Variational Approximations in Bayesian Model Selection

Clare Anne McGrory

A Dissertation Submitted to the
Faculty of Information and Mathematical Sciences
at the University of Glasgow
for the degree of
Doctor of Philosophy

Department of Statistics
October, 2005

Abstract

The research presented in this thesis is on the topic of the Bayesian approach to statistical inference. In particular it focuses on the analysis of mixture models. Mixture models are a useful tool for representing complex data and are widely applied in many areas of statistics (see, for example, Titterington et al. (1985)). The representation of mixture models as missing data models is often useful as it makes more techniques of inference available to us. In addition, it allows us to introduce further dependencies within the mixture model hierarchy leading to the definition of the hidden Markov model and the hidden Markov random field model (see Titterington (1990)).

In the application of mixture models, as well as making inference about model parameters, the determination of the appropriate number of components is a primary objective. Both conventional cross-validation methods (see, for example, Ripley (1996)) and approaches involving Markov chain Monte Carlo methods (for example, Richardson and Green (1997)) have been employed in order to address this task. A main drawback of the cross-validation method is the computational intensity involved in fitting the various competing models using a training data-set and then comparing their results using a validation data-set. This computational expense renders this approach infeasible when there are more than a small number of model parameters to be considered. Markov chain Monte Carlo methods can be used to obtain a posterior distribution over the number of possible components. This approach has been popular in the statistical literature but these methods can be time-consuming and it can be difficult to assess convergence.

In this thesis we will consider how variational methods, that have become popular in some of the neural computing/machine learning literature, can be used to determine a suitable number of components for a mixture model and estimate model parameters. The variational technique is a deterministic approximate method, the practical implementation of which is computationally efficient

and comparitively straightforward.

The issue of model selection is important in Bayesian inference and, with increasingly complicated models, there is a need for more suitable selection criteria. We shall also explore how the Deviance Information Criterion (DIC), a selection criterion for Bayesian model comparison introduced by Spiegelhalter et al. (2002), can be extended to missing data models.

Chapter 1 introduces the main themes of the thesis. It provides an overview of variational methods for approximate Bayesian inference and describes the Deviance Information Criterion for Bayesian model selection.

Chapter 2 reviews the theory of finite mixture models and extends the variational approach and the Deviance Information Criterion to mixtures of Gaussians.

Chapter 3 examines the use of the variational approximation for general mixtures of exponential family models and considers the specific application to mixtures of Poisson and Exponential densities.

Chapter 4 describes how the variational approach can be used in the context of hidden Markov models. It also describes how the Deviance Information Criterion can be used for model selection with this class of model.

Chapter 5 explores the use of variational Bayes and the Deviance Information Criterion in hidden Markov random field analysis. In particular, the focus is on the application to image analysis.

Chapter 6 summarises the research presented in this thesis and suggests some possible avenues of future development.

The material in chapter 2 was presented at the ISBA 2004 world conference in Viña del Mar, Chile and was awarded a prize for best student presentation.

Acknowledgements

I would like to thank everyone who has helped me in completing my thesis in various ways.

Most importantly, I am deeply indebted to my supervisor, Prof. Mike Titterington, whose knowledge, experience and dedication made this research possible. I couldn't have hoped for more in a supervisor, he was always available and gave support whenever it was needed.

I have enjoyed my time in the Department of Statistics, from my days as an undergraduate to my postgraduate studies. It has always been a welcoming and friendly department and throughout the years many people have helped me along the way.

I would like to mention all of my fellow postgraduates who always offered a source of friendly advice as well as company for the many coffee breaks. I also would like to thank Jon Yong for his invaluable help with a C++ application.

On a more personal note, I want to thank my husband, Christopher, who is a constant companion and supportive of everything I do, and who can always make me laugh. Also deserving of thanks is my Aunt Rose who has provided unwavering support of all kinds throughout my life and my sister Catherine who is a great friend.

Lastly, I am very grateful to the EPSRC for funding this research.

