factorises, an approximate posterior distribution over hidden state trajectories can be inferred under an *ensemble* of model parameters, and how an approximate posterior distribution over parameters can be analytically obtained from the sufficient statistics of the hidden state.

3.4.1 Derivation of the VBEM optimisation procedure

Our choice of priors $p(\boldsymbol{\theta})$ and the complete-data likelihood $p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\theta})$ for HMMs satisfy conditions (2.80) and (2.88) respectively, for membership of the conjugate-exponential (CE) family. Therefore it is possible to apply the results of theorem 2.2 directly to obtain the VBM and VBE steps. The derivation is given here step by step, and the ideas of chapter 2 brought in gradually. We begin with the log marginal likelihood for an HMM (3.36), and lower bound it by introducing any distribution over the parameters and hidden variables $q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T})$:

$$\ln p(\mathbf{y}_{1:T}) = \ln \int d\boldsymbol{\pi} \int dA \int dC \sum_{\mathbf{s}_{1:T}} p(\boldsymbol{\pi}, A, C) p(\mathbf{y}_{1:T}, \mathbf{s}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \tag{3.45}$$

$$\geq \int d\boldsymbol{\pi} \int dA \int dC \sum_{\mathbf{s}_{1:T}} q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T}) \ln \frac{p(\boldsymbol{\pi}, A, C) p(\mathbf{y}_{1:T}, \mathbf{s}_{1:T} \,|\, \boldsymbol{\pi}, A, C)}{q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T})} \,. \tag{3.46}$$

This inequality is tight when $q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T})$ is set to the exact posterior over hidden variables and parameters $p(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T} \,|\, \mathbf{y}_{1:T})$, but it is intractable to compute this distribution. We make progress by assuming that the posterior is factorised:

$$p(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T} \,|\, \mathbf{y}_{1:T}) \approx q(\boldsymbol{\pi}, A, C) q(\mathbf{s}_{1:T}) \tag{3.47}$$

which gives a lower bound of the form

$$\ln p(\mathbf{y}_{1:T}) \geq \int d\boldsymbol{\pi} \int dA \int dC \sum_{\mathbf{s}_{1:T}} q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T}) \ln \frac{p(\boldsymbol{\pi}, A, C) p(\mathbf{y}_{1:T}, \mathbf{s}_{1:T} \,|\, \boldsymbol{\pi}, A, C)}{q(\boldsymbol{\pi}, A, C, \mathbf{s}_{1:T})} \tag{3.48}$$

$$= \int d\boldsymbol{\pi} \int dA \int dC \; q(\boldsymbol{\pi}, A, C) \Bigg[\ln \frac{p(\boldsymbol{\pi}, A, C)}{q(\boldsymbol{\pi}, A, C)} + \sum_{\mathbf{s}_{1:T}} q(\mathbf{s}_{1:T}) \ln \frac{p(\mathbf{y}_{1:T}, \mathbf{s}_{1:T} \,|\, \boldsymbol{\pi}, A, C)}{q(\mathbf{s}_{1:T})} \Bigg] \tag{3.49}$$

$$= \mathcal{F}(q(\boldsymbol{\pi}, A, C), q(\mathbf{s}_{1:T})) \,, \tag{3.50}$$

where the dependence on $\mathbf{y}_{1:T}$ is taken to be implicit. On taking functional derivatives of $\mathcal{F}$ with respect to $q(\boldsymbol{\pi}, A, C)$ we obtain

$$\ln q(\boldsymbol{\pi}, A, C) = \ln p(\boldsymbol{\pi}, A, C) \langle \ln p(\mathbf{y}_{1:T}, \mathbf{s}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \rangle_{q(\mathbf{s}_{1:T})} + c \tag{3.51}$$

$$\begin{aligned} &= \ln p(\boldsymbol{\pi}) + \ln p(A) + \ln p(C) \\ &\quad + \langle \ln p(\mathbf{s}_1 \,|\, \boldsymbol{\pi}) \rangle_{q(\mathbf{s}_1)} + \langle \ln p(\mathbf{s}_{2:T} \,|\, \mathbf{s}_1, A) \rangle_{q(\mathbf{s}_{1:T})} \\ &\quad + \langle \ln p(\mathbf{y}_{1:T} \,|\, \mathbf{s}_{1:T}, C) \rangle_{q(\mathbf{s}_{1:T})} + c \,, \end{aligned} \tag{3.52}$$

where c is a normalisation constant. Given that the prior over the parameters (3.37) factorises, and the log complete-data likelihood (3.14) is a sum of terms involving each of $\boldsymbol{\pi}$, A, and C, the variational posterior over the parameters can be factorised *without further approximation* into:

$$q(\boldsymbol{\pi}, A, C) = q(\boldsymbol{\pi})q(A)q(C) \; . \tag{3.53}$$

Note that sometimes this independence is assumed beforehand and believed to concede accuracy, whereas we have seen that it falls out from a free-form extremisation of the posterior with respect to the entire variational posterior over the parameters $q(\boldsymbol{\pi}, A, C)$, and is therefore exact once the assumption of factorisation between hidden variables and parameters has been made.

The VBM step

The VBM step is obtained by taking functional derivatives of $\mathcal{F}$ with respect to each of these distributions and equating them to zero, to yield Dirichlet distributions:

$$q(\boldsymbol{\pi}) = \mathrm{Dir}(\{\pi_1, \ldots, \pi_k\} \,|\, \{w_1^{(\pi)}, \ldots, w_k^{(\pi)}\}) \tag{3.54}$$

$$\text{with} \quad w_j^{(\pi)} = u_j^{(\pi)} + \langle \delta(s_1, j) \rangle_{q(\mathbf{s}_{1:T})} \tag{3.55}$$

$$q(A) = \prod_{j=1}^{k} \mathrm{Dir}(\{a_{j1}, \ldots, a_{jk}\} \,|\, \{w_{j1}^{(A)}, \ldots, w_{jk}^{(A)}\}) \tag{3.56}$$

$$\text{with} \quad w_{jj'}^{(A)} = u_{j'}^{(A)} + \sum_{t=2}^{T} \langle \delta(s_{t-1}, j)\delta(s_t, j') \rangle_{q(\mathbf{s}_{1:T})} \tag{3.57}$$

$$q(C) = \prod_{j=1}^{k} \mathrm{Dir}(\{c_{j1}, \ldots, c_{jp}\} \,|\, \{w_{j1}^{(C)}, \ldots, w_{jp}^{(C)}\}) \tag{3.58}$$

$$\text{with} \quad w_{jq}^{(C)} = u_q^{(A)} + \sum_{t=1}^{T} \langle \delta(s_t, j)\delta(y_t, q) \rangle_{q(\mathbf{s}_{1:T})} \; . \tag{3.59}$$

These are straightforward applications of the result in theorem 2.2(b), which states that the variational posterior distributions have the same form as the priors with their hyperparameters augmented by sufficient statistics of the hidden state and observations.

The VBE step

Taking derivatives of $\mathcal{F}$ (3.49) with respect to the variational posterior over the hidden state $q(\mathbf{s}_{1:T})$ yields:

$$\ln q(\mathbf{s}_{1:T}) = \langle \ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \rangle_{q(\boldsymbol{\pi})q(A)q(C)} - \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \; , \tag{3.60}$$

where $\tilde{\mathcal{Z}}(\mathbf{y}_{1:T})$ is an important normalisation constant that we will return to shortly. Substituting in the complete-data likelihood from (3.14) yields

$$\ln q(\mathbf{s}_{1:T}) = \left\langle \mathbf{s}_1^\top \ln \boldsymbol{\pi} + \sum_{t=2}^{T} \mathbf{s}_{t-1}^\top \ln A \, \mathbf{s}_t + \sum_{t=1}^{T} \mathbf{s}_t^\top \ln C \, \mathbf{y}_t \right\rangle_{q(\boldsymbol{\pi})q(A)q(C)} - \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \quad (3.61)$$

$$= \mathbf{s}_1^\top \langle \ln \boldsymbol{\pi} \rangle_{q(\boldsymbol{\pi})} + \sum_{t=2}^{T} \mathbf{s}_{t-1}^\top \langle \ln A \rangle_{q(A)} \, \mathbf{s}_t + \sum_{t=1}^{T} \mathbf{s}_t^\top \langle \ln C \rangle_{q(C)} \, \mathbf{y}_t - \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \, . \quad (3.62)$$

Note that (3.62) appears identical to the complete-data likelihood of (3.14) except that expectations are now taken of the logarithm of the parameters. Relating this to the result in corollary 2.2, the natural parameter vector $\boldsymbol{\phi}(\boldsymbol{\theta})$ is given by

$$\boldsymbol{\theta} = (\boldsymbol{\pi} \, , A \, , C) \quad (3.63)$$

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = (\ln \boldsymbol{\pi} \, , \ln A \, , \ln C) \, , \quad (3.64)$$

and the expected natural parameter vector $\overline{\boldsymbol{\phi}}$ is given by

$$\overline{\boldsymbol{\phi}} \equiv \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q(\boldsymbol{\theta})} = (\langle \ln \boldsymbol{\pi} \rangle_{q(\boldsymbol{\pi})}, \; \langle \ln A \rangle_{q(A)}, \; \langle \ln C \rangle_{q(C)}) \, . \quad (3.65)$$

Corollary 2.2 suggests that we can use a modified parameter, $\tilde{\boldsymbol{\theta}}$, in the same inference algorithm (forward-backward) in the VBE step. The modified parameter $\tilde{\boldsymbol{\theta}}$ satisfies $\overline{\boldsymbol{\phi}} = \boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q(\boldsymbol{\theta})}$, and is obtained simply by using the inverse of the $\boldsymbol{\phi}$ operator:

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{\phi}^{-1}(\langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q(\boldsymbol{\theta})}) = (\exp\langle \ln \boldsymbol{\pi} \rangle_{q(\boldsymbol{\pi})} \, , \exp\langle \ln A \rangle_{q(A)} \, , \exp\langle \ln C \rangle_{q(C)}) \quad (3.66)$$

$$= (\tilde{\boldsymbol{\pi}} \, , \tilde{A} \, , \tilde{C}) \, . \quad (3.67)$$

Note that the natural parameter mapping $\boldsymbol{\phi}$ operates separately on each of the parameters in the vector $\boldsymbol{\theta}$, which makes the inversion of the mapping $\boldsymbol{\phi}^{-1}$ straightforward. This is a consequence of these parameters being uncoupled in the complete-data likelihood. For other CE models, the inversion of the natural parameter mapping may not be as simple, since having uncoupled parameters is not necessarily a condition for CE family membership. In fact, in chapter 5 we encounter such a scenario for Linear Dynamical Systems.

It remains for us to calculate the expectations of the logarithm of the parameters under the Dirichlet distributions. We use the result that

$$\int d\boldsymbol{\pi} \; \mathrm{Dir}(\boldsymbol{\pi} \,|\, \mathbf{u}) \ln \pi_j = \psi(u_j) - \psi(\sum_{j=1}^{k} u_j) \, , \quad (3.68)$$

where ψ is the *digamma* function (see appendices A and C.1 for details). This yields

$$\tilde{\boldsymbol{\pi}} = \{\tilde{\pi}_j\} = \exp\left[\psi(w_j^{(\pi)}) - \psi(\sum_{j=1}^{k} w_j^{(\pi)})\right] \;:\; \sum_{j=1}^{k} \tilde{\pi}_j \leq 1 \tag{3.69}$$

$$\tilde{A} = \{\tilde{a}_{jj'}\} = \exp\left[\psi(w_{jj'}^{(A)}) - \psi(\sum_{j'=1}^{k} w_{jj'}^{(A)})\right] \;:\; \sum_{j'=1}^{k} \tilde{a}_{jj'} \leq 1 \quad \forall j \tag{3.70}$$

$$\tilde{C} = \{\tilde{c}_{jm}\} = \exp\left[\psi(w_{jm}^{(C)}) - \psi(\sum_{m=1}^{p} w_{jm}^{(C)})\right] \;:\; \sum_{m=1}^{p} \tilde{c}_{jm} \leq 1 \quad \forall j \;. \tag{3.71}$$

Note that taking geometric averages has resulted in sub-normalised probabilities. We may still use the forward-backward algorithm with these sub-normalised parameters, but should bear in mind that the normalisation constants (scaling factors) change. The forward pass (3.21) becomes

$$\alpha_t(\mathbf{s}_t) = \frac{1}{\tilde{\zeta}_t(\mathbf{y}_t)} \left[\sum_{\mathbf{s}_{t-1}} \alpha_{t-1}(\mathbf{s}_{t-1}) \tilde{p}(\mathbf{s}_t \,|\, \mathbf{s}_{t-1})\right] \tilde{p}(\mathbf{y}_t \,|\, \mathbf{s}_t) \;, \tag{3.72}$$

where $\tilde{p}(\mathbf{s}_t \,|\, \mathbf{s}_{t-1})$ and $\tilde{p}(\mathbf{y}_t \,|\, \mathbf{s}_t)$ are new subnormalised probability distributions according to the parameters $\tilde{A}, \tilde{C}$, respectively. Since $\alpha_t(\mathbf{s}_t)$ is the posterior probability of $\mathbf{s}_t$ given data $\mathbf{y}_{1:t}$, it must sum to one. This implies that, for any *particular* time step, the normalisation $\tilde{\zeta}_t(\mathbf{y}_t)$ must be smaller than if we had used normalised parameters. Similarly the backward pass becomes

$$\beta_t(\mathbf{s}_t) = \sum_{\mathbf{s}_{t+1}} \beta_{t+1}(\mathbf{s}_{t+1}) \tilde{p}(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t) \tilde{p}(\mathbf{y}_{t+1} \,|\, \mathbf{s}_{t+1}) \;. \tag{3.73}$$

Computation of the lower bound $\mathcal{F}$

Recall from (3.22) that the product of the normalisation constants corresponds to the probability of the sequence. Here the product of normalisation constants corresponds to a different quantity:

$$\prod_{t=1}^{T} \tilde{\zeta}_t(\mathbf{y}_t) = \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \tag{3.74}$$

which is the normalisation constant given in (3.60). Thus the modified forward-backward algorithm recursively computes the normalisation constant by integrating out each s_t in $q(\mathbf{s}_{1:T})$, as opposed to $p(\mathbf{s}_{1:T} \,|\, \mathbf{y}_{1:T})$. We now show how $\tilde{\mathcal{Z}}(\mathbf{y}_{1:T})$ is useful for computing the lower bound, just as $\mathcal{Z}(\mathbf{y}_{1:T})$ was useful for computing the likelihood in the ML system.

Using (3.49) the lower bound can be written as

$$\begin{aligned}\mathcal{F}(q(\boldsymbol{\pi}, A, C), q(\mathbf{s}_{1:T})) = &\int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln \frac{p(\boldsymbol{\pi})}{q(\boldsymbol{\pi})} + \int dA\; q(A) \ln \frac{p(A)}{q(A)} + \int dC\; q(C) \ln \frac{p(C)}{q(C)} \\ &+ H(q(\mathbf{s}_{1:T})) \\ &+ \langle \ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \rangle_{q(\boldsymbol{\pi})q(A)q(C)q(\mathbf{s}_{1:T})} \,, \end{aligned} \tag{3.75}$$

where $H(q(\mathbf{s}_{1:T}))$ is the entropy of the variational posterior distribution over hidden state sequences. Straight after a VBE step, the form of the hidden state posterior $q(\mathbf{s}_{1:T})$ is given by (3.60), and the entropy can be written:

$$H(q(\mathbf{s}_{1:T})) = -\sum_{\mathbf{s}_{1:T}} q(\mathbf{s}_{1:T}) \ln q(\mathbf{s}_{1:T}) \tag{3.76}$$

$$= -\sum_{\mathbf{s}_{1:T}} q(\mathbf{s}_{1:T}) \left[\langle \ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \rangle_{q(\boldsymbol{\pi})q(A)q(C)} - \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \right] \tag{3.77}$$

$$= -\sum_{\mathbf{s}_{1:T}} q(\mathbf{s}_{1:T}) \langle \ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T} \,|\, \boldsymbol{\pi}, A, C) \rangle_{q(\boldsymbol{\pi})q(A)q(C)} + \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \,. \tag{3.78}$$

Substituting this into (3.75) cancels the expected log complete-data likelihood terms, giving

$$\begin{aligned}\mathcal{F}(q(\boldsymbol{\pi}, A, C), q(\mathbf{s}_{1:T})) = &\int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln \frac{p(\boldsymbol{\pi})}{q(\boldsymbol{\pi})} + \int dA\; q(A) \ln \frac{p(A)}{q(A)} + \int dC\; q(C) \ln \frac{p(C)}{q(C)} \\ &+ \ln \tilde{\mathcal{Z}}(\mathbf{y}_{1:T}) \end{aligned} \tag{3.79}$$

Therefore computing $\mathcal{F}$ for variational Bayesian HMMs consists of evaluating KL divergences between variational posterior and prior Dirichlet distributions for each row of $\boldsymbol{\pi}$, A, C (see appendix A), and collecting the modified normalisation constants $\{\tilde{\zeta}_t(y_t)\}_{t=1}^T$. In essence we have by-passed the difficulty of trying to compute the entropy of the hidden state by recursively computing it with the VBE step's forward pass. Note that this calculation is then only valid straight after the VBE step.

VB learning with multiple i.i.d. sequences is conceptually straightforward and very similar to that described above for ML learning. For the sake of brevity the reader is referred to the chapter on Linear Dynamical Systems, specifically section 5.3.8 and equation (5.152), from which the implementational details for variational Bayesian HMMs can readily be inferred.

Optimising the hyperparameters of the model is straightforward. Since the hyperparameters appear in $\mathcal{F}$ only in the KL divergence terms, maximising the marginal likelihood amounts to minimising the KL divergence between each parameter's variational posterior distribution and its prior distribution. We did not optimise the hyperparameters in the experiments, but instead examined several different settings.

3.4.2 Predictive probability of the VB model

In the Bayesian scheme, the predictive probability of a test sequence $\mathbf{y}' = \mathbf{y}'_{1:T'}$, given a set of training cases denoted by $\mathbf{y} = \{\mathbf{y}_{i,1:T_i}\}_{i=1}^{n}$, is obtained by averaging the predictions of the HMM with respect to the posterior distributions over its parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, A, C\}$:

$$p(\mathbf{y}' \,|\, \mathbf{y}) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y}) p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \;. \tag{3.80}$$

Unfortunately, for the very same reasons that the marginal likelihood of equation (3.44) is intractable, so is the predictive probability. There are several possible methods for approximating the predictive probability. One such method is to sample parameters from the posterior distribution and construct a Monte Carlo estimate. Should it not be possible to sample directly from the posterior, then importance sampling or its variants can be used. This process can be made more efficient by employing Markov chain Monte Carlo and related methods. Alternatively, the posterior distribution can be approximated with some form which when combined with the likelihood term becomes amenable to integration analytically; it is unclear which analytical forms might yield good approximations.

An alternative is to approximate the posterior distribution with the variational posterior distribution resulting from the VB optimisation:

$$p(\mathbf{y}' \,|\, \mathbf{y}) \approx \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \;. \tag{3.81}$$

The variational posterior is a product of Dirichlet distributions, which is in the same form as the prior, and so we have not gained a great deal because we know this integral is intractable. However we can perform two lower bounds on this quantity to obtain:

$$\begin{aligned} p(\mathbf{y}' \,|\, \mathbf{y}) &\approx \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) p(\mathbf{y}' \,|\, \boldsymbol{\theta}) && (3.82)\\ &\geq \exp \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) \ln \sum_{\mathbf{s}'_{1:T'}} p(\mathbf{s}'_{1:T}, \mathbf{y}'_{1:T} \,|\, \boldsymbol{\theta}) && (3.83)\\ &\geq \exp \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) \sum_{\mathbf{s}'_{1:T'}} q(\mathbf{s}'_{1:T'}) \ln \frac{p(\mathbf{s}'_{1:T}, \mathbf{y}'_{1:T} \,|\, \boldsymbol{\theta})}{q(\mathbf{s}'_{1:T'})} \;. && (3.84) \end{aligned}$$

Equation 3.84 is just the last term in the expression for the lower bound of the marginal likelihood of a training sequence given by (3.49), but with the test sequence in place of the training sequence. This insight provides us with the following method to evaluate the approximation. One simply carries out a VBE step on the test sequence, starting from the result of the last VBM step on the training set, and gathers the normalisation constants $\{\tilde{\mathcal{Z}}'_t\}_{t=1}^{T'_i}$ and takes the product of these. Whilst this is a very straightforward method, it should be remembered that it is only a bound on an approximation.

A different way to obtain the predictive probability is to assume that the model at the mean (or mode) of the variational posterior, with parameter $\boldsymbol{\theta}_{\text{MVB}}$, is representative of the distribution as a whole. The likelihood of the test sequence is then computed under the single model with those parameters, which is tractable:

$$p(\mathbf{y}' \,|\, \mathbf{y})_{\text{MVB}} = \sum_{\mathbf{s}'_{1:T}} p(\mathbf{s}'_{1:T}, \mathbf{y}'_{1:T} \,|\, \boldsymbol{\theta}_{\text{MVB}}) \,. \tag{3.85}$$

This approach is suggested as further work in MacKay (1997), and is discussed in the experiments described below.

3.5 Experiments

In this section we perform two experiments, the first on synthetic data to demonstrate the ability of the variational Bayesian algorithm to avoid overfitting, and the second on a toy data set to compare ML, MAP and VB algorithm performance at discriminating between forwards and backwards English character sequences.

3.5.1 Synthetic: discovering model structure

For this experiment we trained ML and VB hidden Markov models on examples of three types of sequences with a three-symbol alphabet $\{a, b, c\}$. Using standard regular expression notation, the first type of sequence was a substring of the regular grammar $(abc)^*$, the second a substring of $(acb)^*$, and the third from $(a^*b^*)^*$ where a and b symbols are emitted stochastically with probability $\frac{1}{2}$ each. For example, the training sequences included the following:

$$\begin{aligned}
\mathbf{y}_{1,1:T_1} &= (abcabcabcabcabcabcabcabcabcabcabc) \\
\mathbf{y}_{2,1:T_2} &= (bcabcabcabcabcabcabcabcabcabcabc) \\
&\vdots \\
\mathbf{y}_{12,1:T_{12}} &= (acbacbacbacbacbacbacb) \\
\mathbf{y}_{13,1:T_{13}} &= (acbacbacbacbacbacbacbacbacbacbacbac) \\
&\vdots \\
\mathbf{y}_{n-1,1:T_{n-1}} &= (baabaabbabaaaabbabaaabbaabbbaa) \\
\mathbf{y}_{n,1:T_n} &= (abaaabbababaabababbbbbaaabaaabba) \,.
\end{aligned}$$

In all, the training data consisted of 21 sequences of maximum length 39 symbols. Looking at these sequences, we would expect an HMM to require 3 hidden states to model $(abc)^*$, a dif-

ferent 3 hidden states to model $(acb)^*$, and a single self-transitioning hidden state stochastically emitting a and b symbols to model $(a^*b^*)^*$. This gives a total of 7 hidden states required to model the data perfectly. With this foresight we therefore chose HMMs with $k = 12$ hidden states to allow for some redundancy and room for overfitting.

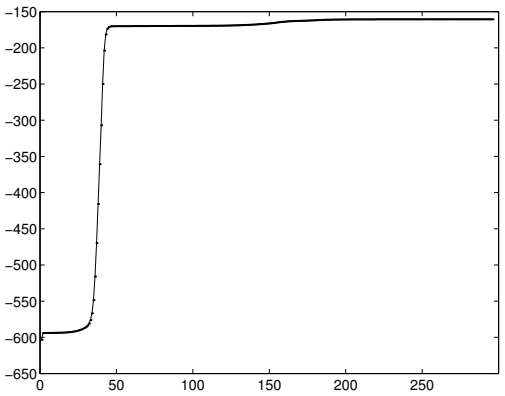
The parameters were initialised by drawing the components of the probability vectors from a uniform distribution and normalising. First the ML algorithm was run to convergence, and then the VB algorithm run *from that point* in parameter space to convergence. This was made possible by initialising each parameter's variational posterior distribution to be Dirichlet with the ML parameter as mean and a strength arbitrarily set to 10. For the MAP and VB algorithms, the prior over each parameter was a symmetric Dirichlet distribution of strength 4.

Figure 3.2 shows the profile of the likelihood of the data under the ML algorithm and the subsequent profile of the lower bound on the marginal likelihood under the VB algorithm. Note that it takes ML about 200 iterations to converge to a local optimum, and from this point it takes only roughly 25 iterations for the VB optimisation to converge — we might expect this as VB is initialised with the ML parameters, and so has less work to do.

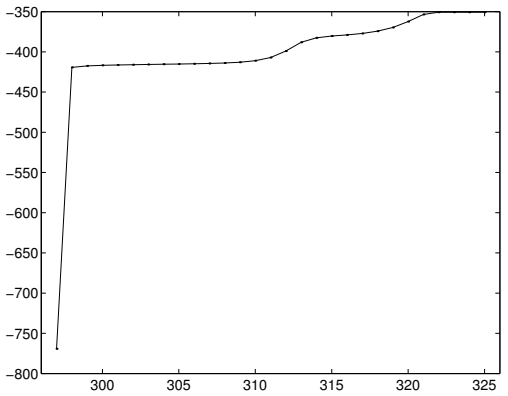
Figure 3.3 shows the recovered ML parameters and VB distributions over parameters for this problem. As explained above, we require 7 hidden states to model the data perfectly. It is clear from figure 3.3(a) that the ML model has used more hidden states than needed, that is to say it has overfit the structure of the model. Figures 3.3(b) and 3.3(c) show that the VB optimisation has removed excess transition and emission processes and, on close inspection, has recovered exactly the model that was postulated above. For example: state (4) self-transitions, and emits the symbols a and b in approximately equal proportions to generate the sequences $(a^*b^*)^*$; states (9,10,8) form a strong repeating path in the hidden state space which (almost) deterministically produce the sequences $(acb)^*$; and lastly the states (3,12,2) similarly interact to produce the sequences $(abc)^*$. A consequence of the Bayesian scheme is that *all* the entries of the transition and emission matrices are necessarily non-zero, and those states (1,5,6,7,11) that are not involved in the dynamics have uniform probability of transitioning to all others, and indeed of generating any symbol, in agreement with the symmetric prior. However these states have small probability of being used at all, as both the distribution $q(\boldsymbol{\pi})$ over the initial state parameter $\boldsymbol{\pi}$ is strongly peaked around high probabilities for the remaining states, and they have very low probability of being transitioned into by the active states.

3.5.2 Forwards-backwards English discrimination

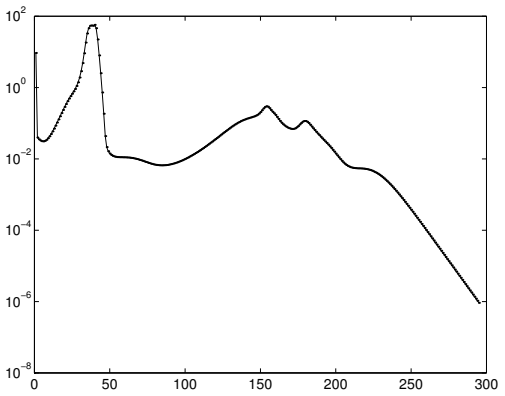
In this experiment, models learnt by ML, MAP and VB are compared on their ability to discriminate between forwards and backwards English text (this toy experiment is suggested in MacKay, 1997). A sentence is classified according to the predictive log probability under each



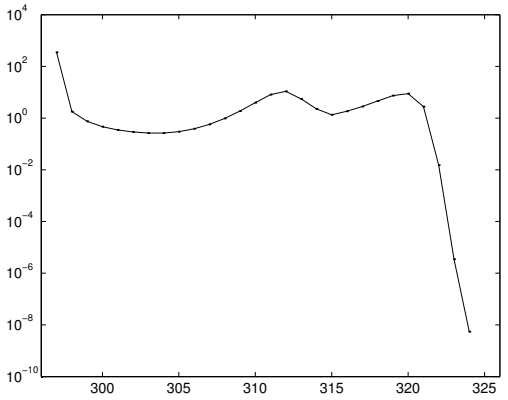
(a) ML: plot of the log likelihood of the data, $p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})$.



(b) VB: plot of the lower bound $\mathcal{F}(q(\mathbf{s}_{1:T}), q(\boldsymbol{\theta}))$.

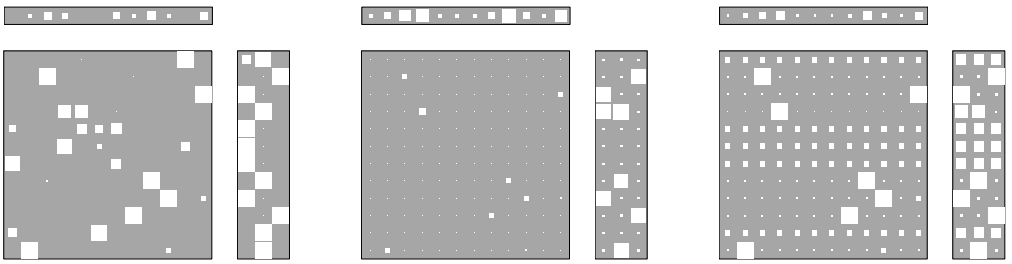


(c) ML: plot of the derivative of the log likelihood in (a).



(d) VB: plot of the derivative of the lower bound in (b).

Figure 3.2: Training ML and VB hidden Markov models on synthetic sequences drawn from $(abc)^*$, $(acb)^*$ and $(a^*b^*)^*$ grammars (see text). Subplots **(a)** & **(c)** show the evolution of the likelihood of the data in the maximum likelihood EM learning algorithm for the HMM with $k = 12$ hidden states. As can be seen in subplot (c) the algorithm converges to a local maximum after by about 296 iterations of EM. Subplots **(b)** & **(d)** plot the marginal likelihood lower bound $\mathcal{F}(q(\mathbf{s}_{1:T}), q(\boldsymbol{\theta}))$ and its derivative, as a *continuation* of learning from the point in parameter space where ML converged (see text) using the variational Bayes algorithm. The VB algorithm converges after about 29 iterations of VBEM.



(a) ML state prior $\boldsymbol{\pi}$, transition A and emission C probabilities.

(b) VB variational posterior parameters for $q(\boldsymbol{\pi})$, $q(A)$ and $q(C)$.

(c) Variational posterior mean probabilities: $\langle q(\boldsymbol{\pi})\rangle$, $\langle q(A)\rangle$ and $\langle q(C)\rangle$.

Figure 3.3: **(a)** Hinton diagrams showing the probabilities learnt by the ML model, for the initial state prior $\boldsymbol{\pi}$, transition matrix A, and emission matrix C. **(b)** Hinton diagrams for the analogous quantities $\mathbf{u}^{(\pi)}$, $\mathbf{u}^{(A)}$ and $\mathbf{u}^{(C)}$, which are the variational parameters (counts) describing the posterior distributions over the parameters $q(\boldsymbol{\pi})$, $q(A)$, and $q(C)$ respectively. **(c)** Hinton diagrams showing the mean/modal probabilities of the posteriors represented in (b), which are simply row-normalised versions of $\mathbf{u}^{(\pi)}$, $\mathbf{u}^{(A)}$ and $\mathbf{u}^{(C)}$.

of the learnt models of forwards and backwards character sequences. As discussed above in section 3.4.2, computing the predictive probability for VB is intractable, and so we approximate the VB solution with the model at the mean of the variational posterior given by equations (3.54–3.59).

We used sentences taken from Lewis Carroll's *Alice's Adventures in Wonderland*. All punctuation was removed to leave 26 letters and the blank space (that is to say $p = 27$). The training data consisted of a maximum of 32 sentences (of length between 10 and 100 characters), and the test data a fixed set of 200 sentences of unconstrained length. As an example, the first 10 training sequences are given below:

(1) 'i shall be late '

(2) 'thought alice to herself after such a fall as this i shall think nothing of tumbling down stairs '

(3) 'how brave theyll all think me at home '

(4) 'why i wouldnt say anything about it even if i fell off the top of the house '

(5) 'which was very likely true '

(6) 'down down down '

(7) 'would the fall never come to an end '

(8) 'i wonder how many miles ive fallen by this time '

(9) 'she said aloud '

(10) 'i must be getting somewhere near the centre of the earth '

ML, MAP and VB hidden Markov models were trained on varying numbers of sentences (sequences), n, varying numbers of hidden states, k, and for MAP and VB, varying prior strengths, u_0, common to all the hyperparameters $\{\mathbf{u}^{(\pi)}, \mathbf{u}^{(A)}, \mathbf{u}^{(C)}\}$. The choices were:

$$n \in \{1, 2, 3, 4, 5, 6, 8, 16, 32\}, \quad k \in \{1, 2, 4, 10, 20, 40, 60\}, \quad u_0 \in \{1, 2, 4, 8\} \; . \tag{3.86}$$

The MAP and VB algorithms were initialised at the ML estimates (as per the previous experiment), both for convenience and fairness. The experiments were repeated a total of 10 times to explore potential multiple maxima in the optimisation.

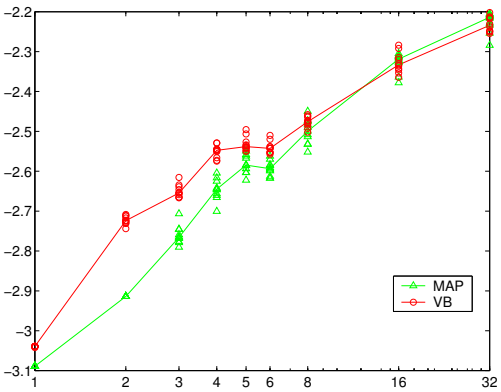
In each scenario two models were learnt, one based on forwards sentences and the other on backwards sentences, and the discrimination performance was measured by the average fraction of times the forwards and backwards models correctly classified forwards and backwards test sentences. This classification was based on the log probability of the test sequence under the forwards and backwards models learnt by each method.

Figure 3.4 presents some of the results from these experiments. Each subplot is an examination of the effect of one of the following: the size of the training set n, the number of hidden states k, or the hyperparameter setting u_0, whilst holding the other two quantities fixed. For the purposes of demonstrating the main trends, the results have been chosen around the canonical values of $n = 2$, $k = 40$, and $u_0 = 2$.

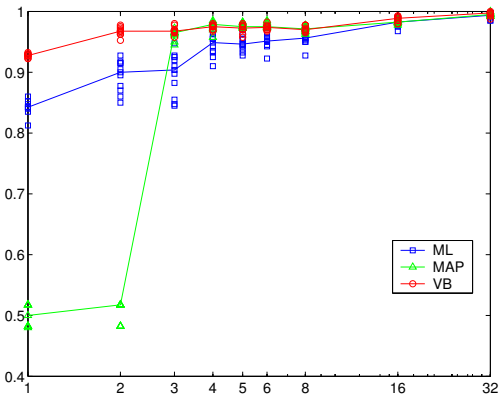
Subplots **(a,c,e)** of figure 3.4 show the average test log probability *per symbol* in the test sequence, for MAP and VB algorithms, as reported on 10 runs of each algorithm. Note that for VB the log probability is measured under the model at the mode of the VB posterior. The plotted curve is the median of these 10 runs. The test log probability for the ML method is omitted from these plots as it is well below the MAP and VB likelihoods (qualitatively speaking, it increases with n in **(a)**, it decreases with k in **(c)**, and is constant with u_0 in **(e)** as the ML algorithm ignores the prior over parameters). Most importantly, in **(a)** we see that VB outperforms MAP when the model is trained on only a few sentences, which suggests that entertaining a distribution over parameters is indeed improving performance. These log likelihoods are those of the forward sequences evaluated under the forward models; we expect these trends to be repeated for reverse sentences as well.

Subplots **(b,d,f)** of figure 3.4 show the fraction of correct classifications of forwards sentences as forwards, and backwards sentences as backwards, as a function of n, k and u_0, respectively.

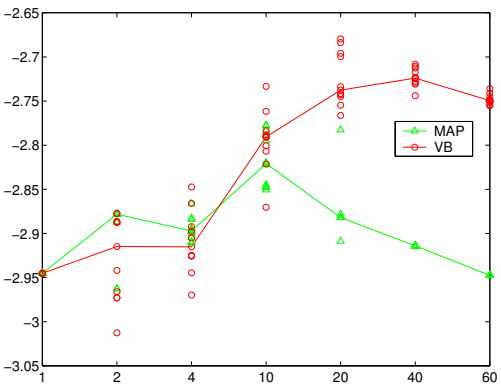
We see that for the most part VB gives higher likelihood to the test sequences than MAP, and also outperforms MAP and ML in terms of discrimination. For large amounts of training data n, VB and MAP converge to approximately the same performance in terms of test likelihood and discrimination. As the number of hidden states k increases, VB outperforms MAP considerably, although we should note that the performance of VB also seems to degrade slightly for $k >$



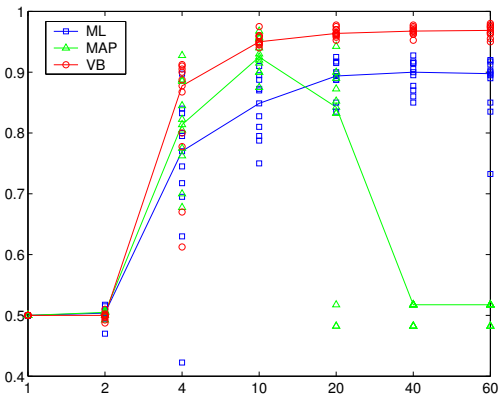
(a) Test log probability per sequence symbol: dependence on n. With $k = 40,\ u_0 = 2$.



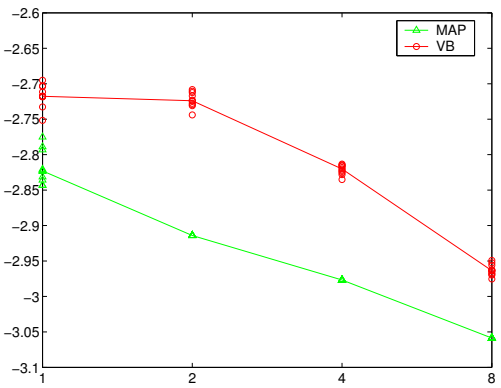
(b) Test discrimination rate dependence on n. With $k = 40,\ u_0 = 2$.



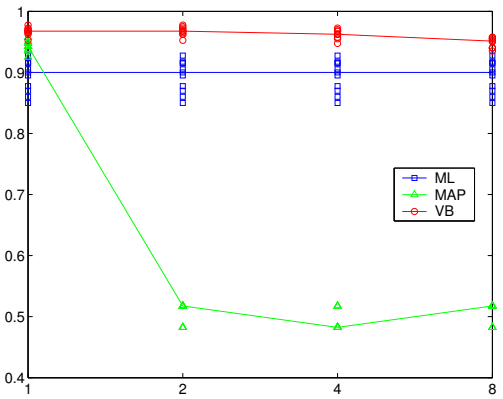
(c) Test log probability per sequence symbol: dependence on k. With $n = 2,\ u_0 = 2$.



(d) Test discrimination rate dependence on k. With $n = 2,\ u_0 = 2$.



(e) Test log probability per sequence symbol: dependence on u_0. With $n = 2,\ k = 40$.



(f) Test discrimination rate dependence on u_0. With $n = 2,\ k = 40$.

Figure 3.4: Variations in performance in terms of test data log predictive probability and discrimination rates of ML, MAP, and VB algorithms for training hidden Markov models. Note that the reported predictive probabilities are per test sequence symbol. Refer to text for details.

20. This decrease in performance with high k corresponds to a solution with the transition matrix containing approximately equal probabilities in all entries, which shows that MAP is over-regularising the parameters, and that VB does so also but not so severely. As the strength of the hyperparameter u_0 increases, we see that both the MAP and VB test log likelihoods decrease, suggesting that $u_0 \leq 2$ is suitable. Indeed at $u_0 = 2$, the MAP algorithm suffers considerably in terms of discrimination performance, despite the VB algorithm maintaining high success rates.

There were some other general trends which were not reported in these plots. For example, in **(b)** the onset of the rise in discrimination performance of MAP away from .5 occurs further to the right as the strength u_0 is increased. That is to say the over-regularising problem is worse with a stronger prior, which makes sense. Similarly, on increasing u_0, the point at which MAP begins to decrease in **(c,d)** moves to the left. We should note also that on increasing u_0, the test log probability for VB **(c)** begins to decrease earlier in terms of k.

The test sentences on which the algorithms tend to make mistakes are the shorter, and more reversible sentences, as to be expected. Some examples are: 'alas ', 'pat ', 'oh ', and 'oh dear '.

3.6 Discussion

In this chapter we have presented the ML, MAP and VB methods for learning HMMs from data. The ML method suffers because it does not take into account model complexity and so can overfit the data. The MAP method performs poorly both from over-regularisation and also because it entertains a single point-parameter model instead of integrating over an ensemble. We have seen that the VB algorithm outperforms both ML and MAP with respect to the likelihood of test sequences and in discrimination tasks between forwards and reverse English sentences. Note however, that a fairer comparison of MAP with VB would include allowing each method to use cross-validation to find the best setting of their hyperparameters. This is fairer because the effective value of u_0 used in the MAP algorithm changes depending on the basis used for the optimisation.

In the experiments the automatic pruning of hidden states by the VB method has been welcomed as a means of inferring useful structure in the data. However, in an ideal Bayesian application one would prefer all states of the model to be active, but with potentially larger uncertainties in the posterior distributions of their transition and emission parameters; in this way all parameters of the model are used for predictions. This point is raised in MacKay (2001) where it is shown that the VB method can inappropriately overprune degrees of freedom in a mixture of Gaussians.

Unless we really believe that our data was generated from an HMM with a finite number of states, then there are powerful arguments for the Bayesian modeller to employ as complex a

model as is computationally feasible, even for small data sets (Neal, 1996, p. 9). In fact, for Dirichlet-distributed parameters, it is possible to mathematically represent the limit of an infinite number of parameter dimensions, with finite resources. This result has been exploited for mixture models (Neal, 1998b), Gaussian mixture models (Rasmussen, 2000), and more recently has been applied to HMMs (Beal et al., 2002). In all these models, sampling is used for inferring distributions over the parameters of a countably infinite number of mixture components (or hidden states). An area of future work is to compare VB HMMs to these infinite HMMs.

Chapter 4

Variational Bayesian Mixtures of Factor Analysers

4.1 Introduction

This chapter is concerned with learning good representations of high dimensional data, with the goal being to perform well in density estimation and pattern classification tasks. The work described here builds on work in Ghahramani and Beal (2000), which first introduced the variational method for Bayesian learning of a mixtures of factor analysers model, resulting in a tractable means of integrating over all the parameters in order to avoid overfitting.

In the following subsections we introduce factor analysis (FA), and the mixtures of factor analysers (MFA) model which can be thought of as a mixture of reduced-parameter Gaussians. In section 4.2 we explain why an exact Bayesian treatment of MFAs is intractable, and present a variational Bayesian algorithm for learning. We show how to learn distributions over the parameters of the MFA model, how to optimise its hyperparameters, and how to automatically determine the dimensionality of each analyser using automatic relevance determination (ARD) methods. In section 4.3 we propose heuristics for efficiently exploring the (one-dimensional) space of the number of components in the mixture, and in section 4.5 we present synthetic experiments showing that the model can simultaneously learn the number of analysers and their intrinsic dimensionalities. In section 4.6 we apply the VBMFA to the real-world task of classifying digits, and show improved performance over a BIC-penalised maximum likelihood approach. In section 4.7 we examine the tightness of the VB lower bound using importance sampling estimates of the exact marginal likelihood, using as importance distributions the posteriors from the VB optimisation. We also investigate the effectiveness of using heavy-tailed and mixture distributions in this procedure. We then conclude in section 4.8 with a brief outlook on recent research progress in this area.

4.1.1 Dimensionality reduction using factor analysis

Factor analysis is a method for modelling correlations in multidimensional data, by expressing the correlations in a lower-dimensional, oriented subspace. Let the data set be $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. The model assumes that each p-dimensional data vector $\mathbf{y}_i$ was generated by first linearly transforming a $k < p$ dimensional vector of unobserved independent zero-mean unit-variance Gaussian sources (factors), $\mathbf{x}_i = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ik}]$, translating by a fixed amount $\boldsymbol{\mu}$ in the data space, followed by adding p-dimensional zero-mean Gaussian noise, $\mathbf{n}_i$, with diagonal covariance matrix Ψ (whose entries are sometimes referred to as the *uniquenesses*). Expressed mathematically, we have

$$\mathbf{y}_i = \Lambda \mathbf{x}_i + \boldsymbol{\mu} + \mathbf{n}_i \tag{4.1}$$

$$\mathbf{x}_i \sim \mathrm{N}(\mathbf{0}, \mathrm{I}), \qquad \mathbf{n}_i \sim \mathrm{N}(\mathbf{0}, \Psi)\ , \tag{4.2}$$

where Λ $(p \times k)$ is the linear transformation known as the *factor loading* matrix, and $\boldsymbol{\mu}$ is the mean of the analyser. Integrating out $\mathbf{x}_i$ and $\mathbf{n}_i$, it is simple to show that the marginal density of $\mathbf{y}_i$ is Gaussian about the displacement $\boldsymbol{\mu}$,

$$p(\mathbf{y}_i \,|\, \Lambda, \boldsymbol{\mu}, \Psi) = \int d\mathbf{x}_i \; p(\mathbf{x}_i) p(\mathbf{y}_i \,|\, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) = \mathrm{N}(\mathbf{y}_i \,|\, \boldsymbol{\mu}, \Lambda\Lambda^\top + \Psi)\ , \tag{4.3}$$

and the probability of an i.i.d. data set $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^n$ is given by

$$p(\mathbf{y} \,|\, \Lambda, \boldsymbol{\mu}, \Psi) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \Lambda, \boldsymbol{\mu}, \Psi)\ . \tag{4.4}$$

Given a data set $\mathbf{y}$ having covariance matrix Σ^* and mean $\boldsymbol{\mu}^*$, factor analysis finds the Λ, $\boldsymbol{\mu}$ and Ψ that optimally fit Σ^* in the maximum likelihood sense. Since $k < p$, a factor analyser can be seen as a reduced parameterisation of a full-covariance Gaussian. The (diagonal) entries of the Ψ matrix concentrate on fitting the axis-aligned (sensor) noise in the data, leaving the factor loadings in Λ to model the remaining (assumed-interesting) covariance structure.

The effect of the mean term $\boldsymbol{\mu}$ can be assimilated into the factor loading matrix by augmenting the vector of factors with a constant bias dimension of 1, and adding a corresponding column $\boldsymbol{\mu}$ to the matrix Λ. With these modifications, learning the Λ matrix incorporates learning the mean; in the equations of this chapter we keep the parameters separate, although the implementations consider the combined quantity.

Dimensionality of the latent space, k

A central problem in factor analysis is deciding on the dimensionality of the latent space. If too low a value of k is chosen, then the model has to discard some of the covariance in the data as noise, and if k is given too high a value this causes the model to fit spurious correlations in the data. Later we describe a Bayesian technique to determine this value automatically, but here we first give an understanding for an upper bound on the required value for k, by comparing the number of degrees of freedom in the covariance specification of the data set and the degrees of freedom that the FA parameterisation has in its parameters. We need to distinguish between the number of parameters and the degrees of freedom, which is really a measure of how many independent directions in parameter space there are that affect the generative probability of the data. The number of degrees of freedom in a factor analyser with latent space dimensionality k cannot exceed the number of degrees of freedom of a full covariance matrix, $\frac{1}{2}p(p+1)$, nor can it exceed the degrees of freedom offered by the parameterisation of the analyser, which is given by $d(k)$,

$$d(k) = kp + p - \frac{1}{2}k(k-1)\ . \tag{4.5}$$

The first two terms on the right hand side are the degrees of freedom in the Λ and Ψ matrices respectively, and the last term is the degrees of freedom in a $(k \times k)$ orthonormal matrix. This last term needs to be subtracted because it represents a redundancy in the factor analysis parameterisation, namely that an arbitrary rotation or reflection of the latent vector space leaves the covariance model of the data unchanged:

$$\text{under } \Lambda \to \Lambda U, \qquad \Lambda\Lambda^\top + \Psi \to \Lambda U(\Lambda U)^\top + \Psi \tag{4.6}$$

$$= \Lambda U U^\top \Lambda^\top + \Psi \tag{4.7}$$

$$= \Lambda\Lambda^\top + \Psi\ . \tag{4.8}$$

That is to say we must subtract the degrees of freedom from degeneracies in Λ associated with arbitrary arrangements of the (a priori identical) hidden factors $\{\mathbf{x}_{ij}\}_{j=1}^{k}$. Since a p-dimensional covariance matrix contains $p(p+1)/2$ pieces of information, in order to be able to perfectly capture the covariance structure of the data the number of degrees of freedom in the analyser (4.5) would have to exceed this. This inequality is a simple quadratic problem, for $k \leq p$

$$kp + p - \frac{1}{2}k(k-1) \geq \frac{1}{2}p(p+1) \tag{4.9}$$

whose solution is given by

$$k_{\text{max}} = \left\lceil p + \frac{1}{2}\left[1 - \sqrt{1+8p}\right]\right\rceil\ . \tag{4.10}$$

We might be tempted to conclude that we only need k_{max} factors to model an arbitrary covariance in p dimensions. However this neglects the constraint that all the diagonal elements of Ψ have

to be positive. We conjecture that because of this constraint the number of factors needed to model a full covariance matrix is $p - 1$. This implies that for high dimensional data, if we want to be able to model a full covariance structure, we cannot expect to be able to reduce the number of parameters by that much at all using factor analysis. Fortunately, for many real data sets we have good reason to believe that, at least locally, the data lies on a low dimensional manifold which we can capture with only a few factors. The fact that this is a good approximation only locally, when the manifold may be globally non-linear, is the motivation for mixture models, discussed next.

4.1.2 Mixture models for manifold learning

It is often the case that apparently high dimensional data in fact lies, to a good approximation, on a low dimensional manifold. For example, consider the data set consisting of many different images of the same digit, given in terms of the pixel intensities. This data has as many dimensions as there are pixels in each image. To explain this data we could first specify a mean digit image, which is a point in this high dimensional space representing a set of pixel intensities, and then specify a small number of transformations away from that digit that would cover small variations in style or perhaps intensity. In factor analysis, each factor dictates the amount of each linear transformation on the pixel intensities. However, with factor analysis we are restricted to linear transformations, and so any one analyser can only explain well a small region of the manifold in which it is locally linear, even though the manifold is globally non-linear.

One way to overcome this is to use mixture models to tile the data manifold. A mixture of factor analysers models the density for a data point $\mathbf{y}_i$ as a weighted average of factor analyser densities

$$p(\mathbf{y}_i \,|\, \boldsymbol{\pi}, \Lambda, \boldsymbol{\mu}, \Psi) = \sum_{s_i=1}^{S} p(s_i \,|\, \boldsymbol{\pi}) p(\mathbf{y}_i \,|\, s_i, \Lambda, \boldsymbol{\mu}, \Psi) \; . \tag{4.11}$$

Here, S is the number of mixture components in the model, $\boldsymbol{\pi}$ is the vector of mixing proportions, s_i is a discrete indicator variable for the mixture component chosen to model data point i, $\Lambda = \{\Lambda^s\}_{s=1}^S$ is a set of factor loadings with Λ^s being the factor loading matrix for analyser s, and $\boldsymbol{\mu} = \{\boldsymbol{\mu}^s\}_{s=1}^S$ is the set of analyser means. The last term in the above probability is just the single analyser density, given in equation (4.3). The directed acyclic graph for this model is depicted in figure 4.1, which uses the *plate* notation to denote repetitions over a data set of size n. Note that there are different indicator variables s_i and latent space variables $\mathbf{x}_i$ for each plate.

By exploiting the factor analysis parameterisation of covariance matrices, a mixture of factor analysers can be used to fit a mixture of Gaussians to correlated high dimensional data without requiring $O(p^2)$ parameters, or undesirable compromises such as axis-aligned covariance matrices. In an MFA each Gaussian cluster has intrinsic dimensionality k, or k_s if the dimensions are allowed to vary across mixture components. Consequently, the mixture of factor analysers

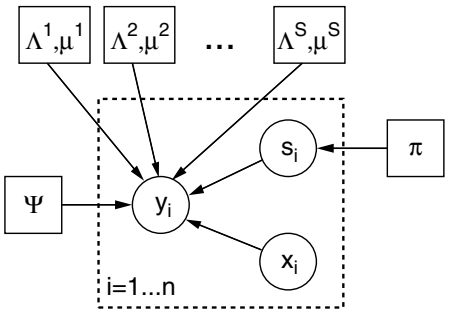


Figure 4.1: Generative model for Maximum Likelihood MFA. Circles denote random variables, solid rectangles parameters, and the dashed rectangle the plate (repetitions) over the data.

simultaneously addresses the problems of clustering and local dimensionality reduction. When Ψ is a multiple of the identity the model becomes a mixture of probabilistic PCAs (pPCA). Tractable maximum likelihood (ML) procedures for fitting MFA and pPCA models can be derived from the Expectation Maximisation algorithm, see for example Ghahramani and Hinton (1996b); Tipping and Bishop (1999). Factor analysis and its relationship to PCA and mixture models is reviewed in Roweis and Ghahramani (1999).

4.2 Bayesian Mixture of Factor Analysers

The maximum likelihood approach to fitting an MFA has several drawbacks. The EM algorithm can easily get caught in local maxima, and often many restarts are required before a good maximum is reached. Technically speaking the log likelihoods in equations (4.3) and (4.11) are not bounded from above, unless constraints are placed on the variances of the components of the mixture. In practice this means that the covariance matrix $\Lambda^s \Lambda^{s\top} + \Psi$ can become singular if a particular factor analyser models fewer points than the degrees of freedom in its covariance matrix. Most importantly, the maximum likelihood approach for fitting MFA models has the severe drawback that it fails to take into account model complexity. For example the likelihood can be increased by adding more analyser components to the mixture, up to the extreme where each component models a single data point, and it can be further increased by supplying more factors in each of the analysers.

A Bayesian approach overcomes these problems by treating the parameters of the model as unknown quantities and averaging over the ensemble of models they produce. Defining $\boldsymbol{\theta} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\pi}, \Psi)$, we write the probability of the data averaged over a prior for parameters:

$$p(\mathbf{y}) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \tag{4.12}$$

$$= \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) \tag{4.13}$$

$$= \int d\boldsymbol{\pi}\; p(\boldsymbol{\pi}) \int d\Lambda\; p(\Lambda) \int d\boldsymbol{\mu}\; p(\boldsymbol{\mu}) \cdot \prod_{i=1}^{n} \left[\sum_{s_i=1}^{S} p(s_i \,|\, \boldsymbol{\pi}) \int d\mathbf{x}_i\; p(\mathbf{x}_i) p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) \right] . \tag{4.14}$$

Equation (4.14) is the marginal likelihood of a dataset (called the marginal probability of the data set by some researchers to avoid confusion with the likelihood of the *parameters*). By integrating out all those parameters whose number increase as the model complexity grows, we effectively penalise models with more degrees of freedom, since they can a priori model a larger range of data sets. By model complexity, we mean the number of components and the dimensionality of each component. Integrating out the parameters naturally embodies the principle of Occam's razor (MacKay, 1992; Jefferys and Berger, 1992). As a result no parameters are ever *fit* to the data, but rather their posterior *distributions* are inferred and used to make predictions about new data. For this chapter, we have chosen not to integrate over Ψ, although this could also be done (see, for example, chapter 5). Since the number of degrees of freedom in Ψ does not grow with the number of analysers or their dimensions, we treat it as a hyperparameter and optimise it, even though this might result in some small degree of overfitting.

4.2.1 Parameter priors for MFA

While arbitrary choices can be made for the priors in (4.14), choosing priors that are conjugate to the likelihood terms greatly simplifies inference and interpretability. Therefore we choose a symmetric Dirichlet prior for the mixing proportion $\boldsymbol{\pi}$, with strength α^*,

$$p(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) = \mathrm{Dir}(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) \,, \qquad \text{such that} \quad \mathbf{m}^* = \left[\frac{1}{S}, \ldots, \frac{1}{S} \right] . \tag{4.15}$$

In this way the prior has a single hyperparameter, its strength α^*, regardless of the dimensionality of $\boldsymbol{\pi}$. This hyperparameter is a measure of how we expect the mixing proportions to deviate from being equal. One could imagine schemes in which we have non-symmetric prior mixing proportion; an example could be making the hyperparameter in the Dirichlet prior an exponentially decaying vector with a single decay rate hyperparameter, which induces a natural ordering in the mixture components and so removes some identifiability problems. Nevertheless for our

purposes a symmetric prior suffices, and expresses the notion that each component has equal a priori chance of being used to generate each data point.

For the entries of the factor loading matrices, $\{\Lambda^s\}_{s=1}^S$, we choose a hierarchical prior in order to perform automatic relevance determination (ARD). Each column of each factor loading matrix has a Gaussian prior with mean zero and a different precision parameter (drawn from a gamma distribution with fixed hyperparameters, see equation (4.18) below):

$$p(\Lambda \,|\, \boldsymbol{\nu}) = \prod_{s=1}^{S} p(\Lambda^s \,|\, \boldsymbol{\nu}^s) = \prod_{s=1}^{S} \prod_{j=1}^{k_s} p(\Lambda^s_{\cdot j} \,|\, \nu^s_j) = \prod_{s=1}^{S} \prod_{j=1}^{k_s} \mathrm{N}(\Lambda^s_{\cdot j} \,|\, \mathbf{0}, \mathrm{I}/\nu^s_j) \,, \tag{4.16}$$

where $\Lambda^s_{\cdot j}$ denotes the vector of entries in the jth column of the sth analyser in the mixture, and ν^s_j is the same scalar precision for each entry in the corresponding column. The role of these precision hyperparameters is explained in section 4.2.2. Note that because the spherical Gaussian prior is separable into each of its p dimensions, the prior can equivalently be thought of as a Gaussian with axis-aligned elliptical covariance on each row of each analyser:

$$p(\Lambda \,|\, \boldsymbol{\nu}) = \prod_{s=1}^{S} \prod_{q=1}^{p} p(\Lambda^s_{q\cdot} \,|\, \boldsymbol{\nu}^s) = \prod_{s=1}^{S} \prod_{q=1}^{p} \mathrm{N}(\Lambda^s_{q\cdot} \,|\, \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\nu}^s)^{-1}) \,, \tag{4.17}$$

where here $\Lambda^s_{q\cdot}$ is used to denote the qth row of the sth analyser. It will turn out to be simpler to have the prior in this form conceptually for learning, since the likelihood terms for Λ factor across its rows.

Since the number of hyperparameters in $\boldsymbol{\nu} = \{\{\nu^s_j\}_{j=1}^{k_s}\}_{s=1}^S$ increases with the number of analysers and also with the dimensionality of each analyser, we place a hyperprior on every element of each $\boldsymbol{\nu}^s$ precision vector, as follows:

$$p(\boldsymbol{\nu} \,|\, a^*, b^*) = \prod_{s=1}^{S} p(\boldsymbol{\nu}^s \,|\, a^*, b^*) = \prod_{s=1}^{S} \prod_{j=1}^{k_s} p(\nu^s_j \,|\, a^*, b^*) = \prod_{s=1}^{S} \prod_{j=1}^{k_s} \mathrm{Ga}(\nu^s_j \,|\, a^*, b^*) \,, \tag{4.18}$$

where a^* and b^* are shape and inverse-scale hyperhyperparameters for a gamma distribution (see appendix A for a definition and properties of the gamma distribution). Note that the same hyperprior is used for every element in $\boldsymbol{\nu}$. As a point of interest, combining the priors for Λ and $\boldsymbol{\nu}$, and integrating out $\boldsymbol{\nu}$, we find that the marginal prior over each Λ^s is Student-t distributed. We will not need to make use of this result right here, but will return to it in section 4.7.1.

Lastly, the means of each analyser in the mixture need to be integrated out. A Gaussian prior with mean $\boldsymbol{\mu}^*$ and axis-aligned precision $\mathrm{diag}\,(\boldsymbol{\nu}^*)$ is placed on each mean $\boldsymbol{\mu}^s$. Note that these

hyperparameters hold $2p$ degrees of freedom, which is not a function of the size of the model. The prior is the same for every analyser:

$$p(\boldsymbol{\mu} \mid \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) = \prod_{s=1}^{S} p(\boldsymbol{\mu}^s \mid \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) = \prod_{s=1}^{S} \mathrm{N}(\boldsymbol{\mu}^s \mid \boldsymbol{\mu}^*, \mathrm{diag}\,(\boldsymbol{\nu}^*)^{-1}) \tag{4.19}$$

Note that this prior has a different precision for each dimension of the output, whereas the prior over the entries in the factor loading matrix uses the same precision on each row, and is different only for each column of each analyser.

If we are to use the implementational convenience of augmenting the latent space with a constant bias dimension, and adding a further column to each factor loading matrix to represent its mean, then the prior over all the entries in the augmented factor loading matrix no longer factorises over rows (4.17) or columns (4.18), but has to be expressed as a product of terms over every entry of the matrix. This point will be made clearer when we derive the posterior distribution over the augmented factor loading matrix.

We use Θ to denote the set of hyperparameters of the model:

$$\Theta = (\alpha^* \mathbf{m}^*, a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \Psi)\ . \tag{4.20}$$

The directed acyclic graph for the generative model for this Bayesian MFA is shown graphically in figure 4.2. Contrasting with the ML graphical model in figure 4.1, we can see that all the model parameters (with the exception of the sensor noise Ψ) have been replaced with uncertain variables, denoted with circles, and now have hyperparameters governing their prior distributions. The generative model for the data remains the same, with the plate over the data denoting i.i.d. instances of the hidden factors $\mathbf{x}_i$, each of which gives rise to an output $\mathbf{y}_i$. We keep the graphical model concise by also using a plate over the S analysers, which clearly shows the role of the hyperpriors.

As an aside, we do not place a prior on the number of components, S. We instead place a symmetric Dirichlet prior over the mixing proportions. Technically, we should include a (square boxed) node S, as the parent of both the plate over analysers and the hyperparameter $\alpha\mathbf{m}$. We have also not placed priors over the number of factors of each analyser, $\{k_s\}_{s=1}^{S}$; this is intentional as there exists an explicit penalty for using more dimensions — the extra entries in factor loading matrix Λ^s need to be explained under a hyperprior distribution (4.16) which is governed by a new hyperparameter $\boldsymbol{\nu}^s$, which itself has to be explained under the hyperhyperprior $p(\boldsymbol{\nu}^s \mid a, b)$ of equation (4.18).

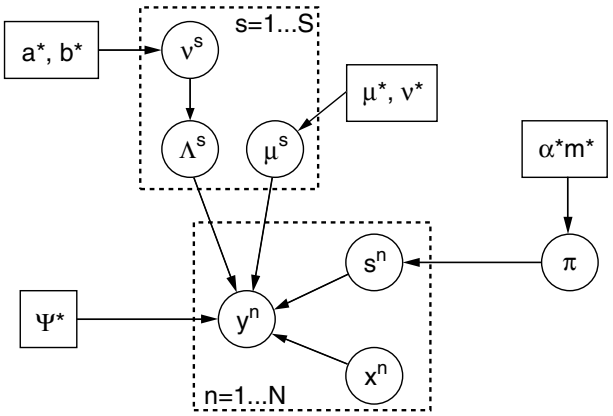


Figure 4.2: A Bayesian formulation for MFA. Here the plate notation is used to denote repetitions over data n and over the S analysers in the generative model. Note that all the parameters in the ML formulation, except Ψ, have now become uncertain random variables in the Bayesian model (circled nodes in the graph), and are governed by hyperparameters (square boxes). The number of hyperparameters in the model is constant and is not a function of the number of analysers or their dimensionalities.

4.2.2 Inferring dimensionality using ARD

Each factor analyser s in the MFA models its local data as a linear projection of k_s-dimensional spherical Gaussian noise into the p-dimensional space. If a maximum dimensionality k_{max} is set, then there exist $k_{\text{max}} \times \cdots \times k_{\text{max}} = (k_{max})^S$ possible subspace configurations amongst the S analysers. Thus determining the optimal configuration is exponentially intractable if a discrete search is employed over analyser dimensionalities. Automatic relevance determination (ARD) solves this discrete search problem with the use of continuous variables that allow a *soft blend* of dimensionalities. Each factor analyser's dimensionality is set to k_{max} and we use priors that discourage large factor loadings. The width of each prior is controlled by a hyperparameter (explained below), and the result of learning with this method is that only those hidden factor dimensions that are required remain active after learning — the remaining dimensions are effectively 'switched off'. This general method was proposed by MacKay and Neal (see MacKay, 1996, for example), and was used in Bishop (1999) for Bayesian PCA, and is closely related to the method given in Neal (1998a) for determining the relevance of inputs to a neural network.

Considering for the moment a single factor analyser. The ARD scheme uses a Gaussian prior with a zero mean for the entries of the factor loading matrix, as shown in (4.16), given again here:

$$p(\Lambda^s \,|\, \boldsymbol{\nu}^s) = \prod_{j=1}^{k_{\text{max}}} p(\Lambda^s_{\cdot j} \,|\, \nu^s_j) = \prod_{j=1}^{k_{\text{max}}} \mathrm{N}(\Lambda^s_{\cdot j} \,|\, \mathbf{0}, \mathrm{I}/\nu^s_j) \;, \tag{4.21}$$

where $\boldsymbol{\nu}^s = \{\nu^s_1, \ldots, \nu^s_{k_{\text{max}}}\}$ are the precisions on the columns of Λ^s, which themselves are denoted by $\{\Lambda_{\cdot 1}, \ldots, \Lambda_{\cdot k_{\text{max}}}\}$. This zero-mean prior couples within-column entries in Λ^s, favouring lower magnitude values.

If we apply this prior to each analyser in the mixture, each column of each factor loading matrix is then governed by a separate ν_l^s parameter. If one of these precisions $\boldsymbol{\nu}_l^s \to \infty$ then the outgoing weights (column l entries in Λ^s) for the lth factor in the sth analyser will have to be very close to zero in order to maintain a high likelihood under this prior, and this in turn leads the analyser to ignore this factor, and thus allows the model to reduce the intrinsic dimensionality of $\mathbf{x}$ in the *locale of that analyser* if the data does not warrant this added dimension. We have not yet explained how some of these precisions come to tend to infinity; this will be made clearer in the derivations of the learning rules in section 4.2.5.

The fully Bayesian application requires that we integrate out all parameters that scale with the number of analyser components and their dimensions; for this reason we use the conjugate prior for a precision variable, a gamma distribution with shape a^* and inverse scale b^*, to integrate over the ARD hyperparameters. Since we are integrating over the hyperparameters, it now makes sense to consider removing a redundant factor loading when the *posterior distribution* over the hyperparameter $\boldsymbol{\nu}_l^s$ has most of its mass near infinity. In practice we take the mean of this posterior to be indicative of its position, and perform removal when it becomes very large. This reduces the coding cost of the parameters, and as a redundant factor is not used to model the data, this must increase the marginal likelihood $p(\mathbf{y})$. We can be harsher still, and prematurely remove those factors which have $\boldsymbol{\nu}_l^s$ escaping to infinity, provided the resulting marginal likelihood is better (we do not implement this scheme in our experiments).

4.2.3 Variational Bayesian derivation

Now that we have priors over the parameters of our model, we can set about computing the marginal likelihood of data. But unfortunately, computing the marginal likelihood in equation (4.14) is intractable because integrating over the parameters of the model induces correlations in the posterior distributions between the hidden variables in all the n plates. As mentioned in section 1.3, there are several methods that are used to approximate such integrals, for example MCMC sampling techniques, the Laplace approximation, and the asymptotic BIC criterion.

For MFA and similar models, MCMC methods for Bayesian approaches have only recently been applied by Fokoué and Titterington (2003), with searches over model complexity in terms of both the number of components and their dimensionalities carried out by reversible jump techniques (Green, 1995). In related models, Laplace and asymptotic approximations have been used to approximate Bayesian integration in mixtures of Gaussians (Roberts et al., 1998). Here our focus is on analytically tractable approximations based on lower bounding the marginal likelihood.

We begin with the log marginal likelihood of the data and first construct a lower bound using a variational distribution over the parameters $\{\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}\}$, and then perform a similar lower

bounding using a variational distribution for the hidden variables $\{s_i, \mathbf{x}_i\}_{i=1}^n$. As a point of nomenclature, just as we have been using the same notation $p(\cdot)$ for every prior distribution, even though they may be Gaussian, gamma, Dirichlet etc., in what follows we also use the same $q(\cdot)$ to denote different variational distributions for different parameters. The form of $q(\cdot)$ will be clear from its arguments.

Combining (4.14) with the priors discussed above including the hierarchical prior on Λ, we obtain the log marginal likelihood of the data, denoted $\mathcal{L}$,

$$\mathcal{L} \equiv \ln p(\mathbf{y}) = \ln \left(\int d\boldsymbol{\pi}\; p(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) \int d\boldsymbol{\nu}\; p(\boldsymbol{\nu} \,|\, a^*, b^*) \int d\Lambda\; p(\Lambda \,|\, \boldsymbol{\nu}) \int d\boldsymbol{\mu}\; p(\boldsymbol{\mu} \,|\, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) \cdot \prod_{i=1}^{n} \left[\sum_{s_i=1}^{S} p(s_i \,|\, \boldsymbol{\pi}) \int d\mathbf{x}_i\; p(\mathbf{x}_i) p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) \right] \right) . \tag{4.22}$$

The marginal likelihood $\mathcal{L}$ is in fact a function of the hyperparameters $(\alpha^* \mathbf{m}^*, a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)$, and the sensor noise Ψ; this dependence is left implicit in this derivation. We introduce an arbitrary distribution $q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu})$ to lower bound (4.22), followed by a second set of distributions $\{q(s_i, \mathbf{x}_i)\}_{i=1}^n$ to further lower bound the bound,

$$\begin{aligned} \mathcal{L} &\geq \int d\boldsymbol{\pi}\, d\boldsymbol{\nu}\, d\Lambda\, d\boldsymbol{\mu}\; q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \left(\ln \frac{p(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) p(\boldsymbol{\nu} \,|\, a^*, b^*) p(\Lambda \,|\, \boldsymbol{\nu}) p(\boldsymbol{\mu} \,|\, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)}{q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu})} \right. \\ &\qquad \left. + \sum_{i=1}^{n} \ln \left[\sum_{s_i=1}^{S} p(s_i \,|\, \boldsymbol{\pi}) \int d\mathbf{x}_i\; p(\mathbf{x}_i) p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) \right] \right) \qquad (4.23) \\ &\geq \int d\boldsymbol{\pi}\, d\boldsymbol{\nu}\, d\Lambda\, d\boldsymbol{\mu}\; q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \left(\ln \frac{p(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) p(\boldsymbol{\nu} \,|\, a^*, b^*) p(\Lambda \,|\, \boldsymbol{\nu}) p(\boldsymbol{\mu} \,|\, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)}{q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu})} \right. \\ &\qquad \left. + \sum_{i=1}^{n} \left[\sum_{s_i=1}^{S} \int d\mathbf{x}_i\; q(s_i, \mathbf{x}_i) \left(\ln \frac{p(s_i \,|\, \boldsymbol{\pi}) p(\mathbf{x}_i)}{q(s_i, \mathbf{x}_i)} + \ln p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) \right) \right] \right) . \qquad (4.24) \end{aligned}$$

In the first inequality, the term on the second line is simply the log likelihood of $\mathbf{y}_i$ for a fixed setting of parameters, which is then further lower bounded in the second inequality using a set of distributions over the hidden variables $\{q(s_i, \mathbf{x}_i)\}_{i=1}^n$. These distributions are *independent* of the settings of the parameters $\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda$, and $\boldsymbol{\mu}$, and they correspond to the standard variational approximation of the factorisation between the parameters and the hidden variables:

$$p(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}, \{s_i, \mathbf{x}_i\}_{i=1}^n \,|\, \mathbf{y}) \approx q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \prod_{i=1}^{n} q(s_i, \mathbf{x}_i) . \tag{4.25}$$

The distribution of hidden variables factorises across the plates because both the generative model is i.i.d. *and* we have made the approximation that the parameters and hidden variables are independent (see proof of theorem 2.1 in section 2.3.1). Here we use a further variational ap-

proximation amongst the parameters, which can be explained by equating the functional derivatives of equation (4.24) with respect to $q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu})$ to zero. One finds that

$$q(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \propto p(\boldsymbol{\pi} \mid \alpha^* \mathbf{m}^*) p(\boldsymbol{\nu} \mid a^*, b^*) p(\Lambda \mid \boldsymbol{\nu}) p(\boldsymbol{\mu} \mid \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) \cdot \exp\left[\sum_{i=1}^{n} \sum_{s_i=1}^{S} \langle \ln p(s_i \mid \boldsymbol{\pi}) p(\mathbf{y}_i \mid s_i, \mathbf{x}_i, \Lambda, \boldsymbol{\mu}, \Psi) \rangle_{q(s_i, \mathbf{x}_i)} \right] \tag{4.26}$$

$$= q(\boldsymbol{\pi}) q(\boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \tag{4.27}$$

$$\approx q(\boldsymbol{\pi}) q(\boldsymbol{\nu}) q(\Lambda, \boldsymbol{\mu}) \; . \tag{4.28}$$

In the second line, the approximate posterior factorises exactly into a contribution from the mixing proportions and the remaining parameters. Unfortunately it is not easy to take expectations with respect to the joint distribution over Λ and its parent parameter $\boldsymbol{\nu}$, and therefore we make the second variational approximation in the last line, equation (4.28). The very last term $q(\Lambda, \boldsymbol{\mu})$ turns out to be jointly Gaussian, and so is of tractable form.

We should note that except for the initial factorisation between the hidden variables and the parameters, the factorisation $q(\boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \approx q(\boldsymbol{\nu}) q(\Lambda, \boldsymbol{\mu})$ is the only other approximating factorisation we make; all other factorisations fall out naturally from the conditional independencies in the model. Note that the complete-data likelihood for mixtures of factor analysers is in the exponential family, even after the inclusion of the precision parameters $\boldsymbol{\nu}$. We could therefore apply the results of section 2.4, but this would entail finding expectations over gamma-Gaussian distributions jointly over $\boldsymbol{\nu}$ and Λ. Although it is possible to take these expectations, for convenience we choose a separable variational posterior on $\boldsymbol{\nu}$ and Λ.

From this point on we assimilate each analyser's mean position $\boldsymbol{\mu}^s$ into its factor loading matrix, in order to keep the presentation concise. The derivations use $\tilde{\Lambda}$ to denote the concatenated result $[\Lambda \; \boldsymbol{\mu}]$. Therefore the prior over the entire factor loadings $\tilde{\Lambda}$ is now a function of the precision parameters $\{\boldsymbol{\nu}^s\}_{s=1}^S$ (which themselves have hyperparameters a, b) and the hyperparameters $\boldsymbol{\mu}^*, \boldsymbol{\nu}^*$. Also, the variational posterior $q(\Lambda, \boldsymbol{\mu})$ becomes $q(\tilde{\Lambda})$.

Substituting the factorised approximations (4.25) and (4.28) into the lower bound (4.24) results in the following lower bound for the marginal likelihood,

$$\begin{aligned}
\mathcal{L} \geq & \int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln \frac{p(\boldsymbol{\pi} \,|\, \alpha^*, \mathbf{m}^*)}{q(\boldsymbol{\pi})} \\
& + \sum_{s=1}^{S} \int d\boldsymbol{\nu}^s\; q(\boldsymbol{\nu}^s) \left[\ln \frac{p(\boldsymbol{\nu}^s \,|\, a^*, b^*)}{q(\boldsymbol{\nu}^s)} + \int d\tilde{\Lambda}^s\; q(\tilde{\Lambda}^s) \ln \frac{p(\tilde{\Lambda}^s \,|\, \boldsymbol{\nu}^s, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)}{q(\tilde{\Lambda}^s)} \right] \\
& + \sum_{i=1}^{n} \sum_{s_i=1}^{S} q(s_i) \left[\int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln \frac{p(s_i \,|\, \boldsymbol{\pi})}{q(s_i)} + \int d\mathbf{x}_i\; q(\mathbf{x}_i \,|\, s_i) \ln \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i \,|\, s_i)} \right. \\
& \qquad \left. + \int d\tilde{\Lambda}\; q(\tilde{\Lambda}) \int d\mathbf{x}_i\; q(\mathbf{x}_i \,|\, s_i) \ln p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \tilde{\Lambda}, \Psi) \right] \qquad (4.29) \\
\equiv & \mathcal{F}(q(\boldsymbol{\pi}), \{q(\boldsymbol{\nu}^s), q(\tilde{\Lambda}^s), \{q(s_i), q(\mathbf{x}_i \,|\, s_i)\}_{i=1}^{n}\}_{s=1}^{S}, \alpha^* \mathbf{m}^*, a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \Psi, \mathbf{y}) \qquad (4.30) \\
= & \mathcal{F}(q(\boldsymbol{\theta}), q(\mathbf{s}, \mathbf{x}), \Theta)\;. \qquad (4.31)
\end{aligned}$$

Thus the lower bound is a functional of the variational posterior distributions over the parameters, collectively denoted $q(\boldsymbol{\theta})$, a functional of the variational posterior distribution over the hidden variables of every data point, collectively denoted $q(\mathbf{s}, \mathbf{x})$, and also a function of the set of hyperparameters in the model Θ, as given in (4.20). In the last line above, we have dropped $\mathbf{y}$ as an argument for the lower bound since it is fixed. The full variational posterior is

$$p(\boldsymbol{\pi}, \boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}, \mathbf{s}, \mathbf{x} \,|\, \mathbf{y}) \approx q(\boldsymbol{\pi}) \prod_{s=1}^{S} q(\boldsymbol{\nu}^s) q(\tilde{\Lambda}^s) \cdot \prod_{i=1}^{n} \prod_{s_i=1}^{S} q(s_i) q(\mathbf{x}_i \,|\, s_i)\;. \qquad (4.32)$$

Note that if we had not made the factorisation $q(\boldsymbol{\nu}, \Lambda, \boldsymbol{\mu}) \approx q(\boldsymbol{\nu}) q(\Lambda, \boldsymbol{\mu})$, then the last term in $\mathcal{F}$ would have required averages not over $q(\tilde{\Lambda})$, but also over the combined $q(\boldsymbol{\nu}, \tilde{\Lambda})$, which would have become fairly cumbersome, although not intractable.

Decomposition of $\mathcal{F}$

The goal of learning is then to maximise $\mathcal{F}$, thus increasing the lower bound on $\mathcal{L}$, the exact marginal likelihood. Note that there is an interesting trade-off at play here. The last term in equation (4.29) is the log likelihood of the data set averaged over the uncertainty we have in the hidden variables and parameters. We can increase this term by altering Ψ and the variational posterior distributions $q(\boldsymbol{\theta})$ and $q(\mathbf{s}, \mathbf{x})$ so as to maximise this contribution. However the first three lines of (4.29) contain terms that are negative Kullback-Leibler (KL) divergences between the approximate posteriors over the parameters and the priors we hold on them. So to increase the lower bound on the marginal likelihood (which does not necessarily imply that the marginal likelihood itself increases, since the bound is not tight), we should also consider moving our approximate posteriors towards the priors, thus decreasing the respective KL divergences. In this manner $\mathcal{F}$ elegantly incorporates the trade-off between modelling the data and remaining

consistent with our prior beliefs. Indeed if there were no contributions from the data (i.e. the last term in equation (4.29) were zero) then the optimal approximate posteriors would default to the prior distributions.

At this stage it is worth noting that, with the exception of the first term in equation (4.29), $\mathcal{F}$ can be broken down into contributions from each component of the mixture (indexed by s). This fact that will be useful later when we wish to compare how well each component of the mixture is modelling its respective data.

4.2.4 Optimising the lower bound

To optimise the lower bound we simply take functional derivatives with respect to each of the $q(\cdot)$ distributions and equate these to zero to find the distributions that extremise $\mathcal{F}$ (see chapter 2). Synchronous updating of the variational posteriors is not guaranteed to increase $\mathcal{F}$ but consecutive updating of dependent distributions is. The result is that each update is guaranteed to monotonically and maximally increase $\mathcal{F}$.

The update for the variational posterior over mixing proportions $\boldsymbol{\pi}$:

$$\frac{\partial \mathcal{F}}{\partial q(\boldsymbol{\pi})} = \ln p(\boldsymbol{\pi} \,|\, \alpha^* \mathbf{m}^*) + \sum_{i=1}^{n} \sum_{s_i=1}^{S} q(s_i) \ln p(s_i \,|\, \boldsymbol{\pi}) - \ln q(\boldsymbol{\pi}) + c \tag{4.33}$$

$$= \ln \left[\prod_{s=1}^{S} \pi_s^{\alpha^* m_s^* - 1} \cdot \prod_{i=1}^{n} \prod_{s_i=1}^{S} \pi_{s_i}^{q(s_i)} \right] - \ln q(\boldsymbol{\pi}) + c \tag{4.34}$$

$$= \ln \left[\prod_{s=1}^{S} \pi_s^{\alpha^* m_s^* + \sum_{i=1}^{n} q(s_i) - 1} \right] - \ln q(\boldsymbol{\pi}) + c \tag{4.35}$$

$$\implies q(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi} \,|\, \alpha \mathbf{m}) \,, \tag{4.36}$$

where each element of the variational parameter $\alpha \mathbf{m}$ is given by:

$$\alpha m_s = \alpha^* \mathbf{m}_s^* + \sum_{i=1}^{n} q(s_i) \,, \tag{4.37}$$

which gives $\alpha = \alpha^* + n$. Thus the strength of our posterior belief in the mean $\mathbf{m}$ increases with the amount of data in a very simple fashion. For this update we have taken $m_s^* = 1/S$ from (4.15), and used $\sum_{s=1}^{S} m_s = 1$.

The variational posterior in the precision parameter for the l^{th} column of the s^{th} factor loading matrix Λ^s,

$$\frac{\partial \mathcal{F}}{\partial q(\nu_l^s)} = \ln p(\nu_l^s \mid a^*, b^*) + \int d\Lambda^s \; q(\Lambda^s) \ln p(\Lambda_l^s \mid \nu_l^s) - \ln q(\nu_l^s) + c \tag{4.38}$$

$$= (a^* - 1) \ln \nu_l^s - b^* \nu_l^s + \frac{1}{2} \sum_{q=1}^{p} \left[\ln \nu_l^s - \nu_l^s \left\langle {\Lambda_{ql}^s}^2 \right\rangle_{q(\Lambda^s)} \right] - \ln q(\nu_l^s) + c \,, \tag{4.39}$$

which implies that the precision is Gamma distributed:

$$q(\nu_l^s) = \mathrm{Ga}(\nu_l^s \mid \; a^* + \frac{p}{2}, \; b^* + \frac{1}{2} \sum_{q=1}^{p} \left\langle {\Lambda_{ql}^s}^2 \right\rangle_{q(\Lambda^s)}) = \mathrm{Ga}(\nu_l^s \mid a, b_l^s) \,, \tag{4.40}$$

Note that these updates constitute the key steps for the ARD mechanisms in place over the columns of the factor loading matrices.

The variational posterior over the centres and factor loadings of each analyser is obtained by taking functional derivatives with respect to $q(\tilde{\Lambda})$:

$$\frac{\partial \mathcal{F}}{\partial q(\tilde{\Lambda}^s)} = \int d\boldsymbol{\nu}^s \; q(\boldsymbol{\nu}^s) \ln p(\tilde{\Lambda}^s \mid \boldsymbol{\nu}^s, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)$$

$$+ \sum_{i=1}^{n} q(s_i) \int d\mathbf{x}_i \; q(\mathbf{x}_i \mid s_i) \ln p(\mathbf{y}_i \mid s_i, \mathbf{x}_i, \tilde{\Lambda}^{s_i}, \Psi) - \ln q(\tilde{\Lambda}^s) + c \tag{4.41}$$

$$= \frac{1}{2} \int d\boldsymbol{\nu}^s \; q(\boldsymbol{\nu}^s) \sum_{q=1}^{p} \sum_{l=1}^{k} \left[\ln \nu_l^s - \nu_l^s {\Lambda_{ql}^s}^2 \right]$$

$$+ \frac{1}{2} \sum_{q=1}^{p} \left[\ln \nu_q^* - \nu_q^* \left(\mu_q^s - \mu_q^* \right)^2 \right] \qquad - \ln q(\Lambda^s, \boldsymbol{\mu}^s) + c$$

$$- \frac{1}{2} \sum_{i=1}^{n} q(s_i) \mathrm{tr} \left[\Psi^{-1} \left\langle \left(\mathbf{y}_i - \begin{bmatrix} \Lambda^s & \boldsymbol{\mu}^s \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right) \left(\mathbf{y}_i - \begin{bmatrix} \Lambda^s & \boldsymbol{\mu}^s \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right)^\top \right\rangle_{q(\mathbf{x}_i \mid s_i)} \right] \tag{4.42}$$

where were have moved from the $\tilde{\Lambda}$ notation to using both Λ and $\boldsymbol{\mu}$ separately to express the different prior form separately. In (4.42), there are two summations over the rows of the factor loading matrix, and a trace term, which can also be written as a sum over rows. Therefore the posterior factorises over the rows of $\tilde{\Lambda}^s$,

$$q(\tilde{\Lambda}^s) = \prod_{q=1}^{p} q(\tilde{\Lambda}_{q\cdot}^s) = \prod_{q=1}^{p} \mathrm{N}(\tilde{\Lambda}_{q\cdot}^s \mid \overline{\tilde{\Lambda}}_{q\cdot}^s, \tilde{\Gamma}_q^s) \,, \tag{4.43}$$

where $\tilde{\Lambda}^s_{q\cdot}$ denotes the column vector corresponding to the qth row of $\tilde{\Lambda}^s$, which has $k_s + 1$ dimensions. To clarify the notation, this vector then has mean $\overline{\tilde{\Lambda}}^s_{q\cdot}$, and covariance matrix $\tilde{\Gamma}^s_q$. These variational posterior parameters are given by:

$$\tilde{\Gamma}^s_q = \begin{bmatrix} \Sigma^{q,s\,-1}_{\Lambda\Lambda} & \Sigma^{q,s\,-1}_{\Lambda\mu} \\ \Sigma^{q,s\,-1}_{\mu\Lambda} & \Sigma^{q,s\,-1}_{\mu\mu} \end{bmatrix}^{-1} \qquad \text{of size } (k_s+1)\times(k_s+1) \tag{4.44}$$

$$\overline{\tilde{\Lambda}}^s_{q\cdot} = \begin{bmatrix} \overline{\Lambda}^s_{q\cdot} \\ \overline{\mu}^s_q \end{bmatrix} \qquad \text{of size } (k_s+1)\times 1 \tag{4.45}$$

with

$$\Sigma^{q,s\,-1}_{\Lambda\Lambda} = \operatorname{diag}\left\langle \boldsymbol{\nu}^s \right\rangle_{q(\boldsymbol{\nu}^s)} + \Psi^{-1}_{qq} \sum_{i=1}^{n} q(s_i) \left\langle \mathbf{x}_i \mathbf{x}_i{}^\top \right\rangle_{q(\mathbf{x}_i \mid s_i)} \tag{4.46}$$

$$\Sigma^{q,s-1}_{\mu\mu} = \nu^*_q + \Psi^{-1}_{qq} \sum_{i=1}^{n} q(s_i) \tag{4.47}$$

$$\Sigma^{q,s-1}_{\Lambda\mu} = \Psi^{-1}_{qq} \sum_{i=1}^{n} q(s_i) \left\langle \mathbf{x}_i \right\rangle_{q(\mathbf{x}_i \mid s_i)} = \Sigma^{q,s-1\,\top}_{\mu\Lambda} \tag{4.48}$$

$$\overline{\Lambda}^s_{q\cdot} = \left[\tilde{\Gamma}^s_q\right]_{\Lambda\Lambda} \left(\Psi^{-1}_{qq} \sum_{i=1}^{n} q(s_i) \mathbf{y}_{i,q} \left\langle \mathbf{x}_i \right\rangle_{q(\mathbf{x}_i \mid s_i)} \right) \tag{4.49}$$

$$\overline{\mu}^s_q = \left[\tilde{\Gamma}^s_q\right]_{\mu\mu} \left(\Psi^{-1}_{qq} \sum_{i=1}^{n} q(s_i) \mathbf{y}_{i,q} + \nu^*_q \mu^*_q \right) . \tag{4.50}$$

This somewhat complicated posterior is the result of maintaining a tractable joint over the centres and factor loadings of each analyser. Note that the optimal distribution for each $\tilde{\Lambda}^s$ matrix as a whole now has block diagonal covariance structure: even though each $\tilde{\Lambda}^s$ is a $(p \times (k_s+1))$ matrix, its covariance only has $O(p(k_s+1)^2)$ parameters — a direct consequence of the likelihood factorising over the output dimensions.

The variational posterior for the hidden factors $\mathbf{x}_i$, conditioned on the indicator variable s_i, is given by taking functional derivatives with respect to $q(\mathbf{x}_i \mid s_i)$:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial q(\mathbf{x}_i \mid s_i)} &= q(s_i) \ln p(\mathbf{x}_i) + \int d\tilde{\Lambda}^{s_i}\; q(\tilde{\Lambda}^{s_i}) q(s_i) \ln p(\mathbf{y}_i \mid s_i, \mathbf{x}_i, \tilde{\Lambda}^{s_i}, \Psi) \\ &\quad - q(s_i) \ln q(\mathbf{x}_i \mid s_i) + c \qquad (4.51) \\ &= q(s_i) \left[-\frac{1}{2} \mathbf{x}_i{}^\top I\, \mathbf{x}_i - \frac{1}{2} \mathrm{tr} \left[\Psi^{-1} \left\langle \left(\mathbf{y}_i - \tilde{\Lambda}^{s_i} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right) \left(\mathbf{y}_i - \tilde{\Lambda}^{s_i} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right)^\top \right\rangle_{q(\Lambda^{s_i})} \right] \right. \\ &\quad \left. - \ln q(\mathbf{x}_i \mid s_i) \right] + c \qquad (4.52) \end{aligned}$$

which, regardless of the value of $q(s_i)$, produces the Gaussian posterior in $\mathbf{x}_i$ for each setting of s_i:

$$q(\mathbf{x}_i \,|\, s) = \mathrm{N}(\mathbf{x}_i \,|\, \overline{\mathbf{x}}_i^s, \Sigma^s) \tag{4.53}$$

$$\text{with} \quad [\Sigma^s]^{-1} = \mathrm{I} + \left\langle \Lambda^{s\top} \Psi^{-1} \Lambda^s \right\rangle_{q(\tilde{\Lambda}^s)} \tag{4.54}$$

$$\overline{\mathbf{x}}_i^s = \Sigma^s \left\langle \Lambda^{s\top} \Psi^{-1} (\mathbf{y}_i - \boldsymbol{\mu}^s) \right\rangle_{q(\tilde{\Lambda}^s)} \tag{4.55}$$

Note that the covariance Σ^s of the hidden state is the same for every data point, and is not a function of the posterior responsibility $q(s_i)$, as in ordinary factor analysis — only the *mean* of the posterior over $\mathbf{x}_i$ is a function of the data $\mathbf{y}_i$. Note also that the $\overline{\mathbf{x}}_i^s$ depend indirectly on the $q(s_i)$ through (4.49), which is the update for the factor loadings and centre position of analyser s.

The variational posterior for the set of indicator variables $\mathbf{s} = \{s_i\}_{i=1}^n$ is given by

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial q(s_i)} = &\int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln p(s_i \,|\, \boldsymbol{\pi}) - \int d\mathbf{x}_i\; q(\mathbf{x}_i \,|\, s_i) \ln q(\mathbf{x}_i \,|\, s_i) \\ &+ \int d\tilde{\Lambda}^{s_i}\; q(\tilde{\Lambda}^{s_i}) \int d\mathbf{x}_i\; q(\mathbf{x}_i \,|\, s_i) \ln p(\mathbf{y}_i \,|\, s_i, \mathbf{x}_i, \tilde{\Lambda}^{s_i}, \Psi) - \ln q(s_i) + c \end{aligned} \tag{4.56}$$

which, utilising a result of Dirichlet distributions given in appendix A, yields

$$\begin{aligned} q(s_i) = \frac{1}{\mathcal{Z}_i} \exp \Bigg[&\psi(\alpha m_{s_i}) - \psi(\alpha) + \frac{1}{2} \ln |\Sigma^{s_i}| \\ &- \frac{1}{2} \mathrm{tr} \left[\Psi^{-1} \left\langle \left(\mathbf{y}_i - \tilde{\Lambda}^{s_i} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right) \left(\mathbf{y}_i - \tilde{\Lambda}^{s_i} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right)^\top \right\rangle_{q(\tilde{\Lambda}^{s_i}) q(\mathbf{x}_i \,|\, s_i)} \right] \Bigg], \end{aligned} \tag{4.57}$$

where $\mathcal{Z}_i$ is a normalisation constant for each data point, such that $\sum_{s_i=1}^S q(s_i) = 1$, and $\psi(\cdot)$ is the digamma function.

By examining the dependencies of each variational posterior's update rules on the other distributions, it becomes clear that certain update orderings are more efficient than others in increasing $\mathcal{F}$. For example, the $q(\mathbf{x}_i \,|\, s_i)$, $q(\tilde{\Lambda})$ and $q(s_i)$ distributions are highly coupled and it therefore might make sense to perform these updates several times before updating $q(\boldsymbol{\pi})$ or $q(\boldsymbol{\nu})$.

4.2.5 Optimising the hyperparameters

The hyperparameters for a Bayesian MFA are $\Theta = (\alpha^* \mathbf{m}^*, a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \Psi)$.

Beginning with Ψ, we simply take derivatives of $\mathcal{F}$ with respect to Ψ^{-1}, leading to:

$$\frac{\partial \mathcal{F}}{\partial \Psi^{-1}} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{s_i=1}^{S} q(s_i)\int d\tilde{\Lambda}^{s_i}\; q(\tilde{\Lambda}^{s_i})\int d\mathbf{x}_i\; q(\mathbf{x}_i \mid s_i)\cdot \\ \frac{\partial}{\partial \Psi^{-1}}\left[\left(\mathbf{y}_i - \tilde{\Lambda}^{s_i}\begin{bmatrix}\mathbf{x}_i \\ 1\end{bmatrix}\right)^{\top}\Psi^{-1}\left(\mathbf{y}_i - \tilde{\Lambda}^{s_i}\begin{bmatrix}\mathbf{x}_i \\ 1\end{bmatrix}\right) + \ln|\Psi|\right] \tag{4.58}$$

$$\implies \Psi^{-1} = \operatorname{diag}\left[\frac{1}{N}\sum_{i=1}^{n}\left\langle\left(\mathbf{y}_i - \tilde{\Lambda}^{s}\begin{bmatrix}\mathbf{x}_i \\ 1\end{bmatrix}\right)\left(\mathbf{y}_i - \tilde{\Lambda}^{s}\begin{bmatrix}\mathbf{x}_i \\ 1\end{bmatrix}\right)^{\top}\right\rangle_{q(\tilde{\Lambda}^s)q(s_i)q(\mathbf{x}_i \mid s_i)}\right] \tag{4.59}$$

where here we use diag as the operator which sets off-diagonal terms to zero.

By writing $\mathcal{F}$ as a function of a^* and b^* only, we can differentiate with respect to these hyper-parameters to yield the fixed point equations:

$$\mathcal{F}(a^*, b^*) = \sum_{s=1}^{S}\int d\boldsymbol{\nu}^s\; q(\boldsymbol{\nu}^s)\ln p(\boldsymbol{\nu}^s \mid a^*, b^*) + c \tag{4.60}$$

$$= \sum_{s=1}^{S}\sum_{l=1}^{k}\int d\boldsymbol{\nu}_l^s\; q(\boldsymbol{\nu}_l^s)\left[a^*\ln b^* - \ln\Gamma(a^*) + (a^*-1)\ln\nu_l^s - b^*\nu_l^s\right] + c\,, \tag{4.61}$$

$$\frac{\partial \mathcal{F}}{\partial a^*} = 0 \quad \implies \quad \psi(a^*) = \ln(b^*) + \frac{1}{Sk}\sum_{s=1}^{S}\sum_{l=1}^{k}\langle\ln\nu_l^s\rangle_{q(\nu_l^s)} \tag{4.62}$$

$$\frac{\partial \mathcal{F}}{\partial b^*} = 0 \quad \implies \quad b^{*-1} = \frac{1}{a^* Sk}\sum_{s=1}^{S}\sum_{l=1}^{k}\langle\nu_l^s\rangle_{q(\nu_l^s)}\;. \tag{4.63}$$

Solving for the fixed point amounts to setting the prior distribution's first moment and first logarithmic moment to the respective averages of those quantities over the factor loading matrices. The expectations for the gamma random variables are given in appendix A.

Similarly, by writing $\mathcal{F}$ as a function of α^* and $\mathbf{m}^*$ only, we obtain

$$\mathcal{F}(\alpha^*, \mathbf{m}^*) = \int d\boldsymbol{\pi}\; q(\boldsymbol{\pi})\ln p(\boldsymbol{\pi} \mid \alpha^*\mathbf{m}^*) \tag{4.64}$$

$$= \int d\boldsymbol{\pi}\; q(\boldsymbol{\pi})\left[\ln\Gamma(\alpha^*) - \sum_{s=1}^{S}\left[\ln\Gamma(\alpha^* m_s^*) - (\alpha^* m_s^* - 1)\ln\pi_s\right]\right]\;. \tag{4.65}$$

Bearing in mind that $q(\boldsymbol{\pi})$ is Dirichlet with parameter $\alpha\mathbf{m}$, and that we have a scaled prior $m_s^* = 1/S$ as given in (4.15), we can express the lower bound as a function of α^* only:

$$\mathcal{F}(\alpha^*) = \ln\Gamma(\alpha^*) - S\ln\Gamma\left(\frac{\alpha^*}{S}\right) + \left(\frac{\alpha^*}{S} - 1\right)\sum_{s=1}^{S}\left[\psi(\alpha m_s) - \psi(\alpha)\right] \tag{4.66}$$

Taking derivatives of this quantity with respect to α^* and setting to zero, we obtain:

$$\psi(\alpha^*) - \psi(\frac{\alpha^*}{S}) = \frac{1}{S}\sum_{s=1}^{S}\left[\psi(\alpha) - \psi(\alpha m_s)\right]. \tag{4.67}$$

The second derivative with respect to α^* of (4.66) is negative for $\alpha^* > 0$, which implies the solution of (4.67) is a maximum. This maximum can be found using gradient following techniques such as Newton-Raphson. The update for $\mathbf{m}^*$ is not required, since we assume that the prior over the mixing proportions is symmetric.

The update for the prior over the centres $\{\boldsymbol{\mu}^s\}_{S=1}^{S}$ of each of the factor analysers is given by considering terms in $\mathcal{F}$ that are functions of $\boldsymbol{\mu}^*$ and $\boldsymbol{\nu}^*$:

$$\mathcal{F}(\boldsymbol{\mu}^*, \boldsymbol{\nu}^*) = \int d\boldsymbol{\mu}\; q(\boldsymbol{\mu}) \ln p(\boldsymbol{\mu} \,|\, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) \tag{4.68}$$

$$= \frac{1}{2}\sum_{s=1}^{S}\int d\boldsymbol{\mu}^s\; q(\boldsymbol{\mu}^s)\left[\ln|\mathrm{diag}\,(\boldsymbol{\nu}^*)| - (\boldsymbol{\mu}^s - \boldsymbol{\mu}^*)^\top \mathrm{diag}\,(\boldsymbol{\nu}^*)\,(\boldsymbol{\mu}^s - \boldsymbol{\mu}^*)\right]. \tag{4.69}$$

Taking derivatives with respect to $\boldsymbol{\mu}^*$ first, and then $\boldsymbol{\nu}^*$, equating each to zero yields the updates

$$\boldsymbol{\mu}^* = \frac{1}{S}\sum_{s=1}^{S}\langle\boldsymbol{\mu}^s\rangle_{q(\boldsymbol{\mu}^s)} \tag{4.70}$$

$$\boldsymbol{\nu}^* = [\nu_1^*, \ldots, \nu_p^*], \text{ with } \boldsymbol{\nu}_q^* = \frac{1}{S}\sum_{s=1}^{S}\left\langle(\mu_q^s - \mu_q^*)(\mu_q^s - \mu_q^*)\right\rangle_{q(\boldsymbol{\mu}^s)}, \tag{4.71}$$

where the update for $\boldsymbol{\nu}^*$ uses the already updated $\boldsymbol{\mu}^*$.

4.3 Model exploration: birth and death

We already have an ARD mechanism in place to discover the local dimensionality for each analyser in the mixture, as part of the inference procedure over the precisions $\boldsymbol{\nu}$. However we have not yet addressed the problem of inferring the number of analysers.

The advantage of the Bayesian framework is that different model structures can be compared without having to rely on heuristic penalty or cost functions to compare their complexities;

ideally different model structures m and m' should be compared using the difference of log marginal likelihoods $\mathcal{L}(m)$ and $\mathcal{L}(m')$. In this work we use $\mathcal{F}(m)$ and $\mathcal{F}(m')$ as guides to the intractable log marginal likelihoods.

This has advantages over unpenalised maximum likelihood methods where, for example, in the split and merge algorithm described in Ueda et al. (2000) changes to model complexity are limited to simultaneous split and merge operations such that the number of components in the mixture remain the same. Whilst this approach is unable to explore differing sizes of models, it is successful in avoiding some local maxima in the optimisation process. For example, a Gaussian component straddled between two distinct clusters of data is an ideal candidate for a split operation — unfortunately their method requires that this split be accompanied with a merging of two other components elsewhere to keep the number of components fixed.

In our Bayesian model, though, we are allowed to propose any changes to the number of components in the mixture. We look at the simple cases of incremental and decremental changes to the total number, S, since we do not expect wild changes to the model structure to be an efficient method for exploring the space. This is achieved through *birth* and *death* 'moves', where a component is removed from or introduced into the mixture model. This modified model is then trained further as described in section 4.2.4 until a measure of convergence is reached (see below), at which point the proposal is accepted or rejected based on the change in $\mathcal{F}$. Another proposal is then made and the procedure repeated, up to a point when no further proposals are accepted. In this model (although not in a general application) component death occurs naturally as a by-product of the optimisation; the following sections explain the death mechanism, and address some interesting aspects of the birth process, which we have more control over.

Our method is similar to that of Reversible Jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) applied to mixture models, where birth and death moves can also be used to navigate amongst different sized models (Richardson and Green, 1997). By sampling in the full space of model parameters for all structures, RJMCMC methods converge to the exact posterior distribution over structures. However, in order to ensure reversibility of the Markov chain, complicated Metropolis-Hastings acceptance functions need to be derived and evaluated for each proposal from one parameter subspace to another. Moreover, the method suffers from the usual problems of MCMC methods, namely difficulty in assessing convergence and long simulation run time. The variational Bayesian method attempts to estimate the posterior distribution directly, not by obtaining samples of parameters and structures, but by attempting to directly integrate over the parameters using a lower bound arrived at deterministically. Moreover, we can obtain a surrogate for the posterior distribution over model structures, $p(m \mid \mathbf{y})$, which is not represented as some large set of samples, but is obtained using a quantity proportional to $p(m) \exp\{\mathcal{F}(m)\}$, where $\mathcal{F}(m)$ is the optimal (highest) lower bound achieved for a model m of particular structure.

4.3.1 Heuristics for component death

There are two routes for a component death occurring in this model, the first is by natural causes and the second through intervention. Each is explained in turn below.

When optimising $\mathcal{F}$, occasionally one finds that for some mixture component s': $\sum_{i=1}^{n} q(s'_i) = 0$ (to machine precision), even though the component still has non-zero prior probability of being used in the mixture, $p(s'_i) = \int d\boldsymbol{\pi} p(\boldsymbol{\pi}) p(s'_i \,|\, \boldsymbol{\pi})$. This is equivalent to saying that it has no responsibility for any of the data, and as a result its parameter posterior distributions have defaulted exactly to the priors. For example, the mean location of the centre of the analyser component is at the centre of the prior distribution (this can be deduced from examining (4.50) for the case of $q(s'_i) = 0 \;\forall\; i$), and the factor loadings have mean zero and high precisions $\boldsymbol{\nu}^{s'}$, referring to (4.40). If the mean of the prior over analyser centres is not located near data (see next removal method below), then this component is effectively redundant (it cannot even model data with the uniquenesses matrix Ψ, say), and can be removed from the model. How does the removal of this component affect the lower bound on the marginal likelihood, $\mathcal{F}$? Since the posterior responsibility of the component is zero it does not contribute to the last term of (4.29), which sums over the data, n. Also, since its variational posteriors over the parameters are all in accord with the priors, then the KL divergence terms in (4.29) are all zero, *except* for the very first term which is the negative KL divergence between the variational posterior and prior distribution over the mixing proportions $\boldsymbol{\pi}$. Whilst the removal of the component leaves all other terms in $\mathcal{F}$ unchanged, not having this 'barren' dimension s' to integrate over should increase this term.

It seems counter-intuitive that the mean of the prior over factor analyser centres might be far from data, as suggested in the previous paragraph, given that the hyperparameters of the prior are updated to reflect the position of the analysers. However, there are cases in which the distribution of data is 'hollow' (see, for example, the spiral data set of section 4.5.3), and in this case redundant components are very easily identified with zero responsibilities, and removed. If the redundant components default to a position which is close to data, their posterior responsibilities may not fall to exactly zero, being able to still use the covariance given in Ψ to model the data. In this case a more aggressive pruning procedure is required, where we examine the change in $\mathcal{F}$ that occurs after removing a component we suspect is becoming, or has become, redundant. We gain by not having to code its parameters, but we may lose if the data in its locale are being uniquely modelled by it, in which case $\mathcal{F}$ may drop. If $\mathcal{F}$ should drop, there is the option of continuing the optimisation to see if $\mathcal{F}$ eventually improves (see next section on birth processes), and rejecting the removal operation if it does not. We do not implement this 'testing' method in our experiments, and rely solely on the first method and remove components once their total posterior responsibilities fall below a reasonable level (in practice less than one data point's worth).

This mechanism for (automatic) removal of components is useful as it allows the data to dictate how many mixture components are required. However we should note that if the data is not distributed as a mixture of Gaussian components, the size of the data set will affect the returned number of components. Thus the number of components should not be taken to mean the number of 'clusters'.

4.3.2 Heuristics for component birth

Component birth does not happen spontaneously during learning, so we have to introduce a heuristic. Even though changes in model structure may be proposed at any point during learning, it makes sense only to do so when learning has plateaued, so as to exploit (in terms of $\mathcal{F}$) the current structure to the full. We define an *epoch* as that period of learning beginning with a proposal of a model alteration, up to the point of convergence of the variational learning rules.

One possible heuristic for deciding at which point to end an epoch can be constructed by looking at the rate of change of the lower bound with iterations of variational EM. If $\Delta\mathcal{F} = \mathcal{F}^{(t)} - \mathcal{F}^{(t-1)}$ falls below a critical value then we can assume that we have plateaued. However it is not easy to define such simple thresholds in a manner that scales appropriately with both model complexity and amount of data. An alternative (implemented in the experiments) is to examine the rate of change of the posterior class-conditional responsibilities, as given in the $q(s_i)$ matrix $(n \times S)$. A suitable function of this sort can be such that it does not depend directly on the data size, dimensionality, or current model complexity. In this work we consider the end of an epoch to be when the *rate of change of responsibility* for each analyser, averaged over all data, falls below a tolerance — this has the intuitive interpretation that the components are no longer 'in flux' and are modelling their data as best they can in that configuration. We shall call this quantity the *agitation*:

$$agitation(s)^{(t)} \equiv \frac{\sum_{i=1}^{n} \left| q(s_i)^{(t)} - q(s_i)^{(t-1)} \right|}{\sum_{i=1}^{n} q(s_i)^{(t)}} , \tag{4.72}$$

where (t) denotes the iteration number of VBEM. We can see that the agitation of each analyser does not directly scale with number of analysers, data points, or dimensionality of the data. Thus a fixed tolerance for this quantity can be chosen that is applicable throughout the optimisation process. We should note that this measure is one of many possible, such as using squared norms etc.

A sensible way to introduce a component into the model is to create that component in the image of an existing component, which we shall call the *parent*. Simply reproducing the exact parameters of the parent does not suffice as the symmetry of the resulting pair needs to be broken for them to model the data differently.

One possible approach would be to remove the parent component, s', and *replace* it with two components, the 'children', with their means displaced symmetrically about the parent's mean, by a vector sampled from the parent's distribution, its covariance ellipsoid given by $\Lambda^{s'}\Lambda^{s'\top}+\Psi$. We call this a *spatial* split. This appeals to the notion that one might expect areas of data that are currently being modelled by one elongated Gaussian to be modelled better by two, displaced most probably along the major axis of variance of that data. However this approach is hard to fine tune so that it scales well with the data dimensionality, p. For example, if the displacement is slightly too large then it becomes very likely in high dimensions that both children model the data poorly and die naturally as a result. If it is too small then the components will diverge very slowly.

Again appealing to the class-conditional responsibilities for the data, we can define a procedure for splitting components that is not directly a function of the dimensionality, or any length scale of the local data. The approach taken in this work uses a partition of the parent's posterior responsibilities for each of the data, $q(s_i = s')$, along a direction $\mathbf{d}^{s'}$ sampled from the parent's covariance ellipsoid. Those data having a positive dot product with the sampled direction donate their responsibilities to one child s^a, and vice-versa for the other child s^b. Mathematically, we sample a direction $\mathbf{d}$ and define an allocation indicator variable for each data point,

$$\mathbf{d} \sim \mathrm{N}(\mathbf{d} \,|\, \langle \boldsymbol{\mu}^{s'} \rangle_{q(\boldsymbol{\mu}^{s'})}, \langle \Lambda^{s'}\Lambda^{s'\top} \rangle_{q(\Lambda^{s'})} + \Psi) \tag{4.73}$$

$$r_i = \begin{cases} 1 & \text{if} \quad (\mathbf{y}_i - \boldsymbol{\mu}^{s'})^\top \mathbf{d} \geq 0 \\ 0 & \text{if} \quad (\mathbf{y}_i - \boldsymbol{\mu}^{s'})^\top \mathbf{d} < 0 \end{cases} \qquad \text{for } i = 1, \ldots, n \,. \tag{4.74}$$

We then set the posterior probabilities in $q(s_i)$ to reflect these assignments, introducing a *hardness* parameter α_h, ranging from .5 to 1:

$$q(s_i^a) = q(s_i') \left[\alpha_h r_i + (1-\alpha_h)(1-r_i)\right] \tag{4.75}$$

$$q(s_i^b) = q(s_i') \left[(1-\alpha_h) r_i + \alpha_h (1-r_i)\right] \tag{4.76}$$

When $\alpha_h = 1$, all the responsibility is transferred to the assigned child, and when $\alpha_h = .5$ the responsibility is shared equally. In the experiments in this chapter we use $\alpha_h = 1$.

The advantage of this approach is that the birth is made in *responsibility* space rather than data-space, and is therefore dimension-insensitive. The optimisation then continues, with the s' analyser removed and the s^a and s^b analysers in its place. The first variational updates should be for $q(\Lambda^{s^a})$ and $q(\Lambda^{s^b})$ since these immediately reflect the change (note that the update for $q(\mathbf{x}_i)$ is not a function of the responsibilities — see equation (4.53)).

The mechanism that chooses which component is to be the parent of a pair-birth operation must allow the space of models to be explored fully. A simple method would be to pick the component at random amongst those present. This has an advantage over a deterministic method, in that

the latter could preclude some components from ever being considered. Interestingly though, there is information in $\mathcal{F}$ that can be used to guide the choice of component to split: with the exception of the first term in equation (4.29), the remaining terms can be decomposed into component-specific contributions, $\mathcal{F}_s$. An ordering for parent choice can be defined using $\mathcal{F}_s$, with the result that is it possible to concentrate attempted births on those components that are not currently modelling their data well. This mirrors the approach taken in Ueda et al. (2000), where the criterion was the (KL) discrepancy between each analyser's local density model and the empirical density of the data.

If, at the end of an epoch, we reject the proposed birth so returning to the original configuration, we may either attempt to split the same component again, but with a new randomly sampled direction, or move on to the next 'best' component in the ordering. We use the following function to define $\mathcal{F}_s$, from which the ordering is recalculated after every successful epoch:

$$\begin{aligned}
\mathcal{F}_s &= \mathcal{F}(\{Q\}, \alpha^*\mathbf{m}^*, a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*, \Psi \mid Y) \\
&= \int d\boldsymbol{\nu}^s \; q(\boldsymbol{\nu}^s) \left[\ln \frac{p(\boldsymbol{\nu}^s \mid a^*, b^*)}{q(\boldsymbol{\nu}^s)} + \int d\tilde{\Lambda}^s \; q(\tilde{\Lambda}^s) \frac{p(\tilde{\Lambda}^s \mid \boldsymbol{\nu}^s, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)}{q(\tilde{\Lambda}^s)} \right] \\
&\quad + \frac{1}{\sum_{i=1}^n q(s_i)} \sum_{i=1}^n q(s_i) \left[\int d\boldsymbol{\pi} \; q(\boldsymbol{\pi}) \ln \frac{p(s_i \mid \boldsymbol{\pi})}{q(s_i)} + \int d\mathbf{x}_i \; q(\mathbf{x}_i \mid s_i) \ln \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i \mid s_i)} \right. \\
&\quad \left. + \int d\tilde{\Lambda}^s \; q(\tilde{\Lambda}^s) \int d\mathbf{x}_i \; q(\mathbf{x}_i \mid s_i) \ln p(\mathbf{y}_i \mid s_i, \mathbf{x}_i, \tilde{\Lambda}^s, \Psi) \right]
\end{aligned} \tag{4.77}$$

This has the intuitive interpretation as being the likelihood of the data (weighted by its data responsibilities) under analyser s, normalised by its overall responsibility, with the relevant (KL) penalty terms as in $\mathcal{F}$. Those components with lower $\mathcal{F}_s$ are preferentially split. The optimisation completes when all existing mixture components have been considered as parents, with no accepted epochs.

Toward the end of an optimisation, the remaining required changes to model structure are mainly local in nature and it becomes computationally wasteful to update the parameters of all the components of the mixture model at each iteration of the variational optimisation. For this reason only those components whose responsibilities are in flux (to some threshold) are updated. This partial optimisation approach still guarantees an increase in $\mathcal{F}$, as we simply perform updates that guarantee to increase parts of the $\mathcal{F}$ term in 4.29.

It should be noted that no matter which heuristics are used for birth and death, ultimately the results are always compared in terms of $\mathcal{F}$, the lower bound on the log marginal likelihood $\mathcal{L}$. Therefore different choices of heuristic can only affect the *efficiency* of the search over model structures and not the theoretical validity of the variational approximation. For example, although it is perfectly possible to start the model with many components and let them die, it

is computationally more efficient and equally valid to start with one component and allow it to spawn more when necessary.

4.3.3 Heuristics for the optimisation endgame

In the previous subsection we proposed a heuristic for terminating the optimisation, namely that every component should be unsuccessfully split a number of times. However, working in the space of components seems very inefficient. Moreover, there are several pathological birth-death scenarios which raise problems when counting the number of times each component has been split; for example, the identities of nearby components can be switched during an epoch (parent splits into two children, first child usurps an existing other component and models its data, whilst that component switches to model the old parent's data, and the second child dies).

One possible solution (personal communication, Y. Teh) is based on a responsibility accumulation method. Whenever a component s is chosen for a split, we store its responsibility vector (of length n) for all the data points $q(\mathbf{s}) = [q(s_1)\ q(s_2)\ \ldots\ q(s_n)]$, and proceed with the optimisation involving its two children. At the end of the epoch, if we have not increased $\mathcal{F}$, we add $q(\mathbf{s})$ to a running total of 'split data' responsibilities, $\mathbf{t} = (t_1, t_2, \ldots, t_n)$. That is $\forall i :\ t_i \leftarrow \min(t_i + q(s_i), t_{\text{max}})$, where t_{max} is some saturation point. If by the end of the epoch we have managed to increase $\mathcal{F}$, then the accumulator $\mathbf{t}$ is reset to zero for every data point.

From this construction we can derive a stochastic procedure for choosing which component to split, using the softmax of the quantity $c(s) = \beta \sum_{i=1}^{n} (t_{\text{max}} - t_i) q(s_i)$. If $c(s)$ is large for some component s, then the data it is responsible for has not 'experienced' many birth attempts, and so it should be a strong candidate for a split. Here $\beta \geq 0$ is a temperature parameter to be set as we wish. As β tends to infinity the choice of component to split becomes deterministic, and is based on which has least responsibility overlap with already-split data. If β is very small (but non-zero) the splits become more random. Whatever setting of β, attempted splits will be automatically focused on those components with more data and unexplored regions of data space. Furthermore, a termination criterion is automatic: continue splitting components until every entry of the $\mathbf{t}$ vector has reached saturation — this corresponds to splitting every *data point* a certain number of times (in terms of its responsibility under the split parent), before we terminate the entire optimisation. This idea was conceived of only after the experiments were completed, and so has not been thoroughly investigated.

4.4 Handling the predictive density

In this section we set about trying to get a handle on the predictive density of VBMFA models using bounds on approximations (in section 4.7.1 we will show how to estimate the density

using sampling methods). In order to perform density estimation or classification of a new test example, we need to have access to the predictive density

$$p(\mathbf{y}' \,|\, \mathbf{y}) = \frac{p(\mathbf{y}', \mathbf{y})}{p(\mathbf{y})} = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y}) p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \tag{4.78}$$

where $\mathbf{y}'$ is a set of test examples $\mathbf{y}' = \{\mathbf{y}'_1, \ldots, \mathbf{y}'_{n'}\}$, and $\mathbf{y}$ is the training data. This quantity is simply the probability of observing the test examples for a particular setting of the model parameters, averaged over the posterior distribution of the parameters given a training set. Unfortunately, the very intractability of the marginal likelihood in equation (4.14) means that the predictive density is also intractable to compute exactly.

A poor man's approximation uses the variational posterior distribution in place of the posterior distribution:

$$p(\mathbf{y}' \,|\, \mathbf{y}) \approx \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \;. \tag{4.79}$$

However we might expect this to overestimate the density of $\mathbf{y}'$ in typical regions of space (in terms of where the training data lie), as the variational posterior tends to over-neglect areas of low posterior probability in parameter space. This is a result of the asymmetric KL divergence measure penalty in the optimisation process.

Substituting the form for MFAs given in (4.14) into (4.79)

$$p(\mathbf{y}' \,|\, \mathbf{y}) \approx \int d\boldsymbol{\pi} \int d\tilde{\Lambda}\; q(\boldsymbol{\pi}, \tilde{\Lambda}) \left[\prod_{i=1}^{n'} \sum_{s_i=1}^{S} p(s_i \,|\, \boldsymbol{\pi}) p(\mathbf{y}'_i \,|\, s_i, \tilde{\Lambda}, \Psi) \right] \;, \tag{4.80}$$

which is still intractable for the same reason that the marginal likelihoods of training set were so. We can lower bound the log of the predictive density using variational distributions over the hidden variables corresponding to each test case:

$$\ln p(\mathbf{y}' \mid \mathbf{y}) \approx \ln \int d\boldsymbol{\pi} \int d\tilde{\Lambda}\; q(\boldsymbol{\pi}, \tilde{\Lambda}) \left[\prod_{i=1}^{n'} \sum_{s_i=1}^{S} p(s_i \mid \boldsymbol{\pi}) p(\mathbf{y}'_i \mid s_i, \tilde{\Lambda}, \Psi) \right] \tag{4.81}$$

$$\geq \sum_{i=1}^{n'} \int d\boldsymbol{\pi} \int d\tilde{\Lambda}\; q(\boldsymbol{\pi}, \tilde{\Lambda}) \left[\ln \sum_{s_i=1}^{S} p(s_i \mid \boldsymbol{\pi}) p(\mathbf{y}'_i \mid s_i, \tilde{\Lambda}, \Psi) \right] \tag{4.82}$$

$$= \sum_{i=1}^{n'} \int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \int d\tilde{\Lambda}\; q(\tilde{\Lambda}) \left[\ln \sum_{s_i=1}^{S} q(s_i) \frac{p(s_i \mid \boldsymbol{\pi}) p(\mathbf{y}'_i \mid s_i, \tilde{\Lambda}, \Psi)}{q(s_i)} \right] \tag{4.83}$$

$$\geq \sum_{i=1}^{n'} \int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \int d\tilde{\Lambda}\; q(\tilde{\Lambda}) \sum_{s_i=1}^{S} q(s_i) \ln \frac{p(s_i \mid \boldsymbol{\pi}) p(\mathbf{y}'_i \mid s_i, \tilde{\Lambda}, \Psi)}{q(s_i)} \tag{4.84}$$

$$\geq \sum_{i=1}^{n'} \sum_{s_i=1}^{S} q(s_i) \left[\int d\boldsymbol{\pi}\; q(\boldsymbol{\pi}) \ln \frac{p(s_i \mid \boldsymbol{\pi})}{q(s_i)} + \int d\mathbf{x}_i\; q(\mathbf{x}_i \mid s_i) \ln \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i \mid s_i)} \right.$$
$$\left. + \int d\tilde{\Lambda}^{s_i}\; q(\tilde{\Lambda}^{s_i}) \int d\mathbf{x}_i\; q(\mathbf{x}_i \mid s_i) \ln p(\mathbf{y}'_i \mid s_i, \mathbf{x}_i, \tilde{\Lambda}^{s_i}, \Psi) \right]. \tag{4.85}$$

The first inequality is a simple Jensen bound, the second is another which introduces a set of variational distributions $q(s_i)$, and the third a further set of distributions over the hidden variables $q(\mathbf{x}_i \mid s_i)$. Note that these distributions correspond to the *test* data, indexed from $i = 1, \ldots, n'$. This estimate of the predictive density is then very similar to the lower bound of the marginal likelihood of the training data (4.29), except that the training data $\mathbf{y}_i$ has been replaced with the test data $\mathbf{y}'_i$, and the KL penalty terms on the parameters have been removed. This carries the interpretation that the distribution over parameters of the model is decided upon and fixed (i.e. the variational posterior), and we simply need to explain the test data under this ensemble of models.

This lower bound on the approximation to the predictive density can be optimised in just *two updates* for each test point. First, infer the distribution $q(\mathbf{x}_i \mid s_i)$ for each test data point, using the analogous form of update (4.53). Then update the distribution $q(s_i)$ based on the resulting distributions over $q(\mathbf{x}_i \mid s_i)$ using the analogous form of update (4.57). Since the $q(\mathbf{x}_i \mid s_i)$ update was not a function of $q(s_i)$, we do not need to iterate the optimisation further to improve the bound.

4.5 Synthetic experiments

In this section we present three toy experiments on synthetic data which demonstrate certain features of a Bayesian mixture of factor analysers. The first experiment shows the ability of the

algorithm's birth and death processes to find the number of clusters in a dataset. The second experiment shows more ambitiously how we can simultaneously recover the number of clusters and their dimensionalities, and how the complexity of the model depends on the amount of data support. The last synthetic experiment shows the ability of the model to fit a low dimensional manifold embedded in three-dimensional space.

4.5.1 Determining the number of components

In this toy example we tested the model on synthetic data generated from a mixture of 18 Gaussians with 50 points per cluster, as shown in figure 4.3(a). The algorithm was initialised with a single analyser component positioned at the mean of the data. Birth proposals were made using spatial splits (as described above). Also shown is the progress of the algorithm after 7, 14, 16 and 22 accepted epochs (figures 4.3(b)-4.3(e)). The variational algorithm has little difficulty finding the correct number of components and the birth heuristics are successful at avoiding local maxima.

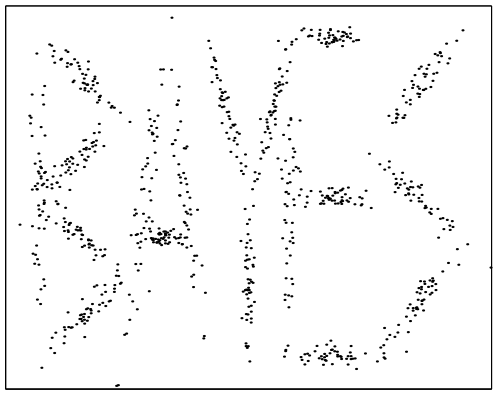
After finding the 18 Gaussians repeated splits are attempted and mostly rejected. Those epochs that are accepted always involve the birth of a component followed at some point by the death of another component, such that the number of components remain 18; the increase in $\mathcal{F}$ over these epochs is extremely small, usually due to the refinement of other components.

4.5.2 Embedded Gaussian clusters

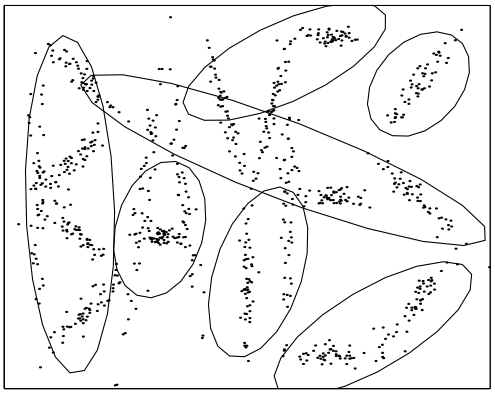
In this experiment we examine the ability of the Bayesian mixture of factor analysers to automatically determine the local dimensionality of high dimensional data. We generated a synthetic data set consisting of 300 data points drawn from each of 6 Gaussian clusters with intrinsic dimensionalities (7 4 3 2 2 1), embedded at random orientations in a 10-dimensional space. The means of the Gaussians were drawn uniformly under $[0, 3]$ in each of the data dimensions, all Gaussian variances set to 1, and sensor noise of covariance .01 added in each dimension.

A Bayesian MFA was initialised with one mixture component centred about the data mean, and trained for a total of 200 iterations of variational EM with spatial split heuristics for the birth proposals. All the analysers were created with a maximum dimensionality of 7. The variational Bayesian approach correctly inferred both the number of Gaussians and their intrinsic dimensionalities, as shown in figure 4.4. The dimensionalities were determined by examining the posterior distributions over the precisions of each factor analyser's columns, and thresholding on the mean of each distribution.