Contents

Abstract i

Acknowledgements iii

List of Figures viii

List of Tables x

1 Introduction 1

1.1 The Variational Approach to Approximate Bayesian Inference . . 1

1.2 The Deviance Information Criterion (DIC) 5

1.2.1 Some Applications of the DIC 9

2 Bayesian Analysis of Finite Mixture Models 11

2.1 Finite Mixture Models 11

2.2 Variational Methods for Analysis of Finite Mixture Distributions in Machine Learning 16

2.3 Mixture Models Interpreted as Missing-Data Models 19

2.4 DIC for Mixture Models 19

2.5 Mixture of K Univariate Gaussian Distributions 20

2.5.1 The Variational Approach 20

2.6 Multivariate Case 24

2.6.1 The Variational Approach 25

2.7 Practical Implementation 28

2.8 Performance of the Method on Simulated and Real Data Sets . . 30

2.8.1 Results of Analysis of Simulated Data from Mixtures of Multivariate Gaussians 30

2.8.2 Results of Analysis of Simulated Data from Mixtures of Univariate Gaussians 37
2.8.3 Analysis of Real Data Sets 44
2.9 Conclusions 49

3 Application of the Variational Approach to Mixtures of Exponential Family Models **51**
3.1 The Poisson Distribution 54
3.2 The Exponential Distribution 56

4 Bayesian Analysis of Hidden Markov Models **59**
4.1 An Introduction to Hidden Markov Models 60
4.1.1 Origins and Applications of Hidden Markov Modelling . . 60
4.1.2 The Characteristics of a Hidden Markov Model 61
4.1.3 Problems of Inference for Hidden Markov Models and their Solutions 62
4.2 Inference for Hidden Markov Models with an Unknown Number of States 64
4.3 The Variational Approach to Inference about Hidden Markov Models with an Unknown Number of States 66
4.4 Model Specification 68
4.5 Practical Implementation 72
4.6 Performance of the Method on Simulated and Real Data Sets . . 74
4.6.1 Application to Simulated Examples 74
4.6.2 Application to Real Data Sets 83
4.7 Conclusions 97

5 Hidden Markov Random Fields and Image Analysis **99**
5.1 Introduction and Background 100
5.2 Markov Random Fields and the Hammersley-Clifford Theorem . . 103
5.2.1 Markov Random Fields 103
5.2.2 The Ising Model and the Potts Model 106
5.3 Estimating the Parameters of a Hidden Markov Random Field . . 108
5.4 Hidden Binary Markov Random Field 110
5.5 Hidden K-State Markov Random Field 113
5.5.1 Optimisation of $q_z(z)$ 116

5.5.2 Optimisation of $q_\beta(\beta)$ 118
5.5.3 Approximating the Expected Value of β 122
5.5.4 Obtaining the DIC and p_D Values 122
5.6 Practical Implementation 125
5.7 Simulating a Binary Image Using Gibbs Sampling 126
5.7.1 The Gibbs Sampler (or Alternating Conditional Sampling) for Sampling from a Posterior Distribution 126
5.7.2 Using a Gibbs Sampler to Sample from a Binary Markov Random Field 127
5.8 Results from the Analysis of Simulated Binary Images with Added White Gaussian Noise 128
5.8.1 Data generated from an Ising Model with $\beta = 0.45$ 129
5.8.2 Data generated from an Ising Model with $\beta = 0.6$ 136
5.9 Other Interesting Applications of Hidden Markov Random Fields 140
5.10 Conclusions 143

6 Conclusions and Possible Areas for Future Research 145

A 147
A.1 Reformulation of the DIC 147

B 148
B.1 'Monotonicity' of Variational Bayes 148

C 150
C.1 Finding the Hyperparameters of the Variational Posterior for a Mixture of Univariate Gaussian Distributions 150
C.2 Derivation of the Formulae for p_D and DIC for a Mixture of Univariate Gaussian Distributions 155

D 158
D.1 Finding the Hyperparameters of the Variational Posterior for a Mixture of Multivariate Gaussian Distributions 158
D.2 Derivation of the Formulae for p_D and DIC for a Mixture of Multivariate Gaussian Distributions 163

E **167**

E.1 Real Data Sets Used in Section 2.8.3 167

F **170**

F.1 The Poisson Distribution 170

F.2 The Exponential Distribution 172

G **175**

G.1 Finding the Form of the Variational Posterior for $q_z(z)$ in the case of a Hidden Markov Model with Gaussian Noise 175

G.2 The Forward Backward Algorithm 176

G.3 Obtaining Formulae for p_D and DIC in the case of a Hidden Markov Model with Gaussian Noise 179