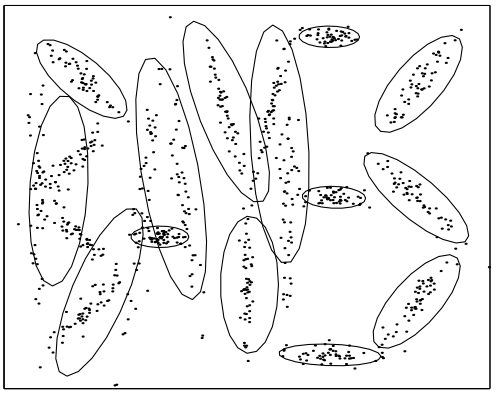
We then varied the number of data points in each cluster and trained models on successively smaller data sets. Table 4.1 shows how the Bayesian MFA partitioned the data set. With large



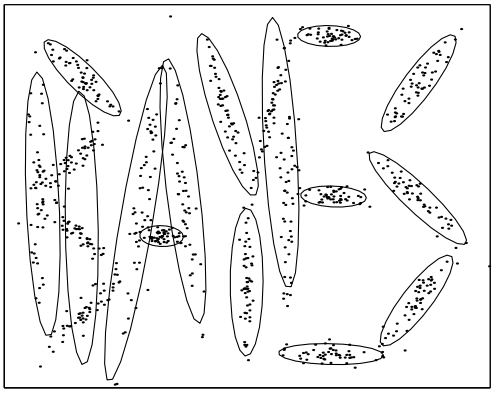
(a) The data, consisting of 18 Gaussian clusters.



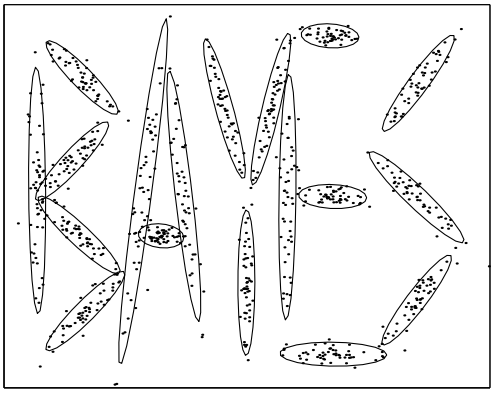
(b) After 7 accepted epochs.



(c) After 14 accepted epochs.



(d) After 16 accepted epochs.



(e) After 22 accepted epochs.

Figure 4.3: The original data, and the configuration of the mixture model at points during the optimisation process. Plotted are the 2 s.d. covariance ellipsoids for each analyser in the mixture. To be more precise, the centre of the ellipsoid is positioned at the mean of the variational posterior over the analyser's centre, and each covariance ellipsoid is the expected covariance under the variational posterior.

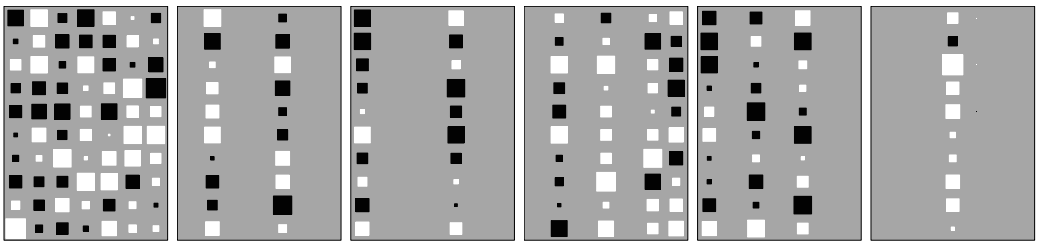


Figure 4.4: Learning the local intrinsic dimensionality. The maximum dimensionality of each analyser was set to 7. Shown are Hinton diagrams for the means of the factor loading matrices $\{\overline{\Lambda}^s\}_{s=1}^S$ for each of the 6 components, after training on the data set with 300 data points per cluster. Note that empty columns correspond to unused factors where the mass of $q(\nu_l^s)$ is at very high values, so the learnt dimensionalities are (7,2,2,4,3,1).

number of points per cluster	intrinsic dimensionalities					
	1	7	4	3	2	2
8	2				1	
8	1		2			
16	1	4			2	
32	1	6	3	3	2	2
64	1	7	4	3	2	2
128	1	7	4	3	2	2

Table 4.1: The recovered number of analysers and their intrinsic dimensionalities. The numbers in the table are the dimensionalities of the analysers and the boxes represent analysers modelling data from more than one cluster. For a large number of data points per cluster ($\geq$ 64), the Bayesian MFA recovers the generative model. As we decrease the amount of data, the model reduces the dimensionality of the analysers and begins to model data from different clusters with the same analyser. The two entries for 8 data points are two observed configurations that the model converged on.

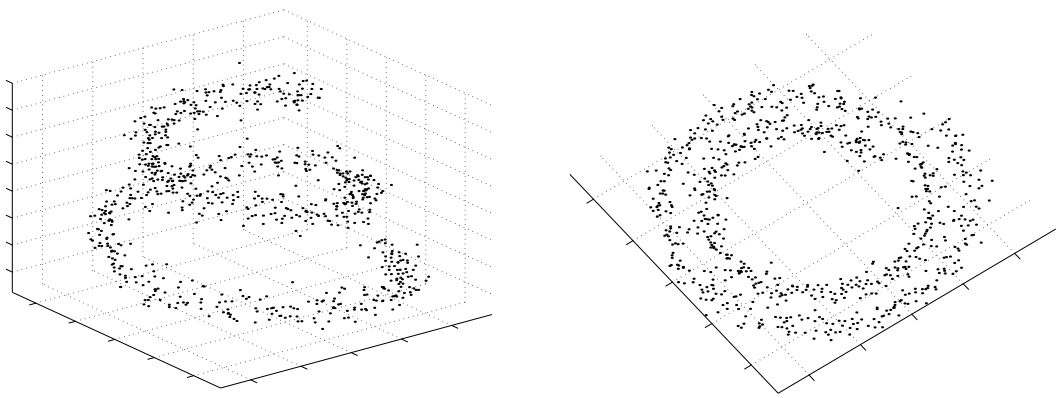
amounts of data the model agrees with the true model, both in the number of analysers and their dimensionalities. As the number of points per cluster is reduced there is insufficient evidence to support the full intrinsic dimensionality, and with even less data the number of analysers drop and they begin to model data from more than one cluster.

4.5.3 Spiral dataset

Here we present a simple synthetic example of how Bayesian MFA can learn locally linear models to tile a manifold for globally non-linear data. We used the dataset of 800 data points from a noisy shrinking spiral, as used in Ueda et al. (2000), given by

$$\mathbf{y}_i = [(13 - 0.5t_i)\cos t_i, \quad -(13 - 0.5t_i)\sin t_i, \quad t_i)] + \mathbf{w}_i \tag{4.86}$$

$$\text{where} \quad t_i \in [0, 4\pi]\,, \qquad \mathbf{w}_i \sim \mathrm{N}(\mathbf{0}, \mathrm{diag}\,([.5\ .5\ .5])) \tag{4.87}$$



(a) An elevated view of the spiral data set (see text for reference).

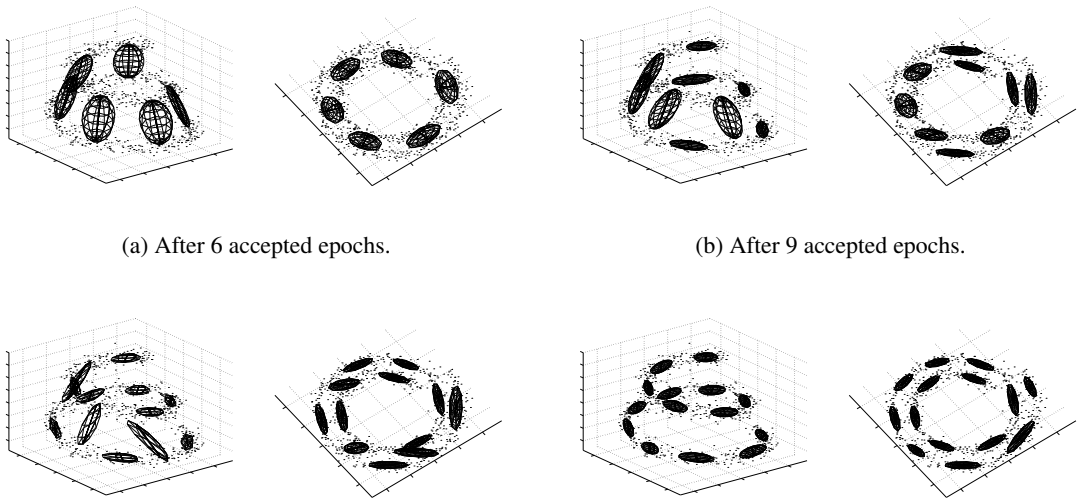
(b) The same data set viewed perpendicular to the third axis.

Figure 4.5: The spiral data set as used in Ueda et al. (2000). Note that the data lie on a 1-dimensional manifold embedded non-linearly in the 3-dimensional data space.

where the parameter t determines the point along the spiral in one dimension. The spiral is shown in figure 4.5, viewed from two angles. Note the spiral data set is really a 1-dimensional manifold embedded non-linearly in the 3-dimensional data space and corrupted by noise.

As before we initialised a variational Bayesian MFA model with a single analyser at the mean of the data, and imposed a maximum dimensionality of $k = 2$ for each analyser. For this experiment, as for the previous synthetic experiments, the spatial splitting heuristic was used. Again local maxima did not pose a problem and the algorithm always found between 12-14 Gaussians. This result was repeatable even when the algorithm was initialised with 200 randomly positioned analysers. The run starting from a single analyser took about 3-4 minutes on a 500MHz Alpha EV6 processor. Figure 4.6 shows the state of the algorithm after 6, 9, 12 and 17 accepted epochs.

Figure 4.7 shows the evolution of the lower bound used to approximate the marginal likelihood of the data. Thick and thin lines in the plot correspond to accepted and rejected epochs, respectively. There are several interesting aspects one should note. First, at the beginning of most of the epochs there is a drop in $\mathcal{F}$ corresponding to a component birth. This is because the model now has to code the parameters of the new analyser component, and initially the model is not fit well to the data. Second, most of the compute time is spent on accepted epochs, suggesting that our heuristics for choosing which components to split, and how to split them, are good. Referring back to figure 4.6, it turns out that it is often components that are straddling arms of the spiral that have low $\mathcal{F}_s$, as given by (4.77), and these are being correctly chosen for splitting ahead of other components modelling their local data better (for example, those aligned on the spiral). Third, after about 1300 iterations, most of the proposed changes to model structure are rejected, and those that are accepted give only a small increase in $\mathcal{F}$.



(a) After 6 accepted epochs.

(b) After 9 accepted epochs.

(c) After 12 accepted epochs.

(d) After 17 accepted epochs.

Figure 4.6: The evolution of the variational Bayesian MFA algorithm over several epochs. Shown are the 1 s.d. covariance ellipses for each analyser: these are the *expected* covariances, since the analysers have distributions over their factor loadings. After 17 accepted epochs the algorithm has converged to a solution with 14 components in the mixture. Local optima, where components are straddled across two arms of the spiral (see **(b)** for example), are successfully avoided by the algorithm.

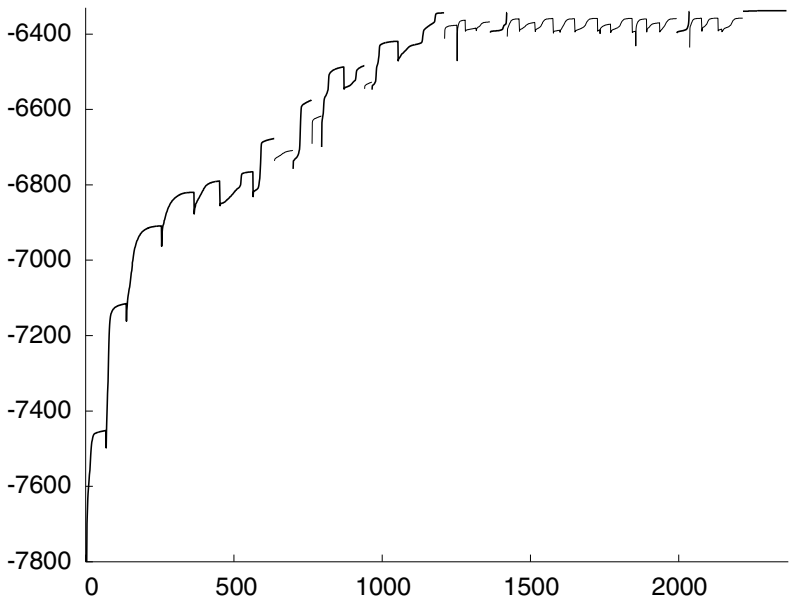


Figure 4.7: Evolution of the lower bound $\mathcal{F}$, as a function of iterations of variational Bayesian EM, for the spiral problem on a typical run. Drops in $\mathcal{F}$ constitute component births. The thick and thin lines represent whole epochs in which a change to model structure was proposed and then eventually accepted or rejected, respectively.

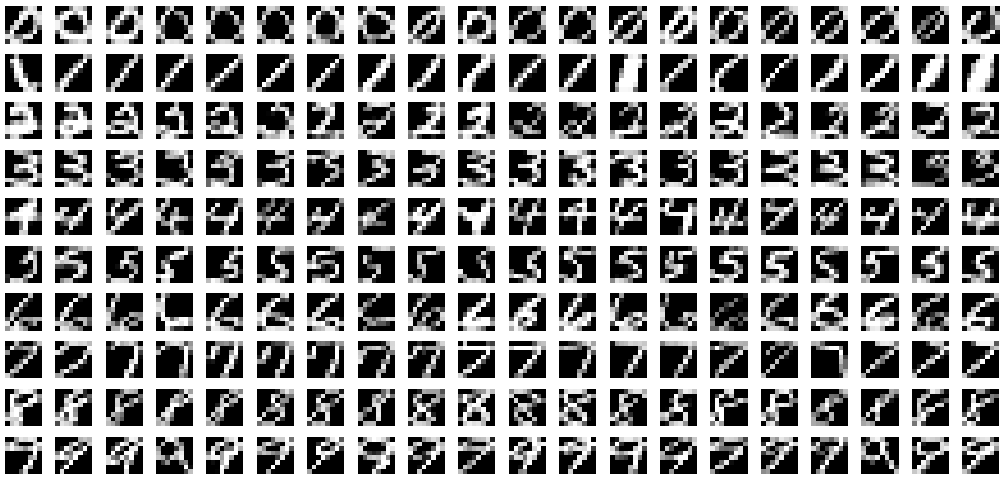


Figure 4.8: Some examples of the digits 0-9 in the training and test data sets. Each digit is 8×8 pixels with gray scale 0 to 255. This data set was normalised before passing to VBMFA for training.

4.6 Digit experiments

In this section we present results of using variational Bayesian MFA to learn both supervised and unsupervised models of images of 8×8 digits taken from the CEDAR database (Hull, 1994). This data set was collected from hand-written digits from postal codes, and are labelled with the classes 0 through to 9. Examples of these digits are given in figure 4.8. The entire data set was normalised before being passed to the VBMFA algorithm, by first subtracting the mean image from every example, and then rescaling each individual pixel to have variance 1 across all the examples. The data set was then partitioned into 700 training and 200 test examples for each digit. Based on density models learnt from the digits, we can build classifiers for a test data set. Histograms of the pixel intensities after this normalisation are quite non-Gaussian, and so factor analysis is perhaps not a good model for this data. Before normalising, we could have considered taking the logarithm or some other non-linear transformation of the intensities to improve the non-Gaussianity, but this was not done.

4.6.1 Fully-unsupervised learning

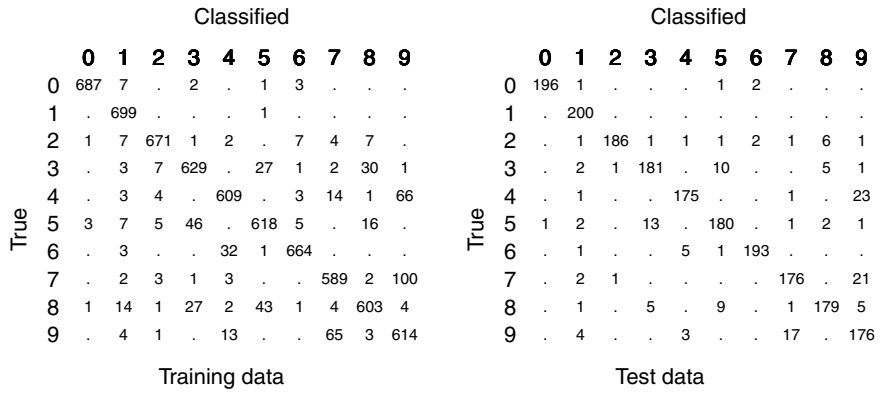
A *single* VBMFA model was trained on 700 examples of every digit 0-9, using birth proposals and death processes as explained in section 4.3. The maximum dimensionality for each analyser $k_{\max}$ was set to 6, and the number of components initialised to be 1. Responsibility-based splits were used for the birth proposals (section 4.3.2) as we would expect these to perform better than spatial-splits given the high dimensionality of the data (using the fraction of accepted splits as a criterion, this was indeed confirmed in preliminary experiments with high dimensional data sets). The choice of when to finish an epoch of learning was based on the rate of change of the

Figure 4.9: A typical model learnt by fully-unsupervised VBMFA using the birth and death processes. Each digit shown represents an analyser in the mixture, and the pixel intensities are the means of the posterior distribution over the centre of the analyser, $\langle \boldsymbol{\mu}^s \rangle_{q(\boldsymbol{\mu}^s)}$. These means can be thought of as *templates*. These intensities have been inversely-processed to show pixel intensities with the same scalings as the training data. The number to the right of each image is that analyser's dimensionality. In this experiment the maximum dimensionality of the latent space was set to $k_{\max} = 6$. As can be seen from these numbers, the highest required dimensionality was 5. The within-row ordering indicates the creation order of the analysers during learning, and we have arranged the templates across different rows according to the 10 different digits in 4.8. This was done by performing a sort of higher-level clustering which the unsupervised algorithm cannot in fact do. Even though the algorithm itself was not given the labels of the data, we as experimenters can examine the posterior responsibilities of each analyser for every item in the training set (whose labels we have access to), and find the majority class for that analyser, and then assign that analyser to the row corresponding to the class label. This is purely a visual aid — in practice if the data is not labelled we have no choice but to call each mixture component in the model a separate class, and have the mean of each analyser as the class template.

component posterior responsibilities (section 4.3.2). The optimisation was terminated when no further changes to model structure managed to increase $\mathcal{F}$ (based on three unsuccessful splits for every component in the model).

Figure 4.9 shows the final model returned from the optimisation. In this figure, each row corresponds to a different digit, and each digit image in the row corresponds to the mean of the posterior over the centre position of each factor analyser component of the mixture. We refer to these as 'templates' because they represent the mean of clusters of similar examples of the same digit. The number to the right of each template is the dimensionality of the analyser, determined from examining the posterior over the precisions governing that factor loading matrix's columns $q(\boldsymbol{\nu}^s) = [q(\nu_1^s), \ldots, q(\nu_{k_{\max}}^s)]$.

For some digits the VBMFA needs to use more templates than others. These templates represent distinctively different styles for the same digit. For example, some 1's are written slanting to the left and others to the right, or the digit 2 may or may not contain a loop. These different styles are in very different areas of the high dimensional data space; so each template explains all the



True \ Classified	0	1	2	3	4	5	6	7	8	9
0	687	7	.	2	.	1	3	.	.	.
1	.	699	.	.	.	1	.	.	.	.
2	1	7	671	1	2	.	7	4	7	.
3	.	3	7	629	.	27	1	2	30	1
4	.	3	4	.	609	.	3	14	1	66
5	3	7	5	46	.	618	5	.	16	.
6	.	3	.	.	32	1	664	.	.	.
7	.	2	3	1	3	.	.	589	2	100
8	1	14	1	27	2	43	1	4	603	4
9	.	4	1	.	13	.	.	65	3	614

Training data

True \ Classified	0	1	2	3	4	5	6	7	8	9
0	196	1	.	.	.	1	2	.	.	.
1	.	200	.	.	.	.	.	.	.	.
2	.	1	186	1	1	1	2	1	6	1
3	.	2	1	181	.	10	.	.	5	1
4	.	1	.	.	175	.	.	1	.	23
5	1	2	.	13	.	180	.	1	2	1
6	.	1	.	.	5	1	193	.	.	.
7	.	2	1	.	.	.	.	176	.	21
8	.	1	.	5	.	9	.	1	179	5
9	.	4	.	.	3	.	.	17	.	176

Test data

Figure 4.10: Confusion tables for digit classification on the training (700) and test (200) sets. The mixture of factor analysers with 92 components obtains 8.8% and 7.9% training and test classification errors respectively.

examples of that style that can be modelled with a linear transformation of the pixel intensities. The number of dimensions of each analyser component for each digit template corresponds very roughly to the number of degrees of freedom there are for that template, and the degree with which each template's factor analyser's linear transformation can extrapolate to the data between the different templates. By using a few linear operations on the pixel intensities of the template image, the analyser can mimic small amounts of shear, rotation, scaling, and translation, and so can capture the main trends in its local data.

When presented with a test example digit from 0-9, we can classify it by asking the model which analyser has the highest posterior responsibility for the test example (i.e. a hard assignment), and then finding which digit class that analyser is clustered into (see discussion above). The result of classifying the training and test data sets are shown in figure 4.10, in confusion matrix form. Each row corresponds to the true class labelling of the digit, and each column corresponds to the digit cluster that the example was assigned to, via the most-responsible analyser in the trained VBMFA model. We see that, for example, about $1/7$ of the training data 8's are misclassified as a variety of classes, and about $1/7$ of the training data 7's are misclassified as 9's (although the converse result is not as poor). These trends are also seen in the classifications of the test data.

The overall classification performance of the model was 91.2% and 92.1% for the training and test sets respectively. This can be compared to simple K-means (using an isotropic distance measure on the identically pre-processed data), with the number of clusters set to the same as inferred in the VBMFA optimisation. The result is that K-means achieves only 87.8% and 86.7% accuracy respectively, despite being initialised with part of the VB solution.

Computation time

The full optimisation for the VBMFA model trained on all 7000 64-dimensional digit examples took approximately 4 CPU days on a Pentium III 500 MHz laptop computer. We would expect the optimisation to take considerably less time if any of the following heuristics were employed. First, one could use partial VBEM updates for $\mathcal{F}$ to update the parameter distributions of only those components that are currently in flux; this corresponds to assuming that changing the modelling configuration of a few analysers in one part of the data space often does not affect the parameter distributions of the overwhelming majority of remaining analysers. In fact, partial updates can be derived that are guaranteed to increase $\mathcal{F}$, simply by placing constraints on the posterior responsibilities of the fixed analysers. Second, the time for each iteration of VBEM can be reduced significantly by removing factors that have been made extinct by the ARD priors; this can even be done prematurely if it increases $\mathcal{F}$. In the implementation used for these experiments all analysers always held factor loading matrix sizes of $(p \times k_{\text{max}})$, despite many of them having far fewer active factors.

4.6.2 Classification performance of BIC and VB models

In these experiments VBMFA was compared to a BIC-penalised maximum likelihood MFA model, in a digit classification task. Each algorithm learnt separate models for each of the digits 0-9, and attempted to classify a data set of test examples based on the predictive densities under each of the learnt digit models. For the VB model, computing the predictive density is intractable (see section 4.4) and so an approximation is required. The experiment was carried out for 7 different training data set sizes ranging from $(100, 200, \ldots 700)$, and repeated 10 times with different parameter initialisations and random subsets of the full 700 images for each digit.

The maximum dimensionality of any analyser component for BIC or VB was set to $k_{\text{max}} = 5$. This corresponds to the maximum dimensionality required by the fully-unsupervised VB model in the previous section's experiments. For the BIC MFA implementation there is no mechanism to prune the factors from the analysers, so all 5 dimensions in each BIC analyser are used all the time.

The same heuristics were used for model search in both types of model, as described in section 4.3. In order to compute a component split ordering, the ML method used the empirical KL divergence to measure the quality of each analyser's fit to its local data (see Ueda et al., 2000, for details). The criterion for ending any particular epoch was again based on the rate of change of component posterior responsibilities. The termination criterion for both algorithms was, as before, three unsuccessful splits of every mixture component in a row. For the ML model, a constraint had to be placed on the Ψ matrix, allowing a minimum variance of 10^{-5} in any direction in the normalised space in which the data has identity covariance. This constraint was

	% correct test classifications	
n	BIC MLMFA	VBMFA
100	88.8 ± .3	89.3 ± .5
200	90.6 ± .4	91.9 ± .3
300	91.1 ± .3	92.7 ± .2
400	91.6 ± .3	92.8 ± .2
500	92.2 ± .3	92.9 ± .2
600	93.0 ± .2	93.3 ± .1
700	93.2 ± .2	93.4 ± .2

Table 4.2: Test classification performance of BIC ML and VB mixture models with increasing data. The standard errors are derived from 10 repetitions of learning with randomly selected training subsets.

introduced to prevent the data likelihood from diverging as a result of the covariance collapsing to zero about any data points.

For the BIC-penalised likelihood, the approximation to the marginal likelihood is given by

$$\ln p(\mathbf{y}) \approx \ln p(\mathbf{y} \,|\, \boldsymbol{\theta}_{\mathrm{ML}}) - \frac{D}{2} \ln n \tag{4.88}$$

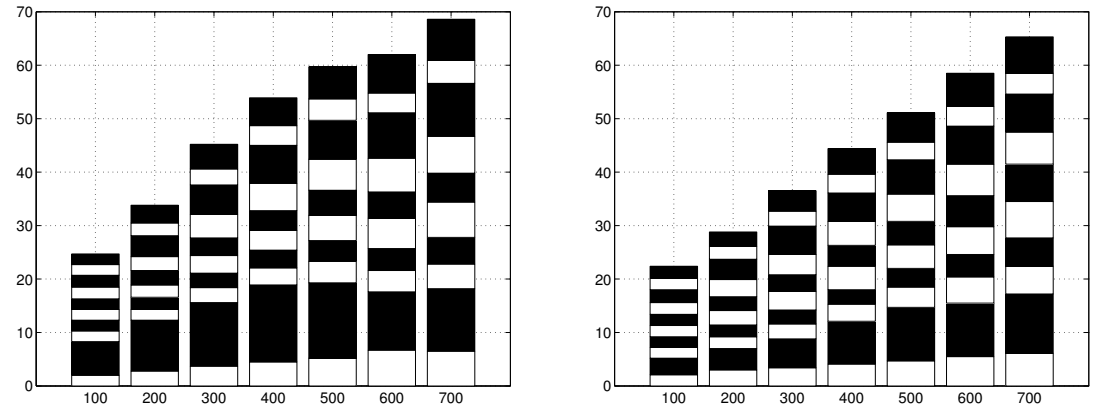
where n is the number of training data (which varied from 100 to 700), and D is the number of degrees of freedom in an MFA model with S analysers with dimensionalities $\{k_s\}_{s=1}^{S}$ (see $d(k)$ of equation (4.5)), which we approximate by

$$D = S - 1 + p + \sum_{s=1}^{S} \left[p + pk_s - \frac{1}{2} k_s(k_s - 1) \right] . \tag{4.89}$$

This quantity is derived from: $S - 1$ degrees of freedom in the prior mixture proportions $\boldsymbol{\pi}$, the number of parameters in the output noise covariance (constrained to be diagonal), p, and the degrees of freedom in the mean and factor loadings of each analyser component. Note that D is only an approximation to the number of degrees of freedom, as discussed in section 4.1.1.

The results of classification experiments for BIC ML and VB are given in table 4.2. VB consistently and significantly outperforms BIC, and in fact surpasses the 92.1% test error performance of the fully-unsupervised VB model on 700 training points. The latter comment is not surprising given that this algorithm receives labelled data. We should note that neither method comes close to state-of-the-art discriminative methods such as support vector machines and convolutional networks, for example *LeNet* (LeCun and Bengio, 1995). This may indicate limitations of the mixture of factor analysers as a generative model for digits.

Figure 4.11 displays the constituents of the mixture models for both BIC and VB for training set sizes {100,200,...,700}. On average, BIC ML tends to use models with slightly more components than does VB, which does not coincide with the common observation that the BIC



(a) BIC. (b) VB.

Figure 4.11: The average number of components used for each digit class by the **(a)** BIC and **(b)** VB models, as the size of the training set increases from 100 to 700 examples. As a visual aid, alternate digits are shaded black and white. The white bottom-most block in each column corresponds to the '0' digit and the black top-most block to the '9' digit. Note that BIC consistently returns a greater total number of components than VB (see text).

penalty over-penalises model complexity. Moreover, BIC produces models with a disproportionate number of components for the '1' digit. VB also does this, but not nearly to the same extent. There may be several reasons for these results, listed briefly below.

First, it may be that the criterion used for terminating the epoch is not operating in the same manner in the VB optimisation as in the ML case — if the ML criterion is ending epochs too early this could easily result in the ML model carrying over some of that epoch's un-plateaued optimisation into the next epoch, to artificially improve the penalised likelihood of the next more complicated model. An extreme case of this problem is the epoch-ending criterion that says "end this epoch just as soon as the penalised likelihood reaches what it was before we added the last component". In this case we are performing a purely exploratory search, as opposed to an exploitative search which plateaus before moving on. Second, the ML model may be concentrating analysers on single data points, despite our precision limit on the noise model. Third, there is no mechanism for component death in the ML MFA model, since in these experiments we did not intervene at any stage to test whether the removal of low responsibility components improved the penalised likelihood (see section 4.3.1). It would be interesting to include such tests, for both ML MFA and VB methods.

4.7 Combining VB approximations with Monte Carlo

In this and other chapters, we have assumed that the variational lower bound is a reliable guide to the log marginal likelihood, using it to infer hidden states, to learn distributions over parameters and especially in this chapter to guide a search amongst models of differing complexity. We have not yet addressed the question of how reliable the bounds are. For example, in section 2.3.2 we mentioned that by using $\mathcal{F}$ for model selection we are implicitly assuming that the KL divergences between the variational and exact posterior distributions over parameters and hidden variables are constant between models. It turns out that we can use the technique of importance sampling to obtain consistent estimators of several interesting quantities, including this KL divergence. In this technique the variational posterior can be used as an importance distribution from which to sample points, as it has been optimised to be representative of the exact posterior distribution.

This section builds on basic claims first presented in Ghahramani and Beal (2000). There it was noted that importance sampling can easily fail for poor choices of importance distributions (personal communication with D. MacKay, see also Miskin, 2000, chapter 4). We also present some extensions to simple importance sampling, including using mixture distributions from several runs of VBEM, and also using heavy-tailed distributions derived from the variational posteriors.

4.7.1 Importance sampling with the variational approximation

Section 4.4 furnishes us with an estimate of the predictive density. Unfortunately this does not even constitute a bound on the predictive density, but a bound on an *approximation* to it. However it is possible to approximate the integrals for such quantities by *sampling*. In this subsection we show how by importance sampling from the variational approximation we can obtain estimators of three important quantities: the exact predictive density, the exact log marginal likelihood $\mathcal{L}$, and the KL divergence between the variational posterior and the exact posterior.

The expectation ε of a function $f(\boldsymbol{\theta})$ under the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y})$ can be written as

$$\varepsilon = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y})\, f(\boldsymbol{\theta}) \;. \tag{4.90}$$

Given that such integrals are usually analytically intractable, they can be approximated by the Monte Carlo average:

$$\hat{\varepsilon}(M) \simeq \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{\theta}^{(m)}) \;, \qquad \boldsymbol{\theta}^{(m)} \sim p(\boldsymbol{\theta} \,|\, \mathbf{y}) \;. \tag{4.91}$$

where $\boldsymbol{\theta}^{(m)}$ are random draws from the posterior $p(\boldsymbol{\theta} \,|\, \mathbf{y})$. In the limit of large number of samples M, $\hat{\varepsilon}$ converges to ε:

$$\lim_{M \to \infty} \hat{\varepsilon}(M) = \varepsilon \; . \tag{4.92}$$

In many models it is not possible to sample directly from the posterior, and so a Markov Chain Monte Carlo approach is usually taken to help explore regions of high posterior probability. In most applications this involves designing tailored Metropolis-Hastings acceptance rules for moving about in the space whilst still maintaining detailed balance.

An alternative to finding samples using MCMC methods is to use *importance sampling*. In this method we express the integral as an expectation over an *importance distribution* $g(\boldsymbol{\theta})$:

$$\varepsilon = \int d\boldsymbol{\theta} \; p(\boldsymbol{\theta} \,|\, \mathbf{y}) \, f(\boldsymbol{\theta}) \tag{4.93}$$

$$= \int d\boldsymbol{\theta} \; g(\boldsymbol{\theta}) \, \frac{p(\boldsymbol{\theta} \,|\, \mathbf{y})}{g(\boldsymbol{\theta})} f(\boldsymbol{\theta}) \tag{4.94}$$

$$\hat{\varepsilon}(M) \simeq \frac{1}{M} \sum_{m=1}^{M} \frac{p(\boldsymbol{\theta}^{(m)} \,|\, \mathbf{y})}{g(\boldsymbol{\theta}^{(m)})} f(\boldsymbol{\theta}^{(m)}) \; , \qquad \boldsymbol{\theta}^{(m)} \sim g(\boldsymbol{\theta}) \; , \tag{4.95}$$

so that now the Monte Carlo estimate (4.95) is taken using samples drawn from $g(\boldsymbol{\theta})$. Weighting factors are required to account for each sample from $g(\boldsymbol{\theta})$ over- or under-representing the actual density we wish to take the expectation under. These are called the *importance weights*

$$\omega^{(m)} = \frac{1}{M} \frac{p(\boldsymbol{\theta} \,|\, \mathbf{y})}{g(\boldsymbol{\theta})} \; . \tag{4.96}$$

This discretisation of the integral then defines a weighted sum of densities:

$$\hat{\varepsilon}(M) = \sum_{m=1}^{M} \omega^{(m)} f(\boldsymbol{\theta}^{(m)}) \; . \tag{4.97}$$

Again, if $g(\boldsymbol{\theta})$ is non-zero wherever $p(\boldsymbol{\theta} \,|\, \mathbf{y})$ is non-zero, it can be shown that $\hat{\varepsilon}$ converges to ε in the limit of large M.

Having used the VBEM algorithm to find a lower bound on the marginal likelihood, we have at our disposal the resulting variational approximate posterior distribution $q(\boldsymbol{\theta})$. Whilst this distribution is not equal to the posterior, it should be a good candidate for an importance distribution because it contains valuable information about the shape and location of the exact posterior, as it was chosen to minimise the KL divergence between it and the exact posterior (setting aside local optima concerns). In addition it usually has a very simple form and so can be sampled from easily. We now describe several quantities that can be estimated with importance sampling using the variational posterior.

Exact predictive density

An asymptotically exact predictive distribution $p(\mathbf{y}' \,|\, \mathbf{y})$ is that given by a weighted average of the likelihood under a set of parameters drawn from the variational posterior $q(\boldsymbol{\theta})$,

$$p(\mathbf{y}' \,|\, \mathbf{y}) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, \mathbf{y})\, p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \tag{4.98}$$

$$= \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta})\, \frac{p(\boldsymbol{\theta} \,|\, \mathbf{y})}{q(\boldsymbol{\theta})} p(\mathbf{y}' \,|\, \boldsymbol{\theta}) \quad \Big/ \quad \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta} \,|\, \mathbf{y})}{q(\boldsymbol{\theta})} \tag{4.99}$$

$$\simeq \frac{1}{M} \sum_{m=1}^{M} \frac{p(\boldsymbol{\theta}^{(m)} \,|\, \mathbf{y})}{q(\boldsymbol{\theta}^{(m)})} p(\mathbf{y}' \,|\, \boldsymbol{\theta}^{(m)}) \quad \Big/ \quad \frac{1}{M} \sum_{o=1}^{M} \frac{p(\boldsymbol{\theta}^{(o)} \,|\, \mathbf{y})}{q(\boldsymbol{\theta}^{(o)})} \tag{4.100}$$

$$= \sum_{m=1}^{M} \omega^{(m)}\, p(\mathbf{y}' \,|\, \boldsymbol{\theta}^{(m)}) \;, \tag{4.101}$$

where $\boldsymbol{\theta}^{(m)} \sim q(\boldsymbol{\theta})$ are samples from the variational posterior, and the ω_m are given by

$$\omega^{(m)} = \frac{p(\boldsymbol{\theta}^{(m)} \,|\, \mathbf{y})}{q(\boldsymbol{\theta}^{(m)})} \quad \Big/ \quad \sum_{o=1}^{M} \frac{p(\boldsymbol{\theta}^{(o)} \,|\, \mathbf{y})}{q(\boldsymbol{\theta}^{(o)})} \tag{4.102}$$

$$= \frac{p(\boldsymbol{\theta}^{(m)}, \mathbf{y})}{q(\boldsymbol{\theta}^{(m)})} \quad \Big/ \quad \sum_{o=1}^{M} \frac{p(\boldsymbol{\theta}^{(o)}, \mathbf{y})}{q(\boldsymbol{\theta}^{(o)})} \tag{4.103}$$

$$= \frac{1}{Z_\omega} \frac{p(\boldsymbol{\theta}^{(m)}, \mathbf{y})}{q(\boldsymbol{\theta}^{(m)})} \;, \tag{4.104}$$

and Z_ω is defined as

$$Z_\omega = \sum_{m=1}^{M} \frac{p(\boldsymbol{\theta}^{(m)}, \mathbf{y})}{q(\boldsymbol{\theta}^{(m)})} \;. \tag{4.105}$$

In the case of MFAs, each such sample $\boldsymbol{\theta}^{(m)}$ is an instance of a mixture of factor analysers with predictive density $p(\mathbf{y}' \,|\, \boldsymbol{\theta}^{(m)})$ as given by (4.11). Since the $\omega^{(m)}$ are normalised to sum to 1, the predictive density for MFAs given in (4.101) represents a *mixture* of mixture of factor analysers.

Note that the step from (4.102) to (4.103) is important because we cannot evaluate the exact posterior density $p(\boldsymbol{\theta}^{(m)} \,|\, \mathbf{y})$, but we can evaluate the *joint* density $p(\boldsymbol{\theta}^{(m)}, \mathbf{y}) = p(\boldsymbol{\theta}^{(m)}) p(\mathbf{y} \,|\, \boldsymbol{\theta}^{(m)})$. Furthermore, note that Z_ω is a function of the weights, and so the estimator in equation (4.101) is really a *ratio* of Monte Carlo estimates. This means that the estimate for $p(\mathbf{y}' \,|\, \mathbf{y})$ is no longer guaranteed to be unbiased. It is however a *consistent* estimator (provided the variances of the numerator and denominator are converging) meaning that as the number of samples tends to infinity its expectation will tend to the exact predictive density.

Exact marginal likelihood

The exact marginal likelihood can be written as

$$\ln p(\mathbf{y}) = \ln\left(\int d\boldsymbol{\theta}\; q(\boldsymbol{\theta})\frac{p(\boldsymbol{\theta},\mathbf{y})}{q(\boldsymbol{\theta})}\right) \tag{4.106}$$

$$= \ln\langle\omega\rangle_{q(\boldsymbol{\theta})} + \ln Z_\omega \tag{4.107}$$

where $\langle\cdot\rangle$ denotes averaging with respect to the distribution $q(\boldsymbol{\theta})$. This gives us an unbiased estimate of the marginal likelihood, but a biased estimate of the log marginal likelihood. Both estimators are consistent however.

KL divergence

This measure of the quality of the variational approximation can be derived by writing $\mathcal{F}$ in the two ways

$$\mathcal{F} = \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta},\mathbf{y})}{q(\boldsymbol{\theta})} \tag{4.108}$$

$$= \langle \ln\omega\rangle_{q(\boldsymbol{\theta})} + \ln Z_\omega, \quad \text{or} \tag{4.109}$$

$$\mathcal{F} = \int d\boldsymbol{\theta}\; q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}\,|\,\mathbf{y})}{q(\boldsymbol{\theta})} + \ln p(\mathbf{y}) \tag{4.110}$$

$$= -\mathrm{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}\,|\,\mathbf{y})) + \ln\langle\omega\rangle_{q(\boldsymbol{\theta})} + \ln Z_\omega. \tag{4.111}$$

By equating these two expressions we obtain a measure of the divergence between the approximating and exact parameter posteriors,

$$\mathrm{KL}(q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}\,|\,\mathbf{y})) = \ln\langle\omega\rangle_{q(\boldsymbol{\theta})} - \langle\ln\omega\rangle_{q(\boldsymbol{\theta})} \tag{4.112}$$

Note that this quantity is not a function of Z_ω, since it was absorbed into the difference of two logarithms. This means that we need not use normalised weights for this measure, and base the importance weights on $p(\boldsymbol{\theta},\mathbf{y})$ rather than $p(\boldsymbol{\theta}\,|\,\mathbf{y})$, and the estimator is unbiased.

Three significant observations should be noted. First, the same importance weights can be used to estimate all three quantities. Second, while importance sampling can work very poorly in high dimensions for *ad hoc* proposal distributions, here the variational optimisation is used in a principled manner to provide a $q(\boldsymbol{\theta})$ that is a good approximation to $p(\boldsymbol{\theta}\,|\,\mathbf{y})$, and therefore hopefully a good proposal distribution. Third, this procedure can be applied to any variational approximation.

4.7.2 Example: Tightness of the lower bound for MFAs

In this subsection we use importance sampling to estimate the tightness of the lower bound in a digits learning problem. In the context of a mixture of factor analysers, $\boldsymbol{\theta} = (\boldsymbol{\pi}, \Lambda, \boldsymbol{\mu}) = \{\pi_s, \tilde{\Lambda}^s\}_{s=1}^S$, and we sample $\boldsymbol{\theta}^{(m)} \sim q(\boldsymbol{\theta}) = q(\boldsymbol{\pi})q(\tilde{\Lambda})$. Each such sample is an instance of a mixture of factor analysers with predictive density given by equation (4.11). Note that Ψ is treated as a hyperparameter so need not be sampled (although we could envisage doing so if we were integrating over Ψ). We weight these predictive densities by the importance weights $w^{(m)} = p(\boldsymbol{\theta}^{(m)}, \mathbf{y})/q(\boldsymbol{\theta}^{(m)})$, which are easy to evaluate. When sampling the parameters $\boldsymbol{\theta}$, one needs only to sample $\boldsymbol{\pi}$ vectors and $\tilde{\Lambda}$ matrices, as these are the only parameters that are required to replicate the generative model of mixture of factor analysers (in addition to the hyperparameter Ψ which has no distribution in our model). Thus the numerator in the importance weights are obtained by calculating

$$p(\boldsymbol{\theta}, \mathbf{y}) = p(\boldsymbol{\pi}, \tilde{\Lambda})p(\mathbf{y} \mid \boldsymbol{\pi}, \tilde{\Lambda}) \tag{4.113}$$

$$= p(\boldsymbol{\pi} \mid \alpha^*, \mathbf{m}^*) \int d\boldsymbol{\nu}\; p(\tilde{\Lambda} \mid \boldsymbol{\nu}, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*)p(\boldsymbol{\nu} \mid a^*, b^*) \prod_{i=1}^n p(\mathbf{y}_i \mid \boldsymbol{\pi}, \tilde{\Lambda}) \tag{4.114}$$

$$= p(\boldsymbol{\pi} \mid \alpha^*, \mathbf{m}^*)p(\tilde{\Lambda} \mid a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) \prod_{i=1}^n p(\mathbf{y}_i \mid \boldsymbol{\pi}, \tilde{\Lambda})\ . \tag{4.115}$$

On the second line we express the prior over the factor loading matrices as a hierarchical prior involving the precisions $\{\boldsymbol{\nu}^s\}_{s=1}^S$. It is not difficult to show that marginalising out the precision for a Gaussian variable yields a multivariate Student-t prior distribution for each row of each $\tilde{\Lambda}^s$, from which we can sample directly. Substituting in the density for an MFA given in (4.11) results in:

$$p(\boldsymbol{\theta}, \mathbf{y}) = p(\boldsymbol{\pi} \mid \alpha^*, \mathbf{m}^*)p(\tilde{\Lambda} \mid a^*, b^*, \boldsymbol{\mu}^*, \boldsymbol{\nu}^*) \prod_{i=1}^n \left[\sum_{s_i=1}^S p(s_i \mid \boldsymbol{\pi})p(\mathbf{y}_i \mid s_i, \tilde{\Lambda}, \Psi) \right]\ . \tag{4.116}$$

The importance weights are then obtained after evaluating the density under the variational distribution $q(\boldsymbol{\pi})q(\tilde{\Lambda})$, which is simple to calculate. Even though we require all the training data to generate the importance weights, once these are made, the importance weights $\{\omega^{(m)}\}_{m=1}^M$ and their locations $\{\boldsymbol{\pi}^{(m)}, \tilde{\Lambda}^{(m)}\}_{m=1}^M$ then capture all the information about the posterior distribution that we will need to make predictions, and so we can discard the training data.

A training data set consisting of 700 examples of each of the digits 0, 1, and 2 was used to train a VBMFA model in a fully-unsupervised fashion. After every successful epoch, the variational posterior distributions over the parameters $\tilde{\Lambda}$ and $\boldsymbol{\pi}$ were recorded. These were then used off-line to produce $M = 100$ importance samples from which a set of importance weights $\{\omega^{(m)}\}_{m=1}^M$ were calculated. Using results of the previous section, these weights were used to estimate the following quantities: the log marginal likelihood, the KL divergence between the variational

posterior $q(\boldsymbol{\pi})q(\tilde{\Lambda})$ and the exact posterior $p(\boldsymbol{\pi}, \tilde{\Lambda} \,|\, \mathbf{y})$, and the KL divergence between the full variational posterior over all hidden variables and parameters and the exact full posterior. The latter quantity is simply the difference between the estimate of the log marginal likelihood and the lower bound $\mathcal{F}$ used in the optimisation (see equation (4.29)).

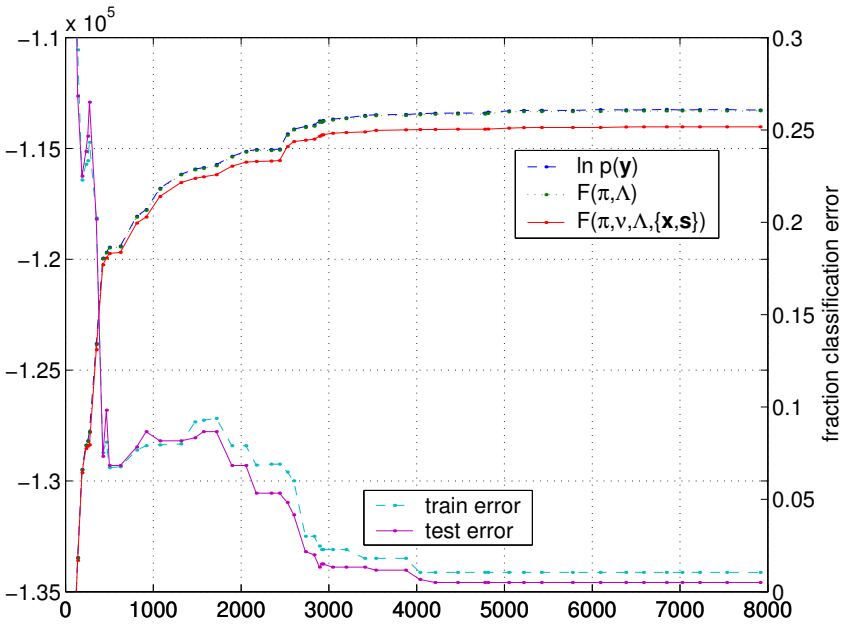
Figure 4.12(a) shows these results plotted alongside the training and test classification errors. We can see that for the most part the lower bound, calculated during the optimisation and denoted $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$ to indicate that it is computed from variational distributions over parameters and hidden variables, is close to the estimate of the log marginal likelihood $\ln p(\mathbf{y})$, and more importantly remains roughly in tandem with it throughout the optimisation. The training and test errors are roughly equal and move together, suggesting that the variational Bayesian model is not overfitting the data. Furthermore, upward changes to the log marginal likelihood are for the most part accompanied by downward changes to the test error rate, suggesting that the marginal likelihood is a good measure for classification performance in this scenario. Lastly, the estimate of the lower bound $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$, which is computed by inserting the importance weights into (4.109), is very close to the estimate of the log marginal likelihood (the difference is made more clear in the accompanying figure 4.12(b)). This means that the KL divergence between the variational and exact posteriors over $(\boldsymbol{\pi}, \tilde{\Lambda})$ is fairly small, suggesting that the majority of the gap between $\ln p(\mathbf{y})$ and $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$ is due to the KL divergence between the variational posterior and exact posteriors over the hidden variables $(\boldsymbol{\nu}, \mathbf{x}, \mathbf{s})$.

Aside: efficiency of the structure search

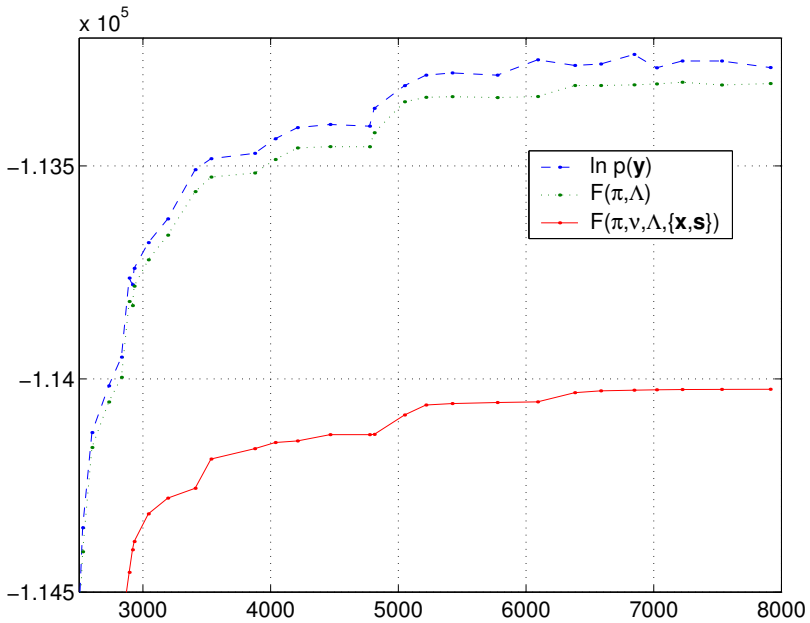
During the optimisation, there were 52 accepted epochs, and a total of 692 proposed component splits (an acceptance rate of only about 7%), resulting in 36 components. However it is clear from the graph (see also figure 4.13(c)) that the model structure does not change appreciably after about 5000 iterations, at which point 41 epochs have been accepted from 286 proposals. This corresponds to an acceptance rate of 14% which suggests that our heuristics for choosing which component to split and how to split it are performing well, given the number of components to chose from and the dimensionality of the data space.

Analysis of the lower bound gap

Given that 100 samples may be too few to obtain reliable estimates, the experiment was repeated with 6 runs of importance sampling, each with 100 samples as before. Figures 4.13(a) and 4.13(b) show the KL divergence measuring the distance between the log marginal likelihood estimate and the lower bounds $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$ and $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$, respectively, as the optimisation proceeds. Figure 4.13(c) plots the number of components, S, in the mixture with iterations of EM, and it is quite clear that the KL divergences in the previous two graphs correlate closely



(a)



(b)

Figure 4.12: **(a)** Log marginal likelihood estimates from importance sampling with iterations of VBEM. Each point corresponds to the model at the end of a successful epoch of learning. The fraction of training and test classification errors are shown on the right vertical axis, and the lower bound $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$ that guides the optimisation on the left vertical axis. Also plotted is $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$, but this is indistinguishable from the other lower bound. The second plot **(b)** is exactly the same as **(a)** except the log marginal likelihood axis has been rescaled to make clear the difference between the log marginal likelihood and the bound $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$.

with the number of components. This observation is borne out explicitly in figures 4.13(d) and 4.13(d) where it is clear that the KL divergence between the lower bound $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$ and the marginal likelihood is roughly proportional to the number of components in the mixture. This is true to an extent also for the lower bound estimate $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$ although this quantity is more noisy. These two observations are unlikely to be artifacts of the sampling process, as the variances are much smaller than the trend. In section 2.3.2 we noted that if the KL discrepancy increases with S then the model exploration may be biased to simpler models. Here we have found some evidence of this, which suggests that variational Bayesian methods may suffer from a tendency to underfit the model structure.

4.7.3 Extending simple importance sampling

Why importance sampling is dangerous

Unfortunately, the importance sampling procedure that we have used is notoriously bad in high dimensions. Moreover, it is easy to show that importance sampling can fail even for just one dimension: consider computing expectations under a one dimensional Gaussian $p(\theta)$ with precision ν_p using an importance distribution $q(\theta)$ which is also a Gaussian with precision ν_q and the same mean. Although importance sampling can give us unbiased estimates, it is simple to show that if $\nu_q > 2\nu_p$ then the variance of the importance weights will be infinite! We briefly derive this result here. The importance weight for the sample drawn from $q(\theta)$ is given by

$$\omega(\theta) = \frac{p(\theta)}{q(\theta)} \,, \tag{4.117}$$

and the variance of the importance weights can be written

$$\begin{aligned} var(\omega) &= \langle \omega^2 \rangle_{q(\theta)} - \langle \omega \rangle^2_{q(\theta)} && (4.118)\\ &= \int d\theta \; q(\theta) \left(\frac{p(\theta)}{q(\theta)} \right)^2 - \left(\int d\theta \; q(\theta) \frac{p(\theta)}{q(\theta)} \right)^2 && (4.119)\\ &= \frac{\nu_p}{\nu_q^{1/2}} \int d\theta \; \exp\left[-\left(\nu_p - \frac{1}{2}\nu_q \right) \theta^2 + k\theta + k' \right] \quad - 1 \,, && (4.120)\\ &= \begin{cases} \nu_p \nu_q^{-1/2} \left(2\nu_p - \nu_q \right)^{-1/2} - 1 & \text{for } \; 2\nu_p > \nu_q \\ \infty & \text{for } \; 2\nu_p \leq \nu_q \end{cases} \,. && (4.121) \end{aligned}$$

where k and k' are constants independent of x. For $2\nu_p \leq \nu_q$, the integral diverges and the variance of the weights is infinite. Indeed this problem is exacerbated in higher dimensions, where if this condition is not met in any dimension of parameter space, then the importance weights will have infinite variance. The intuition behind this is that we need the tails of the sampling distribution $q(\theta)$ to fall off slower than the true distribution $p(\theta)$, otherwise there exists some probability

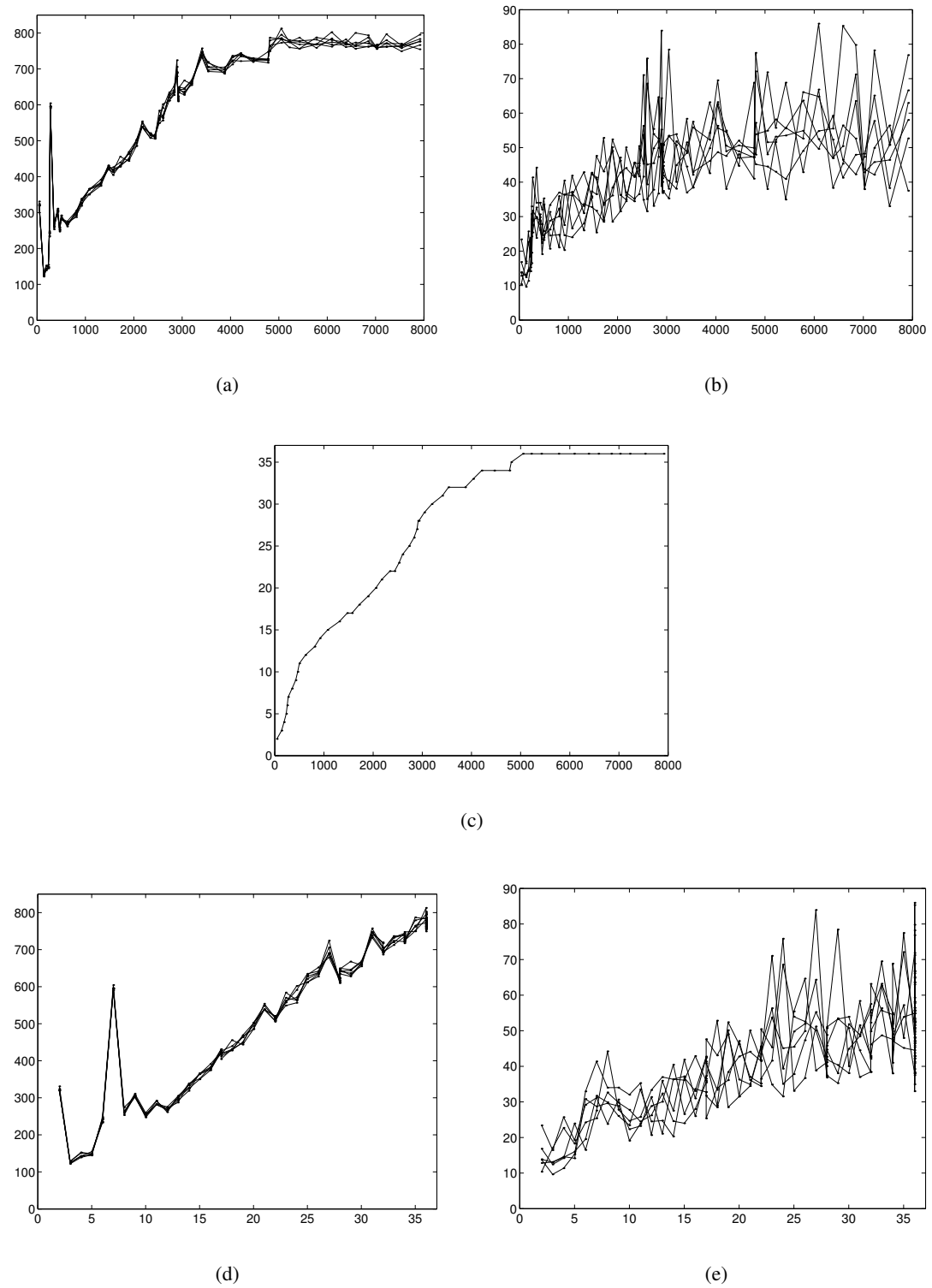


Figure 4.13: At the end of every accepted epoch 6 estimates of the log marginal likelihood were calculated (see text). **(a)** Differences between the log marginal likelihood estimate and the lower bound $\mathcal{F}(\boldsymbol{\pi}, \boldsymbol{\nu}, \tilde{\Lambda}, \mathbf{x}, \mathbf{s})$, as a function of iterations of VBEM. **(b)** Differences between the log marginal likelihood estimate and the lower bound $\mathcal{F}(\boldsymbol{\pi}, \tilde{\Lambda})$. **(c)** Number of components S in the mixture model with iterations of VBEM. **(d)** The same data as in (a), plotted against the number of components S, as given in (c). **(e)** As for (d) but using the data from (b) instead of (a).

that we obtain a very high importance weight. This result is clearly a setback for importance sampling using the variational posterior distribution, since the variational posterior tends to be tighter than the exact posterior, having neglected correlations between some parameters in order to make inference tractable. To complete the argument, we should mention that importance sampling becomes very difficult in high dimensions even if this condition is met, since: firstly, samples from the typical set of the $q(\theta)$ are unlikely to have high probability under $p(\theta)$, unless the distributions are very similar; secondly, even if the distributions are well matched, the weights have a wide range that scales order $\exp(r^{1/2})$, where r is the dimensionality (MacKay, 1999).

The above result (4.121) is extended in Miskin (2000, chapter 4), where the finite variance condition is derived for general $p(\theta)$ and $q(\theta)$ in the exponential family. Also in that work, a bound is derived for the variance of the importance weights when using a finite mixture distribution as the importance distribution (equation 4.31 of that manuscript). This mixture is made from the variational posterior distribution mixed with a set of broader distributions from the *same* exponential family. The rationale for this approach is precisely to create heavier-tailed importance distributions. Unfortunately the bound is not very tight, and the simulations therein report no increase in convergence to the correct expectation.

In addition to these problems, the exact posterior over the parameters can be very multi-modal. The most benign form of such multi-modality is due to aliases arising from having likelihood functions which are invariant to exchanges of labelling of hidden variables, for example indicator variables for components in a mixture. In such cases the variational posterior tends to lock on to one mode and so, when used in an importance sampler, the estimate represents only a fraction of the marginal likelihood. If the modes are well-separated then simple degeneracies of this sort can be accounted for by multiplying the result by the number of aliases. If the modes are overlapping, then a correction should not be needed as we expect the importance distribution to be broad enough. However if the modes are only partially separated then the correction factor is difficult to compute. In general, these corrections cannot be made precise and should be avoided.

Using heavy-tailed and mixture distributions

Here we investigate the effect of two modifications to the naive use of the variational posterior as importance distribution. The first modification considers replacing the variational posterior entirely by a related heavy-tailed Student-t distribution. The second modification uses a stochastic *mixture* distribution for the importance distribution, with each component being the variational posterior obtained from a different VBEM optimisation.

The Student-t can be derived by considering the marginal probability of Gaussian distributed variables under a conjugate gamma distribution for the precision, γ, which is for the univariate case:

$$q_{\mathrm{St}}(\theta) = \int d\gamma \; p(\gamma \,|\, a, b) p(\theta \,|\, \mu, \gamma^{-1}) \tag{4.122}$$

$$= \int d\gamma \; \mathrm{Ga}(\gamma \,|\, a, b) \mathrm{N}(\theta \,|\, \mu, \gamma^{-1}) \tag{4.123}$$

$$= \frac{b^a}{\Gamma(a)\sqrt{2\pi}} \int d\gamma \; e^{-(b+(\theta-\mu)^2/2)\gamma} \gamma^{a-1/2} \tag{4.124}$$

$$= \frac{1}{Z_{\mathrm{St}}(a,b)} \left(1 + \frac{(\theta-\mu)^2}{2b}\right)^{-(a+1/2)} \tag{4.125}$$

where a and b are the shape and inverse-scale respectively of the precision distribution, and Z_{St} is given by

$$Z_{\mathrm{St}}(a,b) = \frac{\Gamma(a+\frac{1}{2})}{\Gamma(a)\sqrt{2\pi b}} \,, \qquad \text{for } \; a > 0, \; b > 0 \,. \tag{4.126}$$

It is straightforward to show that the variance of θ is given by $b/(a-1)$ and the kurtosis by $3(a-1)/(a-2)$ (see appendix A). The degrees of freedom ν and dispersion parameter σ^2 can be arrived at with the following equivalence:

$$\nu = 2a \,, \qquad \sigma^2 = \frac{b}{a} \,. \tag{4.127}$$

The attraction of using this distribution for sampling is that it has heavier tails, with a polynomial rather than exponential decay. In the limit of $\nu \to \infty$ the Student-t is a Gaussian distribution, while for $\nu = 1$ it is a Cauchy distribution.

Three 2-dimensional data sets were generated by drawing 150 samples from 4 Gaussian clusters, with varying separations of their centres, as shown in figure 4.14. For each data set, 10 randomly initialised VBEM algorithms were run to learn a model of the data. If any of the learnt models contained fewer or more than 4 components, that optimisation was discarded and replaced with another. We would expect that for the well-separated data set the exact posterior distribution over the parameters would consist of tight, well-separated modes. Conversely, for the overlapping data set we would expect the posterior to be very broad consisting of several weakly-defined peaks. In the intermediately-spaced data set we would expect the posterior to be mostly separated modes with some overlap.

The following importance samplers were constructed, separately for each data set, and are summarised in table 4.3: (1) a single model out of the 10 that were trained was randomly chosen (once) and its variational posterior $q(\boldsymbol{\pi})q(\tilde{\Lambda})$ used as the importance distribution; (2) the covariance parameters of the variational posterior $q(\tilde{\Lambda})$ of that same model were used as the covariance parameters in t-distributions with 3 degrees of freedom to form $q^{(3)}(\tilde{\Lambda})$, and this used in conjunction with the same $q(\boldsymbol{\pi})$ to form the importance distribution $q(\boldsymbol{\pi})q^{(3)}(\tilde{\Lambda})$; (3) the same as

sampler key	type of importance dist.	each component's			
		form	dof	relative variance	kurtosis
1	single 1	Gaussian	∞	1	3
2	single 1	Student-t	3	3	3.75
3	single 1	Student-t	2	∞	4.5
4	mixture of 10	Gaussian	∞		
5	mixture of 10	Student-t	3	ditto	ditto
6	mixture of 10	Student-t	2		

Table 4.3: The specifications of six importance sampling distributions.

(2) but using 2 degrees of freedom; samplers (4,5,6) are the same as (1,2,3) except the operations are carried out on every one of the 10 models returned, to generate a mixture model with 10 equally weighted mixture components.

Recall that the covariance matrix for the entries of the $\tilde{\Lambda}^s$ matrix for each analyser is of block diagonal form, and so each row can be sampled from independently to produce the importance samples. Furthermore, generating the multivariate Student-t samples from these covariances is a straightforward procedure using standard methods.

Figure 4.14 shows the results of attempting to estimate the marginal likelihood of the three different data sets, using the 6 differently constructed importance samplers given in table 4.3, which are denoted by the labels 1–6. The axis marks F and F' correspond to lower bounds on the log marginal likelihood: F is the lower bound reported by the single model used for the single set of importance samplers (i.e. 1,2,3); and F' is the highest reported lower bound of all 10 of the models trained on that data set. The error bars correspond to the unbiased estimate of the standard deviation in the estimates from five separate runs of importance sampling. We can see several interesting features. First, all the estimates (1-6) using different importance distributions yield estimates greater than the highest lower bound (F'). Second, the use of heavier-tailed and broader Student-t distributions for the most part increases the estimate, whether based on single or mixture importance distributions. Also, the move from 3 to 2 degrees of freedom (i.e. (2) to (3), or (5) to (6) in the plot) for the most part increases the estimate further. These observations suggest that there exists mass outside of the variational posterior that is neglected with the Gaussian implementations (1,4). Third, using mixture distributions increases the estimates. However, this increase from (1,2,3) to (4,5,6) is roughly the same as the increase in lower bounds from F to F'. This implies that the single estimates are affected if using a sub-optimal solution, whereas the mixture distribution can perform approximately as well as its best constituent solution. It should be noted that only the highest lower bound, F', was plotted for each data set, as plotting the remaining 9 lower bounds would have extended the graphs' y-axes too much to be able to visually resolve the differences in the methods (in all three data sets there were at least two poor optimisations).

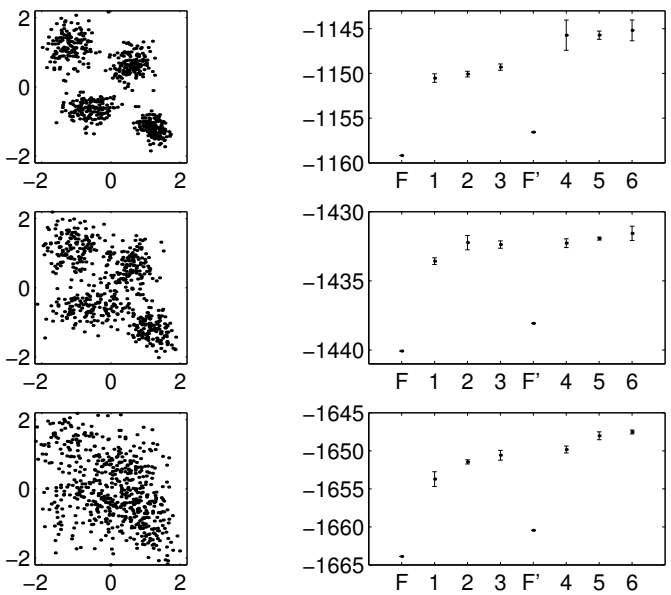


Figure 4.14: **(right)** Importance sampling estimates of the marginal likelihoods of VBMFA models trained on **(left)** three data sets of differently spaced Gaussian clusters. In the plots in the right column, the vertical axis is the log of the marginal likelihood estimate, and the horizontal axis denotes which importance sampling method is used for the estimate, as given in table 4.3. The estimates are taken from five separate runs of importance sampling, with each run consisting of 4000 samples; the error bars are the standard errors in the estimate, assuming the logarithm of the estimates from the five runs are Gaussian distributed. The axis mark F corresponds to the lower bound from the model used for the single samplers (1,2,3), and the mark F' corresponds to the highest lower bound from the 10 models used in the mixture samplers (4,5,6).

4.8 Summary

In this chapter we have shown that how the marginal likelihood of a mixture of factor analysers is intractable, and derived a tractable deterministic variational lower bound which can be optimised using a variational EM algorithm. We can use the lower bound to guide a search among model structures using birth and death moves. We can also use the lower bound to obtain a distribution over structures if desired: $p(m \,|\, \mathbf{y}) \propto p(m)p(\mathbf{y} \,|\, m) \approx p(m) \cdot e^{\mathcal{F}_{opt}(m)}$, with the caveat that there is no guarantee that the best achieved lower bound, $\mathcal{F}_{opt}(m)$, is similarly tight across different models m. Indeed we have found that the KL divergence between the variational and exact posterior over parameters increases approximately linearly with the number of components in the mixture, which suggests a systematic tendency to underfit (refer to page 60).

We have derived a generally applicable importance sampler based on the variational solution, which gives us consistent estimates of the exact marginal likelihood, the exact predictive density, and the KL divergence between the variational posterior and the exact posterior. We have also investigated the use of heavy-tailed and mixture distributions for improving the importance sampler estimates, but there are theoretical reasons for why methods more sophisticated than importance sampling are required for reliable estimates.

It is also possible to integrate the variational optimisation into the proposal distribution for an MCMC sampling method (NIPS workshop: *Advanced Mean Field Methods*, Denver CO, December 1999; personal communication with N. de Freitas, July 2000). The combined procedures combine the relative advantages of the two methods, namely the asymptotic correctness of sampling, and the rapid and deterministic convergence of variational methods. Since the variational optimisation can quickly provide us with an approximation to the shape of the local posterior landscape, the MCMC transition kernel can be adapted to utilise this information to more accurately explore and update that approximation. One would hope that this refined knowledge could then be used to update the variational posterior, and the process iterated. Unfortunately, in its simplest form, this MCMC *adaption* can not be done infinitely often, as it disrupts the stationary distribution of the chain (although see Gilks et al., 1998, for a *regeneration* technique). In de Freitas et al. (2001), a variational MCMC method that includes mixture transition kernels is described and applied to the task of finding the moments of posterior distributions in a sigmoid belief network. There remain plenty of directions of research for such combinations of variational and MCMC methods.

The VB mixtures formalism has been applied to more complicated variants of MFA models recently, with a view to determining the number of components and the local manifold dimensionalities. For example, mixtures of independent components analysers (Choudrey and Roberts, 2002), and mixtures of independent components analysers with non-symmetric sources (Chan et al., 2002).

There have been other Bayesian approaches to modelling densities using mixture distributions. One notable example is the infinite Gaussian mixture model of Rasmussen (2000), which uses sampling to entertain a countably infinite number of mixture components, rather than any particular finite number. In that work, when training on the Spiral data set (examined in section 4.5.3 of this thesis), it was found that on average about 18–20 of the infinitely many Gaussian components had data associated with them. Our VB method usually found between 12–14 analyser components. Examining the differences between the models returned, and perhaps more importantly the predictions made, by these two algorithms is an interesting direction of research.

Search over model structures for MFAs is computationally intractable if each factor analyser is allowed to have different intrinsic dimensionalities. In this chapter we have shown how the variational Bayesian approach can be used to efficiently infer the structure of the model whilst avoiding overfitting and other deficiencies of ML approaches. We have also shown how we can simultaneously infer both the number of analysers and their dimensionalities using birth-death steps and ARD methods, all based on a variational lower bound on the marginal likelihood.

Chapter 5

Variational Bayesian Linear Dynamical Systems

5.1 Introduction

This chapter is concerned with the variational Bayesian treatment of Linear Dynamical Systems (LDSs), also known as linear-Gaussian state-space models (SSMs). These models are widely used in the fields of signal filtering, prediction and control, because: (1) many systems of interest can be approximated using linear systems, (2) linear systems are much easier to analyse than nonlinear systems, and (3) linear systems can be estimated from data efficiently. State-space models assume that the observed time series data was generated from an underlying sequence of unobserved (hidden) variables that evolve with Markovian dynamics across successive time steps. The filtering task attempts to infer the likely values of the hidden variables that generated the current observation, given a sequence of observations up to and including the current observation; the prediction task tries to simulate the unobserved dynamics one or many steps into the future to predict a future observation.

The task of deciding upon a suitable dimension for the hidden state space remains a difficult problem. Traditional methods, such as early stopping, attempt to reduce generalisation error by terminating the learning algorithm when the error as measured on a hold-out set begins to increase. However the hold-out set error is a noisy quantity and for a reliable measure a large set of data is needed. We would prefer to learn from all the available data, in order to make predictions. We also want to be able to obtain posterior distributions over all the parameters in the model in order to quantify our uncertainty.

We have already shown in chapter 4 that we can infer the dimensionality of the hidden variable space (i.e. the number of factors) in a mixture of factor analysers model, by placing priors on

the factor loadings which then implement automatic relevance determination. Linear-Gaussian state-space models can be thought of as factor analysis through time with the hidden factors evolving with noisy linear dynamics. A variational Bayesian treatment of these models provides a novel way to learn their structure, i.e. to identify the optimal dimensionality of their state space.

With suitable priors the LDS model is in the conjugate-exponential family. This chapter presents an example of variational Bayes applied to a conjugate-exponential model, which therefore results in a VBEM algorithm which has an approximate inference procedure with the same complexity as the MAP/ML counterpart, as explained in chapter 2. Unfortunately, the implementation is not as straightforward as in other models, for example the Hidden Markov Model of chapter 3, as some subparts of the parameter-to-natural parameter mapping are non-invertible.

The rest of this chapter is written as follows. In section 5.2 we review the LDS model for both the standard and input-dependent cases, and specify conjugate priors over all the parameters. In 5.3 we use the VB lower bounding procedure to approximate the Bayesian integral for the marginal likelihood of a sequence of data under a particular model, and derive the VBEM algorithm. The VBM step is straightforward, but the VBE step is much more interesting and we fully derive the forward and backward passes analogous to the Kalman filter and Rauch-Tung-Striebel smoothing algorithms, which we call the *variational Kalman filter* and *smoother* respectively. In this section we also discuss hyperparameter learning (including optimisation of automatic relevance determination hyperparameters), and also show how the VB lower bound can be computed. In section 5.4 we demonstrate the model's ability to discover meaningful structure from synthetically generated data sets (in terms of the dimension of the hidden state space etc.). In section 5.5 we present a very preliminary application of the VB LDS model to real DNA microarray data, and attempt to discover underlying mechanisms in the immune response of human T-lymphocytes, starting from T-cell receptor activation through to gene transcription events in the nucleus. In section 5.6 we suggest extensions to the model and possible future work, and in section 5.7 we provide some conclusions.

5.2 The Linear Dynamical System model

5.2.1 Variables and topology

In state-space models (SSMs), a sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ of p-dimensional real-valued observation vectors, denoted $\mathbf{y}_{1:T}$, is modelled by assuming that at each time step t, $\mathbf{y}_t$ was generated from a k-dimensional real-valued hidden state variable $\mathbf{x}_t$, and that the sequence of $\mathbf{x}$'s follow

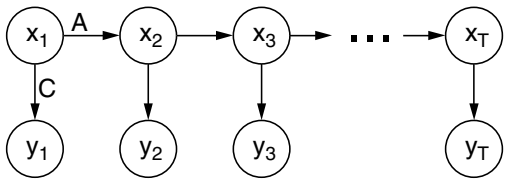


Figure 5.1: Graphical model representation of a state-space model. The hidden variables $\mathbf{x}_t$ evolve with Markov dynamics according to parameters in A, and at each time step generate an observation $\mathbf{y}_t$ according to parameters in C.

a first-order Markov process. The joint probability of a sequence of states and observations is therefore given by:

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{y}_1 \,|\, \mathbf{x}_1) \prod_{t=2}^{T} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})p(\mathbf{y}_t \,|\, \mathbf{x}_t) \;. \tag{5.1}$$

This factorisation of the joint probability can be represented by the graphical model shown in figure 5.1. For the moment we consider just a single sequence, not a batch of i.i.d. sequences. For ML and MAP learning there is a straightforward extension for learning multiple sequences; for VB learning the extensions are outlined in section 5.3.8.

The form of the distribution $p(\mathbf{x}_1)$ over the first hidden state is Gaussian, and is described and explained in more detail in section 5.2.2. We focus on models where both the dynamics, $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, and output functions, $p(\mathbf{y}_t \,|\, \mathbf{x}_t)$, are linear and time-invariant and the distributions of the state evolution and observation noise variables are Gaussian, i.e. linear-Gaussian state-space models:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{w}_t \;, \qquad \mathbf{w}_t \sim \mathrm{N}(\mathbf{0}, Q) \tag{5.2}$$

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t \;, \qquad \mathbf{v}_t \sim \mathrm{N}(\mathbf{0}, R) \tag{5.3}$$

where A $(k \times k)$ is the state dynamics matrix, C $(p \times k)$ is the observation matrix, and Q $(k \times k)$ and R $(p \times p)$ are the covariance matrices for the state and output noise variables $\mathbf{w}_t$ and $\mathbf{v}_t$. The parameters A and C are analogous to the transition and emission matrices respectively in a Hidden Markov Model (see chapter 3). Linear-Gaussian state-space models can be thought of as factor analysis where the low-dimensional (latent) factor vector at one time step diffuses linearly with Gaussian noise to the next time step.

We will use the terms 'linear dynamical system' (LDS) and 'state-space model' (SSM) interchangeably throughout this chapter, although they emphasise different properties of the model. LDS emphasises that the dynamics are linear – such models can be represented either in state-space form or in input-output form. SSM emphasises that the model is represented as a latent-variable model (i.e. the observables are generated via some hidden states). SSMs can be non-

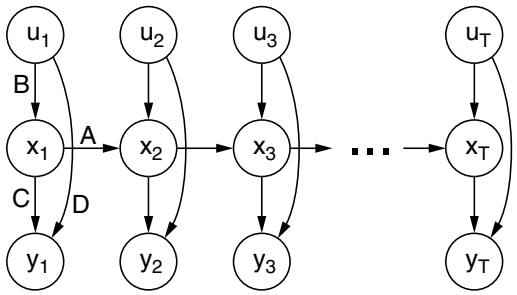


Figure 5.2: The graphical model for linear dynamical systems with inputs.

linear in general; here it should be assumed that we refer to linear models with Gaussian noise except if stated otherwise.

A straightforward extension to this model is to allow both the dynamics and observation model to include a dependence on a series of d-dimensional driving inputs $\mathbf{u}_{1:T}$:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{w}_t \tag{5.4}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t + \mathbf{v}_t \ . \tag{5.5}$$

Here B $(k \times d)$ and D $(p \times d)$ are the input-to-state and input-to-observation matrices respectively. If we now augment the driving inputs with a constant bias, then this input driven model is able to incorporate an arbitrary origin displacement for the hidden state dynamics, and also can induce a displacement in the observation space. These displacements can be learnt as parameters of the input-to-state and input-to-observation matrices.

Figure 5.2 shows the graphical model for an input-dependent linear dynamical system. An input-dependent model can be used to model control systems. Another possible way in which the inputs can be utilised is to feedback the outputs (data) from previous time steps in the sequence into the inputs for the current time step. This means that the hidden state can concentrate on modelling hidden factors, whilst the Markovian dependencies between successive *outputs* are modelled using the output-input feedback construction. We will see a good example of this type of application in section 5.5, where we use it to model gene expression data in a DNA microarray experiment.

On a point of notational convenience, the probability statements in the later derivations leave implicit the dependence of the dynamics and output processes on the driving inputs, since for each sequence they are fixed and merely modulate the processes at each time step. Their omission keeps the equations from becoming unnecessarily complicated.

Without loss of generality we can set the hidden state evolution noise covariance, Q, to the identity matrix. This is possible since an arbitrary noise covariance can be incorporated into the state dynamics matrix A, and the hidden state rescaled and rotated to be made commensurate with

this change (see Roweis and Ghahramani, 1999, page 2 footnote); these changes are possible since the hidden state is unobserved, by definition. This is the case in the maximum likelihood scenario, but in the MAP or Bayesian scenarios this degeneracy is lost since various scalings in the parameters will be differently penalised under the parameter priors (see section 5.2.2 below).

The remaining parameter of a linear-Gaussian state-space model is the covariance matrix, R, of the Gaussian output noise, $\mathbf{v}_t$. In analogy with factor analysis we assume this to be diagonal. Unlike the hidden state noise, Q, there is no degeneracy in R since the data is observed, and therefore its scaling is fixed and needs to be learnt.

For notational convenience we collect the above parameters into a single parameter vector for the model: $\boldsymbol{\theta} = (A, B, C, D, R)$.

We now turn to considering the LDS model for a Bayesian analysis. From (5.1), the complete-data likelihood for linear-Gaussian state-space models is Gaussian, which is in the class of exponential family distributions, thus satisfying condition 1 (2.80). In order to derive a variational Bayesian algorithm by applying the results in chapter 2 we now build on the model by defining conjugate priors over the parameters according to condition 2 (2.88).

5.2.2 Specification of parameter and hidden state priors

The description of the priors in this section may be made more clear by referring to figure 5.3. The forms of the following prior distributions are motivated by conjugacy (condition 2, (2.88)). By writing every term in the complete-data likelihood (5.1) explicitly, we notice that the likelihood for state-space models factors into a product of terms for every *row* of each of the dynamics-related and output-related matrices, and the priors can therefore be factorised over the hidden variable and observed data dimensions.

The prior over the output noise covariance matrix R, which is assumed diagonal, is defined through the precision vector $\boldsymbol{\rho}$ such that $R^{-1} = \mathrm{diag}\,(\boldsymbol{\rho})$. For conjugacy, each dimension of $\boldsymbol{\rho}$ is assumed to be gamma distributed with hyperparameters a and b:

$$p(\boldsymbol{\rho} \,|\, a, b) = \prod_{s=1}^{p} \frac{b^a}{\Gamma(a)} \rho_s^{a-1} \exp\{-b\rho_s\}. \tag{5.6}$$

More generally, we could let R be a full covariance matrix and still be conjugate: its inverse $V = R^{-1}$ would be given a Wishart distribution with parameter S and degrees of freedom ν:

$$p(V \,|\, \nu, S) \propto |V|^{(\nu-p-1)/2} \exp\left[-\frac{1}{2}\mathrm{tr}\; VS^{-1}\right], \tag{5.7}$$

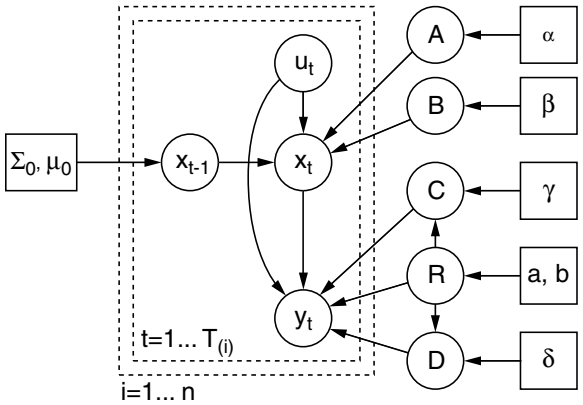


Figure 5.3: Graphical model representation of a Bayesian state-space model. Each sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_{T_i}\}$ is now represented succinctly as the (inner) plate over T_i pairs of hidden variables, each presenting the cross-time dynamics and output process. The second (outer) plate is over the data set of size n sequences. For the most part of the derivations in this chapter we restrict ourselves to $n = 1$, and $T_n = T$. Note that the plate notation used here is non-standard since both $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ have to be included in the plate to denote the dynamics.

where tr is the matrix trace operator. This more general form is not adopted in this chapter as we wish to maintain a parallel between the output model for state-space models and the factor analysis model (as described in chapter 4).

Priors on A, B, C and D

The row vector $\mathbf{a}_{(j)}^\top$ is used to denote the jth row of the dynamics matrix, A, and is given a zero mean Gaussian prior with precision equal to $\mathrm{diag}\,(\boldsymbol{\alpha})$, which corresponds to axis-aligned covariance and can possibly be non-spherical. Each row of C, denoted $\mathbf{c}_{(s)}^\top$, is given a zero-mean Gaussian prior with precision matrix equal to $\mathrm{diag}\,(\rho_s \boldsymbol{\gamma})$. The dependence of the precision of $\mathbf{c}_{(s)}$ on the noise output precision ρ_s is motivated by conjugacy (as can be seen from the explicit complete-data likelihood), and intuitively this prior links the scale of the signal to the noise. We place similar priors on the rows of the input-related matrices B and D, introducing two more hyperparameter vectors $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$. A useful notation to summarise these forms is

$$p(\mathbf{a}_{(j)} \,|\, \boldsymbol{\alpha}) = \mathrm{N}(\mathbf{a}_{(j)} \,|\, \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\alpha})^{-1}) \tag{5.8}$$

$$p(\mathbf{b}_{(j)} \,|\, \boldsymbol{\beta}) = \mathrm{N}(\mathbf{b}_{(j)} \,|\, \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\beta})^{-1}) \qquad \text{for } j = 1, \ldots, k \tag{5.9}$$

$$p(\mathbf{c}_{(s)} \,|\, \rho_s, \boldsymbol{\gamma}) = \mathrm{N}(\mathbf{c}_{(s)} \,|\, \mathbf{0}, \rho_s^{-1}\, \mathrm{diag}\,(\boldsymbol{\gamma})^{-1}) \tag{5.10}$$

$$p(\mathbf{d}_{(s)} \,|\, \rho_s, \boldsymbol{\delta}) = \mathrm{N}(\mathbf{d}_{(s)} \,|\, \mathbf{0}, \rho_s^{-1}\, \mathrm{diag}\,(\boldsymbol{\delta})^{-1}) \tag{5.11}$$

$$p(\rho_s \,|\, a, b) = \mathrm{Ga}(\rho_s \,|\, a, b) \qquad \text{for } s = 1, \ldots, p \tag{5.12}$$

such that $\mathbf{a}_{(j)}$ etc. are column vectors.

The Gaussian priors on the transition (A) and output (C) matrices can be used to perform 'automatic relevance determination' (ARD) on the hidden dimensions. As an example consider the matrix C which contains the linear embedding factor loadings for each factor in each of its columns: these factor loadings induce a high dimensional oriented covariance structure in the data ($CC^\top$), based on an embedding of low-dimensional axis-aligned (unit) covariance. Let us first fix the hyperparameters $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_k\}$. As the parameters of the C matrix are learnt, the prior will favour entries close to zero since its mean is zero, and the degree with which the prior enforces this zero-preference varies across the columns depending on the size of the precisions in $\boldsymbol{\gamma}$. As learning continues, the burden of modelling the covariance in the p output dimensions will be gradually shifted onto those hidden dimensions for which the entries in $\boldsymbol{\gamma}$ are smallest, thus resulting in the least penalty under the prior for non-zero factor loadings. When the hyperparameters are updated to reflect this change, the unequal sharing of the output covariance is further exacerbated. The limiting effect as learning progresses is that some columns of C become zero, coinciding with the respective hyperparameters tending to infinity. This implies that those hidden state dimensions do not contribute to the covariance structure of data, and so can be removed entirely from the output process.

Analogous ARD processes can be carried out for the dynamics matrix A. In this case, if the jth column of A should become zero, this implies that the jth hidden dimension at time $t-1$ is not involved in generating the hidden state at time t (the rank of the transformation A is reduced by 1). However the jth hidden dimension may still be of use in producing covariance structure in the data via the modulatory input at each time step, and should not necessarily be removed unless the entries of the C matrix also suggest this.

For the input-related parameters in B and D, the ARD processes correspond to selecting those particular inputs that are relevant to driving the dynamics of the hidden state (through $\boldsymbol{\beta}$), and selecting those inputs that are needed to directly modulate the observed data (through $\boldsymbol{\delta}$). For example the (constant) input bias that we use here to model an offset in the data mean will almost certainly always remain non-zero, with a correspondingly small value in $\boldsymbol{\delta}$, unless the mean of the data is insignificantly far from zero.

Traditionally, the prior over the hidden state sequence is expressed as a Gaussian distribution directly over the first hidden state $\mathbf{x}_1$ (see, for example Ghahramani and Hinton, 1996a, equation (6)). For reasons that will become clear when later analysing the equations for learning the parameters of the model, we choose here to express the prior over the first hidden state indirectly through a prior over an auxiliary hidden state at time $t = 0$, denoted $\mathbf{x}_0$, which is Gaussian distributed with mean $\boldsymbol{\mu}_0$ and covariance Σ_0:

$$p(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) = \mathrm{N}(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) \,. \tag{5.13}$$

This induces a prior over $\mathbf{x}_1$ via the the state dynamics process:

$$p(\mathbf{x}_1 \mid \boldsymbol{\mu}_0, \Sigma_0, \boldsymbol{\theta}) = \int d\mathbf{x}_0 \; p(\mathbf{x}_0 \mid \boldsymbol{\mu}_0, \Sigma_0) p(\mathbf{x}_1 \mid \mathbf{x}_0, \boldsymbol{\theta}) \tag{5.14}$$

$$= \mathrm{N}(\mathbf{x}_1 \mid A\boldsymbol{\mu}_0 + B\mathbf{u}_1, A^\top \Sigma_0 A + Q) \; . \tag{5.15}$$

Although not constrained to be so, in this chapter we work with a prior covariance Σ_0 that is a multiple of the identity.

The marginal likelihood can then be written

$$p(\mathbf{y}_{1:T}) = \int dA \, dB \, dC \, dD \, d\boldsymbol{\rho} \, d\mathbf{x}_{0:T} \; p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \; . \tag{5.16}$$

All hyperparameters can be optimised during learning (see section 5.3.6). In section 5.4 we present results of some experiments in which we show the variational Bayesian approach successfully determines the structure of state-space models learnt from synthetic data, and in section 5.5 we present some very preliminary experiments in which we attempt to use hyperparameter optimisation mechanisms to elucidate underlying interactions amongst genes in DNA microarray time-series data.

A fully hierarchical Bayesian structure

Depending on the task at hand we should consider how full a Bayesian analysis we require. As the model specification stands, there is the problem that the number of free parameters to be 'fit' increases with the complexity of the model. For example, if the number of hidden dimensions were increased then, even though the parameters of the dynamics (A), output (C), input-to-state (B), and input-to-observation (D) matrices are integrated out, the size of the $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ hyperparameters have increased, providing more parameters to fit. Clearly, the more parameters that are fit the more one departs from the Bayesian inference framework and the more one risks overfitting. But, as pointed out in MacKay (1995), these extra hyperparameters themselves cannot overfit the noise in the data, since it is only the parameters that can do so.

If the task at hand is structure discovery, then the presence of extra hyperparameters should not affect the returned structure. However if the task is model comparison, that is comparing the marginal likelihoods for models with different numbers of hidden state dimensions for example, or comparing differently structured Bayesian models, then optimising over more hyperparameters will introduce a bias favouring more complex models, unless they themselves are integrated out.

The proper marginal likelihood to use in this latter case is that which further integrates over the hyperparameters with respect to some hyperprior which expresses our subjective beliefs over

the distribution of these hyperparameters. This is necessary for the ARD hyperparameters, and also for the hyperparameters governing the prior over the hidden state sequence, $\boldsymbol{\mu}_0$ and Σ_0, whose number of free parameters are functions of the dimensionality of the hidden state, k. For example, the ARD hyperparameter for each matrix A, B, C, D would be given a separate spherical gamma hyperprior, which is conjugate:

$$\boldsymbol{\alpha} \sim \prod_{j=1}^{k} \mathrm{Ga}(\alpha_j \,|\, a_\alpha, b_\alpha) \tag{5.17}$$

$$\boldsymbol{\beta} \sim \prod_{c=1}^{d} \mathrm{Ga}(\beta_c \,|\, a_\beta, b_\beta) \tag{5.18}$$

$$\boldsymbol{\gamma} \sim \prod_{j=1}^{k} \mathrm{Ga}(\gamma_j \,|\, a_\gamma, b_\gamma) \tag{5.19}$$

$$\boldsymbol{\delta} \sim \prod_{c=1}^{d} \mathrm{Ga}(\delta_c \,|\, a_\delta, b_\delta)\ . \tag{5.20}$$

The hidden state hyperparameters would be given spherical Gaussian and spherical inverse-gamma hyperpriors:

$$\boldsymbol{\mu}_0 \sim \mathrm{N}(\boldsymbol{\mu}_0 \,|\, \mathbf{0}, b_{\boldsymbol{\mu}_0}\mathrm{I}) \tag{5.21}$$

$$\Sigma_0 \sim \prod_{j=1}^{k} \mathrm{Ga}(\Sigma_{0\,jj}^{-1} \,|\, a_{\Sigma_0}, b_{\Sigma_0})\ . \tag{5.22}$$

Inverse-Wishart hyperpriors for Σ_0 are also possible. For the most part of this chapter we omit this fuller hierarchy to keep the exposition clearer, and only perform experiments aimed at structure discovery using ARD as opposed to model comparison between this and other Bayesian models. Towards the end of the chapter there is a brief note on how the fuller Bayesian hierarchy affects the algorithms for learning.

Origin of the intractability with Bayesian learning

Since A, B, C, D, $\boldsymbol{\rho}$ and $\mathbf{x}_{0:T}$ are all unknown, given a sequence of observations $\mathbf{y}_{1:T}$, an exact Bayesian treatment of SSMs would require computing marginals of the posterior over parameters and hidden variables, $p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$. This posterior contains interaction terms up to *fifth order*; we can see this by considering the terms in (5.1) for the case of LDS models which, for example, contain terms in the exponent of the form $-\frac{1}{2}\mathbf{x}_t^\top C^\top \mathrm{diag}\,(\boldsymbol{\rho})\, C\mathbf{x}_t$. Integrating over these coupled hidden variables and parameters is not analytically possible. However, since the model is conjugate-exponential we can apply theorem 2.2 to derive a vari-

ational Bayesian EM algorithm for state-space models analogous to the maximum-likelihood EM algorithm of Shumway and Stoffer (1982).

5.3 The variational treatment

This section covers the derivation of the results for the variational Bayesian treatment of linear-Gaussian state-space models. We first derive the lower bound on the marginal likelihood, using only the usual approximation of the factorisation of the hidden state sequence from the parameters. Due to some resulting conditional independencies between the parameters of the model, we see how the approximate posterior over parameters can be separated into posteriors for the dynamics and output processes. In section 5.3.1 the VBM step is derived, yielding approximate distributions over all the parameters of the model, each of which is analytically manageable and can be used in the VBE step.

In section 5.3.2 we justify the use of existing propagation algorithms for the VBE step, and the following subsections derive in some detail the forward and backward recursions for the variational Bayesian linear dynamical system. This section is concluded with results for hyper-parameter optimisation and a note on the tractability of the calculation of the lower bound for this model.

The variational approximation and lower bound

The full joint probability for parameters, hidden variables and observed data, given the inputs is

$$p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, \mathbf{u}_{1:T}) \;, \tag{5.23}$$

which written fully is

$$\begin{aligned} &p(A \,|\, \boldsymbol{\alpha}) p(B \,|\, \boldsymbol{\beta}) p(\boldsymbol{\rho} \,|\, a, b) p(C \,|\, \boldsymbol{\rho}, \boldsymbol{\gamma}) p(D \,|\, \boldsymbol{\rho}, \boldsymbol{\delta}) \cdot \\ &\qquad p(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) \prod_{t=1}^{T} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, A, B, \mathbf{u}_t) p(\mathbf{y}_t \,|\, \mathbf{x}_t, C, D, \boldsymbol{\rho}, \mathbf{u}_t) \;. \end{aligned} \tag{5.24}$$

From this point on we drop the dependence on the input sequence $\mathbf{u}_{1:T}$, and leave it implicit. By applying Jensen's inequality we introduce any distribution $q(\boldsymbol{\theta}, \mathbf{x})$ over the parameters and hidden variables, and lower bound the log marginal likelihood

$$\ln p(\mathbf{y}_{1:T}) = \ln \int dA\, dB\, dC\, dD\, d\boldsymbol{\rho}\, d\mathbf{x}_{0:T}\; p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \tag{5.25}$$

$$\geq \int dA\, dB\, dC\, dD\, d\boldsymbol{\rho}\, d\mathbf{x}_{0:T} \cdot q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}) \ln \frac{p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T})}{q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T})} \tag{5.26}$$

$$= \mathcal{F} \,.$$

The next step in the variational approximation is to assume some approximate form for the distribution $q(\cdot)$ which leads to a tractable bound. First, we factorise the parameters from the hidden variables giving $q(A, B, C, D, \rho, \mathbf{x}_{0:T}) = q_{\boldsymbol{\theta}}(A, B, C, D, \rho) q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Writing out the expression for the exact log posterior $\ln p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{1:T}, \mathbf{y}_{0:T})$, one sees that it contains interaction terms between $\boldsymbol{\rho}$, C and D but none between $\{A, B\}$ and any of $\{\boldsymbol{\rho}, C, D\}$. This observation implies a further factorisation of the posterior parameter distributions,

$$q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}) = q_{AB}(A, B) q_{CD\boldsymbol{\rho}}(C, D, \boldsymbol{\rho}) q_{\mathbf{x}}(\mathbf{x}_{0:T}) \,. \tag{5.27}$$

It is important to stress that this latter factorisation amongst the parameters falls out of the initial factorisation of hidden variables from parameters, and from the *resulting* conditional independencies given the hidden variables. Therefore the variational approximation does not concede any accuracy by the latter factorisation, since it is exact given the first factorisation of the parameters from hidden variables.

We choose to write the factors involved in this joint parameter distribution as

$$q_{AB}(A, B) = q_B(B)\, q_A(A \,|\, B) \tag{5.28}$$

$$q_{CD\boldsymbol{\rho}}(C, D, \boldsymbol{\rho}) = q_{\boldsymbol{\rho}}(\boldsymbol{\rho})\, q_D(D \,|\, \boldsymbol{\rho})\, q_C(C \,|\, D, \boldsymbol{\rho}) \,. \tag{5.29}$$

Now the form for $q(\cdot)$ in (5.27) causes the integral (5.26) to separate into the following sum of terms:

$$\begin{aligned}
\mathcal{F} = &\int dB\; q_B(B) \ln \frac{p(B \,|\, \boldsymbol{\beta})}{q_B(B)} + \int dB\; q_B(B) \int dA\; q_A(A \,|\, B) \ln \frac{p(A \,|\, \boldsymbol{\alpha})}{q_A(A \,|\, B)} \\
&+ \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \ln \frac{p(\boldsymbol{\rho} \,|\, a, b)}{q_{\boldsymbol{\rho}}(\boldsymbol{\rho})} + \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \ln \frac{p(D \,|\, \boldsymbol{\rho}, \boldsymbol{\delta})}{q_D(D \,|\, \boldsymbol{\rho})} \\
&+ \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \int dC\; q_C(C \,|\, \boldsymbol{\rho}, D) \ln \frac{p(C \,|\, \boldsymbol{\rho}, \boldsymbol{\gamma})}{q_C(C \,|\, \boldsymbol{\rho}, D)} \\
&- \int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \\
&+ \int dB\; q_B(B) \int dA\; q_A(A \,|\, B) \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \int dC\; q_C(C \,|\, \boldsymbol{\rho}, D) \cdot \\
&\qquad \int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \qquad (5.30) \\
= &\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}_{0:T}), q_B(B), q_A(A \,|\, B), q_{\boldsymbol{\rho}}(\boldsymbol{\rho}), q_D(D \,|\, \boldsymbol{\rho}), q_C(C \,|\, \boldsymbol{\rho}, D)) \,. \qquad (5.31)
\end{aligned}$$

Here we have left implicit the dependence of $\mathcal{F}$ on the hyperparameters. For variational Bayesian learning, $\mathcal{F}$ is the key quantity that we work with. Learning proceeds with iterative updates of the variational posterior distributions $q_\cdot(\cdot)$, each locally maximising $\mathcal{F}$.

The optimum forms of these approximate posteriors can be found by taking functional derivatives of $\mathcal{F}$ (5.30) with respect to each distribution over parameters and hidden variable sequences. In the following subsections we describe the straightforward VBM step, and the somewhat more complicated VBE step. We do not need to be able to compute $\mathcal{F}$ to produce the learning rules, only calculate its derivatives. Nevertheless its calculation at each iteration can be helpful to ensure that we are monotonically increasing a lower bound on the marginal likelihood. We finish this section on the topic of how to calculate $\mathcal{F}$ which is hard to compute because it contains the a term which is the entropy of the posterior distribution over hidden state sequences,

$$H(q_{\mathbf{x}}(\mathbf{x}_{0:T})) = -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \,. \qquad (5.32)$$

5.3.1 VBM step: Parameter distributions

Starting from some arbitrary distribution over the hidden variables, the VBM step obtained by applying theorem 2.2 finds the variational posterior distributions over the parameters, and from these computes the expected natural parameter vector, $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle$, where the expectation is taken under the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (A, B, C, D, \boldsymbol{\rho})$.

We omit the details of the derivations, and present just the forms of the distributions that extremise $\mathcal{F}$. As was mentioned in section 5.2.2, given the approximating factorisation of the

posterior distribution over hidden variables and parameters, the approximate posterior over the parameters can be factorised without further assumption or approximation into

$$q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho}) = \prod_{j=1}^{k} q(\mathbf{b}_{(j)}) q(\mathbf{a}_{(j)} \,|\, \mathbf{b}_{(j)}) \prod_{s=1}^{p} q(\rho_s) q(\mathbf{d}_{(s)} \,|\, \rho_s) q(\mathbf{c}_{(s)} \,|\, \rho_s, \mathbf{d}_{(s)}) \tag{5.33}$$

where, for example, the row vector $\mathbf{b}_{(j)}^{\top}$ is used to denote the jth row of the matrix B (similarly so for the other parameter matrices).

We begin by defining some statistics of the input and observation data:

$$\ddot{U} \equiv \sum_{t=1}^{T} \mathbf{u}_t \mathbf{u}_t^{\top} \,, \qquad U_Y \equiv \sum_{t=1}^{T} \mathbf{u}_t \mathbf{y}_t^{\top} \,, \qquad \ddot{Y} \equiv \sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^{\top} \,. \tag{5.34}$$

In the forms of the variational posteriors given below, the matrix quantities W_A, G_A, $\tilde{M}$, S_A, and W_C, G_C, S_C are exactly the expected complete data sufficient statistics, obtained in the VBE step — their forms are given in equations (5.126-5.132).

The natural factorisation of the variational posterior over parameters yields these forms for A and B:

$$q_B(B) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{b}_{(j)} \,|\, \Sigma_B \overline{\mathbf{b}}_{(j)},\, \Sigma_B\right) \tag{5.35}$$

$$q_A(A \,|\, B) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{a}_{(j)} \,|\, \Sigma_A \left[\mathbf{s}_{A,(j)} - G_A \mathbf{b}_{(j)}\right],\, \Sigma_A\right) \tag{5.36}$$

with

$$\Sigma_A{}^{-1} = \mathrm{diag}\,(\boldsymbol{\alpha}) + W_A \tag{5.37}$$

$$\Sigma_B{}^{-1} = \mathrm{diag}\,(\boldsymbol{\beta}) + \ddot{U} - G_A^{\top} \Sigma_A G_A \tag{5.38}$$

$$\overline{B} = \tilde{M}^{\top} - S_A^{\top} \Sigma_A G_A \,, \tag{5.39}$$

and where $\overline{\mathbf{b}}_{(j)}^{\top}$ and $\mathbf{s}_{A,(j)}$ are vectors used to denote the jth row of $\overline{B}$ and the jth column of S_A respectively. It is straightforward to show that the marginal for A is given by:

$$q_A(A) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{a}_{(j)} \,|\, \Sigma_A \left[\mathbf{s}_{A,(j)} - G_A \Sigma_B \overline{\mathbf{b}}_{(j)}\right],\, \hat{\Sigma}_A\right) \,, \tag{5.40}$$

$$\text{where} \quad \hat{\Sigma}_A = \Sigma_A + \Sigma_A G_A \Sigma_B G_A^{\top} \Sigma_A \,. \tag{5.41}$$

In the case of either the A and B matrices, for both the marginal and conditional distributions, each row has the same covariance.

The variational posterior over ρ, C and D is given by:

$$q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{Ga}\left(\rho_s \,|\, a + \frac{T}{2},\ b + \frac{1}{2} G_{ss}\right) \tag{5.42}$$

$$q_D(D \,|\, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{d}_{(s)} \,|\, \Sigma_D \overline{\mathbf{d}}_{(s)},\ \rho_s^{-1} \Sigma_D\right) \tag{5.43}$$

$$q_C(C \,|\, D, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{c}_{(s)} \,|\, \Sigma_C \left[\mathbf{s}_{C,(s)} - G_C \mathbf{d}_{(s)}\right],\ \rho_s^{-1} \Sigma_C\right) \tag{5.44}$$

with

$$\Sigma_C{}^{-1} = \mathrm{diag}\left(\boldsymbol{\gamma}\right) + W_C \tag{5.45}$$

$$\Sigma_D{}^{-1} = \mathrm{diag}\left(\boldsymbol{\delta}\right) + \ddot{U} - G_C^{\top} \Sigma_C G_C \tag{5.46}$$

$$G = \ddot{Y} - S_C^{\top} \Sigma_C S_C - \overline{D} \Sigma_D \overline{D}^{\top} \tag{5.47}$$

$$\overline{D} = U_Y^{\top} - S_C^{\top} \Sigma_C G_C \ , \tag{5.48}$$

and where $\overline{\mathbf{d}}_{(s)}^{\top}$ and $\mathbf{s}_{C,(s)}$ are vectors corresponding to the sth row of $\overline{D}$ and the sth column of S_C respectively. Unlike the case of the A and B matrices, the covariances for each row of the C and D matrices can be very different due to the appearance of the ρ_s term, as so they should be. Again it is straightforward to show that the marginal for C given $\boldsymbol{\rho}$, is given by:

$$q_C(C \,|\, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{c}_{(s)} \,|\, \Sigma_C \left[\mathbf{s}_{C,(s)} - G_C \Sigma_D \overline{\mathbf{d}}_{(s)}\right],\ \rho_s^{-1} \hat{\Sigma}_C\right) \ , \tag{5.49}$$

$$\text{where} \quad \hat{\Sigma}_C = \Sigma_C + \Sigma_C G_C \Sigma_D G_C^{\top} \Sigma_C \ . \tag{5.50}$$

Lastly, the full marginals for C and D after integrating out the precision $\boldsymbol{\rho}$ are Student-t distributions.

In the VBM step we need to calculate the expected natural parameters, $\overline{\boldsymbol{\phi}}$, as mentioned in theorem 2.2. These will then be used in the VBE step which infers the distribution $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ over hidden states in the system. The relevant natural parameterisation is given by the following:

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\phi}(A, B, C, D, R) = \Big[A,\ A^{\top}A,\ B,\ A^{\top}B,\ C^{\top}R^{-1}C,\ R^{-1}C,\ C^{\top}R^{-1}D$$
$$B^{\top}B,\ R^{-1},\ \ln\left|R^{-1}\right|,\ D^{\top}R^{-1}D,\ R^{-1}D\Big] \ . \tag{5.51}$$

The terms in the expected natural parameter vector $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$, where $\langle \cdot \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$ denotes expectation with respect to the variational posterior, are then given by:

$$\langle A \rangle = \left[S_A - G_A \Sigma_B \overline{B}^\top \right]^\top \Sigma_A \tag{5.52}$$

$$\langle A^\top A \rangle = \langle A \rangle^\top \langle A \rangle + k \left[\Sigma_A + \Sigma_A G_A \Sigma_B G_A^\top \Sigma_A \right] \tag{5.53}$$

$$\langle B \rangle = \overline{B} \Sigma_B \tag{5.54}$$

$$\langle A^\top B \rangle = \Sigma_A \left[S_A \langle B \rangle - G_A \left\{ \langle B \rangle^\top \langle B \rangle + k \Sigma_B \right\} \right] \tag{5.55}$$

$$\langle B^\top B \rangle = \langle B \rangle^\top \langle B \rangle + k \Sigma_B \ , \tag{5.56}$$

and

$$\langle \rho_s \rangle = \overline{\rho_s} = \frac{a_{\boldsymbol{\rho}} + T/2}{b_{\boldsymbol{\rho}} + G_{ss}/2} \tag{5.57}$$

$$\langle \ln \rho_s \rangle = \overline{\ln \rho_s} = \psi(a_{\boldsymbol{\rho}} + T/2) - \ln(b_{\boldsymbol{\rho}} + G_{ss}/2) \tag{5.58}$$

$$\langle R^{-1} \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \ , \tag{5.59}$$

$$\tag{5.60}$$

and

$$\langle C \rangle = \left[S_C - G_C \Sigma_D \overline{D}^\top \right]^\top \Sigma_C \tag{5.61}$$

$$\langle D \rangle = \overline{D} \Sigma_D \tag{5.62}$$

$$\langle C^\top R^{-1} C \rangle = \langle C \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle C \rangle + p \left[\Sigma_C + \Sigma_C G_C \Sigma_D G_C^\top \Sigma_C \right] \tag{5.63}$$

$$\langle R^{-1} C \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle C \rangle \tag{5.64}$$

$$\langle C^\top R^{-1} D \rangle = \Sigma_C \left[S_C \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle - G_C \langle D \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle - p G_C \Sigma_D \right] \tag{5.65}$$

$$\langle R^{-1} D \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle \tag{5.66}$$

$$\langle D^\top R^{-1} D \rangle = \langle D \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle + p \Sigma_D \ . \tag{5.67}$$

Also included in this list are several expectations which are not part of the mean natural parameter vector, but are given here because having them at hand during and after an optimisation is useful.

5.3.2 VBE step: The Variational Kalman Smoother

We now turn to the VBE step: computing $q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Since SSMs are singly connected belief networks corollary 2.2 tells us that we can make use of belief propagation, which in the case of SSMs is known as the Rauch-Tung-Striebel smoother (Rauch et al., 1963). Unfortunately the

implementations of the filter and smoother are not as straightforward as one might expect, as is explained in the following subsections.

In the standard point-parameter linear-Gaussian dynamical system, given the settings of the parameters, the hidden state posterior is jointly Gaussian over the time steps. Reassuringly, when we differentiate $\mathcal{F}$ with respect to $q_{\mathbf{x}}(\mathbf{x}_{0:T})$, the variational posterior for $\mathbf{x}_{0:T}$ is also Gaussian:

$$\ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) = -\ln Z + \langle \ln p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \rangle \tag{5.68}$$

$$= -\ln Z' + \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle \;, \tag{5.69}$$

where

$$Z' = \int d\mathbf{x}_{0:T} \;\; \exp \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle \;, \tag{5.70}$$

and where $\langle \cdot \rangle$ denotes expectation with respect to the variational posterior distribution over parameters, $q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho})$. In this expression the expectations with respect to the approximate parameter posteriors are performed on the logarithm of the complete-data likelihood and, even though this leaves the coefficients on the $\mathbf{x}_t$ terms in a somewhat unorthodox state, the new log posterior still only contains up to quadratic terms in each $\mathbf{x}_t$ and therefore $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ must be Gaussian, as in the point-parameter case. We should therefore still be able to use an algorithm very similar to the Kalman filter and smoother for inference of the hidden state sequence's sufficient statistics (the E-like step). However we can no longer plug in parameters to the filter and smoother, but have to work with the natural parameters throughout the implementation.

The following paragraphs take us through the required derivations for the forward and backward recursions. For the sake of clarity of exposition, we do not at this point derive the algorithms for the input-driven system (though we do present the full input-driven algorithms as pseudocode in algorithms 5.1, 5.2 and 5.3). At each stage, we first we concentrate on the point-parameter propagation algorithms and then formulate the Bayesian analogues.

5.3.3 Filter (forward recursion)

In this subsection, we first derive the well-known forward filtering recursion steps for the case in which the parameters are fixed point-estimates. The variational Bayesian analogue of the forward pass is then presented. The dependence of the filter equations on the inputs $\mathbf{u}_{1:T}$ has been omitted in the derivations, but is included in the summarising algorithms.

Point-parameter derivation

We define $\alpha_t(\mathbf{x}_t)$ to be the posterior over the hidden state at time t given observed data up to and including time t:

$$\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) \,. \tag{5.71}$$

Note that this is slightly different to the traditional form for HMMs which is $\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t, \mathbf{y}_{1:t})$. We then form the recursion with $\alpha_{t-1}(\mathbf{x}_{t-1})$ as follows:

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) &= \int d\mathbf{x}_{t-1}\; p(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1})\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{y}_t \,|\, \mathbf{x}_t) \;/\; p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) && (5.72)\\
&= \frac{1}{\zeta_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\; \alpha_{t-1}(\mathbf{x}_{t-1})\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{y}_t \,|\, \mathbf{x}_t) && (5.73)\\
&= \frac{1}{\zeta_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\; \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1})\; \mathrm{N}(\mathbf{x}_t \,|\; A\mathbf{x}_{t-1}, \mathrm{I})\; \mathrm{N}(\mathbf{y}_t \,|\; C\mathbf{x}_t, R) && (5.74)\\
&= \mathrm{N}(\mathbf{x}_t \,|\; \boldsymbol{\mu}_t, \Sigma_t) && (5.75)
\end{aligned}$$

where

$$\zeta_t(\mathbf{y}_t) \equiv p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) \tag{5.76}$$

is the filtered output probability; this will be useful for computing the likelihood. Within the above integrand the quadratic terms in $\mathbf{x}_{t-1}$ form the Gaussian $\mathrm{N}(\mathbf{x}_{t-1} \,|\, \mathbf{x}^*_{t-1}, \Sigma^*_{t-1})$ with

$$\begin{aligned}
\Sigma^*_{t-1} &= \left(\Sigma^{-1}_{t-1} + A^\top A\right)^{-1} && (5.77)\\
\mathbf{x}^*_{t-1} &= \Sigma^*_{t-1}\left[\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1} + A^\top \mathbf{x}_t\right] \,. && (5.78)
\end{aligned}$$

Marginalising out $\mathbf{x}_{t-1}$ gives the filtered estimates of the mean and covariance of the hidden state as

$$\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \Sigma_t) \tag{5.79}$$

with

$$\begin{aligned}
\Sigma_t &= \left[\mathrm{I} + C^\top R^{-1} C - A\Sigma^*_{t-1}A^\top\right]^{-1} && (5.80)\\
\boldsymbol{\mu}_t &= \Sigma_t\left[C^\top R^{-1}\mathbf{y}_t + A\Sigma^*_{t-1}\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1}\right] \,. && (5.81)
\end{aligned}$$

At each step the normalising constant ζ_t, obtained as the denominator in (5.72), contributes to the calculation of the probability of the data

$$\begin{aligned}
p(\mathbf{y}_{1:T}) &= p(\mathbf{y}_1)p(\mathbf{y}_2 \,|\, \mathbf{y}_1) \ldots p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) \ldots p(\mathbf{y}_T \,|\, \mathbf{y}_{1:T-1}) && (5.82)\\
&= p(\mathbf{y}_1) \prod_{t=2}^{T} p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} \zeta_t(\mathbf{y}_t) \,. && (5.83)
\end{aligned}$$

It is not difficult to show that each of the above terms are Gaussian distributed,

$$\zeta_t(\mathbf{y}_t) = \mathrm{N}(\mathbf{y}_t \,|\, \boldsymbol{\varpi}_t, \varsigma_t) \tag{5.84}$$

with

$$\varsigma_t = \left(R^{-1} - R^{-1}C\Sigma_t C^\top R^{-1}\right)^{-1} \tag{5.85}$$

$$\boldsymbol{\varpi}_t = \varsigma_t R^{-1} C \Sigma_t A \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \,. \tag{5.86}$$

With these distributions at hand we can compute the probability of each observation $\mathbf{y}_t$ given the previous observations in the sequence, and assign a predictive mean and variance to the data at each time step as it arrives. However, this predictive distribution will change once the hidden state sequence has been smoothed on the backward pass.

Certain expressions such as equations (5.80), (5.81), and (5.85) could be simplified using the matrix inversion lemma (see appendix B.2), but here we refrain from doing so because a similar operation is not possible in the variational Bayesian derivation (see comment at end of section 5.3.3).

Variational derivation

It is quite straightforward to repeat the above derivation for variational Bayesian learning, by replacing parameters (and combinations of parameters) with their expectations under the variational posterior distributions which were calculated in the VBM step (section 5.3.1). Equation (5.74) becomes rewritten as

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) = \frac{1}{\zeta'_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\, \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1}) \cdot \\
\exp -\frac{1}{2}\Big\langle (\mathbf{x}_t - A\mathbf{x}_{t-1})^\top \mathrm{I} (\mathbf{x}_t - A\mathbf{x}_{t-1}) + (\mathbf{y}_t - C\mathbf{x}_t)^\top R^{-1} (\mathbf{y}_t - C\mathbf{x}_t) \\
+ k \ln |2\pi| + \ln |2\pi R| \Big\rangle \qquad (5.87) \\
= \frac{1}{\zeta'_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\, \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1}) \cdot \\
\exp -\frac{1}{2}\Big[\mathbf{x}_{t-1}^\top \langle A^\top A \rangle \mathbf{x}_{t-1} - 2\mathbf{x}_{t-1}^\top \langle A \rangle^\top \mathbf{x}_t \\
+ \mathbf{x}_t^\top (\mathrm{I} + \langle C^\top R^{-1} C \rangle) \mathbf{x}_t - 2\mathbf{x}_t^\top \langle C^\top R^{-1} \rangle \mathbf{y}_t + \ldots \Big] \qquad (5.88)
\end{aligned}$$

where the angled brackets $\langle \cdot \rangle$ denote expectation under the variational posterior distribution over parameters, $q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho})$.

After the parameter averaging, the integrand is still log-quadratic in both $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, and so the derivation continues as before but with parameter expectations taking place of the point estimates. Equations (5.77) and (5.78) now become

$$\Sigma^*_{t-1} = \left(\Sigma^{-1}_{t-1} + \langle A^\top A \rangle\right)^{-1} \tag{5.89}$$

$$\mathbf{x}^*_{t-1} = \Sigma^*_{t-1}\left[\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1} + \langle A \rangle^\top \mathbf{x}_t\right] , \tag{5.90}$$

and marginalising out $\mathbf{x}_{t-1}$ yields a Gaussian distribution over $\mathbf{x}_t$,

$$\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \Sigma_t) \tag{5.91}$$

with mean and covariance given by

$$\Sigma_t = \left[\mathrm{I} + \langle C^\top R^{-1} C \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A \rangle^\top\right]^{-1} \tag{5.92}$$

$$\boldsymbol{\mu}_t = \Sigma_t\left[\langle C^\top R^{-1} \rangle \mathbf{y}_t + \langle A \rangle \Sigma^*_{t-1}\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1}\right] . \tag{5.93}$$

This variational α-message evidently resembles the point-parameter result in (5.80) and (5.81). Algorithm 5.1 shows the full implementation for the variational Bayesian forward recursion, including extra terms from the inputs and input-related parameters B and D which were not derived here to keep the presentation concise. In addition it gives the variational Bayesian analogues of equations (5.85) and (5.86).

We now see why, for example, equation (5.85) was not simplified using the matrix inversion lemma — this operation would necessarily split the R^{-1} and C matrices, yet its variational Bayesian counterpart requires that expectations be taken over the combined product $R^{-1}C$. These expectations cannot be passed through the inversion lemma. Included in appendix B.2 is a proof of the matrix inversion lemma which shows clearly how such expectations would become disjoined.

5.3.4 Backward recursion: sequential and parallel

In the backward pass information about future observations is incorporated to update the posterior distribution on the current time step. This recursion begins at the last time step $t = T$ (which has no future observations to take into account) and recurses to the beginning of the sequence to time $t = 0$.

There are two different forms for the backward pass. The *sequential* form makes use of the α-messages from the forward pass and does not need to access information about the current observation in order to calculate the posterior over the hidden state given all the data. The *parallel* form is so-called because it executes all its recursions independently of the forward

Algorithm 5.1: Forward recursion for variational Bayesian state-space models with inputs $\mathbf{u}_{1:T}$ (variational Kalman filter).

---

1. Initialise hyperparameters $\boldsymbol{\mu}_0$ and Σ_0 as the mean and covariance of the auxiliary hidden state $\mathbf{x}_0$

2. For $t = 1$ to T

   (a) Compute $\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \Sigma_t)$

$$\begin{aligned}
\Sigma^*_{t-1} &= \left(\Sigma^{-1}_{t-1} + \langle A^\top A \rangle\right)^{-1} \\
\Sigma_t &= \left(I + \langle C^\top R^{-1} C \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A \rangle^\top\right)^{-1} \\
\boldsymbol{\mu}_t &= \Sigma_t \Big[\langle C^\top R^{-1} \rangle \mathbf{y}_t + \langle A \rangle \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \\
&\qquad + \left(\langle B \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A^\top B \rangle - \langle C^\top R^{-1} D \rangle\right) \mathbf{u}_t\Big]
\end{aligned}$$

   (b) Compute predictive distribution of $\mathbf{y}_t$

$$\begin{aligned}
\varsigma_t &= \left(\langle R^{-1} \rangle - \langle R^{-1} C \rangle \Sigma_t \langle R^{-1} C \rangle^\top\right)^{-1} \\
\boldsymbol{\varpi}_t &= \varsigma_t \Big[\langle R^{-1} C \rangle \Sigma_t \langle A \rangle \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \\
&\qquad + \left(\langle R^{-1} D \rangle + \langle R^{-1} C \rangle \Sigma_t \left\{\langle B \rangle - \langle C^\top R^{-1} D \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A^\top B \rangle\right\}\right) \mathbf{u}_t\Big]
\end{aligned}$$

   (c) Compute $\zeta'_t(\mathbf{y}_t)$ (see (5.87) and also section 5.3.7 for details)

$$\begin{aligned}
\ln \zeta'_t(\mathbf{y}_t) = -\frac{1}{2}\Big[&\langle \ln |2\pi R| \rangle - \ln \left|\Sigma^{-1}_{t-1} \Sigma^*_{t-1} \Sigma_t\right| + \boldsymbol{\mu}^\top_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}^\top_t \Sigma^{-1}_t \boldsymbol{\mu}_t \\
&+ \mathbf{y}^\top_t \langle R^{-1} \rangle \mathbf{y}_t - 2\mathbf{y}^\top_t \langle R^{-1} D \rangle \mathbf{u}_t + \mathbf{u}^\top_t \langle D^\top R^{-1} D \rangle \mathbf{u}_t \\
&- (\Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \langle A^\top B \rangle \mathbf{u}_t)^\top \Sigma^*_{t-1} (\Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \langle A^\top B \rangle \mathbf{u}_t)\Big]
\end{aligned}$$

   End For

3. Output all computed quantities, including

   $\ln Z' = \sum_{t=1}^{T} \ln \zeta'_t(\mathbf{y}_t)$

pass, and then later combines its messages with those from the forward pass to compute the hidden state posterior for each time step.

Sequential implementation: point-parameters

In the sequential implementation we define a set of γ-messages to be the posterior over the hidden state given all the data. In the case of point-parameters, the recursion is then

$$\begin{aligned}
\gamma_t(\mathbf{x}_t) &\equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}) && (5.94)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.95)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:T}) p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.96)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t})\; p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.97)\\
&= \int d\mathbf{x}_{t+1}\; \left[\frac{p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)}{\int d\mathbf{x}'_t\; p(\mathbf{x}'_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t)}\right] p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.98)\\
&= \int d\mathbf{x}_{t+1}\; \left[\frac{\alpha_t(\mathbf{x}_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)}{\int d\mathbf{x}'_t\; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t)}\right] \gamma_{t+1}(\mathbf{x}_{t+1})\;. && (5.99)
\end{aligned}$$

Here the use of Bayes' rule in (5.98) has had the effect of replacing the explicit data dependence with functions of the α-messages computed in the forward pass. Integrating out $\mathbf{x}_{t+1}$ yields Gaussian distributions for the smoothed estimates of the hidden state at each time step:

$$\gamma_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\omega}_t, \Upsilon_{tt}) \tag{5.100}$$

where Σ^*_t is as defined in the forward pass according to (5.77) and

$$\begin{aligned}
K_t &= \left(\Upsilon^{-1}_{t+1,t+1} + A\Sigma^*_t A^\top\right)^{-1} && (5.101)\\
\Upsilon_{tt} &= \left[\Sigma^{*-1}_t - A^\top K_t A\right]^{-1} && (5.102)\\
\boldsymbol{\omega}_t &= \Upsilon_{tt}\left[\Sigma^{-1}_t \boldsymbol{\mu}_t + A^\top K_t \left(\Upsilon^{-1}_{t+1,t+1}\boldsymbol{\omega}_{t+1} - A\Sigma^*_t \Sigma^{-1}_t \boldsymbol{\mu}_t\right)\right]\;. && (5.103)
\end{aligned}$$

Note that K_t given in (5.101) is a different matrix to the Kalman gain matrix as found in the Kalman filtering and smoothing literature, and should not be confused with it.

The sequential version has an advantage in online scenarios: once the data at time t, $\mathbf{y}_t$, has been filtered it can be discarded and is replaced with its message, $\alpha_t(\mathbf{x}_t)$ (see, for example, Rauch, 1963). In this way potentially high dimensional observations can be stored simply as beliefs in the low dimensional state space.

Sequential implementation: variational analysis

Unfortunately the step using Bayes' rule in (5.98) cannot be transferred over to a variational treatment, and this can be demonstrated by seeing how the term $p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t})$ in (5.97) is altered by the lower bound operation. Up to a normalisation factor,

$$p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \overset{\text{VB}}{\to} \exp\Big\langle \ln p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \Big\rangle \tag{5.104}$$

$$= \exp\Big\langle \ln p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) + \ln \alpha_t(\mathbf{x}_t) - \ln \int d\mathbf{x}'_t \; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t) \Big\rangle \tag{5.105}$$

The last term in the above equation results in a precision term in the exponent of the form: $\ln \int d\mathbf{x}'_t \; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t) = -\frac{1}{2}\left[\mathrm{I} - A\left[\Sigma_t^{-1} + A^\top A\right]^{-1} A^\top\right] + c$. Even though this term is easy to express for a known A matrix, its expectation under $q_A(A)$ is difficult to compute. Even with the use of the matrix inversion lemma (see appendix B.2), which yields $\left(\mathrm{I} + A\Sigma_t A^\top\right)^{-1}$, the expression is still not amenable to expectation.