H **182**

H.1 Finding the Forms of the Hyperparameters for the Gaussian Noise Model of a Hidden Markov Random Field 182

H.2 Derivation of Formulae for p_D and DIC for a Hidden Markov Random Field with Gaussian Noise 185

H.3 Mean-Field Approximation 190

List of Figures

2.1 Fitted model and true distribution for data-set 1 33
2.2 Fitted model and true distribution for data-set 2 33
2.3 Fitted model and true distribution for data-set 3 34
2.4 Simulated Values from N(0,1) 37
2.5 Results for Simulation from Mixture of 4 Normals 39
2.6 Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 0.5$ 40
2.7 Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 1$ 41
2.8 Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 2$ 41
2.9 Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 0.5$ 42
2.10 Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 1$ 42
2.11 Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 2$ 43
2.12 Galaxy Data 47
2.13 Acidity Data 47
2.14 Enzyme Data 48

4.1 Results from 800-observation data initialised with number of states ranging from 2 to 15 resulting in a 2-state model with parameter estimates given in Table 4.1 79
4.2 Results from 500-observation data initialised with number of states ranging from 4 to 6 resulting in a 4 state model with parameter estimates given in Table 4.4 82
4.3 Wind data starting with 7 initial states, terminating with 5 states 84

4.4 2-state model fitted to the wind data when the algorithm was initialised with 2 states 87
4.5 2-state model fitted to the geomagnetic data when the algorithm was initialised with 2 states 90
4.6 2-state model fitted to the daily returns data when the algorithm was initialised with 2 states 92
4.7 Density fitted to the wind data by mixture analysis 94
4.8 Density fitted to the geomagnetic data by mixture analysis 95
4.9 Density fitted to the daily returns data by mixture analysis 96

5.1 First-Order Neighbours 104
5.2 Cliques for a First-Order Neighbourhood System 104
5.3 True image used in simulated data-sets 1-3 129
5.4 Noisy Image : Simulated Data Set 1 130
5.5 Recovered Image : Simulated Data Set 1 130
5.6 Noisy Image : Simulated Data Set 2 131
5.7 Recovered Image : Simulated Data Set 2 132
5.8 Noisy Image : Simulated Data Set 3 133
5.9 Recovered Image : Simulated Data Set 3 134
5.10 True image used in simulated data-sets 4 and 5 137
5.11 Noisy Image : Simulated Data Set 4 138
5.12 Noisy Image : Simulated Data Set 5 139

List of Tables

2.1 True and Fitted Means and Covariances for Data-Set 1 32
2.2 True and Fitted Means and Covariances for Data-Set 2 32
2.3 True and Fitted Means and Covariances for Data-Set 3 32
2.4 DIC and (p_D) values for the three simulated data-sets 36
2.5 Results for Simulation from N(0,1) 37
2.6 Results for Simulation from Mixture of 4 Normals. 38
2.7 Results for Simulation from Mixtures of 2 Normals. 40
2.8 DIC and p_D Values for Simulated Mixtures of 2 Normals. 40
2.9 Number of components fitted and posterior estimates of means, variances and mixing weights for the three real data sets 46
2.10 DIC and p_D values for the three real data sets 46

4.1 Results for the 800-observation data set with a lower bound on a^* 77
4.2 Some other possible results obtained by using different initial weights for the q_{ij}'s 78
4.3 p_D and DIC values corresponding to other possible results given in Table 4.2 78
4.4 Results for the 500-observation data set with a lower bound on a^* 81
4.5 DIC and p_D values for the 500-observation data set corresponding to the solutions presented in Table 4.4 81
4.6 Results for the Wind Data starting algorithm with 7 states . . . 85
4.7 DIC and p_D values and number of states selected for the Wind Data with different initial numbers of components 85
4.8 Results for the Wind Data starting algorithm with 2 states . . . 86
4.9 DIC and p_D values and number of states selected for the geomagnetic data with different initial numbers of components 88

4.10 Results for the geomagnetic data, starting the algorithm with 2 states 89
4.11 DIC and p_D values and number of states selected for the daily returns data with different initial numbers of components 91
4.12 Results for the daily returns data starting algorithm with 2 states 91
4.13 Results from mixture analysis of the wind data starting algorithm with 2 components 94
4.14 Results from mixture analysis of the geomagnetic data starting algorithm with 2 components 95
4.15 Results from mixture analysis of the daily returns data starting algorithm with 15 components 96