Parallel implementation: point-parameters

Some of the above problems are ameliorated using the parallel implementation, which we first derive using point-parameters. The parallel recursion produces β-messages, defined as

$$\beta_t(\mathbf{x}_t) \equiv p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \; . \tag{5.106}$$

These are obtained through a recursion analogous to the forward pass (5.72)

$$\beta_{t-1}(\mathbf{x}_{t-1}) = \int d\mathbf{x}_t \; p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \tag{5.107}$$

$$= \int d\mathbf{x}_t \; p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{x}_t) \beta_t(\mathbf{x}_t) \tag{5.108}$$

$$\propto \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\eta}_{t-1}, \Psi_{t-1}) \tag{5.109}$$

with the end condition that $\beta_T(\mathbf{x}_T) = 1$. Omitting the details, the terms for the backward messages are given by:

$$\Psi_t^* = \left(\mathrm{I} + C^\top R^{-1} C + \Psi_t^{-1}\right)^{-1} \tag{5.110}$$

$$\Psi_{t-1} = \left[A^\top A - A^\top \Psi_t^* A\right]^{-1} \tag{5.111}$$

$$\boldsymbol{\eta}_{t-1} = \Psi_{t-1} A^\top \Psi_t^* \left[C^\top R^{-1} \mathbf{y}_t + \Psi_t^{-1} \boldsymbol{\eta}_t\right] \tag{5.112}$$

where $t = \{T, \ldots, 1\}$, and Ψ_T^{-1} set to $\mathbf{0}$ to satisfy the end condition (regardless of $\boldsymbol{\eta}_T$). The last step in this recursion therefore finds the probability of all the data given the setting of the auxiliary $\mathbf{x}_0$ variable.

Parallel implementation: variational analysis

It is straightforward to produce the variational counterpart of the backward parallel pass just described. Omitting the derivation, the results are presented in algorithm 5.2 which also includes the influence of inputs on the recursions.

Algorithm 5.2: Backward parallel recursion for variational Bayesian state-space models with inputs $\mathbf{u}_{1:T}$.

1. Initialise $\Psi_T^{-1} = \mathbf{0}$ to satisfy end condition $\beta_T(\mathbf{x}_T) = 1$
2. For $t = T$ to 1

$$\begin{aligned}
\Psi_t^* &= \left(\mathrm{I} + \langle C^\top R^{-1} C\rangle + \Psi_t^{-1}\right)^{-1} \\
\Psi_{t-1} &= \left(\langle A^\top A\rangle - \langle A\rangle^\top \Psi_t^* \langle A\rangle\right)^{-1} \\
\boldsymbol{\eta}_{t-1} &= \Psi_{t-1}\Big[-\langle A^\top B\rangle \mathbf{u}_t \\
&\qquad + \langle A\rangle^\top \Psi_t^* \left(\langle B\rangle \mathbf{u}_t + \langle C^\top R^{-1}\rangle \mathbf{y}_t - \langle C^\top R^{-1} D\rangle \mathbf{u}_t + \Psi_t^{-1}\boldsymbol{\eta}_t\right)\Big]
\end{aligned}$$

   End For

3. Output $\{\boldsymbol{\eta}_t, \Psi_t\}_{t=0}^T$

5.3.5 Computing the single and joint marginals

The culmination of the VBE step is to compute the sufficient statistics of the hidden state, which are the marginals at each time step and the pairwise marginals across adjacent time steps.

In the point-parameter case, one can use the sequential backward pass, and then the single state marginals are given exactly by the γ-messages, and it only remains to calculate the pairwise marginals. It is not difficult to show that the terms involving $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are best represented with the quadratic term

$$\ln p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) = -\frac{1}{2}\begin{pmatrix} \mathbf{x}_t^\top & \mathbf{x}_{t+1}^\top \end{pmatrix}\begin{pmatrix} \Sigma_t^{*-1} & -A^\top \\ -A & K_t^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{pmatrix} + \text{const.}\,, \quad (5.113)$$

where Σ_t^* is computed in the forward pass (5.77) and K_t is computed in the backward sequential pass (5.101).

We define $\Upsilon_{t,t+1}$ to be the cross-covariance between the hidden states at times t and $t+1$, given all the observations $\mathbf{y}_{1:T}$:

$$\Upsilon_{t,t+1} \equiv \left\langle \left(\mathbf{x}_t - \langle \mathbf{x}_t \rangle\right) \left(\mathbf{x}_{t+1} - \langle \mathbf{x}_{t+1} \rangle\right)^\top \right\rangle , \tag{5.114}$$

where $\langle \cdot \rangle$ denotes expectation with respect to the posterior distribution over the hidden state sequence given all the data. We now make use of the Schur complements (see appendix B.1) of the precision matrix given in (5.113) to obtain

$$\Upsilon_{t,t+1} = \Sigma_t^* A^\top \Upsilon_{t+1,t+1} . \tag{5.115}$$

The variational Bayesian implementation

In the variational Bayesian scenario the marginals cannot be obtained easily with a backward sequential pass, and they are instead computed by combining the α- and β-messages as follows:

$$\begin{aligned} p(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}) &\propto p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) && (5.116)\\ &= \alpha_t(\mathbf{x}_t)\beta_t(\mathbf{x}_t) && (5.117)\\ &= \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\omega}_t, \Upsilon_{tt}) && (5.118) \end{aligned}$$

with

$$\begin{aligned} \Upsilon_{t,t} &= \left[\Sigma_t^{-1} + \Psi_t^{-1}\right]^{-1} && (5.119)\\ \boldsymbol{\omega}_t &= \Upsilon_{t,t}\left[\Sigma_t^{-1}\boldsymbol{\mu}_t + \Psi_t^{-1}\boldsymbol{\eta}_t\right] . && (5.120) \end{aligned}$$

This is computed for $t = \{0, \ldots, T-1\}$. At $t = 0$, $\alpha_0(\mathbf{x}_0)$ is exactly the prior (5.13) over the auxiliary hidden state; at $t = T$, there is no need for a calculation since $p(\mathbf{x}_T \,|\, \mathbf{y}_{1:T}) \equiv \alpha_T(\mathbf{x}_T)$.

Similarly the pairwise marginals are given by

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) &\propto p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_{t+1}) p(\mathbf{y}_{t+2:T} \,|\, \mathbf{x}_{t+1}) && (5.121)\\ &= \alpha_t(\mathbf{x}_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_{t+1}) \beta_{t+1}(\mathbf{x}_{t+1}) , && (5.122) \end{aligned}$$

which under the variational transform becomes

$$\overset{\text{VB}}{\rightarrow} \alpha_t(\mathbf{x}_t)\ \exp\big\langle \ln p(\mathbf{x}_{t+1}\,|\,\mathbf{x}_t) + \ln p(\mathbf{y}_{t+1}\,|\,\mathbf{x}_{t+1})\big\rangle\ \beta_{t+1}(\mathbf{x}_{t+1}) \tag{5.123}$$

$$= \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{bmatrix} \,|\, \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t+1} \end{bmatrix}, \begin{bmatrix} \Upsilon_{t,t} & \Upsilon_{t,t+1} \\ \Upsilon_{t,t+1}^\top & \Upsilon_{t+1,t+1} \end{bmatrix} \right) . \tag{5.124}$$

With the use of Schur complements again, it is not difficult to show that $\Upsilon_{t,t+1}$ is given by

$$\Upsilon_{t,t+1} = \Sigma_t^* \langle A \rangle^\top \left(\mathrm{I} + \langle C^\top R^{-1} C \rangle + \Psi_{t+1}^{-1} - \langle A \rangle \Sigma_t^* \langle A \rangle^\top \right)^{-1} . \tag{5.125}$$

This cross-covariance is then computed for all time steps $t = \{0, \ldots, T-1\}$, which includes the cross-covariance between the zeroth and first hidden states.

In summary, the entire VBE step consists of a forward pass followed by a backward pass, during which the marginals can be computed as well straight after each β-message.

The required sufficient statistics of the hidden state

In the VBE step we need to calculate the expected sufficient statistics of the hidden state, as mentioned in theorem 2.2. These will then be used in the VBM step which infers the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ over parameters of the system (section 5.3.1). The relevant expectations are:

$$W_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \mathbf{x}_{t-1}^\top \rangle = \sum_{t=1}^{T} \Upsilon_{t-1,t-1} + \boldsymbol{\omega}_{t-1} \boldsymbol{\omega}_{t-1}^\top \tag{5.126}$$

$$G_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \rangle \mathbf{u}_t^\top \quad = \sum_{t=1}^{T} \boldsymbol{\omega}_{t-1} \mathbf{u}_t^\top \tag{5.127}$$

$$\tilde{M} = \sum_{t=1}^{T} \mathbf{u}_t \langle \mathbf{x}_t \rangle^\top \quad = \sum_{t=1}^{T} \mathbf{u}_t \boldsymbol{\omega}_t^\top \tag{5.128}$$

$$S_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \mathbf{x}_t^\top \rangle \quad = \sum_{t=1}^{T} \Upsilon_{t-1,t} + \boldsymbol{\omega}_{t-1} \boldsymbol{\omega}_t^\top \tag{5.129}$$

$$W_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \mathbf{x}_t^\top \rangle = \sum_{t=1}^{T} \Upsilon_{t,t} + \boldsymbol{\omega}_t \boldsymbol{\omega}_t^\top \tag{5.130}$$

$$G_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \rangle \mathbf{u}_t^\top = \sum_{t=1}^{T} \boldsymbol{\omega}_t \mathbf{u}_t^\top \tag{5.131}$$

$$S_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \rangle \mathbf{y}_t^\top = \sum_{t=1}^{T} \boldsymbol{\omega}_t \mathbf{y}_t^\top . \tag{5.132}$$

Note that M and G_C are transposes of one another. Also note that all the summations contain T terms (instead of those for the dynamics model containing $T-1$). This is a consequence of our adoption of a slightly unorthodox model specification of linear dynamical systems which includes a fictitious auxiliary hidden variable $\mathbf{x}_0$.

5.3.6 Hyperparameter learning

The hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, a and b, and the prior parameters Σ_0 and $\boldsymbol{\mu}_0$, can be updated so as to maximise the lower bound on the marginal likelihood (5.30). By taking derivatives of $\mathcal{F}$ with respect to the hyperparameters, the following updates can be derived, applicable after a VBM step:

$$\alpha_j^{-1} \leftarrow \frac{1}{k}\Big[k\Sigma_A + \Sigma_A\big[S_A S_A^\top - 2G_A\langle B\rangle^\top S_A^\top + G_A\{k\Sigma_B + \langle B\rangle^\top\langle B\rangle\}G_A^\top\big]\Sigma_A\Big]_{jj} \tag{5.133}$$

$$\beta_j^{-1} \leftarrow \frac{1}{k}\Big[k\Sigma_B + \langle B\rangle^\top\langle B\rangle\Big]_{jj} \tag{5.134}$$

$$\begin{aligned}\gamma_j^{-1} \leftarrow \frac{1}{p}\Big[&p\Sigma_C + \Sigma_C\big[S_C\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)S_C^\top - 2S_C\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle G_C^\top \\ &+ pG_C\Sigma_D G_C' + G_C\langle D\rangle^\top\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle G_C^\top\big]\Sigma_C\Big]_{jj}\end{aligned} \tag{5.135}$$

$$\delta_j^{-1} \leftarrow \frac{1}{p}\Big[p\Sigma_D + \langle D\rangle^\top\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle\Big]_{jj} \tag{5.136}$$

where $[\cdot]_{jj}$ denotes its (j,j)th element.

Similarly, in order to maximise the probability of the hidden state sequence under the prior, the hyperparameters of the prior over the auxiliary hidden state are set according to the distribution of the smoothed estimate of $\mathbf{x}_0$:

$$\Sigma_0 \leftarrow \Upsilon_{0,0}\,, \qquad \boldsymbol{\mu}_0 \leftarrow \boldsymbol{\omega}_0\,. \tag{5.137}$$

Last of all, the hyperparameters a and b governing the prior distribution over the output noise, $R = \mathrm{diag}\left(\boldsymbol{\rho}\right)$, are set to the fixed point of the equations

$$\psi(a) = \ln b + \frac{1}{p}\sum_{s=1}^{p}\overline{\ln\rho_s}\,, \qquad \frac{1}{b} = \frac{1}{pa}\sum_{s=1}^{p}\overline{\rho_s} \tag{5.138}$$

where $\psi(x) \equiv \partial/\partial x \ln\Gamma(x)$ is the *digamma* function (refer to equations (5.57) and (5.58) for required expectations). These fixed point equations can be solved straightforwardly using gradient following techniques (such as Newton's method) in just a few iterations, bearing in mind the positivity constraints on a and b (see appendix C.2 for more details).

5.3.7 Calculation of $\mathcal{F}$

Before we see why $\mathcal{F}$ is hard to compute in this model, we should rewrite the lower bound more succinctly using the following definitions, in the case of a pair of variables J and K:

$$\mathrm{KL}(J) \equiv \int dJ\; q(J) \ln \frac{q(J)}{p(J)} \qquad \text{(KL divergence)} \tag{5.139}$$

$$\mathrm{KL}(J \,|\, K) \equiv \int dJ\; q(J \,|\, K) \ln \frac{q(J \,|\, K)}{p(J \,|\, K)} \qquad \text{(conditional KL)} \tag{5.140}$$

$$\langle \mathrm{KL}(J \,|\, K) \rangle_{q(K)} \equiv \int dK\; q(K) \mathrm{KL}(J \,|\, K) \qquad \text{(expected conditional KL) .} \tag{5.141}$$

Note that in (5.140) the prior over J may need to be a function of K for conjugacy reasons (this is the case for state-space models for the output parameters C and D, and the noise R). The notation $\mathrm{KL}(J \,|\, K)$ is not to be confused with $\mathrm{KL}(J||K)$ which is the KL divergence between distributions $q(J)$ and $q(K)$ (which are marginals). The lower bound $\mathcal{F}$ (5.26) can now be written as

$$\begin{aligned}
\mathcal{F} = &-\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\
&- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho}, D)} \\
&+ \mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \\
&+ \langle \ln p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle_{q(A,B,C,D,\boldsymbol{\rho}) q(\mathbf{x}_{1:T})}
\end{aligned} \tag{5.142}$$

where $\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T}))$ is the entropy of the variational posterior over the hidden state sequence,

$$\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \equiv -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \; . \tag{5.143}$$

The reason why $\mathcal{F}$ can not be computed directly is precisely due to both this entropy term and the last term which takes expectations over all possible hidden state sequences under the variational posterior $q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Fortunately, straight after the VBE step, we know the form of $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ from (5.69), and on substituting this into $\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T}))$ we obtain

$$H(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \equiv -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \tag{5.144}$$

$$\begin{aligned}
= -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \Big[&- \ln Z' \\
&+ \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}, \boldsymbol{\mu}_0, \Sigma_0) \rangle_{q_{\boldsymbol{\theta}}(A,B,C,D,\boldsymbol{\rho})} \Big]
\end{aligned} \tag{5.145}$$

$$= \ln Z' - \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}, \boldsymbol{\mu}_0, \Sigma_0) \rangle_{q_{\boldsymbol{\theta}}(A,B,C,D,\boldsymbol{\rho}) q_{\mathbf{x}}(\mathbf{x}_{0:T})} \tag{5.146}$$

where the last line follows since $\ln Z'$ is not a function of the state sequence $\mathbf{x}_{0:T}$. Substituting this form (5.146) into the above form for $\mathcal{F}$ (5.142) cancels the expected complete-data term in both equations and yields a simple expression for the lower bound

$$\begin{aligned}\mathcal{F} = &-\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\ &- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho},D)} \\ &+ \ln Z' \,. \end{aligned} \tag{5.147}$$

Note that this simpler expression is only valid straight after the VBE step. The various KL divergence terms are straightforward, yet laborious, to compute (see section C.3 for details).

We still have to evaluate the log partition function, $\ln Z'$. It is not as complicated as the integral in equation (5.70) suggests — at least in the point-parameter scenario we showed that $\ln Z' = \sum_{t=1}^{T} \ln \zeta_t(\mathbf{y}_t)$, as given in (5.83). With some care we can derive the equivalent terms $\{\zeta'_t(\mathbf{y}_t)\}_{t=1}^{T}$ for the variational Bayesian treatment, and these are given in part (c) of algorithm 5.1. Note that certain terms cancel across time steps and so the overall computation can be made more efficient if need be.

Alternatively we can calculate $\ln Z'$ from direct integration of the joint (5.70) with respect to each hidden variable one by one. In principal the hidden variables can be integrated out in any order, but at the expense of having to store statistics for many intermediate distributions.

The complete learning algorithm for state-space models is presented in algorithm 5.3. It consists of repeated iterations of the VBM step, VBE step, calculation of $\mathcal{F}$, and hyperparameter updates. In practice one does not need to compute $\mathcal{F}$ at all for learning. It may also be inefficient to update the hyperparameters after every iteration of VBEM, and for some applications in which the user is certain of their prior specifications, then a hyperparameter learning scheme may not be required at all.

5.3.8 Modifications when learning from multiple sequences

So far in this chapter the variational Bayesian algorithm has concentrated on just a data set consisting of a single sequence. For a data set consisting of n i.i.d. sequences with lengths $\{T_1, \ldots, T_n\}$, denoted $\mathbf{y} = \{\mathbf{y}_{1,1:T_1}, \ldots, \mathbf{y}_{n,1:T_n}\}$, it is straightforward to show that the VB algorithm need only be slightly modified to take into account the following changes.

Algorithm 5.3: Pseudocode for variational Bayesian state-space models.

1. Initialisation

   $\boldsymbol{\Theta} \equiv \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}\} \leftarrow$ initialise precision hyperparameters

   $\boldsymbol{\mu}_0, \Sigma_0 \leftarrow$ initialise hidden state priors

   $hss \leftarrow$ initialise hidden state sufficient statistics

2. Variational M step (VBM)

   Infer parameter posteriors $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ using $\{hss, \mathbf{y}_{1:T}, \mathbf{u}_{1:T}, \boldsymbol{\Theta}\}$

   $q(B)$, $q(A \,|\, B)$, $q(\boldsymbol{\rho})$, $q(D \,|\, \boldsymbol{\rho})$, and $q(C \,|\, \boldsymbol{\rho}, D)$

   $\overline{\boldsymbol{\phi}} \leftarrow$ calculate expected natural parameters using equations (5.52-5.67)

3. Variational E step (VBE)

   Infer distribution over hidden state $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ using $\{\overline{\boldsymbol{\phi}}, \mathbf{y}_{1:T}, \mathbf{u}_{1:T}\}$

   compute $\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) \quad t \in \{1, \ldots, T\}$ (forward pass, algorithm 5.1),

   compute $\beta_t(\mathbf{x}_t) \equiv p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \quad t \in \{0, \ldots, T-1\}$ (backward pass, algorithm 5.2),

   compute $\boldsymbol{\omega}_t, \Upsilon_{t,t} \quad t \in \{0, \ldots, T\}$ (marginals), and

   compute $\Upsilon_{t,t+1} \quad t \in \{0, \ldots, T-1\}$ (cross-covariance).

   $hss \leftarrow$ calculate hidden state sufficient statistics using equations (5.126-5.132)

4. Compute $\mathcal{F}$

   Compute various parameter KL divergences (appendix C.3)

   Compute log partition function, $\ln Z'$ (equation (5.70), algorithm 5.1)

   $\mathcal{F} = -\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle - \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle + \ln Z'$

5. Update hyperparameters

   $\boldsymbol{\Theta} \leftarrow$ update precision hyperparameters using equations (5.133-5.136)

   $\{\boldsymbol{\mu}_0, \Sigma_0\} \leftarrow$ update auxiliary hidden state $\mathbf{x}_0$ prior hyperparameters using (5.137)

   $\{a, b\} \leftarrow$ update noise hyperparameters using (5.138)

6. While $\mathcal{F}$ is increasing, go to step 2

In the VBE step, the forward and backward passes of algorithms 5.1 and 5.2 are carried out on each sequence, resulting in a set of sufficient statistics for each of the n hidden state sequences. These are then pooled to form a combined statistic. For example, equation (5.126) becomes

$$W_A^{(i)} = \sum_{t=1}^{T_i} \langle \mathbf{x}_{i,t-1} \mathbf{x}_{i,t-1}^\top \rangle = \sum_{t=1}^{T_i} \Upsilon_{i,t-1,t-1} + \boldsymbol{\omega}_{i,t-1} \boldsymbol{\omega}_{i,t-1}^\top \,, \tag{5.148}$$

$$\text{and then} \qquad W_A = \sum_{i=1}^{n} W_A^{(i)} \,, \tag{5.149}$$

where $\Upsilon_{i,t,t}$ and $\boldsymbol{\omega}_{i,t}$ are the results of the VBE step on the ith sequence. Each of the required sufficient statistics in equations (5.126-5.132) are obtained in a similar fashion. In addition, the number of time steps T is replaced with the total over all sequences $T = \sum_{i=1}^{n} T_i$.

Algorithmically, the VBM step remains unchanged, as do the updates for the hyperparameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, a, b\}$. The updates for the hyperparameters $\boldsymbol{\mu}_0$ and Σ_0, which govern the mean and covariance of the auxiliary hidden state at time $t = 0$ for every sequence, have to be modified slightly and become

$$\boldsymbol{\mu}_0 \leftarrow \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\omega}_{i,0} \,, \tag{5.150}$$

$$\Sigma_0 \leftarrow \frac{1}{n} \sum_{i=1}^{n} \left[\Upsilon_{i,0,0} + (\boldsymbol{\mu}_0 - \boldsymbol{\omega}_{i,0})(\boldsymbol{\mu}_0 - \boldsymbol{\omega}_{i,0})^\top \right] \,, \tag{5.151}$$

where the $\boldsymbol{\mu}_0$ appearing in the update for Σ_0 is the updated hyperparameter. In the case of $n = 1$, equations (5.150) and (5.151) resemble their originals forms given in section 5.3.6. Note that these batch updates trivially extend the analogous result for ML parameter estimation of linear dynamical systems presented by Ghahramani and Hinton (Ghahramani and Hinton, 1996a, equation (25)), since here we do not assume that the sequences are equal in length (it is clear from the forward and backward algorithms in both the ML and VB implementations that the posterior variance of the auxiliary state $\Upsilon_{i,0,0}$ will only be constant if all the sequences have the same length).

Finally the computation of the lower bound $\mathcal{F}$ is unchanged except that it now involves a contribution from each sequence

$$\begin{aligned} \mathcal{F} = &- \mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\ &- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho}, D)} + \sum_{i=1}^{n} \ln Z'^{(i)} \,, \end{aligned}$$

where $\ln Z'^{(i)}$ is computed in the VBE step in algorithm 5.1 for each sequence individually.

5.3.9 Modifications for a fully hierarchical model

As mentioned towards the end of section 5.2.2, the hierarchy of hyperparameters for priors over the parameters is not complete for this model as it stands. There remains the undesirable feature that the parameters Σ_0 and $\boldsymbol{\mu}_0$ contain more free parameters as the dimensionality of the hidden state increases. There is a similar problem for the precision hyperparameters. We refer the reader to chapter 4 in which a similar structure was used for the hyperparameters of the factor loading matrices.

With such variational distributions in place for VB LDS, the propagation algorithms would change, replacing, for example, $\boldsymbol{\alpha}$, with its expectation over its variational posterior, $\langle \boldsymbol{\alpha} \rangle_{q(\boldsymbol{\alpha})}$, and the hyperhyperparameters a_α, b_α of equation (5.17) would be updated to best fit the variational posterior for α, in the same fashion that the hyperparameters a, b are updated to reflect the variational posterior on ρ (section 5.3.6). In addition a similar KL penalty term would arise.

For the parameters Σ_0 and $\boldsymbol{\mu}_0$, again KL terms would crop up in the lower bound, and where these quantities appeared in the propagation algorithms they would have to be replaced with their expectations under their variational posterior distributions.

These modifications were considered too time-consuming to implement for the experiments carried out in the following section, and so we should of course be mindful of their exclusion.

5.4 Synthetic Experiments

In this section we give two examples of how the VB algorithm for linear dynamical systems can discover meaningful structure from the data. The first example is carried out on a data set generated from a simple LDS with no inputs and a small number of hidden states. The second example is more challenging and attempts to learn the number of hidden states and their dynamics in the presence of noisy inputs. We find in both experiments that the ARD mechanism which optimises the precision hyperparameters can be used successfully to determine the structure of the true generating model.

5.4.1 Hidden state space dimensionality determination (no inputs)

An LDS with hidden state dimensionality of $k = 6$ and an output dimensionality of $p = 10$ was set up with parameters randomly initialised according to the following procedure.

The dynamics matrix A $(k \times k)$ was fixed to have eigenvalues of $(.65, .7, .75, .8, .85, .9)$, constructed from a randomly rotated diagonal matrix; choosing fairly high eigenvalues ensures that

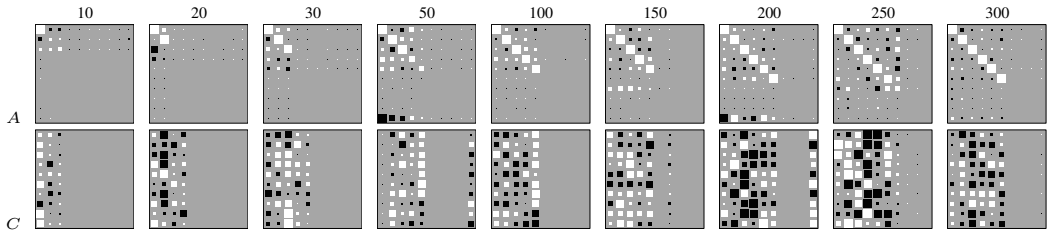
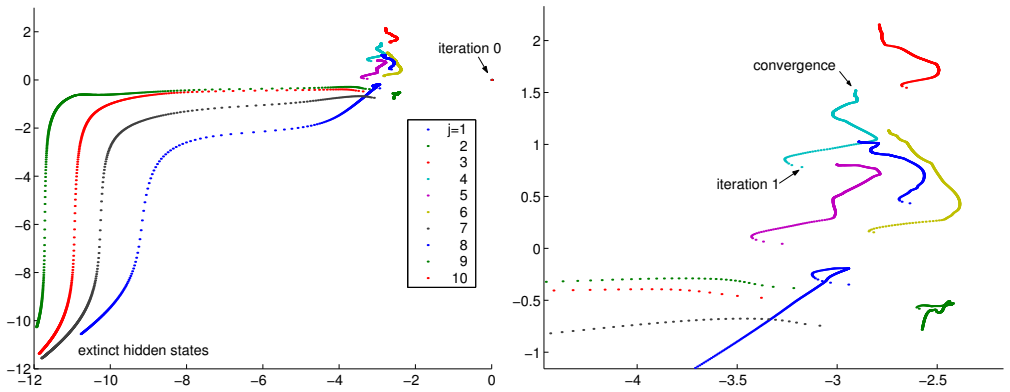


Figure 5.4: Hinton diagrams of the dynamics (A) and output (C) matrices after 500 iterations of VBEM. From left to right, the length of the observed sequence $\mathbf{y}_{1:T}$ increases from $T = 10$ to 300. This true data was generated from a linear dynamical system with $k = 6$ hidden state dimensions, all of which participated in the dynamics (see text for a description of the parameters used). As a visual aid, the entries of A matrix and the columns of the C matrix have been permuted in the order of the size of the hyperparameters in $\boldsymbol{\gamma}$.

every dimension participates in the hidden state dynamics. The output matrix C $(p \times k)$ had each entry sampled from a bimodal distribution made from a mixture of two Gaussians with means at (2,-2) and common standard deviations of 1; this was done in an attempt to keep the matrix entries away from zero, such that every hidden dimension contributes to the output covariance structure. Both the state noise covariance Q and output noise covariance R were set to be the identity matrix. The hidden state at time $t = 1$ was sampled from a Gaussian with mean zero and unit covariance.

From this LDS model several training sequences of increasing length were generated, ranging from $T = 10, \ldots, 300$ (the data sets are incremental). A VBLDS model with hidden state space dimensionality $k = 10$ was then trained on each single sequence, for a total of 500 iterations of VBEM. The resulting A and C matrices are shown in figure 5.4. We can see that for short sequences the model chooses a simple representation of the dynamics and output processes, and for longer sequences the recovered model is the same as the underlying LDS model which generated the sequences. Note that the model learns a predominantly diagonal dynamics matrix, or a self-reinforcing dynamics (this is made obvious by the permutation of the states in the figure (see caption), but is not a contrived observation). The likely reason for this is the prior's preference for the A matrix to have small sum-of-square entries for each column; since the dynamics matrix has to capture a certain amount of power in the hidden dynamics, the least expensive way to do this is to place most of the power on the diagonal entries.

Plotted in figure 5.5 are the trajectories of the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, during the VB optimisation for the sequence of length $T = 300$. For each hidden dimension j the output hyperparameter γ_j (vertical) is plotted against the dynamics hyperparameter α_j. It is in fact the logarithm of the *reciprocal* of the hyperparameter that is plotted on each axis. Thus if a hidden dimension becomes extinct, the reciprocal of its hyperparameter tends to zero (bottom left of plots). Each component of each hyperparameter is initialised to 1 (see annotation for iteration 0, at top right of plot 5.5(a)), and during the optimisation some dimensions become extinct. In this example, four hidden state dimensions become extinct, both in their ability to participate in the dynamics



(a) Hidden state inverse-hyperparameter trajectories (logarithmic axes).

(b) Close-up of top right corner of (a).

Figure 5.5: Trajectories of the hyperparameters for the case $n = 300$, plotted as $\ln \frac{1}{\alpha}$ (horizontal axis) against $\ln \frac{1}{\gamma}$ (vertical axis). Each trace corresponds to one of k hidden state dimensions, with points plotted after each iteration of VBEM. Note the initialisation of $(1, 1)$ for all $(\alpha_j, \gamma_j),\ j = 1, \ldots, k$ (labelled iteration 0). The direction of each trajectory can be determined by noting the spread of positions at successive iterations, which are resolvable at the beginning of the optimisation, but not so towards the end (see annotated close-up). Note especially that four hyperparameters are flung to locations corresponding to very small variances of the prior for both the A and C matrix columns (i.e. this has effectively removed those hidden state dimensions), and six remain in the top right with finite variances. Furthermore, the L-shaped trajectories of the eventually extinct hidden dimensions imply that in this example the dimensions are removed first from the model's dynamics, and then from the output process (see figure 5.8(a,c) also).

and their contribution to the covariance of the output data. Six hyperparameters remain useful, corresponding to $k = 6$ in the true model. The trajectories of these are seen more clearly in figure 5.5(b).

5.4.2 Hidden state space dimensionality determination (input-driven)

This experiment demonstrates the capacity of the input-driven model to use (or not to use) an input-sequence to model the observed data. We obtained a sequence $\mathbf{y}_{1:T}$ of length $T = 100$ by running the linear dynamical system as given in equations (5.4,5.5), with a hidden state space dimensionality of $k = 2$, generating an observed sequence of dimensionality $p = 4$. The input sequence, $\mathbf{u}_{1:T}$, consisted of three signals: the first two were $\frac{\pi}{2}$ phase-lagged sinusoids of period 50, and the third dimension was uniform noise $\sim \mathrm{U}(0, 1)$.

The parameters A, C, and R were created as described above (section 5.4.1). The eigenvalues of the dynamics matrix were set to $(.65, .7)$, and the covariance of the hidden state noise set to the identity. The parameter B $(k \times u)$ was set to the all zeros matrix, so the inputs did not modulate

the hidden state dynamics. The first two columns of the D $(p \times u)$ matrix were sampled from the uniform $\mathrm{U}(-10, 10)$, so as to induce a random (but fixed) displacement of the observation sequence. The third column of the D matrix was set to zeros, so as to ignore the third input dimension (noise). Therefore the only noise in the training data was that from the state and output noise mechanisms (Q and R).

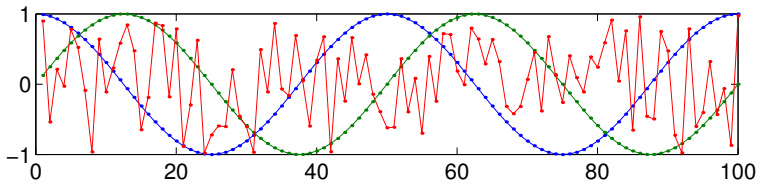
Figure 5.6 shows the input sequence used, the generated hidden state sequence, and the resulting observed data, over $T = 100$ time steps. We would like the variational Bayesian linear dynamical system to be able to identify the number of hidden dimensions required to model the observed data, taking into account the modulatory effect of the input sequence. As in the previous experiment, in this example we attempt to learn an over-specified model, and make use of the ARD mechanisms in place to recover the structure of the underlying model that generated the data.

In full, we would like the model to learn that there are $k = 2$ hidden states, that the third input dimension is irrelevant to predicting the observed data, that all the input dimensions are irrelevant for the hidden state dynamics, and that it is only the two dynamical hidden variables that are being embedded in the data space.

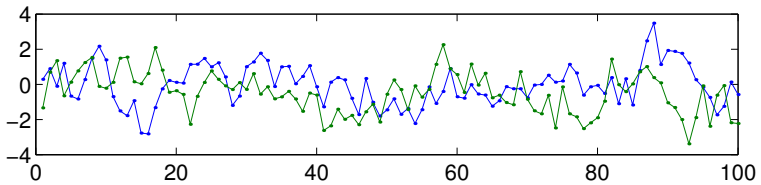
The variational Bayesian linear dynamical system was run with $k = 4$ hidden dimensions, for a total of 800 iterations of VBE and VBM steps (see algorithm 5.3 and its sub-algorithms). Hyperparameter optimisations after each VBM step were introduced on a staggered basis to ease interpretability of the results. The dynamics-related hyperparameter optimisations (i.e. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) were begun after the first 10 iterations, the output-related optimisations (i.e. $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$) after 20 iterations, and the remaining hyperparameters (i.e. a, b, Σ_0 and $\boldsymbol{\mu}_0$) optimised after 30 iterations. After each VBE step, $\mathcal{F}$ was computed and the current state of the hyperparameters recorded.

Figure 5.7 shows the evolution of the lower bound on the marginal likelihood during learning, displayed as both the value of $\mathcal{F}$ computed after each VBE step (figure 5.7(a)), and the *change* in $\mathcal{F}$ between successive iterations of VBEM (figure 5.7(b)). The logarithmic plot shows the onset of each group of hyperparameter optimisations (see caption), and also clearly shows three regions where parameters are being pruned from the model.

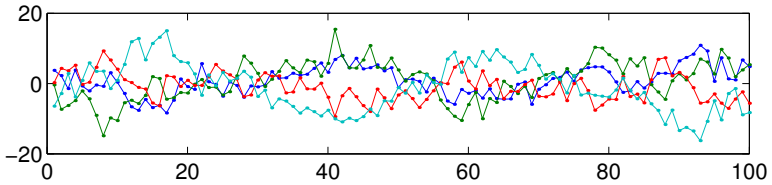
As before we can analyse the change in the hyperparameters during the optimisation process. In particular we can examine the ARD hyperparameter vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$, which contain the prior precisions for the entries of each column of each of the matrices A, B, C and D respectively. Since the hyperparameters are updated to reflect the variational posterior distribution over the parameters, a large value suggest that the relevant column contains entries are close to zero, and therefore can be considered excluded from the state-space model equations (5.4) and (5.5).



(a) 3 dimensional input sequence.

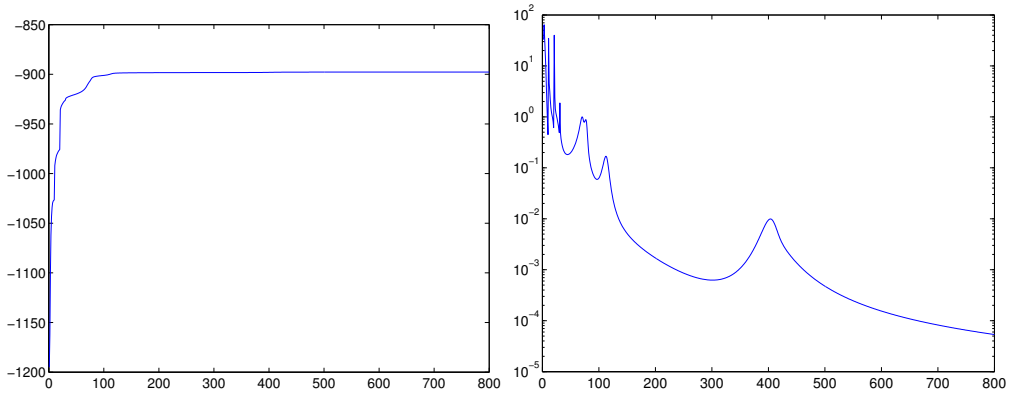


(b) 2 dimensional hidden state sequence.



(c) 4 dimensional observed data.

Figure 5.6: Data for the input-driven example in section 5.4.2. **(a)**: The 3 dimensional input data consists of two phase-lagged sinusoids of period 50, and a third dimension consisting of noise uniformly distributed on $[0, 1]$. Both B and D contain zeros in their third columns, so the noise dimension is not used when generating the synthetic data. **(b)**: The hidden state sequence generated from the dynamics matrix, A, which in this example evolves independently of the inputs. **(c)**: The observed data, generated by combining the embedded hidden state sequence (via the output matrix C) and the input sequence (via the input-output matrix D), and then adding noise with covariance R. Note that the observed data is now a sinusoidally modulated simple linear dynamical system.



(a) Evolution of $\mathcal{F}$ during iterations of VBEM. (b) Change in $\mathcal{F}$ between successive iterations.

Figure 5.7: Evolution of the lower bound $\mathcal{F}$ during learning of the input-dependent model of section 5.4.2. **(a)**: The lower bound $\mathcal{F}$ increases monotonically with iterations of VBEM. **(b)**: Interesting features of the optimisation can be better seen in a logarithmic plot of the change of $\mathcal{F}$ between successive iterations of VBEM. For example, it is quite clear there is a sharp increase in $\mathcal{F}$ at 10 iterations (dynamics-related hyperparameter optimisation activated), at 20 iterations (output-related hyperparameter optimisation activated), and at 30 iterations (the remaining hyperparameter optimisations are activated). The salient peaks around 80, 110, and 400 iterations each correspond to the gradual automatic removal of one or more parameters from the model by hyperparameter optimisation. For example, it is quite probable that the peak at around iteration 400 is due to the recovery of the first hidden state modelling the dynamics (see figure 5.8).

Figure 5.8 displays the components of each of the four hyperparameter vectors throughout the optimisation. The reciprocal of the hyperparameter is plotted since it is more visually intuitive to consider the variance of the parameters falling to zero as corresponding to extinction, instead of the precision growing without bound. We can see that, by 500 iterations, the algorithm has (correctly) discovered that there are only two hidden variables participating in the dynamics (from $\boldsymbol{\alpha}$), these same two variables are used as factors embedded in the output (from $\boldsymbol{\gamma}$), that none of the input dimensions is used to modulate the hidden dynamics (from $\boldsymbol{\beta}$), and that just two dimensions of the input are required to displace the data (from $\boldsymbol{\delta}$). The remaining third dimension of the input is in fact disregarded completely by the model, which is exactly according to the recipe used for generating this synthetic data.

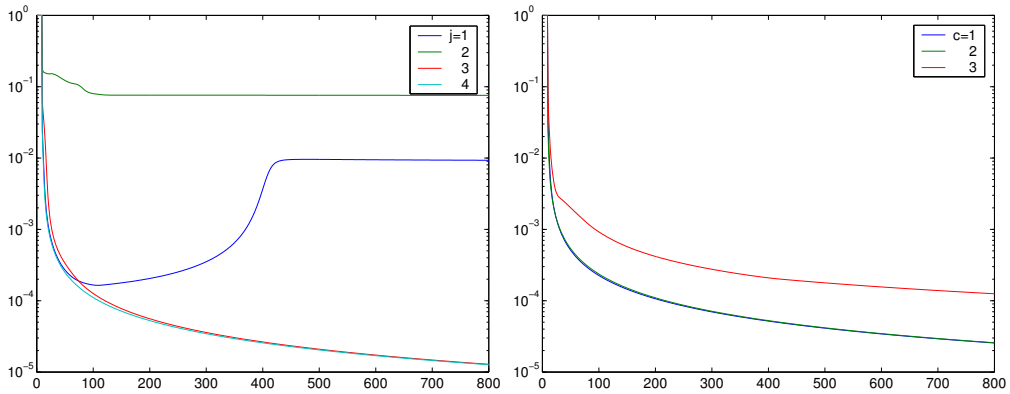
Of course, with a smaller data set, the model may begin to remove some parameters corresponding to arcs of influence between variables across time steps, or between the inputs and the dynamics or outputs. This and the previous experiment suggest that with enough data, the algorithm will generally discover a good model for the data, and indeed recover the true (or equivalent) model if the data was in fact generated from a model within the class of models accessible by the specified input-dependent linear dynamical system.

Although not observed in the experiment presented here, some caution needs to be taken with much larger sequences to avoid local minima in the optimisation. In the larger data sets the problems of local maxima or very long plateau regions in the optimisation become more frequent, with certain dimensions of the latent space modelling either the dynamics or the output processes, but not both (or neither). This problem is due to the presence of a dynamics model coupling the data across each time step. Recall that in the factor analysis model (chapter 4), because of the spherical factor noise model, ARD can rotate the factors into a basis where the outgoing weights for some factors can be set to zero (by taking their precisions to infinity). Unfortunately this degeneracy is not present for the hidden state variables of the LDS model, and so concerted efforts are required to rotate the hidden state along the entire sequence.

5.5 Elucidating gene expression mechanisms

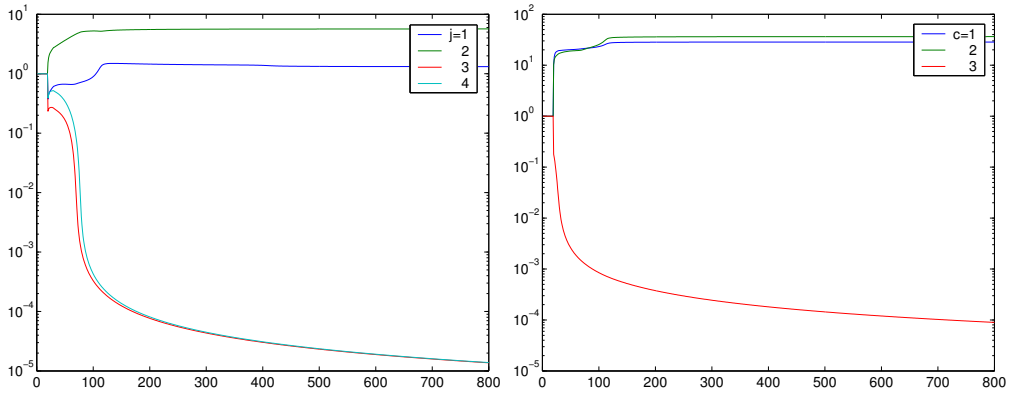
Description of the process and data

The data consists of $n = 34$ time series of the expressions of genes involved in a transcriptional process in the nuclei of human T lymphocytes. Each sequence consists of $T = 10$ measurements of the expressions of $p = 88$ genes, at time points $(0, 2, 4, 6, 8, 18, 24, 32, 48, 72)$ hours after a treatment to initiate the transcriptional process (see Rangel et al., 2001, section 2.1). For each sequence, the expression levels of each gene were normalised to have mean 1, by dividing by the mean gene expression over the 10 time steps. This normalisation reflects our interest in



(a) Prior variance on each column of A, $\frac{1}{\boldsymbol{\alpha}}$.

(b) Prior variance on each column of B, $\frac{1}{\boldsymbol{\beta}}$.



(c) Prior variance on each column of C, $\frac{1}{\boldsymbol{\gamma}}$.

(d) Prior variance on each column of D, $\frac{1}{\boldsymbol{\delta}}$.

Figure 5.8: Evolution of the hyperparameters with iterations of variational Bayesian EM, for the input-driven model trained on the data shown in figure 5.6 (see section 5.4.2). Each plot shows the reciprocal of the components of a hyperparameter vector, corresponding to the prior variance of the entries of each column of the relevant matrix. The hyperparameter optimisation is activated after 10 iterations of VBEM for the dynamics-related hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, after 20 iterations for the output-related hyperparameters $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$, and after 30 for the remaining hyperparmeters. **(a)**: After 150 iterations of VBEM, $\frac{1}{\alpha_3} \to 0$ and $\frac{1}{\alpha_4} \to 0$, which corresponds to the entries in the 3rd and 4th columns of A tending to zero. Thus only the remaining two hidden dimensions (1,2) are being used for the dynamics process. **(b)**: All hyperparameters in the $\boldsymbol{\beta}$ vector grow large, corresponding to each of the column entries in B being distributed about zero with high precision; thus none of the dimensions of the input vector is being used to modulate the hidden state. **(c)**: Similar to the A matrix, two hyperparameters in the vector $\boldsymbol{\gamma}$ remain small, and the remaining two increase without bound, $\frac{1}{\gamma_3} \to 0$ and $\frac{1}{\gamma_4} \to 0$. This corresponds to just two hidden dimensions (factors) causing the observed data through the C embedding. These are the *same* dimensions as used for the dynamics process, agreeing with the mechanism that generated the data. **(d)**: Just one hyperparameter, $\frac{1}{\delta_3} \to 0$, corresponding to the model ignoring the third dimension of the input, which is a confusing input unused in the true generation process (as can be seen from figure 5.6(a)). Thus the model learns that this dimension is irrelevant to modelling the data.

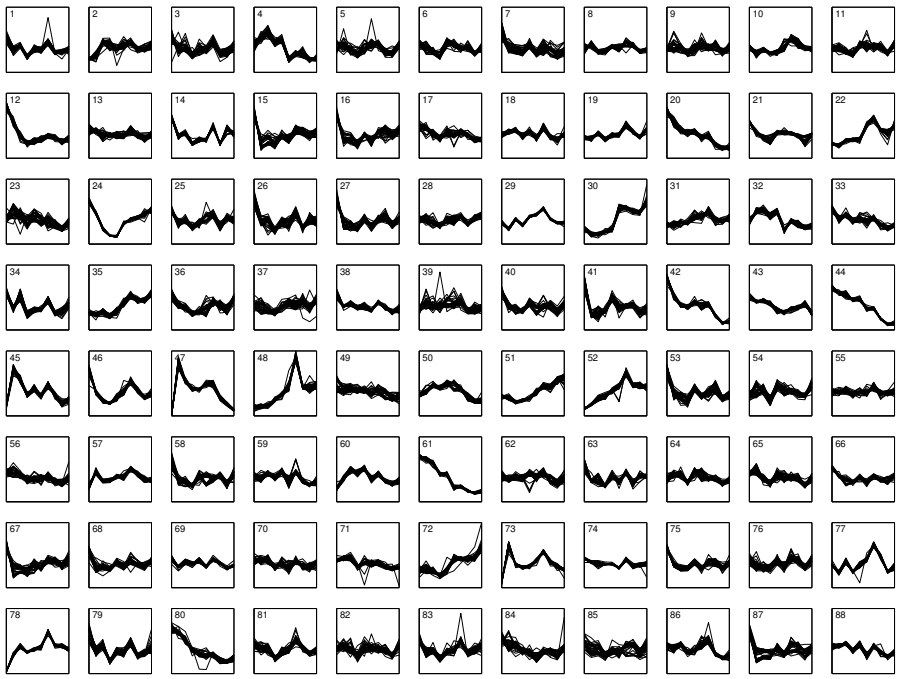


Figure 5.9: The gene expression data of Rangel et al. (2001). Each of the 88 plots corresponds to a particular gene on the array, and contains all of the recorded 34 sequences each of length 10.

the profiles of the genes rather than the absolute expression levels. Figure 5.9 shows the entire collection of normalised expression levels for each gene.

A previous approach to modelling gene expression levels which used graphical models to model the causal relationships between genes is presented in Friedman et al. (2000). However, this approach ignored the temporal dependence of the gene intensities during trials and went only as far as to infer the causal relationships between the genes within one time step. Their method discretised expression levels and made use of efficient candidate proposals and greedy methods for searching the space of model structures. This approach also assumed that all the possibly interacting variables are observed on the microarray. This precludes the existence of hidden causes or unmeasured genes whose involvement might dramatically simplify the network structure and therefore ease interpretability of the mechanisms in the underlying biological process.

Linear dynamical systems and other kinds of possibly nonlinear state-space models are a good class of model to begin modelling this gene expression data. The gene expression measurements are the noisy 88-dimensional outputs of the linear dynamical system, and the hidden states of the model correspond to unobserved factors in the gene transcriptional process which are not recorded in the DNA microarray — they might correspond simply to unmeasured genes, or they could model more abstractly the effect of players other than genes, for example regulatory proteins and background processes such as mRNA degradation.

Some aspects of using the LDS model for this data are not ideal. For example, we make the assumptions that the dynamics and output processes are time invariant, which is unlikely in a real biological system. Furthermore the times at which the data are taken are not linearly-spaced (see above), which might imply that there is some (possibly well-studied) non-linearity in the rate of the transcriptional process; worse still, there may be whole missing time slices which, if they had been included, would have made the dynamics process closer to stationary. There is also the usual limitation that the noise in the dynamics and output processes is almost certainly not Gaussian.

Experiment results

In this experiment we use the input-dependent LDS model, and *feed back* the gene expressions from the previous time step into the input for the current time step; in doing so we attempt to discover gene-gene interactions across time steps (in a causal sense), with the hidden state in this model now really representing unobserved variables. An advantage of this architecture is that we can now use the ARD mechanisms to determine which genes are influential across adjacent time slices, just as before (in section 5.4.2) we determined which inputs were relevant to predicting the data.

A graphical model for this setup is given in figure 5.10. When the input is replaced with the previous time step's observed data, the equations for the state-space model can be rewritten from equations (5.4) and (5.5) into the form:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{y}_{t-1} + \mathbf{w}_t \tag{5.152}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{y}_{t-1} + \mathbf{v}_t \ . \tag{5.153}$$

As a function only of the data at the previous time step, $\mathbf{y}_{t-1}$, the data at time t can be written

$$\mathbf{y}_t = (CB + D)\mathbf{y}_{t-1} + \mathbf{r}_t \ , \tag{5.154}$$

where $\mathbf{r}_t = \mathbf{v}_t + C\mathbf{w}_t + CA\mathbf{x}_{t-1}$ includes all contributions from noise and previous states. Thus to first order the interaction between gene d and gene a can be characterised by the element $[CB + D]_{ad}$ of the matrix. Indeed this matrix need not be symmetric and the element represents activation or inhibition from gene d to gene a at the next time step, depending on its sign. We will return to this quantity shortly.

5.5.1 Generalisation errors

For this experiment we trained both variational Bayesian and MAP LDS models on the first 30 of the 34 gene sequences, with the dimension of the hidden state ranging from $k = 1$ to

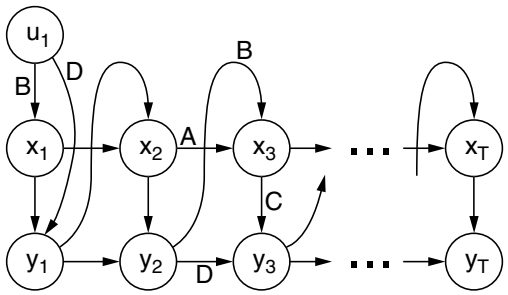


Figure 5.10: The feedback graphical model with outputs feeding into inputs.

20. The remaining 4 sequences were set aside as a test set. Since we required an input at time $t = 1$, $\mathbf{u}_1$, the observed sequences that were learnt began from time step $t = 2$. The MAP LDS model was implemented using the VB LDS with the following two modifications: first, the hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ and a, b were not optimised (however, the auxiliary state prior mean $\boldsymbol{\mu}_0$ and covariance Σ_0 were learnt); second, the sufficient statistics for the parameters were artificially boosted by a large factor to simulate delta functions for the posterior — i.e. in the limit of large n the VBM step recovers the MAP M step estimate of the parameters.

Both algorithms were run for 300 EM iterations, with no restarts. The one-step-ahead mean total square reconstruction error was then calculated for both the training sequences and the test sequences using the learnt models; the reconstruction of the tth observation for the ith sequence, $\mathbf{y}_{i,t}$, was made like so:

$$\hat{\mathbf{y}}_{i,t}^{\text{MAP}} = C_{\text{MAP}} \langle \mathbf{x}_{i,t} \rangle_{q_{\mathbf{x}}} + D_{\text{MAP}} \mathbf{y}_{i,t-1} \tag{5.155}$$

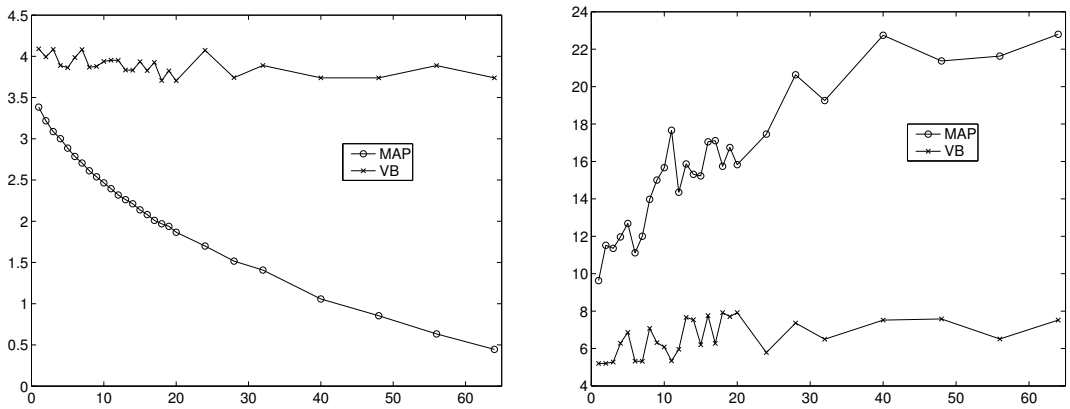
$$\hat{\mathbf{y}}_{i,t}^{\text{VB}} = \langle C \rangle_{q_C} \langle \mathbf{x}_{i,t} \rangle_{q_{\mathbf{x}}} + \langle D \rangle_{q_D} \mathbf{y}_{i,t-1} \ . \tag{5.156}$$

To clarify the procedure: to reconstruct the observations for the ith sequence, we use the entire observation sequence $\mathbf{y}_{i,1:T}$ to first infer the distribution over the hidden state sequence $\mathbf{x}_{i,1:T}$, and then we attempt to reconstruct each $\mathbf{y}_{i,t}$ using just the hidden state $\mathbf{x}_{i,t}$ and $\mathbf{y}_{i,t-1}$. The form given for the VB reconstruction in (5.156) is valid since, subject to the approximate posterior: all of the variational posterior distributions over the parameters and hidden states are Gaussian, C and $\mathbf{x}_t$ are independent, and the noise is Student-t distributed with mean zero.

Thus for each value of k, and for each of the MAP and VB learnt models, the total squared error per sequence is calculated according to:

$$E_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{i \in \text{train}} \sum_{t=2}^{T_i} (\hat{\mathbf{y}}_{i,t} - \mathbf{y}_{i,t})^2 \tag{5.157}$$

$$E_{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{i \in \text{test}} \sum_{t=2}^{T_i} (\hat{\mathbf{y}}_{i,t} - \mathbf{y}_{i,t})^2 \ . \tag{5.158}$$



(a) Training set error per sequence.

(b) Test set error per sequence.

Figure 5.11: The per sequence squared reconstruction error for one-step-ahead prediction (see text), as a function of the dimension of the hidden state, ranging from $k = 1$ to 64, on **(a)** the 30 training sequences, and **(b)** the 4 test sequences.

Figure 5.11 shows the squared reconstruction error for one-step-ahead prediction, as a function of the dimension of the hidden state for both the training and test sequences. We see that the MAP LDS model achieves a decreasing reconstruction error on the training set as the dimensionality of the hidden state is increased, whereas VB produces an approximately constant error, albeit higher. On prediction for the test set, MAP LDS performs very badly and increasingly worse for more complex learnt models, as we would expect; however, the VB performance is roughly constant with increasing k, suggesting that VB is using the ARD mechanism successfully to discard surplus modelling power. The test squared prediction error is slightly more than that on the training set, suggesting that VB is overfitting slightly.

5.5.2 Recovering gene-gene interactions

We now return to the interactions between genes d and a – more specifically the influence of gene d on gene a – in the matrix $[CB + D]$. Those entries in the matrix which are significantly different from zero can be considered as candidates for 'interactions'. Here we consider an entry to be significant if the zero point is more than 3 standard deviations from the posterior mean for that entry (based on the variational posterior distribution for the entry). Calculating the significance for the combined $CB+D$ matrix is laborious, and so here we provide results for only the D matrix. Since there is a degeneracy in the feedback model, we chose to effectively remove the first term, CB, by constraining all (but one) of the hyperparameters in $\boldsymbol{\beta}$ to very high values. The spared hyperparameter in $\boldsymbol{\beta}$ is used to still model an offset in the hidden dynamics using the bias input. This process essentially enforces $[CB]_{ad} = 0$ for all gene-gene pairs, and so simplifies the interpretation of the learnt model.

Figure 5.12 shows the interaction matrix learnt by the MAP and VB models (with the column corresponding the bias removed), for the case of $k = 2$ hidden state dimensions. For the MAP result we simply show $D + CB$. We see that the MAP and VB matrices share some aspects in terms of the signs and size of some of the interactions, but under the variational posterior only a few of the interactions are significantly non-zero, leading to a very sparse interaction matrix (see figure 5.13). Unfortunately, due to proprietary restrictions on the expression data the identities of the genes cannot be published here, so it is hard to give a biological interpretation to the network in figure 5.13. The hope is that these graphs suggest interactions which agree qualitatively with the transcriptional mechanisms already established in the research community. The ultimate result would be to be able to confidently predict the existence of as-yet-undocumented mechanisms to stimulate and guide future biological experiments. The VB LDS algorithm may provide a useful starting point for this research programme.

5.6 Possible extensions and future research

The work in this chapter can be easily extended to linear-Gaussian state-space models on trees, rather than chains, which could be used to model a variety of data. Moreover, for multiply-connected graphs, the VB propagation subroutine can still be used within a structured VB approximation.

Another interesting application of this body of theory could be to a Bayesian version of what we call a *switching state-space model* (SwSSM), which has the following dynamics:

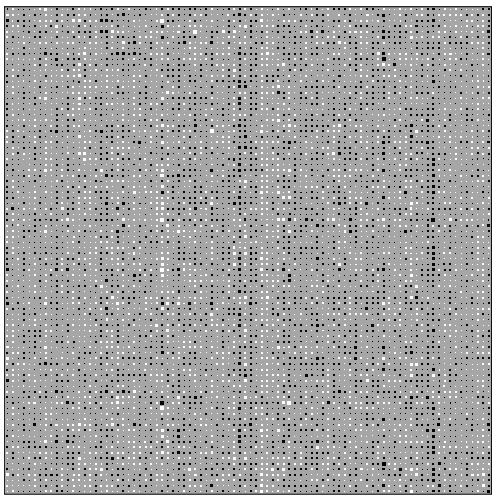
$$\text{a switch variable } s_t \text{ with dynamics} \quad p(s_t = i \,|\, s_{t-1} = j) = T_{ij} \;, \tag{5.159}$$

$$\text{hidden state dynamics} \quad p(\mathbf{x}_t \,|\, s_{t-1}, \mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t \,|\, A_{s_{t-1}}\mathbf{x}_{t-1}, Q_{s_{t-1}}) \;, \tag{5.160}$$

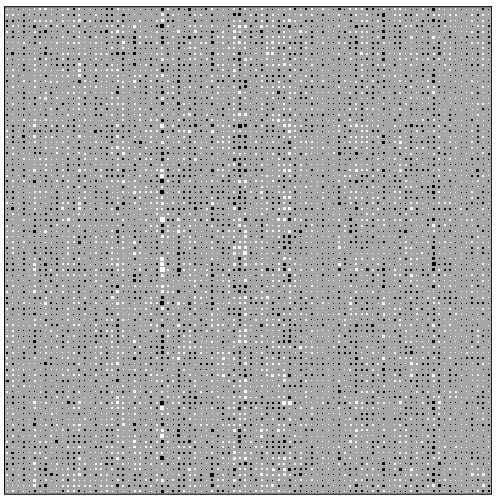
$$\text{and output function} \quad p(\mathbf{y}_t \,|\, s_t, \mathbf{x}_t) = \mathrm{N}(\mathbf{y}_t \,|\; C_{s_t}\mathbf{x}_t, R_{s_t}) \;. \tag{5.161}$$

That is to say we have a non-stationary switching linear dynamical system whose parameters are drawn from a finite set according to a discrete variable with its own dynamics. The appealing aspect of this model is that it contains many models as special cases, including: mixtures of factor analysers, mixtures of linear dynamical systems, Gaussian-output hidden Markov models, and mixtures of Gaussians. With appropriate optimisation of the lower bound on the marginal likelihood, one would hope that the data would provide evidence that one or other, or indeed hybrids, of the above special cases was the underlying generating model, or best approximates the true generating process in some sense. We have seen an example of variational Bayesian learning for hidden Markov models in chapter 3.

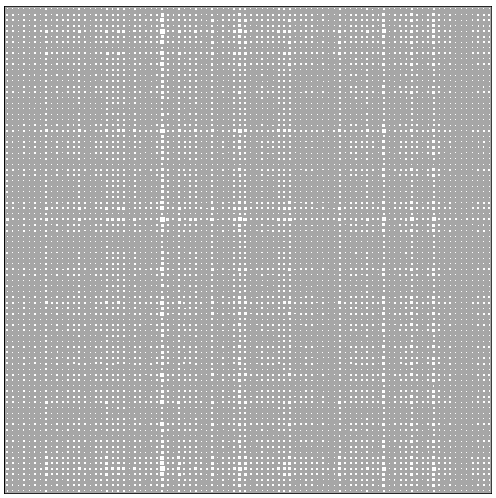
We have not commented on how reliably we expect the variational Bayesian method to approximate the marginal likelihood. Indeed a full analysis of the tightness of the variational bound



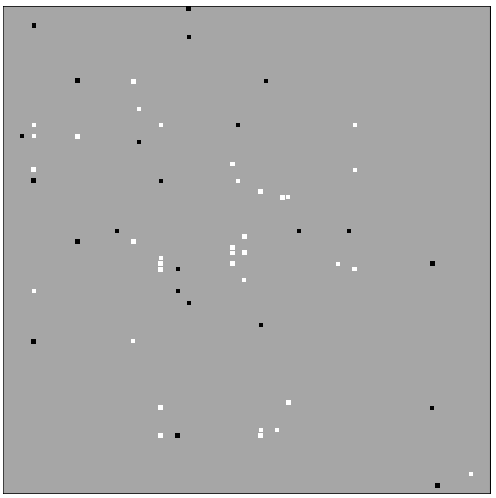
(a) The MAP EM solution $[D + CB]_{ad}$.



(b) Means $\langle D_{ad} \rangle$ after VBEM.



(c) Variances $\langle D_{ad}^2 \rangle - \langle D_{ad} \rangle^2$ after VBEM.



(d) Significant entries of D under $q_D(D)$.

Figure 5.12: The gene-gene interaction matrix learnt from the **(a)** MAP and **(b)** VB models (with the column corresponding to the bias input removed). Note that some of the entries are similar in each of the two matrices. Also shown is **(c)** the covariance of the posterior distribution of each element, which is a separable product of functions of each of the two genes' identities. Show in **(d)** are the entries of $\langle D_{ad} \rangle$ which are significantly far from zero, that is the value of zero is more than 3 standard deviations from the mean of the posterior.

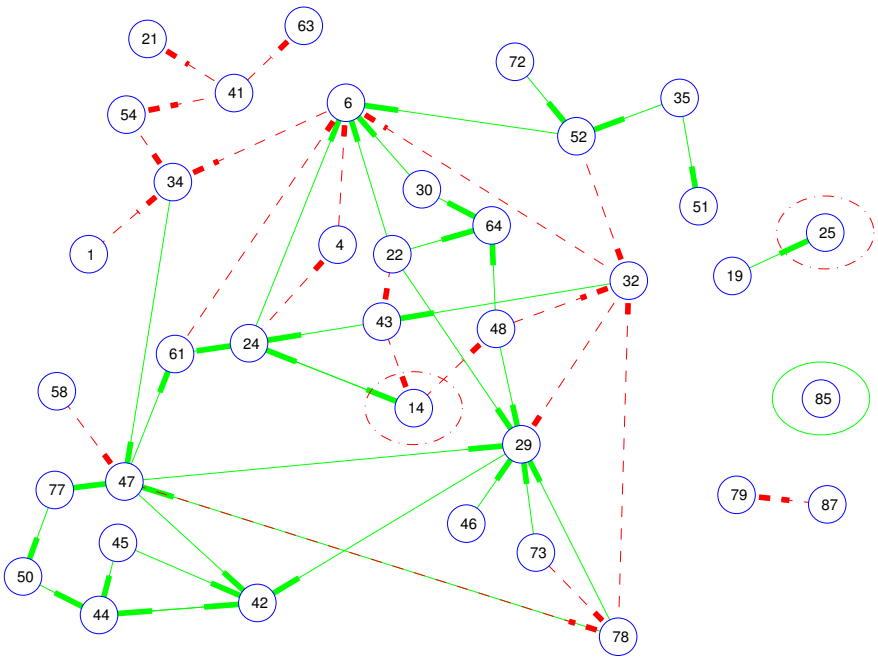


Figure 5.13: An example representation of the recovered interactions in the D matrix, as shown in figure 5.12(d). Each arc between two genes represents a significant entry in D. Red (dotted) and green (solid) denote inhibitory and excitatory influences, respectively. The direction of the influence is from the the thick end of the arc to the thin end. Ellipses denote self-connections. To generate this plot the genes were placed randomly and then manipulated slightly to reduce arc-crossing.

would require sampling for this model (as carried out in Früwirth-Schnatter, 1995, for example). This is left for further work, but the reader is referred to chapter 4 of this thesis and also to Miskin (2000), where sampling estimates of the marginal likelihood are directly compared to the VB lower bound and found to be comparable for practical problems.

We can also model higher than first order Markov processes using this model, by extending the feedback mechanism used in section 5.5. This could be achieved by feeding back concatenated observed data $\mathbf{y}_{t-d:t-1}$ into the current input vector u_t, where d is related to the maximum order present in the data. This procedure is common practice to model higher order data, but in our Bayesian scheme we can also learn posterior uncertainties for the parameters of the feedback, and can entirely remove some of the inputs via the hyperparameter optimisation.

This chapter has dealt solely with the case of linear dynamics and linear output processes with Gaussian noise. Whilst this is a good first approximation, there are many scenarios in which a non-linear model is more appropriate, for one or both of the processes. For example, Särelä et al. (2001) present a model with factor analysis as the output process and a two layer MLP network to model a non-linear dynamics process from one time step to the next, and Valpola and Karhunen (2002) extend this to include a non-linear output process as well. In both, the posterior is assumed to be of (constrained) Gaussian form and a variational optimisation is performed to learn the parameters and infer the hidden factor sequences. However, their model does not exploit the full forward-backward propagation and instead updates the hidden state one step forward and backward in time at each iteration.

5.7 Summary

In this chapter we have shown how to approximate the marginal likelihood of a Bayesian linear dynamical system using variational methods. Since the complete-data likelihood for the LDS model is in the conjugate-exponential family it is possible to write down a VBEM algorithm for inferring the hidden state sequences whilst simultaneously maintaining uncertainty over the parameters of the model, subject to the approximation that the hidden variables and parameters are independent given the data.

Here we have had to rederive the forward and backward passes in the VBE step in order for them to take as input the natural parameter expectations from the VBM step. It is an open problem to prove that for LDS models the natural parameter mapping $\boldsymbol{\phi}(\boldsymbol{\theta})$ is not invertible; that is to say there exists no $\tilde{\boldsymbol{\theta}}$ in general that satisfies $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$. We have therefore derived here the variational Bayesian counterparts of the Kalman filter and Rauch-Tung-Striebel smoother, which can in fact be supplied with *any* distribution over the parameters. As with other conjugate-exponential VB treatments, the propagation algorithms have the same complexity as the MAP point-parameter versions.

We have shown how the algorithm can use the ARD procedure of optimising precision hyperparameters to discover the structure of models of synthetic data, in terms of the number of required hidden dimensions. By feeding back previous data into the inputs of the model we have shown how it is possible to elucidate interactions between genes in a transcription mechanism from DNA microarray data. Collaboration is currently underway to interpret these results (personal communication with D. Wild and C. Rangel).

Chapter 6

Learning the structure of discrete-variable graphical models with hidden variables

6.1 Introduction

One of the key problems in machine learning and statistics is how to learn the structure of graphical models from data. This entails determining the dependency relations amongst the model variables that are supported by the data. Models of differing complexities can be rated according to their posterior probabilities, which by Bayes' rule are related to the marginal likelihood under each candidate model.

In the case of fully observed discrete-variable directed acyclic graphs with Dirichlet priors on the parameters it is tractable to compute the marginal likelihood of a candidate structure and therefore obtain its posterior probability (or a quantity proportional to this). Unfortunately, in graphical models containing hidden variables the calculation of the marginal likelihood is generally intractable for even moderately sized data sets, and its estimation presents a difficult challenge for approximate methods such as asymptotic-data criteria and sampling techniques.

In this chapter we investigate a novel application of the VB framework to approximating the marginal likelihood of discrete-variable directed acyclic graph (DAG) structures that contain hidden variables. We call approximations to a model's marginal likelihood *scores*. We first derive the VB score, which is simply the result of a VBEM algorithm applied to DAGs, and then assess its performance on a model selection task: finding the particular structure (out of a small class of structures) that gave rise to the observed data. We also derive and evaluate the BIC and Cheeseman-Stutz (CS) scores and compare these to VB for this problem.

We also compare the BIC, CS, and VB scoring techniques to annealed importance sampling (AIS) estimates of the marginal likelihood. We consider AIS to be a "gold standard", the best method for obtaining reliable estimates of the marginal likelihoods of models explored in this chapter (personal communication with C. Rasmussen, Z. Ghahramani, and R. Neal). We have used AIS in this chapter to perform the first serious case study of the tightness of variational bounds. An analysis of the limitations of AIS is also provided. The aim of the comparison is to convince us of the reliability of VB as an estimate of the marginal likelihood in the general incomplete-data setting, so that it can be used in larger problems, for example embedded in a (greedy) structure search amongst a much larger class of models.

In section 6.2 we begin by examining the model selection question for discrete directed acyclic graphs, and show how exact marginal likelihood calculation rapidly becomes computationally intractable when the graph contains hidden variables. In section 6.3 we briefly cover the EM algorithm for ML and MAP parameter estimation in DAGs with hidden variables, and discuss the BIC, Laplace and Cheeseman-Stutz asymptotic approximations. We then present the VBEM algorithm for variational Bayesian lower bound optimisation, which in the case of discrete DAGs is a straightforward generalisation of the MAP EM algorithm. In section 6.3.5 we describe in detail an annealed importance sampling method for estimating marginal likelihoods of discrete DAGs. In section 6.4 we evaluate the performance of these different scoring methods on the simple (yet non-trivial) model selection task of determining which of all possible structures within a class generated a data set. Section 6.5 discusses some related topics which expand on the methods used in this chapter: first, we give an analysis of the limitations of the AIS implementation and suggest possible extensions for it; second, we more thoroughly consider the parameter-counting arguments used in the BIC and CS scoring methods, and reformulate a more successful score. Finally we conclude in section 6.6 and suggest directions for future research.

6.2 Calculating marginal likelihoods of DAGs

Consider a data set of size n, $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, modelled by the discrete directed acyclic graph consisting of hidden and observed variables $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\} = \{\mathbf{s}_1, \mathbf{y}_1, \ldots, \mathbf{s}_n, \mathbf{y}_n\}$. The variables in each plate $i = 1, \ldots, n$ are indexed by $j = 1, \ldots, |\mathbf{z}_i|$, of which some $j \in \mathcal{H}$ are hidden and $j \in \mathcal{V}$ are observed variables, i.e. $\mathbf{s}_i = \{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}$ and $\mathbf{y}_i = \{\mathbf{z}_{ij}\}_{j \in \mathcal{V}}$.

On a point of nomenclature, note that $\mathbf{z}_i = \{\mathbf{s}_i, \mathbf{y}_i\}$ contains both hidden and observed variables, and we interchange freely between these two forms where convenient. Moreover, the numbers of hidden and observed variables, $|\mathbf{s}_i|$ and $|\mathbf{y}_i|$, are allowed to vary with the data point index i. An example of such a case could be a data set of sequences of varying length, to be modelled by an HMM. Note also that the meaning of $|\cdot|$ varies depending on the type of its argument, for

example: $|\mathbf{z}|$ is the number of data points, n; $|\mathbf{s}_i|$ is the number of hidden variables (for the ith data point); $|\mathbf{s}_{ij}|$ is the cardinality (number of settings) of the jth hidden variable (for the ith data point).

In a DAG the complete-data likelihood factorises into a product of local probabilities on each variable

$$p(\mathbf{z} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \;, \tag{6.1}$$

where $\mathbf{pa}(j)$ denotes the vector of indices of the parents of the jth variable. Each variable in the model is multinomial, and the parameters of the model are different vectors of probabilities on each variable for each configuration of its parents. For example, the parameter for a binary variable which has two ternary parents is a $3^2 \times 2$ matrix with each row summing to one. Should there be a variable j without any parents ($\mathbf{pa}(j) = \emptyset$), then the parameter associated with variable j is simply a vector of its prior probabilities. If we use θ_{jlk} to denote the probability that variable j takes on value k when its parents are in configuration l, then the complete likelihood can be written out as a product of terms of the form

$$p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) = \prod_{l=1}^{\left|\mathbf{z}_{i\mathbf{pa}(j)}\right|} \prod_{k=1}^{|\mathbf{z}_{ij}|} \theta_{jlk}^{\delta(\mathbf{z}_{ij},k)\delta(\mathbf{z}_{i\mathbf{pa}(j)},l)} \tag{6.2}$$

$$\text{with} \quad \sum_k \theta_{jlk} = 1 \quad \forall \, \{j, l\} \;. \tag{6.3}$$

Here we use $\left|\mathbf{z}_{i\mathbf{pa}(j)}\right|$ to denote the number of joint settings of the parents of variable j. That is to say the probability is a product over both all the $\left|\mathbf{z}_{i\mathbf{pa}(j)}\right|$ possible settings of the parents and the $|\mathbf{z}_{ij}|$ settings of the variable itself. Here we use Kronecker-δ notation which is 1 if its arguments are identical and zero otherwise. The parameters of the model are given independent Dirichlet priors, which are conjugate to the complete-data likelihood above (see equation (2.80), which is Condition 1 for conjugate-exponential models). By independent we mean factorised over variables and parent configurations; these choices then satisfy the *global* and *local* independence assumptions of Heckerman et al. (1995). For each parameter $\boldsymbol{\theta}_{jl} = \{\theta_{jl1}, \ldots, \theta_{jl|\mathbf{z}_{ij}|}\}$, the Dirichlet prior is

$$p(\boldsymbol{\theta}_{jl} \,|\, \boldsymbol{\lambda}_{jl}, m) = \frac{\Gamma(\lambda_{jl}^0)}{\prod_k \Gamma(\lambda_{jlk})} \prod_k \theta_{jlk}^{\lambda_{jlk}-1} \;, \tag{6.4}$$

where $\boldsymbol{\lambda}$ are hyperparameters:

$$\boldsymbol{\lambda}_{jl} = \{\lambda_{jl1}, \ldots, \lambda_{jl|\mathbf{z}_{ij}|}\} \tag{6.5}$$

and

$$\lambda_{jlk} > 0 \quad \forall \, k \;, \qquad \lambda_{jl}^0 = \sum_k \lambda_{jlk} \;. \tag{6.6}$$

This form of prior is assumed throughout the chapter. Since the focus of this chapter is not on optimising these hyperparameters, we use the shorthand $p(\boldsymbol{\theta} \,|\, m)$ to denote the prior from here on. In the discrete-variable case we are considering, the complete-data marginal likelihood is tractable to compute:

$$p(\mathbf{z} \,|\, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m) p(\mathbf{z} \,|\, \boldsymbol{\theta}) \tag{6.7}$$

$$= \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m) \prod_{i=1}^{n} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \tag{6.8}$$

$$= \prod_{j=1}^{|\mathbf{z}_i|} \prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \frac{\Gamma(\lambda^0_{jl})}{\Gamma(\lambda^0_{jl} + N_{jl})} \prod_{k=1}^{|\mathbf{z}_{ij}|} \frac{\Gamma(\lambda_{jlk} + N_{jlk})}{\Gamma(\lambda_{jlk})} \tag{6.9}$$

where N_{jlk} is defined as the count in the data for the number of instances of variable j being in configuration k with parental configuration l:

$$N_{jlk} = \sum_{i=1}^{n} \delta(\mathbf{z}_{ij}, k) \delta(\mathbf{z}_{i\mathbf{pa}(j)}, l), \qquad \text{and} \quad N_{jl} = \sum_{k=1}^{|\mathbf{z}_{ij}|} N_{jlk} \; . \tag{6.10}$$

The incomplete-data likelihood, however, is not as tractable. It results from summing over all settings of the hidden variables and taking the product over i.i.d. presentations of the data:

$$p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{y}_i \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{\{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \; . \tag{6.11}$$

This quantity can be evaluated as the product of n quantities, each of which is a summation over all possible joint configurations of the hidden variables; in the worst case this computation requires $\mathcal{O}(n \prod_{j \in \mathcal{H}} |\mathbf{z}_{ij}|)$ operations (although this can usually be made more efficient with the use of propagation algorithms that exploit the topology of the model). The incomplete-data marginal likelihood for n cases follows from marginalising out the parameters of the model:

$$p(\mathbf{y} \,|\, m) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta} \,|\, m) \prod_{i=1}^{n} \sum_{\{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \; . \tag{6.12}$$

This expression is computationally intractable due to the expectation over the real-valued conditional probabilities $\boldsymbol{\theta}$, which couples the hidden variables across i.i.d. data. In the worst case it can be evaluated as the sum of $\left(\prod_{j \in \mathcal{H}} |\mathbf{z}_{ij}| \right)^n$ Dirichlet integrals. For example, a model with just $|\mathbf{s}_i| = 2$ hidden variables and 100 data points requires the evaluation of 2^{100} Dirichlet integrals. This means that a linear increase in the amount of observed data results in an exponential increase in the cost of inference.

We focus on the task of learning the conditional independence structure of the model, that is, which variables are parents of each variable. We compare structures based on their posterior probabilities. In this chapter we assume that the prior, $p(m)$, is uninformative, and so all our information comes from the intractable marginal likelihood, $p(\mathbf{y} \mid m)$.

In the rest of this chapter we examine several methods to approximate this Bayesian integration (6.12), in order to make learning and inference tractable. For the moment we assume that the cardinalities of the variables, in particular the hidden variables, are fixed beforehand. The related problem of determining the cardinality of the variables from data can be addressed in the same framework, as we have already seen for HMMs in chapter 3.

6.3 Estimating the marginal likelihood

In this section we look at some approximations to the marginal likelihood, which we refer to henceforth as *scores*. We first review ML and MAP parameter learning and briefly present the EM algorithm for a general discrete-variable directed graphical model with hidden variables. From the result of the EM optimisation, we can construct various asymptotic approximations to the marginal likelihood, deriving the BIC and Cheeseman-Stutz scores. We then apply the variational Bayesian framework, which in the case of conjugate-exponential discrete directed acyclic graphs produces a very simple VBEM algorithm, which is a direct extension of the EM algorithm for MAP parameter learning. Finally, we derive an *annealed importance sampling* method (AIS) for this class of graphical model, which is considered to be the current state-of-the-art technique for estimating the marginal likelihood of these models using sampling — we then compare the different scoring methods to it. We finish this section with a brief note on some trivial and non-trivial upper bounds to the marginal likelihood.

6.3.1 ML and MAP parameter estimation

The EM algorithm for ML/MAP estimation can be derived using the lower bound interpretation as was described in section 2.2. We begin with the incomplete-data log likelihood, and lower bound it by a functional $\mathcal{F}(q_{\mathbf{s}}(\mathbf{s}), \boldsymbol{\theta})$ as follows

$$\ln p(\mathbf{y} \mid \boldsymbol{\theta}) = \ln \prod_{i=1}^{n} \sum_{\{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \mid \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \tag{6.13}$$

$$\geq \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{\prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \mid \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta})}{q_{\mathbf{s}_i}(\mathbf{s}_i)} \tag{6.14}$$

$$= \mathcal{F}(\{q_{\mathbf{s}_i}(\mathbf{s}_i)\}_{i=1}^{n}, \boldsymbol{\theta}) \,, \tag{6.15}$$

where we have introduced a distribution $q_{\mathbf{s}_i}(\mathbf{s}_i)$ over the hidden variables $\mathbf{s}_i$ for each data point $\mathbf{y}_i$. We remind the reader that we have used $\mathbf{s}_i = \{\mathbf{z}_{ij}\}_{j\in\mathcal{H}}$ in going from (6.13) to (6.14). On taking derivatives of $\mathcal{F}(\{q_{\mathbf{s}_i}(\mathbf{s}_i)\}_{i=1}^n, \boldsymbol{\theta})$ with respect to $q_{\mathbf{s}_i}(\mathbf{s}_i)$, the optimal setting of the variational posterior is given exactly by the posterior

$$q_{\mathbf{s}_i}(\mathbf{s}_i) = p(\mathbf{s}_i \,|\, \mathbf{y}_i, \boldsymbol{\theta}) \quad \forall\, i\,. \tag{6.16}$$

This is the E step of the EM algorithm; at this setting of the distribution $q_{\mathbf{s}_i}(\mathbf{s}_i)$ it can be easily shown that the bound (6.14) is tight (see section 2.2.2).

The M step of the algorithm is derived by taking derivatives of the bound with respect to the parameters $\boldsymbol{\theta}$. Each $\boldsymbol{\theta}_{jl}$ is constrained to sum to one, and so we enforce this with Lagrange multipliers c_{jl},

$$\frac{\partial}{\partial\theta_{jlk}}\mathcal{F}(q_{\mathbf{s}}(\mathbf{s}), \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i)\frac{\partial}{\partial\theta_{jlk}}\ln p(\mathbf{z}_{ij} \,|\, \mathbf{x}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}_j) + c_{jl} \tag{6.17}$$

$$= \sum_{i=1}^{n}\sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i)\delta(\mathbf{z}_{ij}, k)\delta(\mathbf{z}_{i\mathbf{pa}(j)}, l)\frac{\partial}{\partial\theta_{jlk}}\ln\theta_{jlk} + c_{jl} \tag{6.18}$$

$$= 0\,, \tag{6.19}$$

which upon rearrangement gives

$$\theta_{jlk} \propto \sum_{i=1}^{n}\sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i)\delta(\mathbf{z}_{ij}, k)\delta(\mathbf{z}_{i\mathbf{pa}(j)}, l)\,. \tag{6.20}$$

Due to the normalisation constraint on $\boldsymbol{\theta}_{jl}$ the M step can be written

$$\textbf{M step (ML):} \qquad \theta_{jlk} = \frac{N_{jlk}}{\sum_{k'=1}^{|\mathbf{z}_{ij}|} N_{jlk'}}\,, \tag{6.21}$$

where the N_{jlk} are defined as

$$N_{jlk} = \sum_{i=1}^{n}\left\langle\delta(\mathbf{z}_{ij}, k)\delta(\mathbf{z}_{i\mathbf{pa}(j)}, l)\right\rangle_{q_{\mathbf{s}_i}(\mathbf{s}_i)} \tag{6.22}$$

where angled-brackets $\langle\cdot\rangle_{q_{\mathbf{s}_i}(\mathbf{s}_i)}$ are used to denote expectation with respect to the hidden variable posterior $q_{\mathbf{s}_i}(\mathbf{s}_i)$. The N_{jlk} are interpreted as the expected number of counts for observing simultaneous settings of children and parent configurations over observed and hidden variables. In the cases where both j and $\mathbf{pa}(j)$ are observed variables, N_{jlk} reduces to the simple empirical count as in (6.10). Otherwise if j or its parents are hidden then expectations need be taken over the posterior $q_{\mathbf{s}_i}(\mathbf{s}_i)$ obtained in the E step.

If we require the MAP EM algorithm, we instead lower bound $\ln p(\boldsymbol{\theta})p(\mathbf{y} \mid \boldsymbol{\theta})$. The E step remains the same, but the M step uses augmented counts from the prior of the form in (6.4) to give the following update:

$$\textbf{M step} \text{ (MAP):} \qquad \theta_{jlk} = \frac{\lambda_{jlk} - 1 + N_{jlk}}{\sum_{k'=1}^{|\mathbf{z}_{ij}|} \lambda_{jlk'} - 1 + N_{jlk'}} \,. \tag{6.23}$$

Repeated applications of the E step (6.16) and the M step (6.21, 6.23) are guaranteed to increase the log likelihood (with equation (6.21)) or the log posterior (with equation (6.23)) of the parameters at every iteration, and converge to a local maximum.

As mentioned in section 1.3.1, we note that MAP estimation is basis-dependent. For any particular $\boldsymbol{\theta}^*$, which has non-zero prior probability, it is possible to find a (one-to-one) reparameterisation $\boldsymbol{\phi}(\boldsymbol{\theta})$ such that the MAP estimate for $\boldsymbol{\phi}$ is at $\boldsymbol{\phi}(\boldsymbol{\theta}^*)$. This is an obvious drawback of MAP parameter estimation. Moreover, the use of (6.23) can produce erroneous results in the case of $\lambda_{jlk} < 1$, in the form of negative probabilities. Conventionally, researchers have limited themselves to Dirichlet priors in which every $\lambda_{jlk} \geq 1$, although in MacKay (1998) it is shown how a reparameterisation of $\boldsymbol{\theta}$ into the softmax basis results in MAP updates which do not suffer from this problem (which look identical to (6.23), but without the -1 in numerator and denominator).

6.3.2 BIC

The Bayesian Information Criterion approximation, described in section 1.3.4, is the asymptotic limit to large data sets of the Laplace approximation. It is interesting because it does not depend on the prior over parameters, and attractive because it does not involve the burdensome computation of the Hessian of the log likelihood and its determinant. For the size of structures considered in this chapter, the Laplace approximation would be viable to compute, subject perhaps to a transformation of parameters (see for example MacKay, 1995). However in larger models the approximation may become unwieldy and further approximations would be required (see section 1.3.2).

For BIC, we require the number of free parameters in each structure. In the experiments in this chapter we use a simple counting argument; in section 6.5.2 we discuss a more rigorous method for estimating the dimensionality of the parameter space of a model. We apply the following counting scheme. If a variable j has no parents in the DAG, then it contributes $(|\mathbf{z}_{ij}| - 1)$ free parameters, corresponding to the degrees of freedom in its vector of prior probabilities (constrained to lie on the simplex $\sum_k p_k = 1$). Each variable that has parents contributes

$(|\mathbf{z}_{ij}| - 1)$ parameters for each configuration of its parents. Thus in model m the total number of parameters $d(m)$ is given by

$$d(m) = \sum_{j=1}^{|\mathbf{z}_i|} (|\mathbf{z}_{ij}| - 1) \prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \left|\mathbf{z}_{i\mathbf{pa}(j)l}\right| \ , \tag{6.24}$$

where $\left|\mathbf{z}_{i\mathbf{pa}(j)l}\right|$ denotes the cardinality (number of settings) of the lth parent of the jth variable. We have used the convention that the product over zero factors has a value of one to account for the case in which the jth variable has no parents, i.e.:

$$\prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \left|\mathbf{z}_{i\mathbf{pa}(j)l}\right| = 1 \ , \quad \text{if} \quad \left|\mathbf{z}_{i\mathbf{pa}(j)l}\right| = 0 \ . \tag{6.25}$$

The BIC approximation needs to take into account aliasing in the parameter posterior (as described in section 1.3.3). In discrete-variable DAGs, parameter aliasing occurs from two symmetries: first, a priori identical hidden variables can be permuted; and second, the labellings of the states of each hidden variable can be permuted. As an example, let us imagine the parents of a single observed variable are 3 hidden variables having cardinalities $(3, 3, 4)$. In this case the number of aliases is 1728 ($= 2! \times 3! \times 3! \times 4!$). If we assume that the aliases of the posterior distribution are well separated then the score is given by

$$\ln p(\mathbf{y} \,|\, m)_{\text{BIC}} = \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) - \frac{d(m)}{2} \ln n + \ln S \tag{6.26}$$

where S is the number of aliases, and $\hat{\boldsymbol{\theta}}$ is the MAP estimate as described in the previous section. This correction is accurate only if the modes of the posterior distribution are well separated, which should be the case in the large data set size limit for which BIC is useful. However, since BIC is correct only up to an indeterminant missing factor, we might think that this correction is not necessary. In the experiments we examine the BIC score with and without this correction, and also with and without the prior term included.

6.3.3 Cheeseman-Stutz

The Cheeseman-Stutz approximation uses the following identity for the incomplete-data marginal likelihood:

$$p(\mathbf{y} \,|\, m) = p(\mathbf{z} \,|\, m) \frac{p(\mathbf{y} \,|\, m)}{p(\mathbf{z} \,|\, m)} = p(\mathbf{z} \,|\, m) \frac{\int d\boldsymbol{\theta} \ p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)}{\int d\boldsymbol{\theta} \ p(\boldsymbol{\theta}' \,|\, m) p(\mathbf{z} \,|\, \boldsymbol{\theta}', m)} \tag{6.27}$$

which is true for any completion $\mathbf{z} = \{\hat{\mathbf{s}}, \mathbf{y}\}$ of the data. This form is useful because the complete-data marginal likelihood, $p(\mathbf{z} \,|\, m)$, is tractable to compute for discrete DAGs with

independent Dirichlet priors: it is just a product of Dirichlet integrals (see equation (6.9)). Using the results of section 1.3.2, in particular equation (1.45), we can apply Laplace approximations to both the numerator and denominator of the above fraction to give

$$p(\mathbf{y} \,|\, m) \approx p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) \frac{p(\hat{\boldsymbol{\theta}} \,|\, m) p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) \, |2\pi A|^{-1}}{p(\hat{\boldsymbol{\theta}}' \,|\, m) p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}') \, |2\pi A'|^{-1}} \; . \tag{6.28}$$

We assume that $p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}})$ is computable exactly. If the errors in each of the Laplace approximations are similar, then they should roughly cancel each other out; this will be the case if the shape of the posterior distributions about $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}'$ are similar. We can ensure that $\hat{\boldsymbol{\theta}}' = \hat{\boldsymbol{\theta}}$ by completing the hidden data $\{\mathbf{s}_i\}_{i=1}^n$ with their expectations under their posterior distributions $p(\mathbf{s}_i \,|\, \mathbf{y}, \hat{\boldsymbol{\theta}})$. That is to say the hidden states are completed as follows:

$$\hat{\mathbf{s}}_{ijk} = \langle \delta(\mathbf{s}_{ij}, k) \rangle_{q_{\mathbf{s}_i}(\mathbf{s}_i)} \; , \tag{6.29}$$

which will generally result in non-integer counts N_{jlk} on application of (6.22). Having computed these counts and re-estimated $\hat{\boldsymbol{\theta}}'$ using equation (6.23), we note that $\hat{\boldsymbol{\theta}}' = \hat{\boldsymbol{\theta}}$. The Cheeseman-Stutz approximation then results from taking the BIC-type asymptotic limit of both Laplace approximations in (6.28),

$$\begin{aligned} \ln p(\mathbf{y} \,|\, m)_{\text{CS}} &= \ln p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) + \ln p(\hat{\boldsymbol{\theta}} \,|\, m) + \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) - \frac{d}{2} \ln n \\ &\quad - \ln p(\hat{\boldsymbol{\theta}}' \,|\, m) - \ln p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) + \frac{d'}{2} \ln n \qquad (6.30) \\ &= \ln p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) + \ln p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) - \ln p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) \; , \qquad (6.31) \end{aligned}$$

where the last line follows from the modes of the Gaussian approximations being at the same point, $\hat{\boldsymbol{\theta}}' = \hat{\boldsymbol{\theta}}$, and also the assumption that the number of parameters in the models for complete and incomplete data are the same, i.e. $d = d'$ (Cheeseman and Stutz, 1996, but also see section 6.5.2). Each term of (6.31) can be evaluated individually:

$$\text{from (6.9)} \qquad p(\hat{\mathbf{s}}, \mathbf{y} \,|\, m) = \prod_{j=1}^{|\mathbf{z}_i|} \prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \frac{\Gamma(\lambda^0_{jl})}{\Gamma(\lambda_{jl} + \hat{N}_{jl})} \prod_{k=1}^{|\mathbf{z}_{ij}|} \frac{\Gamma(\lambda_{jlk} + \hat{N}_{jlk})}{\Gamma(\lambda_{jlk})} \tag{6.32}$$

$$\text{from (6.11)} \qquad p(\mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) = \prod_{i=1}^{n} \sum_{\{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}} \prod_{j=1}^{|\mathbf{z}_i|} \prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \prod_{k=1}^{|\mathbf{z}_{ij}|} \hat{\theta}_{jlk}^{\delta(\mathbf{z}_{ij},k)\delta(\mathbf{z}_{i\mathbf{pa}(j)},l)} \tag{6.33}$$

$$\text{from (6.1)} \qquad p(\hat{\mathbf{s}}, \mathbf{y} \,|\, \hat{\boldsymbol{\theta}}) = \prod_{j=1}^{|\mathbf{z}_i|} \prod_{l=1}^{|\mathbf{z}_{i\mathbf{pa}(j)}|} \prod_{k=1}^{|\mathbf{z}_{ij}|} \hat{\theta}_{jlk}^{\hat{N}_{jlk}} \tag{6.34}$$

where the $\hat{N}_{jlk}$ are identical to the N_{jlk} of equation (6.22) if the completion of the data with $\hat{\mathbf{s}}$ is done with the posterior found in the M step of the MAP EM algorithm used to find $\hat{\boldsymbol{\theta}}$. Equation

(6.33) is simply the output of the EM algorithm, equation (6.32) is a function of the counts obtained in the EM algorithm, and equation (6.34) is a simple computation again.

As with BIC, the Cheeseman-Stutz score also needs to be corrected for aliases in the parameter posterior, as described above, and is subject to the same caveat that these corrections are only accurate if the aliases in the posterior are well-separated.

We note that CS is a lower bound on the marginal likelihood, as shown in section 2.6.2 of this thesis. We will return to this point in the discussion of the experimental results.

6.3.4 The VB lower bound

The incomplete-data log marginal likelihood can be written as

$$\ln p(\mathbf{y} \,|\, m) = \ln \int d\boldsymbol{\theta} \; p(\boldsymbol{\theta} \,|\, m) \prod_{i=1}^{n} \sum_{\{\mathbf{z}_{ij}\}_{j \in \mathcal{H}}} \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}) \; . \tag{6.35}$$

We can form the lower bound in the usual fashion using $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $\{q_{\mathbf{s}_i}(\mathbf{s}_i)\}_{i=1}^{n}$ to yield (see section 2.3.1):

$$\begin{aligned} \ln p(\mathbf{y} \,|\, m) &\geq \int d\boldsymbol{\theta} \; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta} \,|\, m)}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \\ &\qquad + \sum_{i=1}^{n} \int d\boldsymbol{\theta} q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln \frac{p(\mathbf{z}_i \,|\, \boldsymbol{\theta}, m)}{q_{\mathbf{s}_i}(\mathbf{s}_i)} \qquad (6.36) \\ &= \mathcal{F}_m(q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), q(\mathbf{s})) \; . \qquad (6.37) \end{aligned}$$

We now take functional derivatives to write down the variational Bayesian EM algorithm (theorem 2.1, page 54). The VBM step is straightforward:

$$\ln q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \ln p(\boldsymbol{\theta} \,|\, m) + \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{z}_i \,|\, \boldsymbol{\theta}, m) + \mathrm{c} \; , \tag{6.38}$$

with c a constant. Given that the prior over parameters factorises over variables as in (6.4), and the complete-data likelihood factorises over the variables in a DAG as in (6.1), equation (6.38) can be broken down into individual derivatives:

$$\ln q_{\boldsymbol{\theta}_{jl}}(\boldsymbol{\theta}_{jl}) = \ln p(\boldsymbol{\theta}_{jl} \,|\, \boldsymbol{\lambda}_{jl}, m) + \sum_{i=1}^{n} \sum_{\mathbf{s}_i} q_{\mathbf{s}_i}(\mathbf{s}_i) \ln p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \boldsymbol{\theta}, m) + c_{jl} \; , \tag{6.39}$$

where $\mathbf{z}_{ij}$ may be either a hidden or observed variable, and each c_{jl} is a Lagrange multiplier from which a normalisation constant is obtained. Equation (6.39) has the form of the Dirichlet distribution. We define the expected counts under the posterior hidden variable distribution

$$N_{jlk} = \sum_{i=1}^{n} \left\langle \delta(\mathbf{z}_{ij}, k)\delta(\mathbf{z}_{i\mathbf{pa}(j)}, l) \right\rangle_{q_{\mathbf{s}_i}(\mathbf{s}_i)} . \tag{6.40}$$

Therefore N_{jlk} is the expected total number of times the jth variable (hidden or observed) is in state k when its parents (hidden or observed) are in state l, where the expectation is taken with respect to the posterior distribution over the hidden variables for each datum. Then the variational posterior for the parameters is given simply by (see theorem 2.2)

$$q_{\boldsymbol{\theta}_{jl}}(\boldsymbol{\theta}_{jl}) = \mathrm{Dir}\left(\lambda_{jlk} + N_{jlk} \;:\; k = 1, \ldots, |\mathbf{z}_{ij}|\right) . \tag{6.41}$$

For the VBE step, taking derivatives of (6.37) with respect to each $q_{\mathbf{s}_i}(\mathbf{s}_i)$ yields

$$\ln q_{\mathbf{s}_i}(\mathbf{s}_i) = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln p(\mathbf{z}_i \,|\, \boldsymbol{\theta}, m) + c_i' = \int d\boldsymbol{\theta}\; q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \boldsymbol{\theta}, m) + c_i' , \tag{6.42}$$

where each c_i' is a Lagrange multiplier for normalisation of the posterior. Since the complete-data likelihood $p(\mathbf{z}_i \,|\, \boldsymbol{\theta}, m)$ is in the exponential family and we have placed conjugate Dirichlet priors on the parameters, we can immediately utilise the results of corollary 2.2 (page 74) which gives simple forms for the VBE step:

$$q_{\mathbf{s}_i}(\mathbf{s}_i) \propto q_{\mathbf{z}_i}(\mathbf{z}_i) = \prod_{j=1}^{|\mathbf{z}_i|} p(\mathbf{z}_{ij} \,|\, \mathbf{z}_{i\mathbf{pa}(j)}, \tilde{\boldsymbol{\theta}}) . \tag{6.43}$$

Thus the approximate posterior over the hidden variables $\mathbf{s}_i$ resulting from a variational Bayesian approximation is identical to that resulting from exact inference in a model with known point parameters $\tilde{\boldsymbol{\theta}}$. Corollary 2.2 also tells us that $\tilde{\boldsymbol{\theta}}$ should be chosen to satisfy $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}$. The natural parameters for this model are the log probabilities $\{\ln \boldsymbol{\theta}_{jlk}\}$, where j specifies which variable, l indexes the possible configurations of its parents, and k the possible settings of the variable. Thus

$$\ln \tilde{\boldsymbol{\theta}}_{jlk} = \boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}_{jlk}) = \overline{\boldsymbol{\phi}}_{jlk} = \int d\boldsymbol{\theta}_{jl}\; q_{\boldsymbol{\theta}_{jl}}(\boldsymbol{\theta}_{jl}) \ln \boldsymbol{\theta}_{jlk} . \tag{6.44}$$

Under a Dirichlet distribution, the expectations are given by differences of digamma functions

$$\ln \tilde{\boldsymbol{\theta}}_{jlk} = \psi(\lambda_{jlk} + N_{jlk}) - \psi(\sum_{k=1}^{|\mathbf{z}_{ij}|} \lambda_{jlk} + N_{jlk}) \quad \forall \{j, l, k\} . \tag{6.45}$$

where the N_{jlk} are defined in (6.40), and the $\psi(\cdot)$ are digamma functions (see appendix C.1). Since this expectation operation takes the geometric mean of the probabilities, the propagation algorithm in the VBE step is now passed sub-normalised probabilities as parameters

$$\sum_{k=1}^{|\mathbf{z}_{ij}|} \tilde{\boldsymbol{\theta}}_{jlk} \leq 1 \qquad \forall \{j, l\} . \tag{6.46}$$

This use of sub-normalised probabilities also occurred in Chapter 3, which is unsurprising given that both models consist of local multinomial conditional probabilities. In that model, the inference algorithm was the forward-backward algorithm (or its VB analogue), and was restricted to the particular topology of a Hidden Markov Model. Our derivation uses belief propagation (section 1.1.2) for any singly-connected discrete DAG.

The expected natural parameters become normalised only if the distribution over parameters is a delta function, in which case this reduces to the MAP inference scenario of section 6.3.1. In fact, if we look at the limit of the digamma function for large arguments (see appendix C.1), we find

$$\lim_{x \to \infty} \psi(x) = \ln x \, , \tag{6.47}$$

and equation (6.45) becomes

$$\lim_{n \to \infty} \ln \tilde{\boldsymbol{\theta}}_{jlk} = \ln(\lambda_{jlk} + N_{jlk}) - \ln(\sum_{k=1}^{|\mathbf{z}_{ij}|} \lambda_{jlk} + N_{jlk}) \tag{6.48}$$

which has recovered the MAP estimator for $\boldsymbol{\theta}$ (6.23), up to the -1 entries in numerator and denominator which become vanishingly small for large data, and vanish completely if MAP is performed in the softmax basis. Thus in the limit of large data VB recovers the MAP parameter estimate.

To summarise, the VBEM implementation for discrete DAGs consists of iterating between the VBE step (6.43) which infers distributions over the hidden variables given a distribution over the parameters, and a VBM step (6.41) which finds a variational posterior distribution over parameters based on the hidden variables' sufficient statistics from the VBE step. Each step monotonically increases a lower bound on the marginal likelihood of the data, and the algorithm is guaranteed to converge to a local maximum of the lower bound.

The VBEM algorithm uses as a subroutine the algorithm used in the E step of the corresponding EM algorithm for MAP estimation, and so the VBE step's computational complexity is the same — there is some overhead in calculating differences of digamma functions instead of ratios of expected counts, but this is presumed to be minimal and fixed.

As with BIC and Cheeseman-Stutz, the lower bound does not take into account aliasing in the parameter posterior, and needs to be corrected as described in section 6.3.2.

6.3.5 Annealed Importance Sampling (AIS)

AIS (Neal, 2001) is a state-of-the-art technique for estimating marginal likelihoods, which breaks a difficult integral into a series of easier ones. It combines techniques from importance sampling, Markov chain Monte Carlo, and simulated annealing (Kirkpatrick et al., 1983). It builds on work in the Physics community for estimating the free energy of systems at different temperatures, for example: thermodynamic integration (Neal, 1993), *tempered transitions* (Neal, 1996), and the similarly inspired *umbrella sampling* (Torrie and Valleau, 1977). Most of these, as well as other related methods, are reviewed in Gelman and Meng (1998).

Obtaining samples from the posterior distribution over parameters, with a view to forming a Monte Carlo estimate of the marginal likelihood of the model, is usually a very challenging problem. This is because, even with small data sets and models with just a few parameters, the distribution is likely to be very peaky and have its mass concentrated in tiny volumes of space. This makes simple approaches such as sampling parameters directly from the prior or using simple importance sampling infeasible. The basic idea behind annealed importance sampling is to move in a *chain* from an easy-to-sample-from distribution, via a series of intermediate distributions, through to the complicated posterior distribution. By annealing the distributions in this way the parameter samples should hopefully come from representative areas of probability mass in the posterior. The key to the annealed importance sampling procedure is to make use of the importance weights gathered at all the distributions up to and including the final posterior distribution, in such a way that the final estimate of the marginal likelihood is unbiased. A brief description of the AIS procedure follows:

We define a series of inverse-temperatures $\{\tau(k)\}_{k=0}^{K}$ satisfying

$$0 = \tau(0) < \tau(1) < \cdots < \tau(K-1) < \tau(K) = 1 \,. \tag{6.49}$$

We refer to temperatures and inverse-temperatures interchangeably throughout this section. We define the function:

$$f_k(\boldsymbol{\theta}) \equiv p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)^{\tau(k)} \,, \qquad k \in \{0, \ldots, K\} \,. \tag{6.50}$$

Thus the set of functions $\{f_k(\boldsymbol{\theta})\}_{k=0}^{K}$ form a series of unnormalised distributions which *interpolate* between the prior and posterior, parameterised by τ. We also define the normalisation constants

$$\mathcal{Z}_k \equiv \int d\boldsymbol{\theta} \; f_k(\boldsymbol{\theta}) = \int d\boldsymbol{\theta} \; p(\boldsymbol{\theta} \,|\, m) p(\mathbf{y} \,|\, \boldsymbol{\theta}, m)^{\tau(k)} \,, \qquad k \in \{0, \ldots, K\} \,. \tag{6.51}$$

We note the following:

$$\mathcal{Z}_0 = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}\,|\,m) = 1 \tag{6.52}$$

from normalisation of the prior, and

$$\mathcal{Z}_K = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}\,|\,m) p(\mathbf{y}\,|\,\boldsymbol{\theta}, m) = p(\mathbf{y}\,|\,m)\;, \tag{6.53}$$

which is exactly the marginal likelihood that we wish to estimate. We can estimate $\mathcal{Z}_K$, or equivalently $\frac{\mathcal{Z}_K}{\mathcal{Z}_0}$, using the identity

$$p(\mathbf{y}\,|\,m) = \frac{\mathcal{Z}_K}{\mathcal{Z}_0} \equiv \frac{\mathcal{Z}_1}{\mathcal{Z}_0}\,\frac{\mathcal{Z}_2}{\mathcal{Z}_1}\,\cdots\,\frac{\mathcal{Z}_K}{\mathcal{Z}_{K-1}} = \prod_{k=1}^{K} \mathcal{R}_k\;, \tag{6.54}$$

Each of the K ratios in this expression can be individually estimated using importance sampling (see section 1.3.6). The kth ratio, denoted $\mathcal{R}_k$, can be estimated from a set of (not necessarily independent) samples of parameters $\{\boldsymbol{\theta}^{(k,c)}\}_{c\in\mathcal{C}_k}$ which are drawn from the higher temperature $\tau(k-1)$ distribution (the importance distribution), i.e. each $\boldsymbol{\theta}^{(k,c)} \sim f_{k-1}(\boldsymbol{\theta})$, and the importance weights are computed at the lower temperature $\tau(k)$. These samples are used to construct the Monte Carlo estimate for $\mathcal{R}_k$:

$$\mathcal{R}_k \equiv \frac{\mathcal{Z}_k}{\mathcal{Z}_{k-1}} = \int d\boldsymbol{\theta}\; \frac{f_k(\boldsymbol{\theta})}{f_{k-1}(\boldsymbol{\theta})}\frac{f_{k-1}(\boldsymbol{\theta})}{\mathcal{Z}_{k-1}} \tag{6.55}$$

$$\approx \frac{1}{C_k}\sum_{c\in\mathcal{C}_k} \frac{f_k(\boldsymbol{\theta}^{(k,c)})}{f_{k-1}(\boldsymbol{\theta}^{(k,c)})}\;, \qquad \text{with } \boldsymbol{\theta}^{(k,c)} \sim f_{k-1}(\boldsymbol{\theta}) \tag{6.56}$$

$$= \frac{1}{C_k}\sum_{c\in\mathcal{C}_k} p(\mathbf{y}\,|\,\boldsymbol{\theta}^{(k,c)}, m)^{\tau(k)-\tau(k-1)}\;. \tag{6.57}$$

Here, the importance weights are the summands in (6.56). The accuracy of each $\mathcal{R}_k$ depends on the constituent distributions $\{f_k(\boldsymbol{\theta}), f_{k-1}(\boldsymbol{\theta})\}$ being sufficiently close so as to produce low-variance weights. The estimate of $\mathcal{Z}_K$ in (6.54) is unbiased if the samples used to compute each ratio $\mathcal{R}_k$ are drawn from the equilibrium distribution at each temperature $\tau(k)$. In general we expect it to be difficult to sample directly from the forms $f_k(\boldsymbol{\theta})$ in (6.50), and so Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970) steps are used at each temperature to generate the set of C_k samples required for each importance calculation in (6.57).

Metropolis-Hastings for discrete-variable models

In the discrete-variable graphical models covered in this chapter, the parameters are multinomial probabilities, hence the support of the Metropolis proposal distributions is restricted to the

simplex of probabilities summing to 1. At first thought one might suggest using a Gaussian proposal distribution in the softmax basis of the current parameters $\boldsymbol{\theta}$:

$$\theta_i \equiv \frac{e^{b_i}}{\sum_j^{|\boldsymbol{\theta}|} e^{b_j}} \,. \tag{6.58}$$

Unfortunately an invariance exists: with β a scalar, the transformation $b_i' \leftarrow b_i + \beta \; \forall i$ leaves the parameter $\boldsymbol{\theta}$ unchanged. Therefore the determinant of the Jacobian of the transformation (6.58) from the vector $\mathbf{b}$ to the vector $\boldsymbol{\theta}$ is zero, and it is hard to construct a reversible Markov chain.

A different and intuitively appealing idea is to use a Dirichlet distribution as the proposal distribution, with its mean positioned at the current parameter. The precision of the Dirichlet proposal distribution at inverse-temperature $\tau(k)$ is governed by its *strength*, $\alpha(k)$, which is a free variable to be set as we wish, provided it is not in any way a function of the sampled parameters. A Metropolis-Hastings acceptance function is required to maintain detailed balance: if $\boldsymbol{\theta}'$ is the sample under the proposal distribution centered around the current parameter $\boldsymbol{\theta}^{(k,c)}$, then the acceptance function is:

$$a(\boldsymbol{\theta}', \boldsymbol{\theta}^{(k,c)}) = \min \left(\frac{f_k(\boldsymbol{\theta}')}{f_k(\boldsymbol{\theta}^{(k,c)})} \frac{\mathrm{Dir}(\boldsymbol{\theta}^{(k,c)} \,|\, \boldsymbol{\theta}', \alpha(k))}{\mathrm{Dir}(\boldsymbol{\theta}' \,|\, \boldsymbol{\theta}^{(k,c)}, \alpha(k))} \,,\; 1 \right) \,, \tag{6.59}$$

where $\mathrm{Dir}(\boldsymbol{\theta} \,|\, \overline{\boldsymbol{\theta}}, \alpha)$ is the probability density of a Dirichlet distribution with mean $\overline{\boldsymbol{\theta}}$ and strength α, evaluated at $\boldsymbol{\theta}$. The next sample is instantiated as follows:

$$\boldsymbol{\theta}^{(k,c+1)} = \begin{cases} \boldsymbol{\theta}' & \text{if } w < a(\boldsymbol{\theta}', \boldsymbol{\theta}^{(k,c)}) \qquad \text{(accept)} \\ \boldsymbol{\theta}^{(k,c)} & \text{otherwise} \qquad \text{(reject)} \,, \end{cases} \tag{6.60}$$

where $w \sim \mathrm{U}(0,1)$ is a random variable sampled from a uniform distribution on $[0,1]$. By repeating this procedure of accepting or rejecting $C_k' \geq C_k$ times at the temperature $\tau(k)$, the MCMC sampler generates a set of (dependent) samples $\{\boldsymbol{\theta}^{(k,c)}\}_{c=1}^{C_k'}$. A subset of these $\{\boldsymbol{\theta}^{(k,c)}\}_{c \in \mathcal{C}_k}$, with $|\mathcal{C}_k| = C_k \leq C_k'$, is then used as the importance samples in the computation above (6.57). This subset will generally not include the first few samples, as these samples are likely not yet samples from the equilibrium distribution at that temperature.

An algorithm to compute all ratios

The entire algorithm for computing all K marginal likelihood ratios is given in algorithm 6.1. It has several parameters, in particular: the number of annealing steps, K; their inverse-temperatures (the annealing schedule), $\{\tau(k)\}_{k=1}^K$; the parameters of the MCMC importance sampler at each temperature $\{C_k', C_k, \alpha(k)\}_{k=1}^K$, which are the number of proposed samples,

Algorithm 6.1: **AIS**. To compute all ratios $\{\mathcal{R}_k\}_{k=1}^K$ for the marginal likelihood estimate.

1. Initialise $\boldsymbol{\theta}_{\text{ini}} \sim f_0(\boldsymbol{\theta})$ i.e. from the prior $p(\boldsymbol{\theta} \,|\, m)$
2. For $k = 1$ to K annealing steps
   (a) Run MCMC at temperature $\tau(k-1)$ as follows:
      i. Initialise $\boldsymbol{\theta}^{(k,0)} \leftarrow \boldsymbol{\theta}_{\text{ini}}$ from previous temp.
      ii. Generate the set $\{\boldsymbol{\theta}^{(k,c)}\}_{c=1}^{C'_k} \sim f_{k-1}(\boldsymbol{\theta})$ as follows:
         A. For $c = 1$ to C'_k
            Propose $\boldsymbol{\theta}' \sim \text{Dir}(\boldsymbol{\theta}' \,|\, \boldsymbol{\theta}^{(k,c-1)}, \alpha(k))$
            Accept $\boldsymbol{\theta}^{(k,c)} \leftarrow \boldsymbol{\theta}'$ according to (6.59) and (6.60)
            End For
         B. Store $\boldsymbol{\theta}_{\text{ini}} \leftarrow \boldsymbol{\theta}^{(k,C'_k)}$
      iii. Store a subset of these $\{\boldsymbol{\theta}^{(k,c)}\}_{c \in \mathcal{C}_k}$ with $|\mathcal{C}_k| = C_k \leq C'_k$
   (b) Calculate $\mathcal{R}_k \equiv \frac{\mathcal{Z}_k}{\mathcal{Z}_{k-1}} \approxeq \frac{1}{C_k} \sum_{c=1}^{C_k} \frac{f_k(\boldsymbol{\theta}^{(k,c)})}{f_{k-1}(\boldsymbol{\theta}^{(k,c)})}$

   End For
3. Output $\{\ln \mathcal{R}_k\}_{k=1}^K$ and $\ln \hat{\mathcal{Z}}_K = \sum_{k=1}^K \ln \mathcal{R}_k$ as the approximation to $\ln \mathcal{Z}_K$

the number used for the importance estimate, and the precision of the proposal distribution, respectively.

Nota bene: In the presentation of AIS thus far, we have shown how to compute estimates of $\mathcal{R}_k$ using a set, $\mathcal{C}_k$, of importance samples (see equation (6.56)), chosen from the larger set, $\mathcal{C}'_k$, drawn using a Metropolis-Hastings sampling scheme. In the original paper by Neal (2001), the size of the set $\mathcal{C}_k$ is *exactly one*, and it is only for this case that the validity of AIS as an unbiased estimate has been proved. Because the experiments carried out in this chapter do in fact only use $C_k = |\mathcal{C}_k| = 1$ (as described in section 6.4.1), we stay in the realm of the proven result. It is open research question to show that algorithm 6.1 is unbiased for $C_k = |\mathcal{C}_k| > 1$ (personal communication with R. Neal).

Algorithm 6.1 produces only a single estimate of the marginal likelihood; the variance of this estimate can be obtained from the results of several annealed importance samplers run in parallel. Indeed a particular attraction of AIS is that one can take averages of the marginal likelihood estimates from a set of G annealed importance sampling runs to form a better (unbiased) estimate:

$$\left[\frac{\mathcal{Z}_K}{\mathcal{Z}_0}\right]^{(G)} = \frac{1}{G} \sum_{g=1}^{G} \prod_{k=1}^{K^{(g)}} \mathcal{R}_k^{(g)} \,. \tag{6.61}$$

However this computational resource might be better spent simulating a single chain with a more finely-grained annealing schedule, since for each k we require each pair of distributions $\{f_k(\boldsymbol{\theta}), f_{k-1}(\boldsymbol{\theta})\}$ to be sufficiently close that the importance weights have low variance. Or perhaps the computation is better invested by having a coarser annealing schedule and taking more samples at each temperature to ensure the Metropolis-Hastings sampler has reached equilibrium. In Neal (2001) an in-depth analysis is presented for these and other similar concerns for estimating the marginal likelihoods in some very simple models, using functions of the variance of the importance weights (i.e. the summands in (6.56)) as guides to the reliability of the estimates.

In section 6.5.1 we discuss the performance of AIS for estimating the marginal likelihood of the graphical models used in this chapter, addressing the specific choices of proposal widths, number of samples, and annealing schedules used in the experiments.

6.3.6 Upper bounds on the marginal likelihood

This section is included to justify comparing the marginal likelihood to scores such as MAP and ML. The following estimates based on the ML parameters and the posterior distribution over parameters represent strict bounds on the true marginal likelihood of a model, $p(\mathbf{y})$,

$$p(\mathbf{y}) = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \;. \tag{6.62}$$

(where we have omitted the dependence on m for clarity).

We begin with the ML estimate:

$$p(\mathbf{y})_{\mathrm{ML}} = \int d\boldsymbol{\theta}\; \boldsymbol{\delta}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{ML}}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \tag{6.63}$$

which is the expectation of the data likelihood under a delta function about the ML parameter setting. This is a strict upper bound only if $\boldsymbol{\theta}_{\mathrm{ML}}$ has found the global maximum of the likelihood. This may not happen due to local maxima in the optimisation process, for example if the model contains hidden variables and an EM-type optimisation is being employed.

The second estimate is that arising from the MAP estimate,

$$p(\mathbf{y})_{\mathrm{MAP}} = \int d\boldsymbol{\theta}\; \boldsymbol{\delta}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{MAP}}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \tag{6.64}$$

which is the expectation of the data likelihood under a delta function about the MAP parameter setting. However is not a strict upper or lower bound on the marginal likelihood, since this depends on how the prior term acts to position the MAP estimate.

The last estimate, based on the posterior distribution over parameters, is for academic interest only, since we would expect its calculation to be intractable:

$$p(\mathbf{y})_{post.} = \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}\,|\,\mathbf{y}) p(\mathbf{y}\,|\,\boldsymbol{\theta})\;. \tag{6.65}$$

This is the expected likelihood under the posterior distribution over parameters. To prove that (6.65) is an upper bound on the marginal likelihood, we use a simple convexity bound as follows:

$$\begin{aligned}
p(\mathbf{y})_{post.} &= \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}\,|\,\mathbf{y}) p(\mathbf{y}\,|\,\boldsymbol{\theta}) && (6.66)\\
&= \int d\boldsymbol{\theta}\; \frac{p(\boldsymbol{\theta}) p(\mathbf{y}\,|\,\boldsymbol{\theta})}{p(\mathbf{y})} p(\mathbf{y}\,|\,\boldsymbol{\theta}) \qquad \text{by Bayes' rule} && (6.67)\\
&= \frac{1}{p(\mathbf{y})} \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) \left[p(\mathbf{y}\,|\,\boldsymbol{\theta})\right]^2 && (6.68)\\
&\geq \frac{1}{p(\mathbf{y})} \left[\int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}) p(\mathbf{y}\,|\,\boldsymbol{\theta})\right]^2 \qquad \text{by convexity of } x^2 && (6.69)\\
&= \frac{1}{p(\mathbf{y})} \left[p(\mathbf{y})\right]^2 = p(\mathbf{y})\;. && (6.70)
\end{aligned}$$

As we would expect the integral (6.65) to be intractable, we could instead estimate it by taking samples from the posterior distribution over parameters and forming the Monte Carlo estimate:

$$\begin{aligned}
p(\mathbf{y}) \leq p(\mathbf{y})_{post.} &= \int d\boldsymbol{\theta}\; p(\boldsymbol{\theta}\,|\,\mathbf{y}) p(\mathbf{y}\,|\,\boldsymbol{\theta}) && (6.71)\\
&\approx \frac{1}{C} \sum_{c=1}^{C} p(\mathbf{y}\,|\,\boldsymbol{\theta}^{(c)}) && (6.72)
\end{aligned}$$

where $\boldsymbol{\theta}^{(c)} \sim p(\boldsymbol{\theta}\,|\,\mathbf{y})$, the exact posterior. Had we taken samples from the prior $p(\boldsymbol{\theta})$, this would have yielded the true marginal likelihood, so it makes sense that by concentrating samples in areas which give rise to high likelihoods we are over-estimating the marginal likelihood; for this reason we would only expect this upper bound to be close for small amounts of data. An interesting direction of thought would be to investigate the mathematical implications of drawing samples from an approximate posterior instead of the exact posterior, such as that obtained in a variational optimisation, which itself is arrived at from a lower bound on the marginal likelihood; this could well give an even higher upper bound since the approximate variational posterior is likely to neglect regions of low posterior density.

6.4 Experiments

In this section we experimentally examine the performance of the variational Bayesian procedure in approximating the marginal likelihood for all the models in a particular class. We first describe the class defining our space of hypothesised structures, then chose a particular mem-

ber of the class as the "true" structure, generate a set of parameters for that structure, and then generate varying-sized data sets from that structure with those parameters. The task is then to estimate the marginal likelihood of every data set under each member of the class, including the true structure, using each of the scores described in the previous section. The hope is that the VB lower bound will be able to find the true model, based on its scoring, as reliably as the gold standard AIS does. We would ideally like the VB method to perform well even with little available data.

Later experiments take the true structure and analyse the performance of the scoring methods under many different settings of the parameters drawn from the parameter prior for the true structure. Unfortunately this analysis does not include AIS, as sampling runs for each and every combination of the structures, data sets, and parameter settings would take a prohibitively large amount of compute time.

A specific class of graphical model. We look at the specific class of discrete directed *bipartite* graphical models, i.e. those graphs in which only hidden variables can be parents of observed variables, and the hidden variables themselves have no parents. We further restrict ourselves to those graphs which have just $k = |\mathcal{H}| = 2$ hidden variables, and $p = |\mathcal{V}| = 4$ observed variables; both hidden variables are binary i.e. $|\mathrm{s}_{ij}| = 2$ for $j \in \mathcal{H}$, and each observed variable has cardinality $|\mathrm{y}_{ij}| = 5$ for $j \in \mathcal{V}$.

The number of distinct graphs. In the class of bipartite graphs described above, with k distinct hidden variables and p observed variables, there are 2^{kp} possible structures, corresponding to the presence or absence of a directed link between each hidden and each conditionally independent observed variable. If the hidden variables are unidentifiable, which is the case in our example model where they have the same cardinality, then the number of possible graphs is reduced. It is straightforward to show in this example that the number of graphs is reduced from $2^{2\times 4} = 256$ down to 136.

The specific model and generating data. We chose the particular structure shown in figure 6.1, which we call the "true" structure. We chose this structure because it contains enough links to induce non-trivial correlations amongst the observed variables, whilst the class as a whole has few enough nodes to allow us to examine exhaustively every possible structure of the class. There are only three other structures in the class which have more parameters than our chosen structure; these are: two structures in which either the left- or right-most visible node has both hidden variables as parents instead of just one, and one structure which is fully connected. As a caveat, one should note that our chosen true structure is at the higher end of complexity in this class, and so we might find that scoring methods that do not penalise complexity do seemingly better than naively expected.

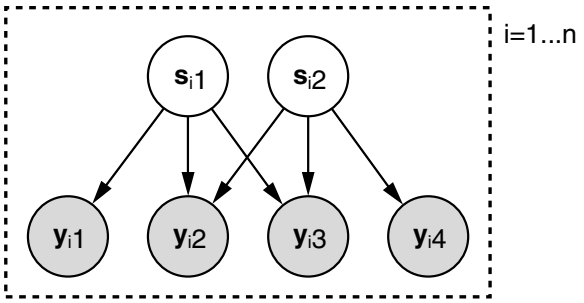


Figure 6.1: The true structure that was used to generate all the data sets used in the experiments. The hidden variables (top) are each binary, and the observed variables (bottom) are each five-valued. This structure has 50 parameters, and is two links away from the fully-connected structure. In total there are 136 possible distinct structures with two (identical) hidden variables and four observed variables.

Evaluation of the marginal likelihood of all possible alternative structures in the class is done for academic interest only; in practice one would embed different structure scoring methods in a greedy model search outer loop (Friedman, 1998) to find probable structures. Here, we are not so much concerned with structure *search* per se, since a prerequisite for a good structure search algorithm is an efficient and accurate method for evaluating any particular structure. Our aim in these experiments is to establish the reliability of the variational bound as a score, compared to annealed importance sampling, and the currently employed asymptotic scores such as BIC and Cheeseman-Stutz.

The parameters of the true model

Conjugate uniform symmetric Dirichlet priors were placed over all the parameters of the model, that is to say in equation (6.4), $\lambda_{jlk} = 1 \; \forall \{jlk\}$. This particular prior was arbitrarily chosen for the purposes of the experiments, and we do not expect it to influence our conclusions much. For the network shown in figure 6.1 parameters were sampled from the prior, once and for all, to instantiate a true underlying model, from which data was then generated. The sampled parameters are shown below (their sizes are functions of each node's and its parents' cardinalities):

$$\boldsymbol{\theta}_1 = \begin{bmatrix} .12 & .88 \end{bmatrix} \quad \boldsymbol{\theta}_3 = \begin{bmatrix} .03 & .03. & .64 & .02 & .27 \\ .18 & .15 & .22 & .19 & .27 \end{bmatrix} \quad \boldsymbol{\theta}_6 = \begin{bmatrix} .10 & .08 & .43 & .03 & .36 \\ .30 & .14 & .07 & .04 & .45 \end{bmatrix}$$

$$\boldsymbol{\theta}_2 = \begin{bmatrix} .08 & .92 \end{bmatrix} \quad \boldsymbol{\theta}_4 = \begin{bmatrix} .10 & .54 & .07 & .14 & .15 \\ .04 & .15 & .59 & .05 & .16 \\ .20 & .08 & .36 & .17 & .18 \\ .19 & .45 & .10 & .09 & .17 \end{bmatrix} \quad \boldsymbol{\theta}_5 = \begin{bmatrix} .11 & .47 & .12 & .30 & .01 \\ .27 & .07 & .16 & .25 & .25 \\ .52 & .14 & .15 & .02 & .17 \\ .04 & .00 & .37 & .33 & .25 \end{bmatrix}$$

where $\{\boldsymbol{\theta}_j\}_{j=1}^{2}$ are the parameters for the hidden variables, and $\{\boldsymbol{\theta}_j\}_{j=3}^{6}$ are the parameters for the remaining four observed variables. Recall that each row of each matrix denotes the

probability of each multinomial setting for a particular configuration of the parents. Each row of each matrix sums to one (up to rounding error). Note that there are only two rows for $\boldsymbol{\theta}_3$ and $\boldsymbol{\theta}_6$ as both these observed variables have just a single binary parent. For variables 4 and 5, the four rows correspond to the parent configurations (in order): $\{[1\ 1], [1\ 2], [2\ 1], [2\ 2]\}$.