Chapter 1

Introduction

The Bayesian approach to statistical inference allows us to incorporate prior knowledge into our analysis and makes use of all of the available data. The approach leads to a posterior distribution over the model framework and so it avoids overfitting of problems. It also provides a basis for model selection. Bayes' rule allows us to update our distribution over parameters from prior to posterior conditioning on the available observed data. The resulting posterior distribution is the key quantity in Bayesian inference. Unfortunately, except in the case of simple models, the associated posterior distributions and predictive densities are generally intractable. In this chapter we shall consider how this problem can be addressed by making use of variational approximations. We also describe the deviance information model selection criterion introduced by Spiegelhalter et al. (2002) and review some other popular selection criteria.

1.1 The Variational Approach to Approximate Bayesian Inference

The diversity of data available to researchers is continually growing due to many advances in computational power. Further to this, the invention of new methods of analysis leads to the creation of more complicated hierarchical models which can better represent the available data. Of course, these developments present new challenges for Bayesian inference. A full Bayesian analysis of our data requires us to specify a prior distribution over our parameters. This prior distribution may then be parameterised further with unknown parameters termed

hyperparameters and this results in a hierarchical structure. Many modern applications involve the use of hierarchical models; indeed, all of the models we will consider in this thesis have a hierarchical structure. The increasing complexity of statistical models and the intractability of posterior distributions is a main focus of much of the literature on Bayesian inference coming from the statistical as well as the machine learning community. In recent years, the most popular approach to this problem among statisticians has involved using the Markov chain Monte Carlo (MCMC) simulation-based approximations to the incalculable distributions (Gelfand and Smith (1990), Geyer (1992), Tierney (1994), Gilks et al. (1995), Green (1995), Gilks et al. (1996), Robert et al. (2000)). The use of MCMC simulation has also spread into the artificial intelligence literature (see Neal (1996), Doucet et al. (2001), Andrieu et al. (2003), for example). The attraction of MCMC methods comes from the fact that approximations are correct provided the model used for sampling does provide a good representation of the true model and the number of simulations carried out is suitably large. The drawback of these methods is that if the model is very complicated then the method can involve substantial computational time as storage of parameters is required throughout the sampling iterations. It can also be difficult to assess the convergence of the algorithm. Variational methods are a fast, deterministic alternative to MCMC methods and recently they have been gaining popularity in the machine learning literature (for instance, Jordan et al. (1999), Corduneanu and Bishop (2001) and Ueda and Ghahramani (2002)).

Variational methods take their name from their roots in the calculus of variations and they describe optimisation problems where the aim is to maximise or minimise an integral over unknown functions. Rustagi (1976) describes the use of variational methods in statistics and traces the origins of the calculus of variations approach back to Sir Isaac Newton who used this method to find the optimal shape for the hull of a ship. Since that time the methods have gone through many stages of development and found application to problems in various disciplines. They have been applied to statistical problems in areas such as operational research and optimal design (see Rustagi (1976)). As mentioned above, lately these methods have enjoyed popularity with the machine learning community for their application to statistical learning problems. This kind of variational Bayesian method aims to construct a tight lower bound on the data marginal likelihood and then seeks to optimise this bound using an iterative scheme. This form of

variational approximation is often referred to as Variational Bayes (or occasionally ensemble learning) in the literature. Modern statistical techniques are widely used by the machine learning community to the extent that statistics plays an integral role in computer science research. There is a growing awareness of the strong links between the two subjects. The connection is emphasised by compilations such as Kay and Titterington (1999), which collects work from researchers in the statistical and the artificial neural-network communities. From a statistical view point, review papers, such as those by Cheng and Titterington (1994) and Titterington (2004), and papers such as Ripley (1993,1994), describe the applications of statistics in computer science. In the machine learning literature, publications such as Bishop (1995) and Ripley (1996) highlight the statistical aspects of artificial intelligence. This expansion of interest in statistical theory and practice has led to new developments and computational techniques and has opened up new avenues of further research for both communities. The work in this thesis is at the interface between machine learning and statistics.