Also note that for this particular instantiation of the parameters, both the hidden variable priors are close to deterministic, causing approximately 80% of the data to originate from the $[2\ 2]$ setting of the hidden variables. This means that we may need many data points before the evidence for two hidden variables outweighs that for one.

Incrementally larger and larger data sets were generated with these parameter settings, with

$$n \in \{10,20,40,80,110,160,230,320,400,430,\\ 480,560,640,800,960,1120,1280,2560,5120,10240\} .$$

The items in the $n = 10$ data set are a subset of the $n = 20$ and subsequent data sets, etc. The particular values of n were chosen from an initially exponentially increasing data set size, followed by inclusion of some intermediate data sizes to concentrate on interesting regions of behaviour.

6.4.1 Comparison of scores to AIS

All 136 possible distinct structures were scored for each of the 20 data set sizes given above, using MAP, BIC, CS, VB and AIS scores. Strictly speaking, MAP is not an approximation to the marginal likelihood, but it is an upper bound (see section 6.3.6) and so is nevertheless interesting for comparison.

We ran EM on each structure to compute the MAP estimate of the parameters, and from it computed the BIC score as described in section 6.3.2. We also computed the BIC score including the parameter prior, denoted BICp, which was obtained by including a term $\ln p(\hat{\boldsymbol{\theta}} \,|\, m)$ in equation (6.26). From the same EM optimisation we computed the CS score according to section 6.3.3. We then ran the variational Bayesian EM algorithm with the same initial conditions to give a lower bound on the marginal likelihood. For both these optimisations, random parameter initialisations were used in an attempt to avoid local maxima — the highest score over three random initialisations was taken for each algorithm; empirically this heuristic appeared to avoid local maxima problems. The EM and VBEM algorithms were terminated after either 1000 iterations had been reached, or the change in log likelihood (or lower bound on the log marginal likelihood, in the case of VBEM) became less than 10^{-6} per datum.

For comparison, the AIS sampler was used to estimate the marginal likelihood (see section 6.3.5), annealing from the prior to the posterior in $K = 16384$ steps. A nonlinear anneal-

ing schedule was employed, tuned to reduce the variance in the estimate, and the Metropolis proposal width was tuned to give reasonable acceptance rates. We chose to have just a single sampling step at each temperature (i.e. $C'_k = C_k = 1$), for which AIS has been proven to give unbiased estimates, and initialised the sampler at each temperature with the parameter sample from the previous temperature. These particular choices are explained and discussed in detail in section 6.5.1. Initial marginal likelihood estimates from single runs of AIS were quite variable, and for this reason several more batches of AIS runs were undertaken, each using a different random initialisation (and random numbers thereafter); the total of G batches of scores were averaged according to the procedure given in section 6.3.5, equation (6.61), to give the AIS$^{(G)}$ score. In total, $G = 5$ batches of AIS runs were carried out.

Scoring all possible structures

Figure 6.2 shows the MAP, BIC, BICp, CS, VB and AIS$^{(5)}$ scores obtained for each of the 136 possible structures against the number of parameters in the structure. Score is measured on the vertical axis, with each scoring method (columns) sharing the same vertical axis range for a particular data set size (rows).

The horizontal axis of each plot corresponds to the number of parameters in the structure (as described in section 6.3.2). For example, at the extremes there is one structure with 66 parameters which is the fully connected structure, and one structure with 18 parameters which is the fully unconnected structure. The structure that generated the data has exactly 50 parameters. In each plot we can see that several structures can occupy the same column, having the same number of parameters. This means that, at least visually, it is not always possible to unambiguously assign each point in the column to a particular structure.

The scores shown here are those corrected for aliases — the difference between the uncorrected and corrected versions is only just perceptible as a slight downward movement of the low parameter structures (those with just one or zero hidden variables), as these have a smaller number of aliases S (see equation (6.26)).

In each plot, the true structure is highlighted by a '$\circ$' symbol, and the structure currently ranked highest by that scoring method is marked with a '$\times$'. We can see the general upward trend for the MAP score which prefers more complicated structures, and the pronounced downward trend for the BIC and BICp scores which (over-)penalise structure complexity. In addition one can see that neither upward or downward trends are apparent for VB or AIS scores. Moreover, the CS score does tend to show a downward trend similar to BIC and BICp, and while this trend weakens with increasing data, it is still present at $n = 10240$ (bottom row). Although not verifiable from these plots, we should note that for the vast majority of the scored structures

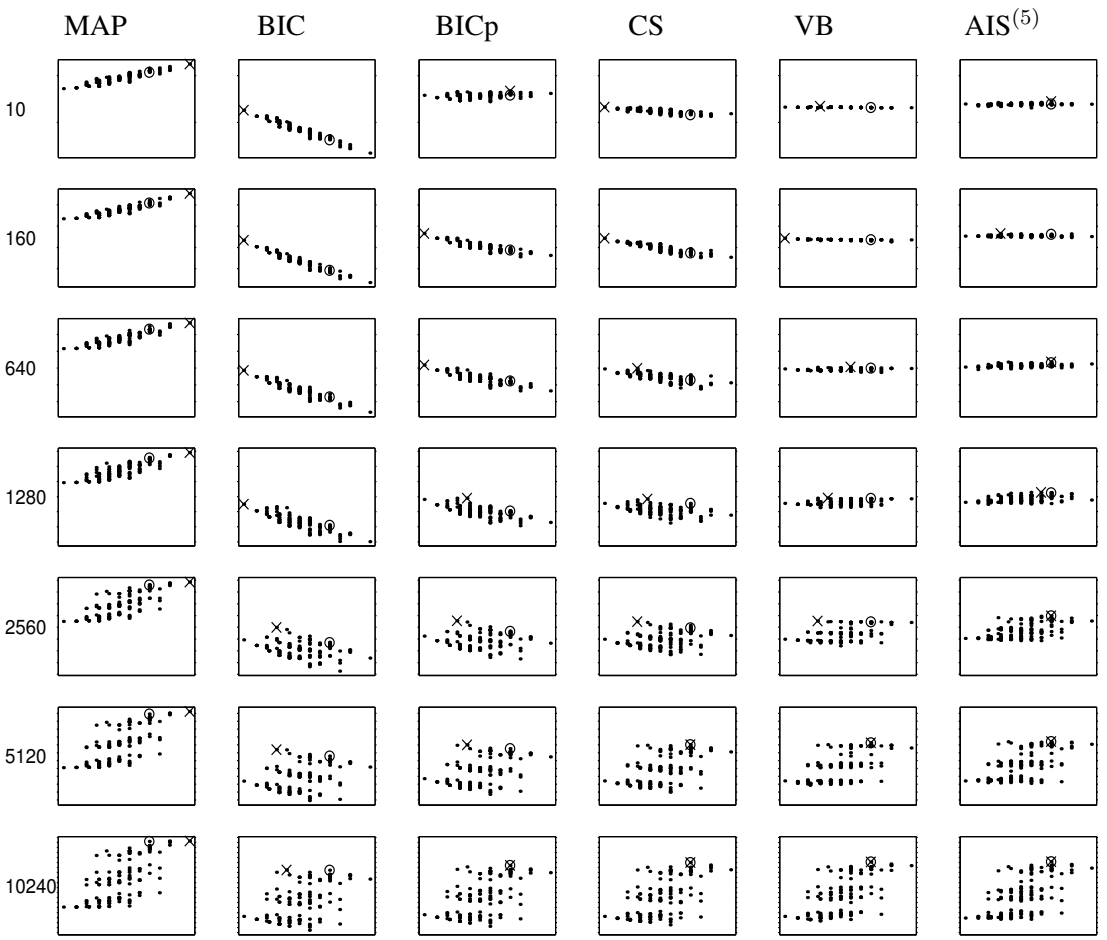


Figure 6.2: Scores for all 136 of the structures in the model class, by each of six scoring methods. Each plot has the score (approximation to the log marginal likelihood) on the vertical axis, with tick marks every 40 nats, and the number of parameters on the horizontal axis (ranging from 18 to 66). The middle four scores have been corrected for aliases (see section 6.3.2). Each row corresponds to a data set of a different size, n: from top to bottom we have $n = 10, 160, 640, 1280, 2560, 5120, 10240$. The true structure is denoted with a '$\circ$' symbol, and the highest scoring structure in each plot marked by the '$\times$' symbol. Every plot in the same row has the same scaling for the vertical score axis, set to encapsulate every structure for all scores. For a description of how these scores were obtained see section 6.4.1.

and data set sizes, the AIS$^{(5)}$ score is higher than the VB lower bound, as we would expect (see section 6.5.1 for exceptions to this observation).

The horizontal bands observed in the plots is an interesting artifact of the particular model used to generate the data. For example, we find on closer inspection some strictly followed trends: all those model structures residing in the upper band have the first three observable variables ($j = 3, 4, 5$) governed by at least one of the hidden variables; and all those structures in the middle band have the third observable ($j = 4$) connected to at least one hidden variable.

In this particular example, AIS finds the correct structure at $n = 960$ data points, but unfortunately does not retain this result reliably until $n = 2560$. At $n = 10240$ data points, BICp, CS, VB and AIS all report the true structure as being the one with the highest score amongst the other contending structures. Interestingly, BIC still does not select the correct structure, and MAP has given a structure with sub-maximal parameters the highest score. The latter observation may well be due to local maxima in the EM optimisation, since for previous slightly smaller data sets MAP chooses the fully-connected structure as expected. Note that as we did not have intermediate data sets it may well be that, for example, AIS reliably found the structure after 1281 data points, but we cannot know this without performing more experiments.

Ranking of the true structure

A somewhat more telling comparison of the scoring methods is given by how they rank the true structure amongst the alternative structures. Thus a ranking of 1 means that the scoring method has given the highest marginal likelihood to the true structure.

Note that a performance measure based on ranking makes several assumptions about our choice of loss function. This performance measure disregards information in the posterior about the structures with lower scores, reports only the number of structures that have higher scores, and not the amount by which the true structure is beaten. Ideally, we would compare a quantity that measured the divergence of all structures' posterior probabilities from the true posterior.

Moreover, we should keep in mind that at least for small data set sizes, there is no reason to assume that the actual posterior over structures has the true structure at its mode. Therefore it is slightly misleading to ask for high rankings at small data set sizes.

Table 6.1 shows the ranking of the true structure, as it sits amongst all the possible structures, as measured by each of the scoring methods MAP, BIC, BICp, CS, VB and AIS$^{(5)}$; this is also plotted in figure 6.3 where the MAP ranking is not included for clarity. Higher positions in the plot correspond to better rankings.

n	MAP	BIC*	BICp*	CS*	VB*	BIC	BICp	CS	VB	AIS$^{(5)}$
10	21	127	55	129	122	127	50	129	115	59
20	12	118	64	111	124	118	64	111	124	135
40	28	127	124	107	113	127	124	107	113	15
80	8	114	99	78	116	114	99	78	116	44
110	8	109	103	98	114	109	103	98	113	2
160	13	119	111	114	83	119	111	114	81	6
230	8	105	93	88	54	105	93	88	54	54
320	8	111	101	90	44	111	101	90	33	78
400	6	101	72	77	15	101	72	77	15	8
430	7	104	78	68	15	104	78	68	14	18
480	7	102	92	80	55	102	92	80	44	2
560	9	108	98	96	34	108	98	96	31	11
640	7	104	97	105	19	104	97	105	17	7
800	9	107	102	108	35	107	102	108	26	23
960	13	112	107	76	16	112	107	76	13	1
1120	8	105	96	103	12	105	96	103	12	4
1280	7	90	59	8	3	90	59	6	3	5
2560	6	25	17	11	11	25	15	11	11	1
5120	5	6	5	1	1	6	5	1	1	1
10240	3	2	1	1	1	2	1	1	1	1

Table 6.1: Ranking of the true structure by each of the scoring methods, as the size of the data set is increased. Asterisks (*) denote scores uncorrected for parameter aliasing in the posterior. Strictly speaking, the MAP score is not an estimate of the marginal likelihood. Note that these results are from data generated from only one instance of parameters under the true structure's prior over parameters.

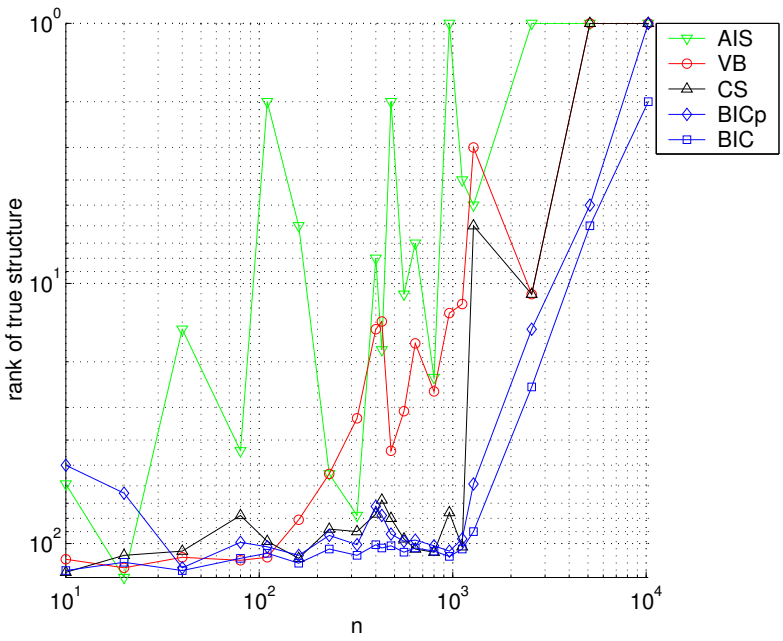


Figure 6.3: Ranking given to the true structure by each scoring method for varying data set sizes (higher in plot is better), by BIC, BICp, CS, VB and AIS$^{(5)}$ methods.

For small n, the AIS score produces a better ranking for the true structure than any of the other scoring methods, which suggests that the AIS sampler is managing to perform the Bayesian parameter averaging process more accurately than other approximations. For almost all n, VB outperforms BIC, BICp and CS, consistently giving a higher ranking to the true structure. Of particular note is the stability of the VB score ranking with respect to increasing amounts of data as compared to AIS (and to some extent CS).

Columns in table 6.1 with asterisks (*) correspond to scores that are not corrected for aliases, and are omitted from the figure. These corrections assume that the posterior aliases are well separated, and are valid only for large amounts of data and/or strongly-determined parameters. In this experiment, structures with two hidden states acting as parents are given a greater correction than those structures with only a single hidden variable, which in turn receive corrections greater than the one structure having no hidden variables. Of interest is that the correction nowhere degrades the rankings of any score, and in fact improves them very slightly for CS, and especially so for the VB score.

Score discrepancies between the true and top-ranked structures

Figure 6.4 plots the differences in score between the true structure and the score of the structure ranked top by BIC, BICp, CS, VB and AIS methods. The convention used means that all the differences are exactly zero or negative, measured from the score of the top-ranked structure — if the true structure is ranked top then the difference is zero, otherwise the true structure's score must be less than the top-ranked one. The true structure has a score that is close to the top-ranked structure in the AIS method; the VB method produces approximately similar-sized differences, and these are much less on the average than the CS, BICp, and BIC scores. For a better comparison of the non-sampling-based scores, see section 6.4.2, and figure 6.6.

Computation Time

Scoring all 136 structures at 480 data points on a 1GHz Pentium III processor took: 200 seconds for the MAP EM algorithms required for BIC/BICp/CS, 575 seconds for the VBEM algorithm required for VB, and 55000 seconds (15 hours) for a single run of the AIS algorithm (using 16384 samples as in the main experiments). All implementations were in MATLAB. Given the massive computational burden of the sampling method (approx 75 hours), which still produces fairly variable scores when averaging over five runs, it does seem as though CS and VB are proving very useful indeed. Can we justify the mild overall computational increase for VB? This increase results from both computing differences between digamma functions as opposed to ratios, and also from an empirically-observed slower convergence rate of the VBEM algorithm as compared to the EM algorithm.

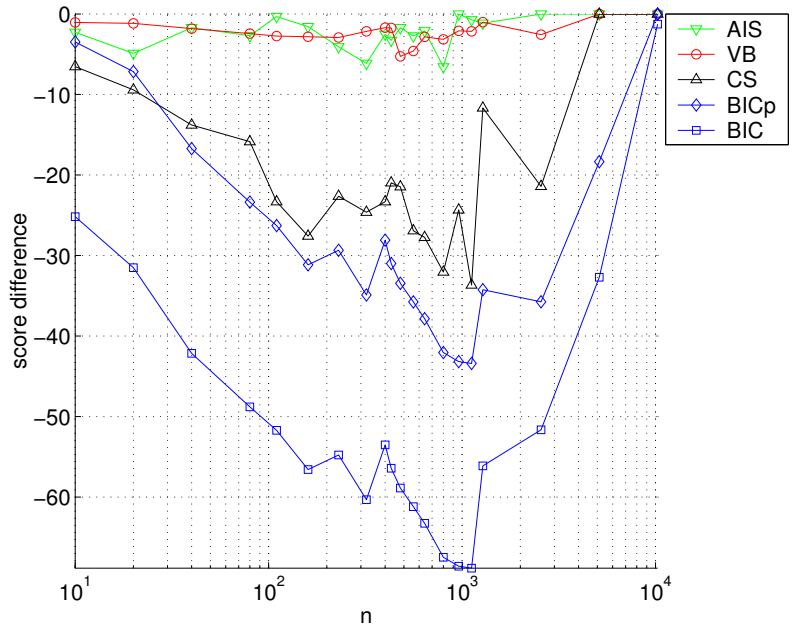


Figure 6.4: Differences in log marginal likelihood estimates (scores) between the top-ranked structure and the true structure, as reported by BIC, BICp, CS, VB and AIS$^{(5)}$ methods. All differences are exactly zero or negative: if the true structure is ranked top then the difference is zero, otherwise the score of the true structure must be less than the top-ranked structure. Note that these score differences are not per-datum scores, and therefore are not normalised for the data n.

6.4.2 Performance averaged over the parameter prior

The experiments in the previous section used a single instance of sampled parameters for the true structure, and generated data from this particular model. The reason for this was that, even for a single experiment, computing an exhaustive set of AIS scores covering all data set sizes and possible model structures takes in excess of 15 CPU days.

In this section we compare the performance of the scores over many different sampled parameters of the true structure (shown in figure 6.1). 106 parameters were sampled from the prior (as done once for the single model in the previous section), and incremental data sets generated for each of these instances as the true model. MAP EM and VBEM algorithms were employed to calculate the scores as described in section 6.4.1. For each instance of the true model, calculating scores for all data set sizes used and all possible structures, using three random restarts, for BIC/BICp/CS and VB took approximately 2.4 and 4.2 hours respectively on an Athlon 1800 Processor machine, which corresponds to about 1.1 and 1.9 seconds for each individual score.

The results are plotted in figure 6.5, which shows the median ranking given to the true structure by each scoring method, computed over 106 randomly sampled parameter settings. This plot corresponds to a smoothed version of figure 6.3, but unfortunately cannot contain AIS averages

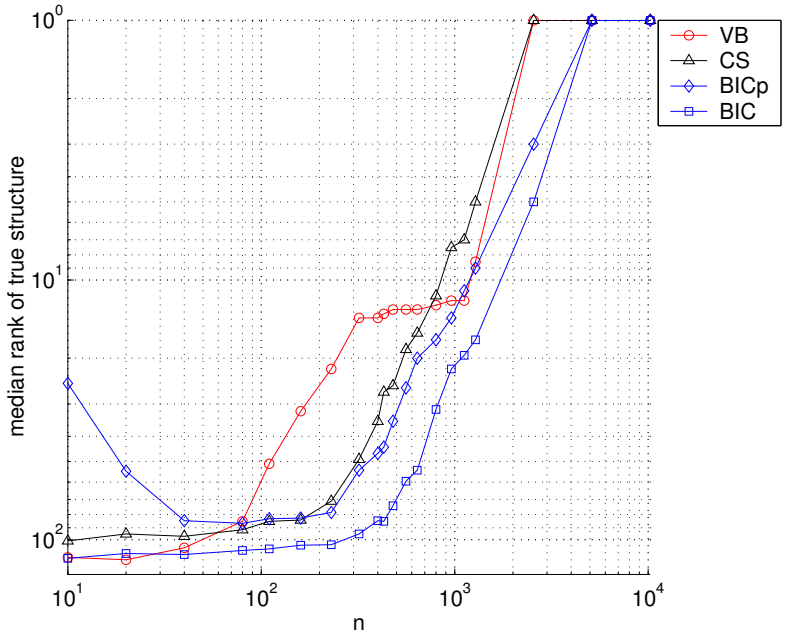


Figure 6.5: Median ranking of the true structure as reported by BIC, BICp, CS and VB methods, against the size of the data set n, taken over 106 instances of the true structure.

% times that \ than	BIC*	BICp*	CS*	CS*†	BIC	BICp	CS	CS†
VB ranks worse	16.9	30.2	31.8	32.8	15.1	29.6	30.9	31.9
same	11.1	15.0	20.2	22.1	11.7	15.5	20.9	22.2
better	72.0	54.8	48.0	45.1	73.2	55.0	48.2	45.9

Table 6.2: Comparison of the VB score to its competitors, using the ranking of the true structure as a measure of performance. The table gives the percentage fraction of times that the true structure was ranked lower, the same, and higher by VB than by the other methods (rounded to nearest .1%). The ranks were collected from all 106 generated parameters and all 20 data set sizes. Note that VB outperforms all competing scores, whether we base our comparison on the alias-corrected or uncorrected (*) versions of the scores. The CS score annotated with † is an improvement on the original CS score, and is explained in section 6.5.2.

for the computational reasons mentioned above. The results clearly show that for the most part VB outperforms all other scores on this task by this measure although there is a region in which VB seems to underperform CS, as measured by the median score.

Table 6.2 shows in more detail the performance of VB and its alias-uncorrected counterpart VB* in terms of the number of times the score correctly selects the true model (i.e. ranks it top). The data was collated from all 106 sampled true model structures, and all 20 data set sizes, giving a total of 288320 structures that needed to be scored by each approximate method. We see that VB outperforms the other scores convincingly, whether we compare the uncorrected (left hand side of table) or corrected (right hand side) scores. The results are more persuasive for the alias-corrected scores, suggesting that VB is benefitting more from this modification — it is not obvious why this should be so.

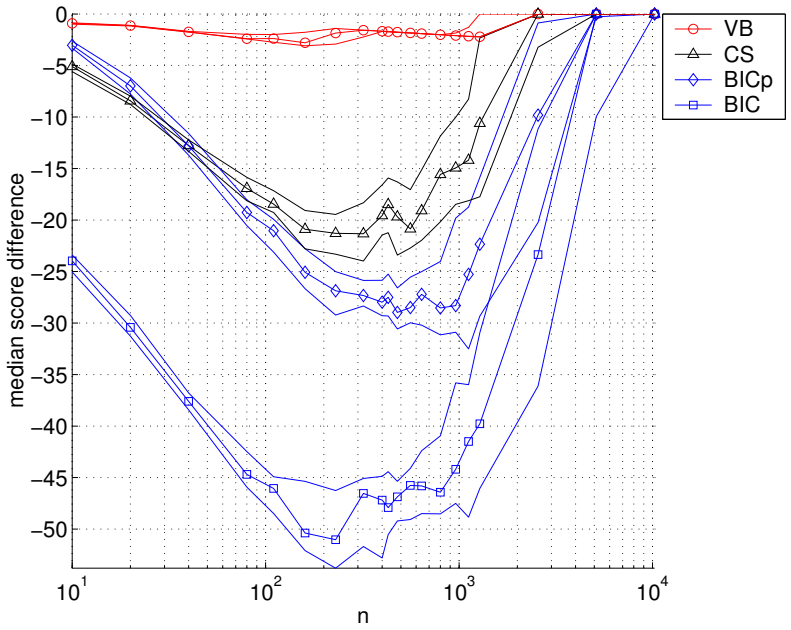


Figure 6.6: Median difference in score between the true and top-ranked structures, under BIC, BICp, CS and VB scoring methods, against the size of the data set n, taken over 106 instances of the true structure. Also plotted are the 40-60% intervals about the medians.

These percentages are likely to be an underestimate of the success of VB, since on close examination of the individual EM and VBEM optimisations, it was revealed that for several cases the VBEM optimisation reached the maximum number of allowed iterations before it had converged, whereas EM always converged. Generally speaking the VBEM algorithm was found to require more iterations to reach convergence than EM, which would be considered a disadvantage if it were not for the considerable performance improvement of VB over BIC, BICp and CS.

We can also plot the smoothed version of figure 6.4 over instances of parameters of the true structure drawn from the prior; this is plotted in figure 6.6, which shows the median difference between the score of the true structure and the structure scoring highest under BIC, BICp, CS and VB. Also plotted is the 40-60% interval around the median. Again, the AIS experiments would have taken an unfeasibly large amount of computation time, and were not carried out.

We can see quite clearly here that the VB score of the true structure is generally much closer to that of the top-ranked structure than is the case for any of the other scores. This observation in itself is not particularly satisfying, since we are comparing scores to scores rather than scores to exact marginal likelihoods; nevertheless it can at least be said that the dynamic range between true and top-ranked structure scores by the VB method is much smaller than the range for the other methods. This observation is also apparent (qualitatively) across structures in the various plots in figure 6.2. We should be wary about the conclusions drawn from this graph comparing VB to the other methods: a completely ignorant algorithm which gives the same score to all

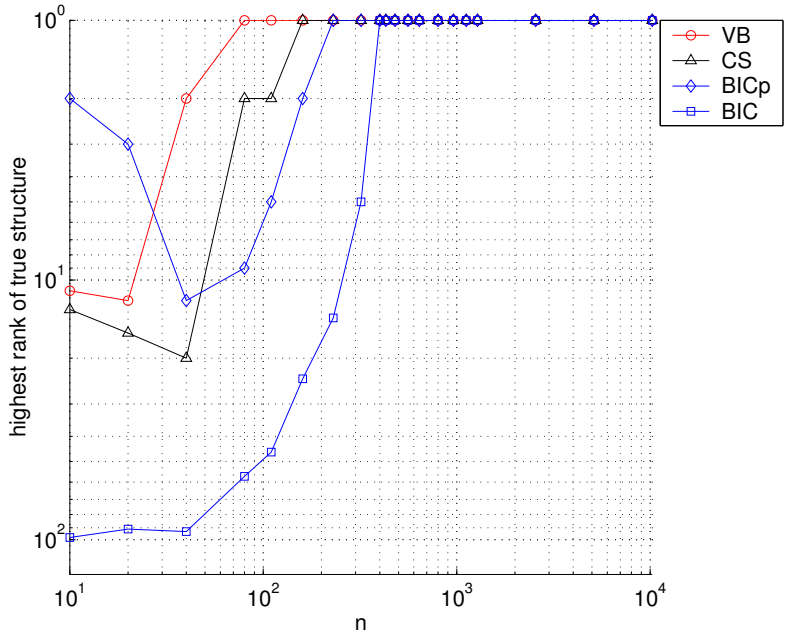


Figure 6.7: The highest ranking given to the true structure under BIC, BICp, CS and VB methods, against the size of the data set n, taken over 106 instances of the true structure. These two traces can be considered as the results of the $\min$ operation on the rankings of all the 106 instances for each n in figure 6.5.

possible structures would look impressive on this plot, giving a score difference of zero for all data set sizes.

Figures 6.7 and 6.8 show the best performance of the BIC, BICp, CS and VB methods over the 106 parameter instances, in terms of the rankings and score differences. These plots can be considered as the extrema of the median ranking and median score difference plots, and reflect the bias in the score.

Figure 6.7 shows the best ranking given to the true structure by all the scoring methods, and it is clear that for small data set sizes the VB and CS scores can perform quite well indeed, whereas the BIC scores do not manage a ranking even close to these. This result is echoed in figure 6.8 for the score differences, although we should bear in mind the caveat mentioned above (that the completely ignorant algorithm can do well by this measure).

We can analyse the expected performance of a naive algorithm which simply picks any structure at random as the guess for the true structure: the best ranking given to the true model in a set of 106 trials where a structure is chosen at random from the 136 structures is, on the average, roughly 1.8. We can see in figure 6.7 that CS and VB surpass this for $n > 30$ and $n > 40$ data points respectively, but that BICp and BIC do so only after 300 and 400 data points. However we should remember that, for small data set sizes, the true posterior over structures may well not have the true model at its mode.

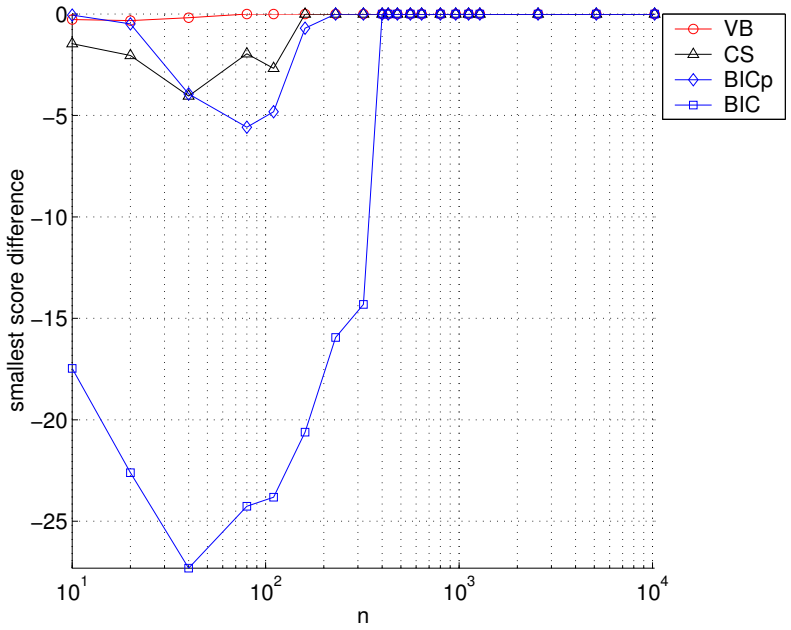


Figure 6.8: The smallest difference in score between the true and top-ranked structures, under BIC, BICp, CS and VB methods, against the size of the data set n, taken over 106 instances of the true structure. These two traces can be considered as the results of the $\max$ operation on the all the 106 differences for each n in figure 6.6.

Lastly, we can examine the success rate of each score at picking the correct structure. Figure 6.9 shows the fraction of times that the true structure is ranked top by the different scoring methods. This plot echoes those results in table 6.2.

6.5 Open questions and directions

This section is split into two parts which discuss some related issues arising from the work in this chapter. In section 6.5.1 we discuss some of the problems experienced when using the AIS approach, and suggest possible ways to improve the methods used in our experiments. In section 6.5.2 we more thoroughly revise the parameter-counting arguments used for the BIC and CS scores, and provide a method for estimating the complete and incomplete-data dimensionalities in arbitrary models, and as a result form a modified score CS†.

6.5.1 AIS analysis, limitations, and extensions

The technique of annealed importance sampling is currently regarded as a state-of-the-art method for estimating the marginal likelihood in discrete-variable directed acyclic graphical models (personal communication with R. Neal, Z. Ghahramani and C. Rasmussen). In this section the

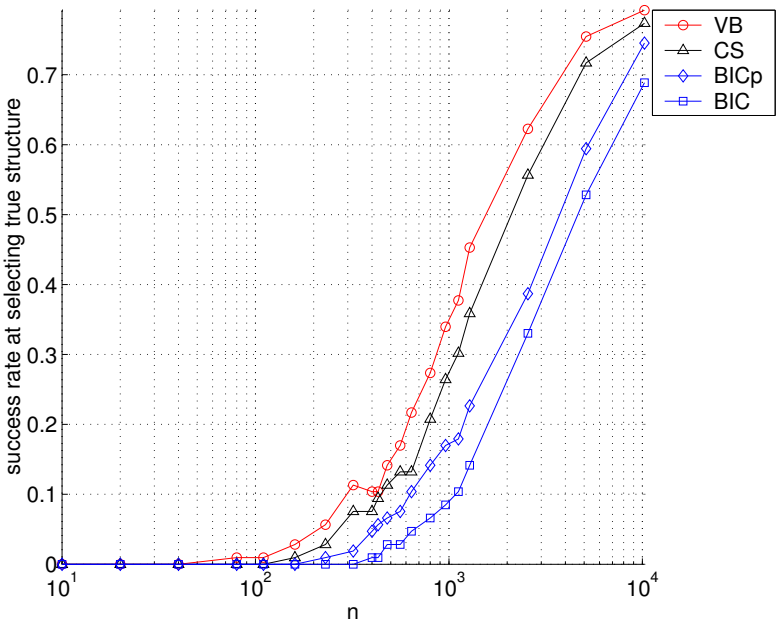


Figure 6.9: The success rate of the scoring methods BIC, BICp, CS and VB, as measured by the fraction of 106 trials in which the true structure was given ranking 1 amongst the 136 candidate structures, plotted as a function of the data set size. See also table 6.2 which presents softer performance rates (measured in terms of relative rankings) pooled from all the data set sizes and 106 parameter samples.

AIS method is critically examined as a reliable tool to judge the performance of the BIC, CS and VB scores.

The implementation of AIS has considerable flexibility; for example the user is left to specify the length, granularity and shape of the annealing schedules, the form of the Metropolis-Hastings sampling procedure, the number of samples taken at each temperature, etc. These and other parameters were described in section 6.3.5; here we clarify our choices of settings and discuss some further ways in which the sampler could be improved. Throughout this subsection we use AIS to refer to the algorithm which provides a single estimate of the marginal likelihood, i.e. $\mathrm{AIS}^{(1)}$.

First off, how can we be sure that the AIS sampler is reporting the correct answer for the marginal likelihood of each structure? To be sure of a correct answer one should use as long and gradual an annealing schedule as possible, containing as many samples at each temperature as is computationally viable (or compare to a very long simple importance sampler). In the AIS experiments in this chapter we always opted for a single sample at each step of the annealing schedule, initialising the parameter at the next temperature at the last accepted sample, and ensured that the schedule itself was as finely grained as we could afford. This reduces the variables at our disposal to a single parameter, namely the total number of samples taken in each run of AIS, which is then directly related to the schedule granularity. Without yet discussing the shape

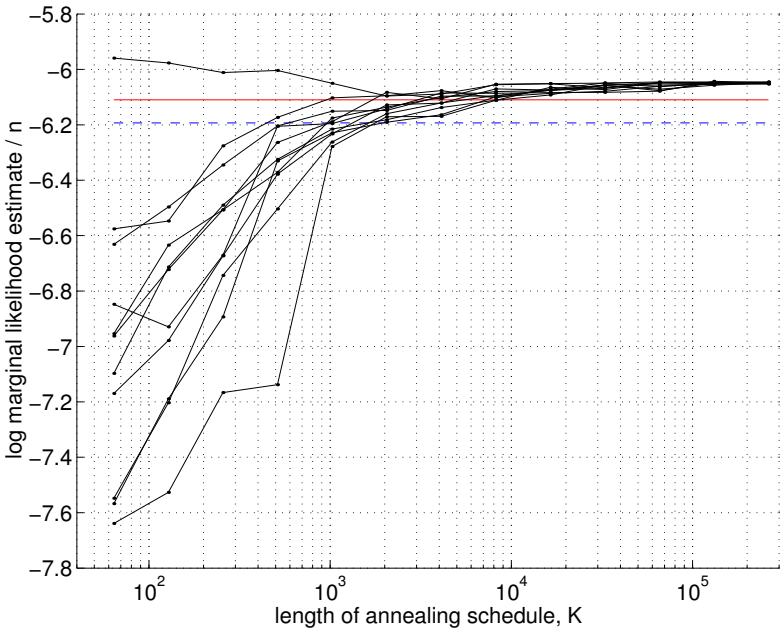


Figure 6.10: Logarithm of AIS estimates (vertical) of the marginal likelihood for different initial conditions of the sampler (different traces) and different duration of annealing schedules (horizontal), for the true structure with $n = 480$ data points. The top-most trace is that corresponding to setting the initial parameters to the true values that generated the data. Shown are also the BIC score (dashed) and the VB lower bound (solid).

of the annealing schedule, we can already examine the performance of the AIS sampler as a function of the number of samples.

Figure 6.10 shows several AIS estimates of the marginal likelihood for the data set of size $n = 480$ under the model having the true structure. Each trace is a result of initialising the AIS sampler at a different position in parameter space sampled from the prior (6.4), except for the top-most trace which is the result of initialising the AIS algorithm at the exact parameters that were used to generate the data (which as the experimenter we have access to). It is important to understand the abscissa of the plot: it is the number of samples in the AIS run and, given the above comments, relates to the granularity of the schedule; thus the points on a particular trace do *not* correspond to progress through the annealing schedule, but in fact constitute the results of runs that are completely different other than in their common parameter initialisation.

Also plotted for reference are the VB and BIC estimates of the log marginal likelihood for this data set under the true structure, which are not functions of the annealing duration. We know that the VB score is a strict lower bound on the log marginal likelihood, and so those estimates from AIS that consistently fall below this score must be indicative of an inadequate annealing schedule shape or duration.

For short annealing schedules, which are necessarily coarse to satisfy the boundary requirements on τ (see equation (6.49)), it is clear that the AIS sampling is badly under-estimating the log marginal likelihood. This can be explained simply because the rapid annealing schedule does not give the sampler time to locate and exploit regions of high posterior probability, forcing it to neglect representative volumes of the posterior mass; this conclusion is further substantiated since the AIS run started from the true parameters (which if the data is representative of the model should lie in a region of high posterior probability) over-estimates the marginal likelihood, because it is prevented from exploring regions of low probability. Thus for coarse schedules of less than about $K = 1000$ samples, the AIS estimate of the log marginal likelihood seems biased and has very high variance. Note that the construction of the AIS algorithm guarantees that the estimates of the marginal likelihood are unbiased, but not necessarily the log marginal likelihood.

We see that all runs converge for sufficiently long annealing schedules, with AIS passing the BIC score at about 1000 samples, and the VB lower bound at about 5000 samples. Thus, loosely speaking, where the AIS and VB scores intersect we can consider their estimates to be roughly equally reliable. We can then compare their computational burdens and make some statement about the advantage of one over the other in terms of compute time. At $n = 480$ the VB scoring method requires about $1.5s$ to score the structure, whereas AIS at $n = 480$ and $K = 2^{13}$ requires about $100s$; thus for this scenario VB is 70 times more efficient at scoring the structures (at its own reliability).

In this chapter's main experiments a value of $K = 2^{14} = 16384$ steps was used, and it is clear from figure 6.10 that we can be fairly sure of the AIS method reporting a reasonably accurate result at this value of K, at least for $n = 480$. However, how would we expect these plots to look for larger data sets in which the posterior over parameters is more peaky and potentially more difficult to navigate during the annealing?

A good indicator of the mobility of the Metropolis-Hastings sampler is the acceptance rate of proposed samples, from which the representative set of importance weights are computed (see (6.60)). Figure 6.11 shows the fraction of accepted proposals during the annealing run, averaged over AIS scoring of all 136 possible structures, plotted against the size of the data set, n; the error bars are the standard errors of the mean acceptance rate across scoring all structures. We can see that at $n = 480$ the acceptance rate is rarely below 60%, and so one would indeed expect to see the sort of convergence shown in figure 6.10. However for the larger data sets the acceptance rate drops to 20%, implying that the sampler is having considerable difficulty obtaining representative samples from the posterior distributions in the annealing schedule. Fortunately this drop is only linear in the logarithm of the data size. For the moment, we defer discussing the temperature dependence of the acceptance rate, and first consider combining AIS sampling runs to reduce the variance of the estimates.

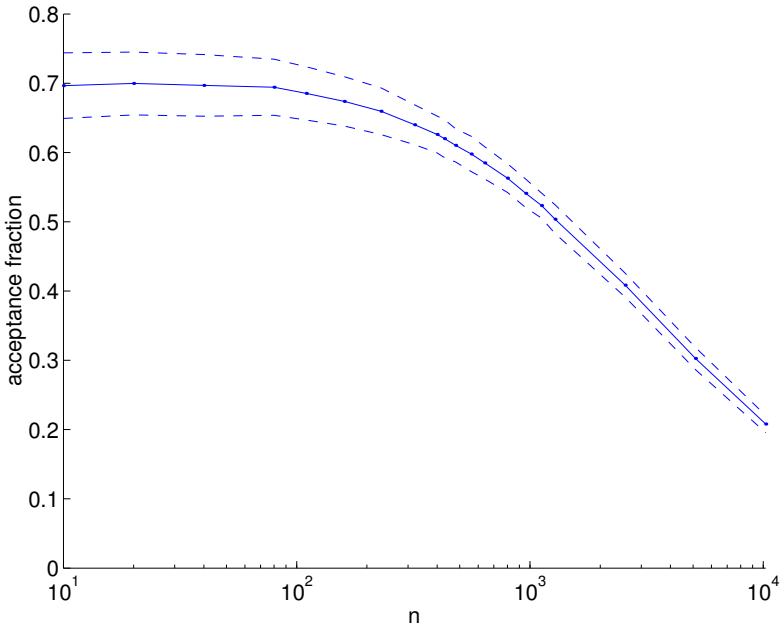


Figure 6.11: Acceptance rates of the Metropolis-Hastings proposals along the entire annealing schedule, for one batch of AIS scoring of all structures, against the size of the data set, n. The dotted lines are the sample standard deviations across all structures for each n.

One way of reducing the variance in our estimate of the marginal likelihood is to pool the results of several AIS samplers run in parallel according to the averaging in equation (6.61). Returning to the specific experiments reported in section 6.4, table 6.3 shows the results of running five AIS samplers in parallel with different random seeds on the entire class of structures and data set sizes, and then using the resulting averaged AIS estimate, $\text{AIS}^{(5)}$, as a score for ranking the structures. In the experiments it is the performance of these averaged scores that are compared to the other scoring methods: BIC, CS and VB. To perform five runs took at least 40 CPU days on an Athlon 1800 Processor machine.

By examining the reported AIS scores, both for single and pooled runs, over the 136 structures and 20 data set sizes, and comparing them to the VB lower bound, we can see how often AIS violates the lower bound. Table 6.4 shows the number of times the reported AIS score is below the VB lower bound, along with the rejection rates of the Metropolis-Hastings sampler that was used in the experiments (which are also plotted in figure 6.11). From the table we see that for small data sets the AIS method reports "valid" results and the Metropolis-Hastings sampler is accepting a reasonable proportion of proposed parameter samples. However at and beyond $n = 560$ the AIS sampler degrades to the point where it reports "invalid" results for more than half the 136 structures it scores. However, since the AIS estimate is noisy and we know that the tightness of the VB lower bound increases with n, this criticism could be considered too harsh — indeed if the bound were tight, we would expect the AIS score to violate the bound on roughly 50% of the runs anyway. The lower half of the table shows that, by combining AIS estimates from separate runs, we obtain an estimate that violates the VB lower bound far less

n	$\text{AIS}^{(1)}$ #1	$\text{AIS}^{(1)}$ #2	$\text{AIS}^{(1)}$ #3	$\text{AIS}^{(1)}$ #4	$\text{AIS}^{(1)}$ #5	$\text{AIS}^{(5)}$
10	27	38	26	89	129	59
20	100	113	88	79	123	135
40	45	88	77	5	11	15
80	10	47	110	41	95	44
110	1	50	8	2	62	2
160	33	2	119	31	94	6
230	103	25	23	119	32	54
320	22	65	51	44	42	78
400	89	21	1	67	10	8
430	29	94	21	97	9	18
480	2	42	14	126	18	2
560	47	41	7	59	7	11
640	12	10	23	2	23	7
800	7	3	126	101	22	23
960	1	4	1	128	8	1
1120	3	53	3	37	133	4
1280	76	2	50	7	12	5
2560	1	1	4	1	1	1
5120	12	1	24	2	16	1
10240	1	1	2	12	1	1

Table 6.3: Rankings resulting from averaging batches of AIS scores. Each one of the five columns correspond to a different initialisation of the sampler, and gives the rankings resulting from a single run of AIS for each of the 136 structures and 20 data set size combinations. The last column is the ranking of the true structure based on the mean of the AIS marginal likelihood estimates from all five runs of AIS of each structure and data set size (see section 6.3.5 for averaging details).

n	10 . . . 560	640	800	960	1120	1280	2560	5120	10240
single									
#$\text{AIS}^{(1)}$<VB*	≤5.7	12.3	8.5	12.3	10.4	17.0	25.5	53.8	71.7
#$\text{AIS}^{(1)}$<VB	≤7.5	15.1	9.4	14.2	12.3	20.8	31.1	59.4	74.5
% M-H rej.	<40.3	41.5	43.7	45.9	47.7	49.6	59.2	69.7	79.2
averaged									
#$\text{AIS}^{(5)}$<VB*	0	0.0	0.0	0.0	0.0	0.7	3.7	13.2	50.0
#$\text{AIS}^{(5)}$<VB	≤1.9	0.0	0.0	0.0	1.5	2.2	5.1	19.9	52.9

Table 6.4: AIS violations: for each size data set, n, we show the percentage of times, over the 136 structures, that a particular *single* AIS run reports marginal likelihoods below the VB lower bound. These are given for the VB scores that are uncorrected (*) and corrected for aliases. Also shown are the average percentage rejection rates of the Metropolis-Hastings sampler used to gather samples for the AIS estimates. The bottom half of the table shows the similar violations by the AIS score that are made from averaging the estimates of marginal likelihoods from five separate runs of AIS (see section 6.3.5). Note that the Metropolis-Hastings rejection rates are still just as high for each of the individual runs (not given here).

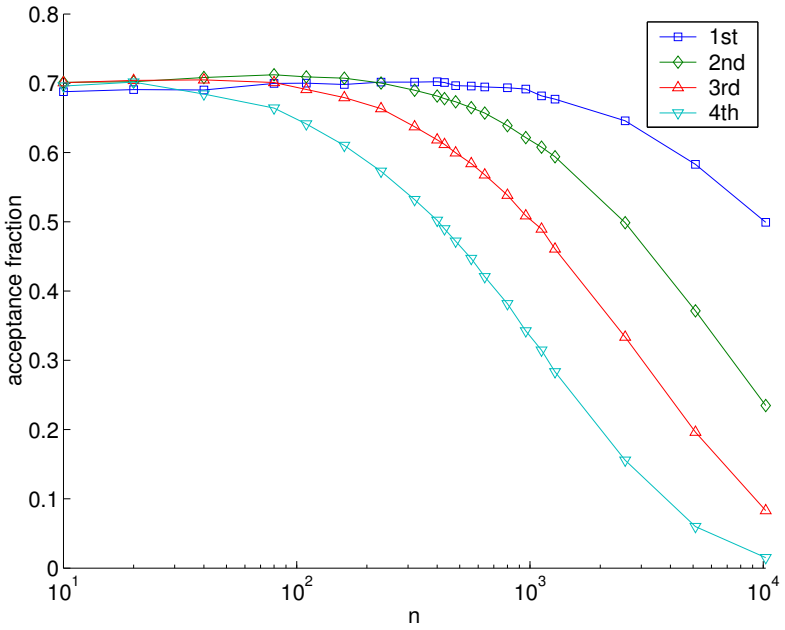


Figure 6.12: Acceptance rates of the Metropolis-Hastings proposals for each of four quarters of the annealing schedule, for one batch of AIS scoring of all structures, against the size of the data set, n. Standard errors of the means are omitted for clarity.

often, and as expected we see the 50% violation rate for large amounts of data. This is a very useful result, and obviates to some extent the Metropolis-Hastings sampler's deficiency in all five runs.

However, considering for the moment a single AIS run, for large data set sizes the VB bound is still violated an unacceptable number of times, suggesting that the Metropolis-Hastings proposals are simply not adequate for these posterior landscapes. This suggests a modification to the proposal mechanism, outlined below. Diagnostically speaking, this hopefully has served as a good example of the use of readily-computable VB lower bounds for evaluating the reliability of the AIS method *post hoc*.

Let us return to examining why the sampler is troubled for large data set sizes. Figure 6.12 shows the fraction of accepted Metropolis-Hastings proposals during each of four quarters of the annealing schedule used in the experiments. The rejection rate tends to increase moving from the beginning of the schedule (the prior) to the end (the posterior), the degradation becoming more pronounced for large data sets. This is most probably due to the proposal width remaining unchanged throughout all the AIS implementations: ideally one would use a predetermined sequence of proposal widths which would be a function of the amount of data, n, and the position along the schedule. This would hopefully eliminate or at least alleviate the pronounced decrease in acceptance rate across the four quarters, but would also cause each individual trace to not drop so severely with n.

We can use a heuristic argument to roughly predict the optimal proposal width to use for the AIS method. From mathematical arguments outlined in sections 1.3.2 and 1.3.4, the precision of the posterior distribution over parameters is approximately proportional to the size of the data set n. Furthermore, the distribution being sampled from at step k of the AIS schedule is effectively that resulting from a fraction $\tau(k)$ of the data. Therefore these two factors imply that the width of the Metropolis-Hastings proposal distribution should be inversely proportional to $\sqrt{n\tau(k)}$. In the case of multinomial variables, since the variance of a Dirichlet distribution is approximately inversely proportional to the strength, α, (see appendix A), then the optimal strength of the proposal distribution should be $\alpha_{opt} \propto n\tau(k)$ if its precision is to match the posterior precision. Note that we are at liberty to set these proposal precisions arbitrarily beforehand without causing the sampler to become biased.

We have not yet discussed the shape of the annealing schedule: should the inverse-temperatures $\{\tau(k)\}_{k=1}^{K}$ change linearly from 0 to 1, or follow some other function? The particular annealing schedule in these experiments was chosen to be nonlinear, lingering at higher temperatures for longer than at lower temperatures, following the relationship

$$\tau(k) = \frac{e_\tau k/K}{1 - k/K + e_\tau} \qquad k \in \{0, \ldots, K\} \,, \tag{6.73}$$

with e_τ set to 0.2 . For any setting of $e_\tau > 0$, the series of temperatures is monotonic and the initial and final temperatures satisfy (6.49):

$$\tau(0) = 0 \,, \qquad \text{and} \qquad \tau(K) = 1 \,. \tag{6.74}$$

For large e_τ, the schedule becomes linear. This is plotted for different values of e_τ in figure 6.13. The particular value of e_τ was chosen to reduce the degree of hysteresis in the annealing ratios, as discussed below.

Hysteresis in the annealing ratios

As presented in section 6.3.5 and algorithm 6.1, the algorithm for computing each and every marginal likelihood ratio in (6.54) did so in a forward manner, carrying over the parameter setting $\boldsymbol{\theta}_{\text{ini}}$ from the calculation of the previous ratio to initialise the sampling procedure for calculating the next ratio. However, whilst it makes sense to move from higher to lower temperatures to avoid local maxima in the posterior in theory, the final estimate of the marginal likelihood is unbiased regardless of the order in which the ratios are tackled. In particular, we can run the AIS algorithm in the *reverse* direction, starting from the posterior and warming the system to the prior, calculating each ratio exactly as before but using the last sample from the lower temperature as an initialisation for the sampling at the next higher temperature in the schedule (note that by doing this we are *not* inverting the fractions appearing in equation (6.54)).

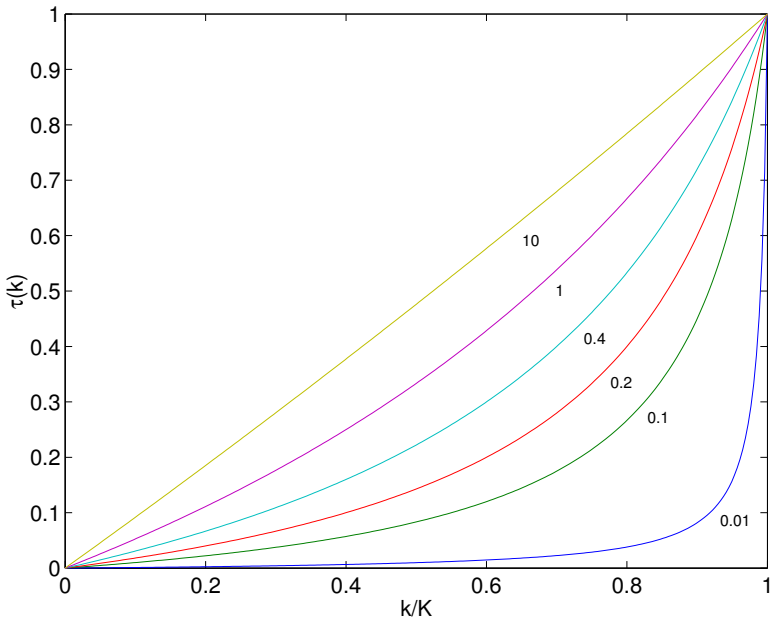


Figure 6.13: Non-linear AIS annealing schedules, plotted for six different values of e_τ. In the experiments performed in this chapter, $e_\tau = 0.2$.

What can this reverse procedure do for us? If we look at figure 6.10 again, we can see that for any random parameter initialisation the reported marginal likelihood is much more often than not an underestimate of the true value. This is because for coarse annealing schedules we are unlikely to locate regions of high posterior probability by the time the system is quenched. If we were then to run the AIS algorithm in a reverse direction, starting from where we had finished the forward pass, we would expect on average to report a higher marginal likelihood than that just reported by the forward pass, simply because the sampler has had longer to explore the high probability regions.

A logical conclusion is that if the forward and reverse passes yield very different values for the marginal likelihood, then we have most likely used too short an annealing schedule. And furthermore, since the marginal likelihood estimates are constructed from the product of many ratios of marginal likelihoods, we can use the discrepancies between the ratios calculated on the forward and reverse passes to choose temperature regions where more sampling is required, and dilate the annealing schedules in these regions accordingly. Of course we should remember that these discrepancies are stochastic quantities, and so we should modify the schedule based on averaged discrepancies over several runs.

This heuristic analysis was used when designing the shape and granularity of the annealing schedule, and we found that more time was required at higher and intermediate temperatures at the expense of lower temperatures. An area of future research is to formalise and more fully investigate this and related arguments. For example, it would be useful to characterise the dependence of the degree of hysteresis along the schedule for different settings of e_τ.

6.5.2 Estimating dimensionalities of the incomplete and complete-data models

The BICp, BIC and CS approximations take the limit of the Laplace approximation as the amount of data tends to infinity, and result in scores that depend on the dimensionalities of the incomplete and complete models, d and d' respectively. In the experiments in this chapter, for BIC d was calculated using a simple counting argument (see equation (6.24) in section 6.3.2), and for CS d and d' were assumed to be equal, which is the assumption made in the original implementation of Cheeseman and Stutz (1996).

In models that have no hidden variables, the value of d required for the BIC approximation can usually be arrived at by adding together the degrees of freedom in each parameter, taking care to take into consideration any parameter degeneracies. However, in models that do have hidden variables the number of free parameters in the incomplete model is much less than that in the complete model. This is because the full effect of each hidden variable cannot always be fully manifest in the functions produced on the observed variables. This situation can be seen in the following discrete example: imagine the model consisting of a single k-valued hidden variable which is the (only) parent of a p-valued observed variable. The naive counting argument would return the complete dimensionality as $d' = (k-1) + (p-1) \times k$. However, the incomplete dimensionality can be no more than $d = (p-1)$, as a model with this many degrees of freedom can exactly model any observed set of counts of the observed variable.

In a general setting, deducing the complete and incomplete model dimensionalities can be complicated (see, for example, Settimi and Smith, 1998; Kočka and Zhang, 2002), since it involves computing the rank of the Jacobian of the transformation for parameters from incomplete to complete models. Geiger et al. (1996) describe a method by which d can be computed in discrete DAGs, by diagonalising the Jacobian symbolically; they also present a theorem that guarantees that a randomised version of the symbolic operation is viable as well. Unfortunately their approach seems difficult to implement efficiently on an arbitrary topology discrete DAG, since both symbolic and randomised versions require diagonalisation. Furthermore it is not clear how, if at all, it can be transferred to DAGs containing continuous variables with arbitrary mappings between the complete and incomplete data models.

For the purposes of this chapter, we have used a simple method to estimate the dimensionalities of each model in our class. It is based on analysing the effect of random perturbations to the model's parameters on the complete and incomplete-data likelihoods. The procedure is presented in algorithm 6.2, and estimates the number of effective dimensions, d and d', by computing the rank of a perturbation matrix. Since the rank operation attempts to find the number of linearly independent rows of the matrices C and C', the random ϵ-perturbations must be small enough such that the change in the log likelihoods are linear with ϵ. Also, the number of samples n should be chosen to be at least as large as the total number of parameters possible in

Algorithm 6.2: $d(m), d'(m)$: To estimate incomplete and complete model parameter dimensionalities.

---

1. For each structure m

   (b) Obtain $\boldsymbol{\theta}_{\text{MAP}}$ using the MAP EM algorithm (section 6.3.1).

   (a) Obtain a representative set of all possible observed data $\{\mathbf{y}_i\}_{i=1}^{n}$.

   (d) Randomly (spherically) ϵ-perturb $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ R times, to form $\{\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_R\}$.

   (e) Compute the matrix $C(n \times R) : C_{ir} = \ln p(\mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}}_r)$ for all (i, r).
   Estimate $d(m) = \text{rank}(C) - 1$.

   (f) Compute the matrix $C'(n \times R) : C'_{ir} = \ln p(\mathbf{s}_i, \mathbf{y}_i \,|\, \hat{\boldsymbol{\theta}}_r)$ for all (i, r), where $\mathbf{s}_i$ is a randomly instantiated hidden state.
   Estimate $d'(m) = \text{rank}(C') - 1$.

End For

the model (as the rank of a matrix can be no more than the smaller of the number of rows or columns), and preferably several times this for reliable estimates.

This procedure was found to give reasonable results when carried out on all the model structures used in this chapter, with a randomly generated data set of size $n = 1000$ and $R = 100$. Without listing all the results, it suffices to say that: for all structures $d \leq d' \leq d+2$, and for the majority of structures $d' = d+|\mathcal{H}|$ — that is to say a further degree of freedom is provided for each binary hidden variable (of which there are at most 2) on top of the incomplete dimensionality. There are some structures for which the discrepancy $d' - d$ is smaller than 2, which is not as we would expect.

There may be several reasons for this discrepancy. First the random perturbations may not have explored certain directions from the MAP estimate, and thus the algorithm could have reported a lower dimensionality than true (unlikely). Second, the data $\mathbf{y}$ only represented a subset of all possible configurations (almost certainly since there are 5^4 possible realisations and 1000 data points are generated randomly), and therefore the effective dimensionality drops.

These results support the use of a more accurate CS† score — see equation (6.30), which modifies the score by adding a term $(d' - d)/2 \cdot \ln n$. The effect of this is to raise the scores for models with 2 hidden variables by $\ln n$, raise those with just 1 hidden variable by $1/2 \cdot \ln n$, and leave unchanged the single model with no hidden states.

Table 6.5 shows the improvement (in terms of ranking) of the more accurate CS† over the original CS approximation, bringing it closer to the performance of the VB score. The table shows the number of times in the 106 samples (see experiments in section 6.4 above) that the

n	BIC	BICp	CS	CS†	VB
10	0	0	0	0	0
20	0	0	0	0	0
40	0	0	0	0	0
80	0	0	0	1	1
110	0	0	0	0	1
160	0	0	1	2	3
230	0	1	3	5	6
320	0	2	8	10	12
400	1	5	8	9	11
430	1	6	10	10	11
480	3	7	12	12	15
560	3	8	14	16	18
640	5	11	14	17	23
800	7	15	22	23	29
960	9	18	28	33	36
1120	11	19	32	33	40
1280	15	24	38	41	48
2560	35	41	59	62	66
5120	56	63	76	76	80
10240	73	79	82	83	84

Table 6.5: Number of times (out of 106) that each score selects the true structure. Shown are the performance of the original BIC, BICp, CS and VB scores, all corrected for aliasing, and also shown is the CS† score, resulting from (further) correcting CS for the difference between complete and incomplete data model dimensionalities.

score successfully selected the true model structure. Is it clear that CS† is an improvement over CS, suggesting that the assumption made above is true. However, we should interpret this experiment with some care, because our original choice of the true model having two hidden variables may be masking a bias in the altered score; it would make sense to perform similar experiments choosing a much simpler model to generate the data.

The improvement in performance of the CS† score, averaged over all data set sizes and all 106 generated parameter sets can be see in table 6.2 (page 233), where it is compared alongside BIC, CS and VB. It can be seen that VB still performs better. Further verification of this result will be left to future work.

6.6 Summary

In this chapter we have presented various scoring methods for approximating the marginal likelihood of discrete directed graphical models with hidden variables. We presented EM algorithms for ML and MAP parameter estimation, showed how to calculate the asymptotic criteria of BIC and Cheeseman-Stutz, derived the VBEM algorithm for approximate Bayesian learning which

maintains distributions over the parameters of the model and has the same complexity as the EM algorithm, and presented a (somewhat impoverished) AIS method designed for discrete-variable DAGs.

We have shown that VB consistently outperforms BIC and CS, and that VB performs respectively as well as and more reliably than AIS for intermediate and large sizes of data. The AIS method has very many parameters to tune and requires extensive knowledge of the model domain to design efficient and reliable sampling schemes and annealing schedules. VB on the other hand has not a single parameter to set or tune, and can be applied without any expert knowledge, at least in the class of singly-connected discrete-variable DAGs with Dirichlet priors which we have considered in this chapter. Section 6.5.1 discussed several ways in which the AIS method could be improved, for example by better matching the Metropolis-Hastings proposal distributions to the annealed posterior; in fact a method based on slice sampling should be able to adapt better to the annealing posterior with little or no expert knowledge of the shape of the annealed posterior (Neal, 2003).

It may be that there exists a better AIS scheme than sampling in parameter space. To be more specific, for any completion of the data the parameters of the model can be integrated out tractably (at least for the class of models examined in this chapter); thus an AIS scheme which anneals in the space of completions of the data may be more efficient than the current scheme which anneals in the space of parameters (personal communication with R. Neal). However, this latter scheme may only be efficient for models with little data compared to the number of parameters, as the sampling space of all completions increases linearly with the amount of data. This avenue of research is left to further work.

This chapter has presented a novel application of variational Bayesian methods to discrete DAGs. In the literature there have been other attempts to solve this long-standing model selection problem. For example the *structural EM* algorithm of Friedman (1998) uses a structure search algorithm which uses a scoring algorithm very similar to the VBEM algorithm presented here, except that for tractability the distribution over $\boldsymbol{\theta}$ is replaced by the MAP estimate, $\boldsymbol{\theta}_{\mathrm{MAP}}$. We have shown here how the VB framework enables us to use the entire distribution over $\boldsymbol{\theta}$ for inference of the hidden variables.