We now outline the basic theory of the variational approximation method for Bayesian inference. Suppose we have observed data y, that we assume a parametric model with parameters θ and that z denotes missing or unobserved values. Of interest is the posterior distribution of θ given y. The idea of the variational approximation is to approximate the joint conditional density of θ and z by a more amenable distribution $q(\theta, z)$, chosen to minimise the Kullback-Leibler (KL) divergence (Kullback and Leibler (1951)) between the approximating density $q(\theta, z)$ and the true joint conditional density, $p(\theta, z|y)$. The motivation for this is that we wish to obtain a tight lower bound on the marginal probability density $p(y)$ of y. We can find a lower bound on $p(y)$ as follows,

$$\begin{aligned}
\log p(y) &= \log \int \int p(y, z, \theta) d\theta dz && (1.1)\\
&= \log \int \sum_{\{z\}} q(\theta, z) \frac{p(y, z, \theta)}{q(\theta, z)} d\theta && (1.2)\\
&\geq \int \sum_{\{z\}} q(\theta, z) \log \frac{p(y, z, \theta)}{q(\theta, z)} d\theta, \quad \text{by Jensen's Inequality.} && (1.3)
\end{aligned}$$

The difference between the right and left hand sides of equation (1.3) is the KL divergence, given by

$$KL(q|p) = \int \sum_{\{z\}} q(\theta, z) \log \frac{q(\theta, z)}{p(\theta, z|y)} d\theta,$$

since

$$\log p(y) = \int \sum_{\{z\}} q(\theta, z) \log \frac{p(y, z, \theta)}{q(\theta, z)} d\theta + KL(q|p).$$

We want the lower bound to be as close as possible to $p(y)$ and clearly, because of the positivity of the KL divergence, maximising the lower bound, (1.3), corresponds to minimising the KL divergence.

The KL divergence is minimised by taking $q(\theta, z) = p(\theta, z|y)$, but this does not simplify the problem. We require a $q(\theta, z)$ which provides a close approximation to the true joint conditional density and yet is simple enough to be computed. Usually $q(\theta, z)$ is restricted to have a factorised form, in particular of the form

$$q(\theta, z) = q_\theta(\theta) q_z(z).$$

The factors are chosen to minimise

$$\int \int q_\theta(\theta) q_z(z) \log \frac{q_\theta(\theta) q_z(z)}{p(y, z, \theta)} d\theta dz. \tag{1.4}$$

Since $q(\theta, z) = q_\theta(\theta) q_z(z)$ is being regarded as an approximation for $p(\theta, z|y)$, the corresponding (marginal) approximation for $p(\theta|y)$ is clearly

$$p(\theta|y) \approx q_\theta(\theta)$$

Observing the relationship

$$p(y, z|\theta) = \frac{p(y, z, \theta)}{p(\theta)} = \frac{p(\theta, z|y) p(y)}{p(\theta)},$$

and then substituting $q_\theta(\theta) q_z(z)$ for $p(\theta, z|y)$, this results in the approximation

$$p(y|\theta) = \int p(y, z|\theta) dz \approx \frac{q_\theta(\theta) p(y)}{p(\theta)}.$$

So we have $q_\theta(\theta)$ which is the variational posterior for the model parameters and $q_z(z)$ which is the variational posterior for the missing variables. The equations for $q_\theta(\theta)$ and $q_z(z)$ resulting from the variational approximation are coupled.

They can be solved by choosing initial values for the sufficient statistics and then iteratively updating the equations for the model parameters and the missing indicator variables alternately until convergence.

As a general rule, if the complete-data model corresponds to an exponential-family model and if the appropriate conjugate prior is chosen, it follows from the properties of the Kullback-Leibler divergence that the optimal $q_\theta(\theta)$ belongs to the conjugate family (see Ghahramani and Beal (2001) and Chapter 3). The relevant optimal hyperparameters are obtained by solving a set of coupled non-linear equations. Although it is clear that the 'correct' posterior density does not belong to the conjugate family, such approximations have been found to be very useful in many contexts. It can also be shown that the Variational Bayes method is monotonic: in a well-specified sense (1.4) 'decreases' (see Appendix B.1). For more general background on this type of variational approximation, see for example Ghahramani and Beal (2001) and Titterington (2004), and see Jordan (2004) for insight into a more general approach to variational approximations based on the duality theory of convex analysis.

1.2 The Deviance Information Criterion (DIC)

The recent expansion of research into complex hierarchical models for better representing real-world data has incurred the need for some suitable criterion for facilitating model comparison. The classical approach to model comparison usually involves a trade-off between how well the model fits the data and the level of complexity involved. In a somewhat similar spirit, Spiegelhalter et al. (2002) devised a selection criterion, called the Deviance Information Criterion, or DIC, based on Bayesian measures of the complexity level and of how well the model fits the data.

Akaike's (1973) well known criterion, the AIC (Akaike's Information Criterion), proposes that the best model of competing set is the one which minimises

$$\text{AIC} = -2\log(\text{maximum likelihood}) + 2(\text{number of parameters}).$$

The log-likelihood tends to favour models with more parameters and this is penalised by the addition of the number of parameters term. In this way the AIC trades off the fit against complexity. However it has been shown by Shibata (1976)

and Katz (1981) that the AIC still tends to select a model with more parameters than is necessary.

For a long time Bayes factors have been the standard criteria for performing Bayesian model comparison. They provide a way of measuring the evidence in favour of a hypothesis and are defined as the ratio of posterior to prior odds of a hypothesis. However, there are some disadvantages to this criterion. Bayes factors are not well suited for models using improper priors (see Kass and Raftery (1995) for more detail). In addition they require the computation of marginal likelihoods which entails performing integration over the parameter space. There are few models for which these integrals can be evaluated exactly and so usually the Bayes factor has to be approximated.

There are various ways of approximating the Bayes factor, the simplest approximation being the Bayesian Information Criterion (BIC), also known as the Schwarz criterion (Schwarz 1978). It roughly approximates the Bayes factor in a way which avoids the computational difficulties described above.

$$\begin{aligned} \text{BIC} &= -2\log(\text{maximum likelihood}) \\ &\quad +(\text{number of parameters})\log(\text{number of observations}). \end{aligned}$$

The BIC is a conservative criterion in that it tends to provide less support for additional parameters or effects (Raftery (1998)). It also tends to prefer simpler models than those selected using the AIC as the penalisation term is larger than that of the AIC. The disadvantage of the BIC is that one has to be able to specify the number of free parameters in advance and it is not clear how to do this for complex hierarchical models.

There are other methods for approximating Bayes factors but when the dimensionality of the parameter space is high, as is often the case in modern applications, the computational expense involved makes them impractical. Han and Carlin (2001) provides further discussion of some of the theoretical and computational barriers associated with using Bayes factors for the comparison of hierarchical models. So, for complex models, the DIC has the advantage that it is relatively straightforward to compute and, unlike the BIC, one does not have to specify the number of unknown parameters in the model to calculate it. This has made it an attractive option for modern applications, some exam-

ples are described in section 1.2.1. As the DIC was introduced fairly recently, its potential for application and properties are still being investigated. In their development, Spiegelhalter et al. (2002) concentrate on the application of the DIC to exponential-family models, with little said about other scenarios such as models for incomplete data. We consider how this criterion can be extended to these types of model by exploiting the use of variational approximations.

Calculating the DIC and Extending it to the Case of Missing-Data Models

The selection criterion devised by Spiegelhalter et al. (2002) combines Bayesian measures of model complexity and fit. They derive a complexity measure, p_D, which is based on a deviance, the key term of which is

$$D(\theta) = -2\log p(y|\theta),$$

where y denotes data and θ are parameters within the parametric density $p(\cdot|\theta)$. The measure p_D is defined as the difference between the posterior mean of the deviance and the deviance evaluated at the posterior mean or mode, $\widetilde{\theta}$, say, of the relevant parameters:

$$p_D = \mathbf{E}_{\theta|y}\{-2\log p(y|\theta)\} + 2\log p(y|\tilde{\theta}).$$

This p_D is a measure of the effective number of parameters in a model. Spiegelhalter et al. (2002) motivate the use of p_D with an information theoretic argument and investigate some of its formal properties. An attraction of p_D is that if it is being used in conjunction with an MCMC analysis, the quantity can easily be obtained without further approximation. Spiegelhalter et al. (2002) point out that, when $\widetilde{\theta}$ is taken to be the posterior mean, $p_D \geq 0$ for likelihoods which are log-concave in θ. However, it is possible to obtain negative p_D's for non-log-concave likelihoods in some instances.