In chapter 2 we proved that the Cheeseman-Stutz score is in fact a lower bound on the marginal likelihood and, more importantly, we proved that there exists a construction which is guaranteed to produce a variational Bayesian lower bound that is *at least as tight* as the Cheeseman-Stutz score (corollary 2.5 to theorem 2.3, page 79). This construction builds a variational Bayesian approximation using the same MAP parameter estimate used to obtain the CS score. However, we did not use this construction in our experiments, and ran both the MAP EM and VB optimisations independently of each other. As a result we cannot guarantee that the VB bound is in all runs tighter than the CS bound, as the dynamics of the optimisations for MAP learning

and VB learning may in general lead even identically initialised algorithms to different optima in parameter space (or parameter distribution space). Nevertheless we have still seen improvement in terms of ranking of the true structure by VB as compared to CS. A tighter bound on the marginal likelihood does not necessarily directly imply that we should have better structure determination, although it certainly suggests this and is supported by the experimental results. Empirically, the reader may be interested to know that the VB lower bound was observed to be *lower* than the CS score in only 173 of the 288320 total scores calculated (about 0.06%). If the construction derived in corollary 2.5 had been used then this number of times would of course be exactly zero.

Chapter 7

Conclusion

7.1 Discussion

In this thesis we have shown how intractable Bayesian learning, inference, and model selection problems can be tackled using variational approximations. We have described a general framework for variational Bayesian learning and shown how it can be applied to several models of interest. We have demonstrated that it is an efficient and trustworthy approximation as compared to other more traditional approaches. Before summarising the contributions of this thesis, we spend the next few paragraphs discussing some of the evolving directions for model selection and variational Bayes, including the use of infinite models, inferring causal relationships using the marginal likelihood, other candidates for approximating the marginal likelihood, and lastly automated algorithm derivation procedures. These areas are expected to be active and fruitful future research directions. We conclude in section 7.2 with a summary of the main contributions of the thesis.

Infinite models

In this thesis we have focused on Bayesian learning in models that can be specified using a finite number of parameters. However, there are powerful arguments for entertaining models with infinitely many parameters, or at least as complex models as can be handled computationally. The process of Bayesian inference yields a unique answer. That is to say, given our prior beliefs, on observing some data all inference is automatic and there is one and only one answer to any prediction of the model. The problems of under- or overfitting by using too simple or too complex a model are simply not a concern if we integrate over all uncertain variables in the model, since applying Bayes' rule correctly at every step is guaranteed to result in coherent and optimal inferences given the prior beliefs. In this way the problem of model selection is

no longer an issue, because the infinite model can entertain a continuum of models and average with respect to all of these simultaneously. This approach to modelling is discussed in Neal (1996) where, for example, neural networks with an infinite number of hidden units are shown (theoretically and empirically) to produce sensible predictions, and on some data sets state-of-the-art performance. In general it is difficult and sometimes impossible to entertain the limit of an infinite model, except where the mathematics lends itself to analytically tractable solutions — this is often the case for mixture models. Examples of Bayesian learning with infinite models include: the neural networks mentioned above, infinite mixtures of Gaussians (Rasmussen, 2000), infinite hidden Markov models (Beal et al., 2002), and infinite mixtures of Gaussian process experts (Rasmussen and Ghahramani, 2002). The basic idea of examining the infinite limit of finite models can be applied to a host of other as yet unexplored models and applications.

Unfortunately, a major drawback for these infinite models is that inference is generally intractable, and one has to resort to Monte Carlo sampling methods which can be computationally costly. Also, representing an infinite number of components in a mixture model, for example, can quickly become cumbersome; even elaborate Markov chain Monte Carlo approaches become very inefficient in models with many parameters. One further disadvantage of employing infinite models is that it is often difficult to find ways of encapsulating prior expert knowledge into the model. Methods such as examining the properties of data drawn from specific prior settings are illuminating but not always entirely satisfactory for designing the prior to articulate one's beliefs.

An alternative to grappling with the conceptual and implementational problems of infinite models is then to restrict ourselves to performing model inference, or selection amongst a finite set of finite-size models. Each individual model is then manageable and often simpler to interpret in terms of its structure. On the basis of the marginal likelihood we can obtain posterior distributions over the different candidate models. The problems discussed in this thesis have emphasised these model selection and structure learning tasks, as well as attempting to obtain full posterior distributions over model structures. We have examined a selection of statistical models, all of which contained hidden variables which cause the marginal likelihood computation to be intractable, and tackled this intractability using variational methods.

Bethe, Kikuchi, and cluster-variation methods

Variational Bayes, as described in this thesis, is just one type of variational approach that could be used to approximate Bayesian inference. It assumes simple forms for the posterior distributions over hidden variables and parameters, and then uses these forms to construct lower bounds on the marginal likelihood that are tractable. Algorithms for inference and learning are then derived as a result of optimising this lower bound by iteratively updating the parameters of these simplified distributions. Most of this thesis has concentrated on the ease with which the model

parameters can be included in the set of uncertain variables to infer and integrate over, at least for the sub-class of conjugate-exponential models.

A promising alternative direction is to explore the Bethe and Kikuchi family of variational methods (Yedidia et al., 2001), sometimes called cluster-variational methods, which may be more accurate but do not provide the assurance of being bounds. These re-express the negative log marginal likelihood as a "free energy" from statistical physics, and then approximate the (intractable) entropy of the posterior distribution over latent variables by neglecting high order terms. In the Bethe approximation, the entropy is approximated with an expression which depends only on functions of single variables and pairs of variables. There are several procedures for minimising the Bethe free energy as a functional of the approximate posterior distributions to obtain estimates of the marginal likelihood. It turns out that for singly-connected graphs the fixed point equations that result from iterative minimisation of this free energy with respect to the single and pairwise functions correspond exactly to the messages that are passed in the junction tree and sum-product algorithms. Thus the Bethe free energy is exact for singly-connected graphs (trees). Interestingly, it has recently been shown that the belief propagation algorithm, even when run on multiply-connected graphs (i.e. 'loopy' graphs), has stable fixed points at the minima of the Bethe free energy (Heskes, 2003). While belief propagation on loopy graphs is not guaranteed to converge, it often works well in practice, and has become the standard approach to decoding state-of-the-art error-correcting codes. Furthermore, convergent algorithms for minimising the Bethe free energy have recently been derived (Yuille, 2001; Welling and Teh, 2001). There are other related methods, such as expectation propagation (EP, Minka, 2001a), approximations which observe higher order correlations in the variables (Leisink and Kappen, 2001), and other more elaborate variational schemes for upper bounds on partition functions (Wainwright et al., 2002).

The question remains open as to whether these methods can be readily applied to Bayesian learning problems. One can view Bayesian learning as simply treating the parameters as hidden variables, and so every method that has been shown to be successful for inference over hidden variables should also do well for integrating over parameters. However, there have been few satisfactory examples of Bayesian learning using any of the other methods described above, and this is an important direction for future research.

Inferring causal relationships

Most research in statistics has focused on inferring probabilistic dependencies between model variables, but more recently people have begun to investigate the more challenging and controversial problem of inferring *causality*. Causality can be understood statistically as a relationship $s \rightarrow t$ which is stable regardless of whether s was set through intervention / experimental manipulation or it occurred randomly. An example of this is smoking (s) causing yellowing of

the teeth (t). Painting the teeth yellow does not change the probability of smoking, but forcing someone to smoke does change the probability of the teeth becoming yellow. Note that both the models $s \rightarrow t$ and $s \leftarrow t$ have the same conditional independence structure, yet they have very different causal interpretations. Unfortunately this has lead many researchers to believe that such causal relationships cannot be inferred from observational data alone, since these models are *likelihood equivalent* (Heckerman et al., 1995). Likelihood equivalent models are those for which an arc reversal can be accompanied by a change in parameters to yield the same likelihood. As a result these researchers then propose that causation can only be obtained by assessing the impact of active manipulation of one variable on another. However, this neglects the fact that the *prior* over parameters may cause the marginal likelihoods to be different even for likelihood equivalent models (D. MacKay, personal communication). In this context, it would be very interesting to explore the reliability with which variational Bayesian methods can be used to infer such causal relationships in general graphical models. In chapter 6 we showed that variational Bayes could determine the presence or absence of arcs from hidden variables to observed variables in a simple graphical model class. Envisaged then is a similar investigation for examining the directionality of arcs in a perhaps more expressive structure class.

Automated algorithm derivation

One of the problems with the variational Bayesian framework is that, despite the relative simplicity of the theory, the effort required to derive the update rules for the VBE and VBM steps is usually considerable and a hindrance to any implementation. Both the derivation and implementation have to be repeated for each new model, and both steps are prone to error. The variational linear dynamical system discussed in chapter 5 is a good example of a simple model for which the implementation is nevertheless cumbersome.

Our contribution of generalising the procedure for conjugate-exponential (CE) family models (section 2.4) is a step in the right direction for automated algorithm derivation. For CE models, we now know that the complexity of inference for variational Bayesian inference is the same as for point-parameter inference, and that for simple models such as HMMs existing propagation algorithms can be used unaltered with *variational Bayes point* parameters (see theorem 2.2).

There are a number of software implementations available or in development for inference and general automated algorithm derivation. The BUGS software package (Thomas et al., 1992) for automated Bayesian inference using Gibbs sampling is the most widely used at present; the graphical model and functional forms of the conditional probabilities involving both discrete and continuous variables can be specified by hand and then the sampling is left to obtain posterior distributions and marginal probabilities. For more generic algorithm derivation, the *AutoBayes* project (Gray et al., 2003) uses symbolic techniques to automatically derive the equations re-

quired for learning and inference in the model and explicitly produces the software to perform the task.

A similar piece of software is being developed in the *VIBES* project (Bishop et al., 2003). This package explicitly uses precisely the CE variational Bayesian results presented in chapter 2 of this thesis to automate the variational inference and learning processes, for (almost) arbitrary models expressed in graphical form. To be fully useful, this package should be able to cope with user-specified further approximation to the posterior, on top of just the parameter / hidden variable factorisation. Furthermore it should be relatively straightforward to allow the user to specify models which have non-CE components, such as logistic sigmoid functions. This would allow for discrete children of continuous parents, and could be made possible by including quadratic lower bounds on the sigmoid function (due to Jaakkola, 1997) to ensure that there is still a valid overall lower bound on the marginal likelihood. Looking further in to the future, these software applications may even be able to suggest 'good' factorisations, or work with a variety of these approximations together or even hierarchically. Also an alternative for coping with non-CE components of the model might be to employ sampling-based inferences in small regions of the graph that are affected.

Combining the variational Bayesian theory with a user-friendly interface in the form of VIBES or similar software could lead to the mass use of variational Bayesian methods in a wide variety of application fields. This would allow the ready comparison of a host of different models, and greatly improve the efficiency of current research on variational Bayes. However there is the caveat, which perhaps has not been emphasised enough in this thesis, that blind applications of variational Bayes may lead to the wrong conclusions, and that any inferences should be considered in the context of the approximations that have been made. This reasoning may not come easily to an automated piece of software, and the only sure answer to the query of whether the variational lower bound is reliable is to compare it to the exact marginal likelihood. It should not be difficult to overlay onto VIBES or similar software a set of sampling components to do exactly this task of estimating the marginal likelihood very accurately for diagnostic purposes; one such candidate for this task could be annealed importance sampling.

7.2 Summary of contributions

The aim of this thesis has been to investigate the variational Bayesian method for approximating Bayesian inference and learning in a variety of statistical models used in machine learning applications. Chapter 1 reviewed some of the basics of probabilistic inference in graphical models, such as the junction tree and belief propagation algorithms for exact inference in both undirected and directed graphs. These algorithms are used for inferring the distribution over hidden variables given observed data, for a *particular setting* of the model parameters. We showed that in

situations where the parameters of the model are unknown the correct Bayesian procedure is to integrate over this uncertainty to form the marginal likelihood of the model. We explained how the marginal likelihood is the key quantity for choosing between models in a model selection task, but also explained that it is intractable to compute for almost all interesting models.

We reviewed a number of current methods for approximating the marginal likelihood, such as Laplace's method, the Bayesian information criterion (BIC), and the Cheeseman-Stutz criterion (CS). We discussed how each of these have significant drawbacks in their approximations. Perhaps the most salient deficiency is that they are based on maximum a posteriori parameter (MAP) estimates of the model parameters, which are arrived at by maximising the posterior density of the parameters, and so the MAP estimate may not be representative of the posterior mass at all. In addition we noted that the MAP optimisation is basis dependent, which means that two different experimenters with the same model and priors, but with different parameterisations, do not produce the same predictions using their MAP estimates. We also discussed a variety of sampling methods, and noted that these are guaranteed to give an exact answer for the marginal likelihood only in the limit of an infinite number of samples, and one often requires infeasibly long sampling runs to obtain accurate and reliable estimates.

In chapter 2 we presented the variational Bayesian method for approximating the marginal likelihood. We first showed how the standard expectation-maximisation (EM) algorithm for learning ML and MAP parameters can be interpreted as a variational optimisation of a lower bound on the likelihood of the data. In this optimisation, the E step can either be exact, in which case the lower bound is tight after each E step, or it can be restricted to a particular family of distributions in which case the bound is loose. The amount by which the bound is loose is exactly the Kullback-Leibler divergence between the variational hidden variable posterior and the exact posterior. We then generalised this methodology to the variational Bayesian EM algorithm which integrates over the parameters. The algorithm alternates between a VBE step which obtains a variational posterior distribution over the hidden variables given a distribution over the parameters, and a VBM step which infers the variational distribution over the parameters given the result of the VBE step. The lower bound gap is then given by the KL divergence between the variational joint posterior over hidden variables and parameters, and the corresponding exact posterior.

Significant progress in understanding the VB EM optimisation was made by considering the form of the update equations in the case of conjugate-exponential (CE) models. We showed that if the complete-data likelihood for the model is in the exponential family and the prior over parameters is conjugate to this likelihood, then the VB update equations take on analytically tractable forms and have attractive intuitive interpretations. We showed that, in theory, it is possible to use existing propagation algorithms for performing the VBE step, even though we have at all times a distribution over the parameters. This is made possible by passing the propagation algorithm the *variational Bayes point* parameter, $\boldsymbol{\theta}_{\mathrm{BP}} \equiv \boldsymbol{\phi}^{-1}(\langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})})$, which

is the result of inverting the exponential family's natural parameter mapping after averaging the natural parameters under the variational posterior. This is a very powerful result as it means that variational Bayesian inference (the VBE step) is possible in the same time complexity as the standard E step for the point-parameter case (with the only overhead being that of inverting the mapping). We also presented corollaries of this result applied to directed (Bayesian) and undirected (Markov) networks — see corollaries 2.2 and 2.4.

In chapter 3 we presented a straightforward example of this important result applied to Bayesian learning in a hidden Markov model. Here the variational Bayes point parameters are sub-normalised transition and emission probabilities for the HMM, and the well-known forward-backward algorithm can be used unchanged with these modified parameters. We carried out experiments (some of which are suggested in MacKay, 1997) which showed that the VB algorithm was capable of determining the number of hidden states used to generate a synthetic data set, and outperforms ML and MAP learning on a task of discriminating between forwards and backwards English sentences. This shows that integrating over the uncertainty in parameters is important, especially for small data set sizes. The linear dynamical system of chapter 5 has the same structure as the HMM, so we might expect it to be equally suitable for the propagation corollary. However for this model it was not found to be possible to invert the natural parameter mapping, but nevertheless a variational Bayesian inference algorithm was derived with the same time complexity as the well-known Rauch-Tung-Striebel smoother. It was then shown that the VB LDS system could use automatic relevance determination methods to successfully determine the dimensionality of the hidden state space in a variety of synthetic data sets, and that the model was able to discard irrelevant driving inputs to the hidden state dynamics and output processes. Some preliminary results on elucidating gene-expression mechanisms were reported, and we expect this to be an active area of future research.

Chapter 4 focused on a difficult model selection problem, that of determining the numbers of mixture components in a mixture of factor analysers model. Search over model structures for MFAs is computationally intractable if each analyser is allowed to have different intrinsic dimensionalities. We derived and implemented the variational Bayesian EM algorithm for this MFA model, and showed that by wrapping the VB EM optimisation within a birth and death process we were able to navigate through the space of number of components using the lower bound as a surrogate for the marginal likelihood. Since all the parameters are integrated out in a Bayesian implementation, we are at liberty to begin the search either from the simplest model or from a model with very many components. Including an automatic relevance determination prior on the entries of each of the factor loading matrices' columns allowed the optimisation to simultaneously find the number of components and their dimensionalities. We demonstrated this on several synthetic data sets, and showed improved performance on a digit classification task as compared to a BIC-penalised ML MFA model. We noted that for this mixture model the death process was an automatic procedure, and also suggested several ways in which the birth processes could be implemented to increase the efficiency of the structure search.

Also in this chapter we presented a generally applicable importance sampling procedure for obtaining estimates of the marginal likelihood, predictive density, and the KL divergence between the variational and exact posterior distributions. In the sampler, the variational posteriors are used as proposal distributions for drawing importance samples. We found that although the lower bound tends to correlate well with the importance sampling estimate of the marginal likelihood, the KL divergence (the bound gap) increases approximately linearly with the number of components in the MFA model, which would suggest that the VB approximation has an inherent bias towards simpler models. We note also that importance sampling can fail for poor choices of proposal distribution and is not ideal for high dimensional parameter spaces. We attempted to improve the estimates by using heavier tailed and mixture distributions derived from the variational posteriors, but any improvements are not very conclusive. The problems with simple importance sampling have motivated attempts at combining variational methods with more sophisticated MCMC methods, but to date there have been few successful implementations, and this is an area of future work.

We showed in chapter 2 that the variational Bayesian EM algorithm is a generalisation of the EM algorithm for ML/MAP optimisation — the standard EM algorithm is recovered by restricting the form of the variational posterior distribution over parameters to a delta function, or a point-estimate. There is also the interesting observation that the VB approximation reduces to the BIC approximation in the limit of an infinitely large data set, for which we provided a brief proof in the case of CE models. However, we have also found intriguing connections between the VB lower bound and Cheeseman-Stutz approximations to the marginal likelihood. In particular we proved with theorem 2.3 that the CS criterion is a strict lower bound on the marginal likelihood for arbitrary models (not just those in the CE family), which was a previously unrecognised fact (although Minka (2001b) makes this observation in a mixture modelling context). We then built on this theorem to show with corollary 2.5 that there is a construction for obtaining a VB approximation which *always* results in *at least as tight a bound* as the CS criterion. This is a very interesting and useful result because it means that all existing implementations using CS approximations can now be made more faithful to the exact marginal likelihood by overlaying a variational Bayesian approximation. This is only a very recent discovery, and as a result has not yet been exploited to the full.

We saw superior performance of the variational Bayesian lower bound over the Cheeseman-Stutz and BIC criteria in chapter 6, where the task was finding the particular structure (out of a small class of structures) that gave rise to an observed data set, via the marginal likelihood. This was despite not making use of the aforementioned construction derived in corollary 2.5 (which we were not aware of when carrying out the chapter's experiments). In these experiments we found that VB outperformed both BIC and CS approximations, and also tended to provide more reliable results than the sampling gold standard, annealed importance sampling. Not only does the VB approximation provide a bound on the marginal likelihood (which in the experiments often showed AIS estimates to be 'invalid'), but it also arrives at this bound in a fraction (about

1%) of the time of the sampling approach. Moreover the VB approximation does not require the tuning of proposal distributions, annealing schedules, nor does it require extensive knowledge of the model domain to produce a reliable algorithm. We presented a number of extensions to the AIS algorithm, including a more general algorithm for computing marginal likelihoods which uses estimates based on more than one sample at each temperature (see algorithm 6.1). In the near future we hope to prove whether estimates using this algorithm are biased or not (personal communication with R. Neal).

To conclude, I hope that this thesis has provided an accessible and coherent account of the widely applicable variational Bayesian approximation. We have derived variational Bayesian algorithms for a variety of statistical models and provided the tools with which new models can be tackled, especially with a view to building software for automated algorithm derivation. This should throw open the doors to Bayesian learning in a host of models other than those investigated here. There are many directions for this research to be taken in and much work left to be done. The hope is that the experimental findings and insights documented in these chapters will stimulate and guide future research on variational Bayes.

Appendix A

Conjugate Exponential family examples

The following two tables present information for a variety of exponential family distributions, and include entropies, KL divergences, and commonly required moments. Where used, tilde symbols (e.g. $\tilde{\boldsymbol{\theta}}$), denote the parameters of a different distribution of the same form. Therefore $\mathrm{KL}(\tilde{\boldsymbol{\theta}}||\boldsymbol{\theta})$ is shorthand for the KL divergence between the distribution with parameter $\tilde{\boldsymbol{\theta}}$ and the distribution with parameter $\boldsymbol{\theta}$ (averaging with respect to the first distribution that is specified). The remainder of the notation should be self-explanatory.

Distribution	Notation & Parameters	Density function	Moments, entropy, KL-divergence, etc.
Exponential Family	$\boldsymbol{\theta} \sim ExpFam(\eta, \boldsymbol{\nu})$ number η and value $\boldsymbol{\nu}$ of pseudo-observations	$p(\boldsymbol{\theta} \mid \eta, \boldsymbol{\nu}) = \frac{1}{Z_{\eta\boldsymbol{\nu}}} g(\boldsymbol{\theta})^{\eta} e^{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top}\boldsymbol{\nu}}$	$H_{\boldsymbol{\theta}} = \ln Z_{\eta\boldsymbol{\nu}} - \eta\langle \ln g(\boldsymbol{\theta})\rangle - \boldsymbol{\nu}^{\top}\langle \boldsymbol{\phi}(\boldsymbol{\theta})\rangle$
Uniform	$\theta \sim U(a, b)$ boundaries a,b with $b > a$	$p(\theta \mid a, b) = \frac{1}{b-a}, \theta \in [a, b]$	$H_{\theta} = \ln(b-a)$ $\langle\theta\rangle = \frac{a+b}{2}, \langle\theta^2\rangle - \langle\theta\rangle^2 = \frac{(b-a)^2}{12}$
Laplace	$\theta \sim Laplace(\mu, \lambda)$ μ mean λ decay scale	$p(\theta \mid \mu, \lambda) = \frac{1}{2\lambda} e^{-\frac{\lvert\theta-\mu\rvert}{\lambda}}$ $\lambda > 0$	$H_{\theta} = 1 + \ln(2\lambda)$
Multivariate normal (Gaussian)	$\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \Sigma)$ $\boldsymbol{\mu}$ mean vector Σ covariance	$p(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \Sigma) = (2\pi)^{-d/2} \lvert\Sigma\rvert^{-1/2} e^{-\frac{1}{2}\mathrm{tr}\left[\Sigma^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})(\boldsymbol{\theta}-\boldsymbol{\mu})^{\top}\right]}$	$H_{\boldsymbol{\theta}} = \frac{d}{2}(\ln 2\pi e) + \frac{1}{2}\ln\lvert\Sigma\rvert$ $\mathrm{KL}(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma} \mid\mid \boldsymbol{\mu}, \Sigma) = -\frac{1}{2}\left(\ln\left\lvert\tilde{\Sigma}\Sigma^{-1}\right\rvert\right.$ $\left. +\mathrm{tr}\left[I - \left[\tilde{\Sigma} + (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})^{\top}\right]\Sigma^{-1}\right]\ln e\right)$ $\langle\boldsymbol{\theta}\rangle = \boldsymbol{\mu}$ $\langle\boldsymbol{\theta}\boldsymbol{\theta}^{\top}\rangle = \Sigma$ $K_{\boldsymbol{\theta}} = \frac{\langle\boldsymbol{\theta}^4\rangle}{\langle\boldsymbol{\theta}^2\rangle^2} - 3 = 0$ (relative kurtosis)
Gamma	$\tau \sim Ga(\alpha, \beta)$ shape $\alpha > 0$ inv. scale $\beta > 0$	$p(\tau \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \tau^{\alpha-1} e^{-\beta\tau}$	$H_{\tau} = \ln\Gamma(\alpha) - \ln\beta + (1-\alpha)\psi(\alpha) + \alpha$ $\langle\tau^n\rangle = \frac{\Gamma(\alpha+n)}{\beta^n\Gamma(\alpha)}$ $\langle(\ln\tau)^n\rangle = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\frac{\partial^n}{\partial\alpha^n}\left(\frac{\Gamma(\alpha)}{\beta^{\alpha}}\right)$ $\langle\tau\rangle = \alpha/\beta$ $\langle\tau^2\rangle - \langle\tau\rangle^2 = \alpha/\beta^2$ $\langle\ln\tau\rangle = \psi(\alpha) - \ln\beta$ $\mathrm{KL}(\tilde{\alpha}, \tilde{\beta} \mid\mid \alpha, \beta) = \tilde{\alpha}\ln\tilde{\beta} - \alpha\ln\beta - \ln\frac{\Gamma(\tilde{\alpha})}{\Gamma(\alpha)}$ $+(\tilde{\alpha} - \alpha)(\psi(\tilde{\alpha}) - \ln\tilde{\beta}) - \tilde{\alpha}(1 - \frac{\beta}{\tilde{\beta}})$

Distribution	Notation & Parameters	Density function	Moments, entropy, KL-divergence, etc.
Wishart	$W \sim Wishart_\nu(S)$ deg. of freedom ν precision matrix S	$p(W \mid \nu, S) = \frac{1}{Z_{\nu S}} \lvert W\rvert^{(\nu-k-1)/2} e^{-\frac{1}{2}\mathrm{tr}\left[S^{-1}W\right]}$ $Z_{\nu S} = 2^{\nu k/2}\pi^{k(k-1)/4} \lvert S\rvert^{\nu/2} \prod_{i=1}^{k} \Gamma\left(\frac{\nu+1-i}{2}\right)$	$H_W = \ln Z_{\nu S} - \frac{\nu-k-1}{2}\langle \ln \lvert W\rvert\rangle + \frac{1}{2}\nu k$ $\langle W\rangle = \nu S$ $\langle \ln \lvert W\rvert\rangle = \sum_{i=1}^{k} \psi\left(\frac{\nu+1-i}{2}\right) + k\ln 2 + \ln \lvert S\rvert$ $\mathrm{KL}(\tilde{\nu}, \tilde{S}\|\nu, S) = \ln \frac{Z_{\nu S}}{Z_{\tilde{\nu}\tilde{S}}}$ $+\frac{\tilde{\nu}-\nu}{2}\langle \ln \lvert W\rvert\rangle_{\tilde{Q}} + \frac{1}{2}\tilde{\nu}\,\mathrm{tr}\left[S^{-1}\tilde{S} - \mathrm{I}\right]$
Inverse-Wishart	$W \sim Inv\text{–}Wishart_\nu(S^{-1})$ deg. of freedom ν covariance matrix S	$p(W \mid \nu, S^{-1}) = \frac{1}{Z} \lvert W\rvert^{-(\nu+k+1)/2} e^{-\frac{1}{2}\mathrm{tr}\left[SW^{-1}\right]}$ $Z = 2^{\nu k/2}\pi^{k(k-1)/4} \prod_{i=1}^{k} \Gamma\left(\frac{\nu+1-i}{2}\right) \times \lvert S\rvert^{-\nu/2}$	$\langle W\rangle = (\nu-k-1)^{-1} S$
Student-t (1)	$\theta \sim t_\nu(\mu, \sigma^2)$ deg. of freedom $\nu > 0$ mean μ; scale $\sigma > 0$	$p(\theta \mid \nu, \mu, \sigma^2) = \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\nu\pi\sigma}} \left(1 + \frac{1}{\nu}\left(\frac{\theta-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$	$\langle\theta\rangle = \mu$, for $\nu > 1$ $\langle\theta^2\rangle - \langle\theta\rangle^2 = \frac{\nu}{\nu-2}\sigma^2$, for $\nu > 2$
Student-t (2)	$\theta \sim t(\mu, \alpha, \beta)$ shape $\alpha > 0$; mean μ scale^2 $\beta > 0$	$p(\theta \mid \mu, \alpha, \beta) = \frac{\Gamma(\alpha+1/2)}{\Gamma(\alpha)\sqrt{2\pi\beta}} \left(1 + \frac{(\theta-\mu)^2}{2\beta}\right)^{-(\alpha+1/2)}$	$H_\theta = \left[\psi(\alpha+\frac{1}{2}) - \psi(\alpha)\right](\alpha+\frac{1}{2})$ $+\ln\sqrt{2\beta}B(\frac{1}{2}, \alpha)$ $K_\theta = \frac{3}{\alpha-2}$ (relative to Gaussian) equiv. $\alpha \to \frac{\nu}{2}; \beta \to \frac{\nu}{2}\sigma^2$
Multivariate Student-t	$\boldsymbol{\theta} \sim t_\nu(\boldsymbol{\mu}, \Sigma)$ deg. of freedom $\nu > 0$ mean $\boldsymbol{\mu}$; scale^2 matrix Σ	$p(\boldsymbol{\theta} \mid \nu, \boldsymbol{\mu}, \Sigma) = \frac{1}{Z}\left(1 + \frac{1}{\nu}\mathrm{tr}\left[\Sigma^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})(\boldsymbol{\theta}-\boldsymbol{\mu})^\top\right]\right)^{-(\nu+d)/2}$ $Z = \frac{\Gamma((\nu+d)/2)}{\Gamma(\nu/2)(\nu\pi)^{d/2}} \lvert\Sigma\rvert^{-1/2}$	$\langle\boldsymbol{\theta}\rangle = \boldsymbol{\mu}$, for $\nu > 1$ $\langle\boldsymbol{\theta}\boldsymbol{\theta}^\top\rangle - \langle\boldsymbol{\theta}\rangle\langle\boldsymbol{\theta}\rangle^\top = \frac{\nu}{\nu-2}\Sigma$, for $\nu > 2$
Beta	$\theta \sim Beta(\alpha, \beta)$ prior sample sizes $\alpha > 0, \beta > 0$	$p(\theta \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$ $\theta \in [0, 1]$	See Dirichlet with $k = 2$
Dirichlet	$\boldsymbol{\pi} \sim Dir(\boldsymbol{\alpha})$ prior sample sizes $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_k\}$ $\alpha_j > 0; \alpha_0 = \sum_{j=1}^{k} \alpha_j$	$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_k)}\pi_1^{\alpha_1-1}\cdots\pi_k^{\alpha_k-1}$ $\pi_1, \ldots, \pi_k \geq 0; \sum_{j=1}^{k} \pi_j = 1$	$\langle\boldsymbol{\pi}\rangle = \boldsymbol{\alpha}/\alpha_0$ $\langle\boldsymbol{\pi}\boldsymbol{\pi}^\top\rangle - \langle\boldsymbol{\pi}\rangle\langle\boldsymbol{\pi}\rangle^\top = \frac{\alpha_0\mathrm{diag}(\boldsymbol{\alpha}) - \boldsymbol{\alpha}\boldsymbol{\alpha}^\top}{\alpha_0^2(\alpha_0+1)}$ $\langle\ln\pi_j\rangle = \psi(\alpha_j) - \psi(\alpha_0)$ $\mathrm{KL}(\tilde{\boldsymbol{\alpha}}\|\boldsymbol{\alpha}) = \ln\frac{\Gamma(\tilde{\alpha}_0)}{\Gamma(\alpha_0)} - \sum_{j=1}^{k}\Big[\ln\frac{\Gamma(\tilde{\alpha}_j)}{\Gamma(\alpha_j)}$ $-(\tilde{\alpha}_j - \alpha_j)\left(\psi(\tilde{\alpha}_j) - \psi(\tilde{\alpha}_0)\right)\Big]$

Appendix B

Useful results from matrix theory

B.1 Schur complements and inverting partitioned matrices

In chapter 5 on Linear Dynamical Systems, we needed to obtain the cross-covariance of states across two time steps from the precision matrix, calculated from combining the forward and backward passes over the sequences. This precision is based on the joint distribution of the states, yet we are interested only in the cross-covariance between states. If A is of 2×2 block form, we can use Schur complements to obtain the following results for the partitioned inverse of A, and its determinant in terms of its blocks' constituents.

The partitioned inverse is given by

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} = \begin{pmatrix} F_{11}^{-1} & -A_{11}^{-1}A_{12}F_{22}^{-1} \\ -F_{22}^{-1}A_{21}A_{11}^{-1} & F_{22}^{-1} \end{pmatrix} \tag{B.1}$$

$$= \begin{pmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}F_{22}^{-1}A_{21}A_{11}^{-1} & -F_{11}^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}F_{11}^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}F_{11}^{-1}A_{12}A_{22}^{-1} \end{pmatrix} \tag{B.2}$$

and the determinant by

$$\begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} = |A_{22}| \cdot |F_{11}| = |A_{11}| \cdot |F_{22}| \,, \tag{B.3}$$

where

$$F_{11} = A_{11} - A_{12}A_{22}^{-1}A_{21} \tag{B.4}$$

$$F_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12} \,. \tag{B.5}$$

Notice that inverses of A_{12} or A_{21} do not appear in these results. There are other Schur complements that are defined in terms of the inverses of these 'off-diagonal' terms, but they are not needed for our purposes, and indeed if the states involved have different dimensionalities or are independent, then these off-diagonal quantities are not invertible.

B.2 The matrix inversion lemma

Here we present a sketch proof of the matrix inversion lemma, included for reference only. In the derivation that follows, it becomes quite clear that there is no obvious way of carrying the sort of expectations encountered in chapter 5 through the matrix inversion process (see comments following equation (5.105)).

The matrix inversion result is most useful when A is a large diagonal matrix and B has few columns (equivalently D has few rows).

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}\,. \tag{B.6}$$

To derive this lemma we use the Taylor series expansion of the matrix inverse

$$(A+M)^{-1} = A^{-1}(I+MA^{-1})^{-1} = A^{-1}\sum_{i=0}^{\infty}(-1)^i(MA^{-1})^i\,, \tag{B.7}$$

where the series is only well-defined when the spectral radius of MA^{-1} is less than unity. We can easily check that this series is indeed the inverse by directly multiplying by $(A+M)$, yielding the identity,

$$\begin{aligned}(A+M)A^{-1}\sum_{i=0}^{\infty}(-1)^i(MA^{-1})^i = {}& AA^{-1}\left[I - MA^{-1} + (MA^{-1})^2 - (MA^{-1})^3 + \ldots\right] \\ & + MA^{-1}\left[\quad I \quad - \quad MA^{-1} \quad + (MA^{-1})^2 - \ldots\right] \qquad \text{(B.8)}\\ = {}& I\,. \qquad \text{(B.9)}\end{aligned}$$

In the series expansion we find an embedded expansion, which forms the inverse matrix term on the right hand side, as follows

$$
\begin{aligned}
(A+BCD)^{-1} &= A^{-1}(I+BCDA^{-1})^{-1} && \text{(B.10)}\\
&= A^{-1}\sum_{i=0}^{\infty}(-1)^i(BCDA^{-1})^i && \text{(B.11)}\\
&= A^{-1}\left(I+\sum_{i=1}^{\infty}(-1)^i(BCDA^{-1})^i\right) && \text{(B.12)}\\
&= A^{-1}\left(I-BC\left[\sum_{i=0}^{\infty}(-1)^i(DA^{-1}BC)^i\right]DA^{-1}\right) && \text{(B.13)}\\
&= A^{-1}\left(I-BC(I+DA^{-1}BC)^{-1}DA^{-1}\right) && \text{(B.14)}\\
&= A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}\,. && \text{(B.15)}
\end{aligned}
$$

In the above equations, we assume that the spectral radii of $BCDA^{-1}$ (B.11) and $DA^{-1}BC$ (B.13) are less than one for the Taylor series to be convergent. Aside from these constraints, we can post-hoc check the result simply by showing that multiplication of the expression by its proposed inverse does in fact yield the identity.

Appendix C

Miscellaneous results

C.1 Computing the digamma function

The digamma function is defined as

$$\psi(x) = \frac{d}{dx} \ln \Gamma(x) \, , \tag{C.1}$$

where $\Gamma(x)$ is the Gamma function given by

$$\Gamma(x) = \int_0^\infty d\tau \; \tau^{x-1} e^{-\tau} \, . \tag{C.2}$$

In the implementations of the models discussed in this thesis, the following expansion is used to compute the $\psi(x)$ for large positive arguments

$$\psi(x) \simeq \ln x - \frac{1}{2x} - \frac{1}{12x^2} + \frac{1}{120x^4} - \frac{1}{252x^6} + \frac{1}{240x^8} + \ldots \, . \tag{C.3}$$

If we have small arguments, then we would expect this expansion to be inaccurate if we only used a finite number of terms. However, we can make use of a recursion of the digamma function to ensure that we always pass this expansion large arguments. The Gamma function has the well known recursion:

$$x! = \Gamma(x+1) = x\Gamma(x) = x(x-1)! \, , \tag{C.4}$$

from which the recursion for the digamma function readily follows:

$$\psi(x+1) = \frac{1}{x} + \psi(x) \, . \tag{C.5}$$

In our experiments we used an expansion (C.3) containing terms as far as $\mathcal{O}(1/x^{14})$, and used the recursion to evaluate this only for arguments of $\psi(x)$ greater than 6. This is more than enough precision.

C.2 Multivariate gamma hyperparameter optimisation

In hierarchical models such as the VB LDS model of chapter 5, there is often a gamma hyperprior over the noise precisions on each dimension of the data. On taking derivatives of the lower bound with respect to the shape a and inverse scale b of this hyperprior distribution, we obtain fixed point equations of this form:

$$\psi(a) = \ln b + \frac{1}{p}\sum_{s=1}^{p}\overline{\ln \rho_s}\,, \qquad \frac{1}{b} = \frac{1}{pa}\sum_{s=1}^{p}\overline{\rho_s} \tag{C.6}$$

where the notation $\overline{\ln \rho_s}$ and $\overline{\rho_s}$ is used to denote the expectations of quantities under the variational posterior distribution (see section 5.3.6 for details). We can rewrite this as:

$$\psi(a) = \ln b + c\,, \qquad \frac{1}{b} = \frac{d}{a}\,, \tag{C.7}$$

where

$$c = \frac{1}{p}\sum_{s=1}^{p}\overline{\ln \rho_s}\,, \qquad \text{and} \quad d = \frac{1}{p}\sum_{s=1}^{p}\overline{\rho_s}\,. \tag{C.8}$$

Equation (C.7) is the generic fixed point equation commonly arrived at when finding the variational parameters a and b which minimise the KL divergence on a gamma distribution.

The fixed point for a is found at the solution of

$$\psi(a) = \ln a - \ln d + c\,, \tag{C.9}$$

which can be arrived at using the Newton-Raphson iterations:

$$a_{\text{new}} \leftarrow a\left[1 - \frac{\psi(a) - \ln a + \ln d - c}{a\psi'(a) - 1}\right]\,, \tag{C.10}$$

where $\psi'(x)$ is the first derivative of the digamma function. Unfortunately, this update cannot ensure that a remains positive for the next iteration (the gamma distribution is only defined for $a > 0$) because the gradient information is taken locally.

There are two immediate ways to solve this. First if a should become negative during the Newton-Raphson iterations, reset it to a minimum value. This is a fairly crude solution. Alter-

natively, we can solve a different fixed point equation for a' where $a = \exp(a')$, resulting in the multiplicative updates:

$$a_{\text{new}} \leftarrow a \exp\left[-\frac{\psi(a) - \ln a + \ln d - c}{a\psi'(a) - 1}\right] . \tag{C.11}$$

This update has the same fixed point but exhibits different (well-behaved) dynamics to reach it. Note that equation C.10 is simply the first two terms in the Taylor series of the exponential function in the above equation.

Once the fixed point a^* is reached, the corresponding b^* is found simply from

$$b^* = \frac{a^*}{d} . \tag{C.12}$$

C.3 Marginal KL divergence of gamma-Gaussian variables

This note is intended to aid the reader in computing the lower bound appearing in equation (5.147) for variational Bayesian state-space models. Terms such as the KL divergence between two Gaussian or two gamma distributions are straightforward to compute and are given in appendix A. However there are more complicated terms involving expectations of KL divergences for joint Gaussian and gamma variables, for which we give results here.

Suppose we have two variables of interest, $\mathbf{a}$ and $\mathbf{b}$, that are jointly Gaussian distributed. To be more precise let the two variables be linearly dependent on each other in this sense:

$$q(\mathbf{a}, \mathbf{b}) = q(\mathbf{b})q(\mathbf{a} \,|\, \mathbf{b}) = \mathrm{N}(\mathbf{b} \,|\, \boldsymbol{\mu}_b, \Sigma_b) \cdot \mathrm{N}(\mathbf{a} \,|\, \boldsymbol{\mu}_a, \Sigma_a) \tag{C.13}$$

$$\text{where} \quad \boldsymbol{\mu}_a = \mathbf{y} - G\mathbf{b} . \tag{C.14}$$

Let us also introduce a prior distribution $p(\mathbf{a} \,|\, \mathbf{b})$ in this way:

$$p(\mathbf{a} \,|\, \mathbf{b}) = \mathrm{N}(\mathbf{a} \,|\, \tilde{\boldsymbol{\mu}}_a, \tilde{\Sigma}_a) \tag{C.15}$$

where neither parameter $\tilde{\boldsymbol{\mu}}_a$ nor $\tilde{\Sigma}_a$ are functions of b.

The first result is the KL divergence between two Gaussian distributions (given in appendix A)

$$\mathrm{KL}\left[q(\mathbf{a} \,|\, \mathbf{b}) \,\|\, p(\mathbf{a} \,|\, \mathbf{b})\right] = \int d\mathbf{a}\, q(\mathbf{a} \,|\, \mathbf{b}) \ln \frac{q(\mathbf{a} \,|\, \mathbf{b})}{p(\mathbf{a} \,|\, \mathbf{b})} \tag{C.16}$$

$$= -\frac{1}{2} \ln \left|\tilde{\Sigma}_a^{-1}\Sigma_a\right| + \frac{1}{2} \mathrm{tr}\, \tilde{\Sigma}_a^{-1} \left[\Sigma_a - \tilde{\Sigma}_a + (\boldsymbol{\mu}_a - \tilde{\boldsymbol{\mu}}_a)(\boldsymbol{\mu}_a - \tilde{\boldsymbol{\mu}}_a)^\top\right] . \tag{C.17}$$

Note that this divergence is written w.r.t. the $q(\mathbf{a}\,|\,\mathbf{b})$ distribution. The dependence on $\mathbf{b}$ is not important here, but will be required later. The important part to note is that it obviously depends on each Gaussian's covariance, but also on the Mahalanobis distance between the means as measured w.r.t. the non-averaging distribution.

Consider now the KL divergence between the full joint posterior and full joint prior:

$$\mathrm{KL}\left[q(\mathbf{a},\mathbf{b})\,\|\,p(\mathbf{a},\mathbf{b})\right] = \int d\mathbf{a}\,d\mathbf{b}\;q(\mathbf{a},\mathbf{b})\ln\frac{q(\mathbf{a},\mathbf{b})}{p(\mathbf{a},\mathbf{b})} \tag{C.18}$$

$$= \int d\mathbf{b}\;q(\mathbf{b})\int d\mathbf{a}\;q(\mathbf{a}\,|\,\mathbf{b})\ln\frac{q(\mathbf{a}\,|\,\mathbf{b})}{p(\mathbf{a}\,|\,\mathbf{b})} + \int d\mathbf{b}\;q(\mathbf{b})\ln\frac{q(\mathbf{b})}{p(\mathbf{b})}\;. \tag{C.19}$$

The last term in this is equation is simply the KL divergence between two Gaussians, which is straightforward, but the first term is the *expected* KL divergence between the conditional distributions, where the expectation is taken w.r.t. the marginal distribution $q(\mathbf{b})$. After some simple manipulation, this first term is given by

$$\left\langle \mathrm{KL}\left[q(\mathbf{a}\,|\,\mathbf{b})\,\|\,p(\mathbf{a}\,|\,\mathbf{b})\right]\right\rangle_{q(\mathbf{b})} = \int d\mathbf{b}\;q(\mathbf{b})\int d\mathbf{a}\;q(\mathbf{a}\,|\,\mathbf{b})\ln\frac{q(\mathbf{a}\,|\,\mathbf{b})}{p(\mathbf{a}\,|\,\mathbf{b})} \tag{C.20}$$

$$= -\frac{1}{2}\ln\left|\tilde{\Sigma}_a^{-1}\Sigma_a\right| + \frac{1}{2}\mathrm{tr}\;\tilde{\Sigma}_a^{-1}\Big[\Sigma_a - \tilde{\Sigma_a} + G\Sigma_b G^\top + (\mathbf{y} - G\boldsymbol{\mu}_b - \tilde{\boldsymbol{\mu}}_a)\,(\mathbf{y} - G\boldsymbol{\mu}_b - \tilde{\boldsymbol{\mu}}_a)^\top\Big]\;. \tag{C.21}$$

Let us now suppose that the covariance terms for the prior $\tilde{\Sigma}$ and posterior Σ_a have the same multiplicative dependence on another variable ρ^{-1}. This is the case in the variational state-space model of chapter 5 where, for example, the uncertainty in the entries for the output matrix C should be related to the setting of the output noise $\boldsymbol{\rho}$ (see equation (5.44) for example). In equation (C.17) it is clear that if both covariances are dependent on the same ρ^{-1}, then the KL divergence will not be a function of ρ^{-1} *provided* that the means of both distributions are the same. If they are different however, then there is a residual dependence on ρ^{-1} due to the $\tilde{\Sigma}_a^{-1}$ term from the non-averaging distribution $p(\mathbf{a}\,|\,\mathbf{b})$. This is important as there will usually be distributions over this ρ variable of the form

$$q(\rho) = \mathrm{Ga}(\rho\,|\,e_\rho, f_\rho) \tag{C.22}$$

with e and f shape and precision parameters of a gamma distribution. The most complicated term to compute is the penultimate term in (5.147), which is

$$\Big\langle\left\langle \mathrm{KL}\left[q(\mathbf{a}\,|\,\mathbf{b},\rho)\,\|\,p(\mathbf{a}\,|\,\mathbf{b},\rho)\right]\right\rangle_{q(\mathbf{b})}\Big\rangle_{q(\rho)} = \int d\rho\;q(\rho)\int d\mathbf{b}\;q(\mathbf{b}\,|\,\rho)\int d\mathbf{a}\;q(\mathbf{a}\,|\,\mathbf{b},\rho)\ln\frac{q(\mathbf{a}\,|\,\mathbf{b},\rho)}{p(\mathbf{a}\,|\,\mathbf{b},\rho)}\;. \tag{C.23}$$

In the variational Bayesian state-space model, the prior and posterior for the parameters of the output matrix C (and D for that matter) are defined in terms of the same noise precision variable

ρ. This means that all terms but the last one in equation (C.21) are not functions of ρ and pass through the expectation in (C.23) untouched. The final term has a dependence on ρ, but on taking expectations w.r.t. $q(\rho)$ this simply yields a multiplicative factor of $\langle\rho\rangle_{q(\rho)}$. It is straightforward to extend this to the case of data with several dimensions, in which case the lower bound is a sum over all p dimensions of similar quantities.

Bibliography

D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985. 2.4.1

N. J. Adams, A. J. Storkey, Z. Ghahramani, and C. K. I. Williams. MFDTs: Mean field dynamic trees. In *Proc. 15th Int. Conf. on Pattern Recognition*, 2000. 2.5.1

S. L. Adler. Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D*, 23:2901–2904, 1981. 1.3.6

H. Attias. Independent Factor Analysis. *Neural Computation*, 11:803–851, 1999a. 2.4.1

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*, 1999b. 2.6.1

H. Attias. A variational Bayesian framework for graphical models. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, Cambridge, MA, 2000. MIT Press. 2.3.2, 2.4.3

L. Bahl and F. Jelinek. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. *IEEE Transactions on Information Theory*, 21(4):404–411, 1975. 3.1

D. Barber and C. M. Bishop. Ensemble learning for multi-layer networks. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 395–401, Cambridge, MA, 1998. MIT Press. 2.3.2

D. Barber and P. Sollich. Gaussian fields for approximate inference in layered sigmoid belief networks. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, Cambridge, MA, 2000. MIT Press. 2.3.2

A. I. Barvinok. Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor. *Random Structures and Algorithms*, 14(1):29–61, 1999. 1.3.3

L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, 1966. 3.1

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41:164–171, 1970. 2.2.2, 3.1, 3.2

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press. 3.6, 7.1

A. J. Bell and T. J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. 2.4.1

J. M. Bernardo and F. J. Giron. A Bayesian analysis of simple mixture problems. In J. M. Bernardo, M. H. Degroot, A. F. Smith, and D. V. Lindley, editors, *Bayesian Statistics 3*, pages 67–78. Clarendon Press, 1988. 2.3.2

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons, Inc., New York, 1994. 1.2.2, 2.6.1

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. 2.4.1

C. M. Bishop. Variational PCA. In *Proc. Ninth Int. Conf. on Artificial Neural Networks. ICANN*, 1999. 2.3.2, 4.2.2

C. M. Bishop, N. D. Lawrence, T. S. Jaakkola, and M. I. Jordan. Approximating posterior distributions in belief networks using mixtures. In *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1998. MIT Press. 2.3.2

C. M. Bishop, D. Spiegelhalter, and J. Winn. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems 15*, S. Becker and S. Thrun and K. Obermayer, 2003. 2.4.3, 7.1

X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, 1998. 2.3.2

S. Brooks. MCMC repository. Statistical Laboratory, University of Cambridge. Accessible on the world wide web at `http://www.statslab.cam.ac.uk/~mcmc`. 1.3.6

W. Buntine. Variational extensions to EM and multinomial PCA. In *ECML*, 2002. 2.4.3

G. Casella, K. L. Mengersen, C. P. Robert, and D. M. Titterington. Perfect slice samplers for mixtures of distributions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 64(4):777–790, 2000. 1.3.6

K. Chan, T. Lee, and T. J. Sejnowski. Variational learning of clusters of undercomplete nonsymmetric independent components. *Journal of Machine Learning Research*, 3:99–114, August 2002. 4.8

P. Cheeseman and J. Stutz. Bayesian classification (Autoclass): Theory and results. In U. M. Fayyad, G. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180, Menlo Park, CA, 1996. AAAI Press/MIT Press. 1.3.1, 1.3.5, 1.3.5, 2.6.2, 2.6.2, 6.3.3, 6.5.2

D. M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29(2–3):181–212, 1997. 2.6.2, 2.6.2

R. Choudrey and S. Roberts. Variational mixture of Bayesian independent component analysers. *Neural Computation*, 15(1), 2002. 4.8

P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994. 2.4.1

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, 1999. 1.1

R. T. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14 (1):1–13, 1946. 1.1

N. de Freitas, P. Højen-Sørensen, M. I. Jordan, and S. Russell. Variational MCMC. In J. S. Breese and D. Koller, editors, *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2001. 4.8

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39: 1–38, 1977. 2.2.2, 2.4.3

R. P. Feynman. *Statistical Mechanics: A Set of Lectures*. Perseus, Reading, MA, 1972. 2.2.1, 2.3.2

J. A. Fill. An interruptible algorithm for perfect sampling via Markov chains. *The Annals of Applied Probability*, 8(1):131–162, 1998. 1.3.6

E. Fokoué and D. M. Titterington. Mixtures of factor analysers. Bayesian estimation and inference by stochastic simulation. *Machine Learning*, 50(1):73–94, January 2003. 4.2.3

B. J. Frey, R. Patrascu, T. S. Jaakkola, and J. Moran. Sequentially fitting"inclusive" trees for inference in noisy-OR networks. In *Advances in Neural Information Processing Systems 13*, 2001. 2.3.2

N. Friedman. The Bayesian structural EM algorithm. In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98*, San Francisco, CA, 1998. Morgan Kaufmann Publishers. 6.4, 6.6

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000. 5.5

S. Früwirth-Schnatter. Bayesian model discrimination and Bayes factors for linear Gaussian state space models. *Journal of the Royal Statistical Society, Series B (Methodological)*, 57: 237–246, 1995. 5.6

D. Geiger, D. Heckerman, and C. Meek. Asymptotic model selection for directed networks with hidden variables. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, 1996. 6.5.2

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995. 2.4.1

A. Gelman and X. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185, 1998. 6.3.5

Z. Ghahramani. Factorial learning and the EM algorithm. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 617–624, Cambridge, MA, 1995. MIT Press. 2.2.3

Z. Ghahramani. An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001. 3.1

Z. Ghahramani and H. Attias. Online variational Bayesian learning, 2000. Slides from talk presented at NIPS 2000 workshop on Online Learning, available at `http://www.gatsby.ucl.ac.uk/~zoubin/papers/nips00w.ps`. 2.4.3

Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems 12*, Cambridge, MA, 2000. MIT Press. 2.3.2, 2.4.3, 4.1, 4.7

Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. MIT Press. 2.4.3, 2.4.3

Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 1996a. 5.2.2, 5.3.8

Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996b. 4.1.2

Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4), 2000. 2.2.3, 3.1

Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29: 245–273, 1997. 2.2.3, 2.2.3, 3.1

W. R. Gilks. Derivative-free adaptive rejection sampling for Gibbs sampling. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages 641–649. Clarendon Press, 1992. 1.3.6

W. R. Gilks, N. G. Best, and K. K. C. Tan. Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics*, 44:455–472, 1995. 1.3.6

W. R. Gilks, G. O. Roberts, and S. K. Sahu. Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association*, 93:1045–1054, 1998. 4.8

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41(2):337–348, 1992. 1.3.6, 1.3.6

A. G. Gray, B. Fischer, J. Schumann, and W. Buntine. Automatic derivation of statistical algorithms: The EM family and beyond. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003. 2.4.3, 7.1

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995. 1.3.6, 4.2.3, 4.3

M. Harvey and R. M. Neal. Inference for belief networks using coupling from the past. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 256–263. Morgan Kaufmann, 2000. 1.3.6

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 1.3.6, 6.3.5

D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06 [ftp://ftp.research.microsoft.com/pub/tr/TR-95-06.PS] , Microsoft Research, 1996. 1.1

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995. 6.2, 7.1

T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press. 7.1

G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Sixth ACM Conference on Computational Learning Theory, Santa Cruz*, 1993. 1.2.1, 2.3.2

G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, San Francisco, CA, 1994. Morgan Kaufmann. 2.2.3

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5), May 1994. 4.6

T. S. Jaakkola. Variational methods for inference and estimation in graphical models. Technical Report Ph.D. Thesis, Department of Brain and Cognitive Sciences, MIT, Cambridge, MA, 1997. 2.2.3, 2.3.2, 2.5.2, 7.1

T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 163–173. Kluwer, 1998. 2.3.2

T. S. Jaakkola and M. I. Jordan. Bayesian logistic regression: a variational approach. *Statistics and Computing*, 10:25–37, 2000. 2.3.2

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003. 1.1

W. H. Jefferys and J. O. Berger. Ockham's razor and Bayesian analysis. *American Scientist*, 80: 64–72, 1992. 1.2.1, 4.2

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007), 1946. 1.2.2

F. V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag, New York, 1996. 1.1

J. L. W. V. Jensen. Sur les fonctions convexes et les inegalités entre les valeurs moyennes. *Acta Mathematica*, 30:175–193, 1906. 2.2.1

M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *ACM Symposium on Theory of Computing*, pages 712–721, 2001. 1.3.3

M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999. 1.1

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods in graphical models. *Machine Learning*, 37:183–233, 1999. 2.3.2, 2.5.2

M. I. Jordan, Z. Ghahramani, and L. K. Saul. Hidden Markov decision trees. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. MIT Press. 3.1

M. I. Jordan and Y. Weiss. Graphical models: Probabilistic inference. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks, 2nd edition*. MIT Press, Cambridge, MA, 2002. 1.1.2

B. H. Juang and L. R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 1991. 3.1

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995. 1.2.1, 1.3.1, 1.3.2

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. 6.3.5

T. Kočka and N. L. Zhang. Dimension correction for hierarchical latent class models. In A. Darwich and N. Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 267–274. Morgan Kaufmann, 2002. 6.5.2

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 50(2):157–224, 1988. 1.1

N. D. Lawrence and M. Azzouzi. A variational Bayesian committee of neural networks. Submitted to *Neural Networks*, 1999. 2.3.2

N. D. Lawrence, C. M. Bishop, and M. I. Jordan. Mixture representations for inference and learning in Boltzmann machines. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 320–327, Madison, Wisconsin, 1998. 2.3.2

Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995. 4.6.2

M. A. R. Leisink and H. J. Kappen. A tighter bound for graphical models. *Neural Computation*, 13(9):2149–2170, 2001. 7.1

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992. 3.3, 4.2

D. J. C. MacKay. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995. 1.2.1, 1.3, 1.3.2, 5.2.2, 6.3.2

D. J. C. MacKay. Bayesian non-linear modelling for the 1993 energy prediction competition. In G. Heidbreder, editor, *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, pages 221–234, Dordrecht, 1996. Kluwer. 4.2.2

D. J. C. MacKay. Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge, 1997. 2.3.2, 2.4.3, 2.5.1, 3.1, 3.4, 3.4.2, 3.5.2, 7.2

D. J. C. MacKay. Choice of basis for Laplace approximation. *Machine Learning*, 33(1), 1998. 1.3.2, 3.3, 3.3, 6.3.1

D. J. C. MacKay. An introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999. 4.7.3

D. J. C. MacKay. A problem with variational free energy minimization, 2001. 3.6

D. J. C. MacKay and L. C. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1995. 3.3

N. Metropolis, A. W. Rosenbluth, M. N. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953. 1.3.6, 6.3.5

T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001a. 2.3.2, 7.1

T. P. Minka. Using lower bounds to approximate integrals, 2001b. 2.6.2, 7.2

T. P. Minka and J. Lafferty. Expectation-Propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2002. 2.3.2

J. W. Miskin. *Ensemble Learning for Independent Component Analysis*. PhD thesis, University of Cambridge, December 2000. 4.7, 4.7.3, 5.6

D. J. Murdoch and P. J. Green. Exact sampling from a continuous state space. *Scandinavian Journal of Statistics*, 25(3):483–502, 1998. 1.3.6

R. M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992. 1.3.6, 2.2.3

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993. 1.3.6, 6.3.5

R. M. Neal. *Bayesian Learning in Neural Networks*. Springer-Verlag, 1996. 1.2.1, 1.3.6, 3.6, 6.3.5, 7.1

R. M. Neal. Assessing relevance determination methods using DELVE. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 97–129. Springer-Verlag, 1998a. 4.2.2

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Department of Statistics, University of Toronto, 1998b. 3.6

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001. 1.3.6, 6.3.5, 6.3.5, 6.3.5

R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3), 2003. With discussion. 6.6

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–369. Kluwer Academic Publishers, 1998. 2.2.1, 2.3.2

A. O'Hagan. Monte Carlo is fundamentally unsound. *Statistician*, 36(2/3):247–249, 1987. Special Issue: Practical Bayesian Statistics. 1.3.6

A. O'Hagan. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3): 245–260, 1991. 1.3.6

G. Parisi. *Statistical Field Theory*. Addison Wesley, 1988. 2.3.2

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Francisco, CA, 1988. 1.1, 1.1.1, 2.5

J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 1&2(9):223–252, 1996. 1.3.6

L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE Acoustics, Speech & Signal Processing Magazine*, 3:4–16, 1986. 3.2

C. Rangel, D. L. Wild, F. Falciani, Z. Ghahramani, and A. Gaiba. Modeling biological responses using gene expression profiling and linear dynamical systems. In *To appear in Proceedings of the 2nd International Conference on Systems Biology*, Madison, WI, 2001. OmniPress. 5.5, 5.9

C. E. Rasmussen. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, Cambridge, MA, 2000. MIT Press. 3.6, 4.8, 7.1

C. E. Rasmussen and Z. Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. MIT Press. 1.2.1

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press. 7.1

C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2003. MIT Press. 1.3.6

H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372, 1963. 5.3.4

H. E. Rauch, F. Tung, and C. T. Striebel. On the maximum likelihood estimates for linear dynamic systems. Technical Report 6-90-63-62, Lockheed Missiles and Space Co., Palo Alto, California, June 1963. 5.3.2

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B (Methodological)*, 59(4): 731–758, 1997. 4.3

J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49:223–239 and 253–265, 1987. With discussion. 1.3.4

C. P. Robert, G. Celeux, and J. Diebolt. Bayesian estimation of hidden Markov chains: a stochastic implementation. *Statistics & Probability Letters*, 16(1):77–83, 1993. 3.3

S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modeling. *IEEE PAMI*, 20(11):1133–1142, 1998. 4.2.3

S. T. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999. 4.1.2, 5.2.1

J. Särelä, H. Valpola, R. Vigário, and E. Oja. Dynamical factor analysis of rhythmic magnetoencephalographic activity. In *Proceedings of the 3rd International Conference on Independent Component Analysis and Blind Signal Separation, ICA 2001*, pages 457–462, San Diego, California, USA, 2001. 5.6

M. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13(7): 1649–1681, 2001. 2.4.3

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996. 2.2.3

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978. 1.3.1, 1.3.4

R. Settimi and J. Q. Smith. On the geometry of Bayesian graphical models with hidden variables. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 472–479, Madison, Wisconsin, 1998. 6.5.2

R. D. Shachter. Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 480–487, Madison, Wisconsin, 1998. Morgan Kaufmann. 1.1.1

R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982. 5.2.2

P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9:227–269, 1997. 3.1

M. Stephens. *Bayesian methods for mixtures of normal distributions*. PhD thesis, Oxford University, 1997. 2.3.2

A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 11–18, San Francisco, CA, 1993. Morgan Kaufmann. 3.3

A. M. Storkey. Dynamic trees: A structured variational method giving efficient propagation rules. In *Proc. 16th Conf. on Uncertainty in Artificial Intelligence. UAI*, San Francisco, CA, 2000. Morgan Kaufmann Publishers. 2.5.1

A. Thomas, D. J. Spiegelhalter, and W. R. Gilks. BUGS: A program to perform Bayesian inference using Gibbs sampling. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages 837–842. Clarendon Press, 1992. 7.1

M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999. 4.1.2

G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in Monte Carlo free energy estimation: Umbrella sampling. *J. Comp. Phys.*, 23:187–199, 1977. 6.3.5

N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000. 4.3, 4.3.2, 4.5.3, 4.5, 4.6.2

H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002. 5.6

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. 3.3

M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2002. 7.1

C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49(3):240–265, 1987. With discussion. 1.3.4

S. Waterhouse, D. J. C. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1996. MIT Press. 2.3.2

M. Welling and Y. W. Teh. Belief Optimisation for binary networks: A stable alternative to loopy belief propagation. In *UAI 2001*, Seattle, Washington, 2001. 7.1

C. K. I. Williams and N. J. Adams. DTs: Dynamic trees. In *Advances in Neural Information Processing Systems 11*, Cambridge, MA, 1999. MIT Press. 2.5.1

C. K. I. Williams and G. E. Hinton. Mean field networks that learn to discriminate temporally distorted strings. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Connectionist Models: Proceedings of the 1990 Summer School*, pages 18–22. Morgan Kaufmann Publishers, San Francisco, CA, 1991. 2.2.3

J. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2001. MIT Press. 7.1

A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. Technical report, Smith-Kettlewell Eye Research Institute, 2001. 7.1