To measure the fit of the model, the posterior mean deviance,

$$\overline{D(\theta)} = \mathbf{E}_{\theta|y}\{-2\log p(y|\theta)\},$$

is used. Using the posterior mean deviance as measure of fit was suggested by Dempster (1974) who proposed using the posterior mean of the classical deviance statistic to perform Bayesian model selection. It has been used since then by other authors for informal model comparison, but none of these has proposed

any formal method of trading off this quantity against the model complexity. Spiegelhalter et al. (2002) propose such a formal comparison criterion and their Deviance Information Criterion, or DIC, is formed by adding p_D and $\overline{D(\theta)}$:

$$\text{DIC} = \overline{D(\theta)} + p_D.$$

Models which provide a good fit to the data should have larger likelihood, so since the measure of fit, $\overline{D(\theta)}$, is minus twice the posterior mean log-likelihood, a natural choice for a suitable model is one that minimises the DIC. An alternative, equivalent version (see Appendix A.1 for more detail), reminiscent of Akaike's AIC, is

$$\text{DIC} = -2\log p(y|\tilde{\theta}) + 2p_D. \tag{1.5}$$

Spiegelhalter et al. (2002) justify the use of the DIC via a decision-theoretic argument and draw parallels between the DIC and other non-Bayesian selection criteria. Spiegelhalter et al. (2002) point out that the DIC and the AIC are approximately equivalent for models having negligible prior information. The DIC can be thought of as a generalisation of the AIC as it is motivated in a similar way but it can be applied to any type of model.

We briefly list some of the practicalities involved in using the DIC and refer the reader to Spiegelhalter et al. (2002) for more detail.

Invariance to Parameterisation

The p_D is not invariant to the model parameterisation since changing posterior means corresponding to different choices of θ can result in different values of $D(\tilde{\theta})$. p_D may only be approximately invariant to these parameterisation changes.

Focus of Analysis

DIC may be sensitive to changes in the model structure.

Nuisance Parameters

Nuisance parameters such as variances which are not initially integrated out of the likelihood add to the complexity estimation.

Significance of Differences Between DIC's

Spiegelhalter et al. (2002) suggest the rule of thumb that models with a DIC within one to two of the 'best' model are worth considering and that others are not well supported. This follows the suggestion made by Burnham and Anderson (1998) for comparing AIC values.

Asymptotic consistency

The DIC does not consistently choose the correct model from a fixed set with growing sample sizes. This is the same as for the AIC and the authors are not greatly perturbed by this.

As previously mentioned, Spiegelhalter et al. (2002) say little about using the DIC to compare models such as those for incomplete data. Celeux et al. (2006) propose a number of adaptations for dealing with such cases. A crucial issue is the fact that $\overline{D(\theta)}$ is not known explicitly, and when necessary Celeux et al. (2006) replace such expectations with sample means based on a large number of MCMC realisations from $p(\theta|y)$. Our approach is, instead, to exploit the use of variational approximations which allows us to express p_D in the following computable form

$$p_D \approx -2 \int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta + 2 \log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}, \tag{1.6}$$

and hence allows us to extend the DIC to missing data models.

1.2.1 Some Applications of the DIC

The DIC has been applied to various problems involving model comparison for complex models. For instance, Berg et al. (2004) apply the DIC to model selection for stochastic volatility models which are used to analyse financial time series data. This is an example of an area where recent advances have led to increasingly complex models for which the standard selection criteria are unsuitable. Berg et al. (2004) point out that there is no straightforward, computationally efficient way of using Bayes factors for these models, since they involve a high-dimensional parameter space, and suggest that the DIC is more convenient for model comparison. Since stochastic volatility models are hierarchical, it is not straightforward to specify the number of free parameters to allow the calculation of the BIC which approximates the Bayes factor. The DIC has the advantage that this specification is not required. The authors found that, for a simulated data set, the DIC identified the correct model out of 8 possibilities. For a real

data example, they compared the DIC selection to that made using Bayes factors (taking this as the "gold-standard" criterion) and found that both methods selected the same model for the data. They also noted that the DIC seemed to be robust against changes in priors.

Zhu and Carlin (2000) apply the DIC to model selection for hierarchical models in medical applications. They take a Bayesian approach to smoothing crude maps of disease risk. Developments in this area have led to the analysis of spatial data involving variables which are clustered across varying sets of regional boundaries (i.e. spatially misaligned). There is also a temporal aspect to be considered. This type of data can be modelled using MCMC but a formal method for comparing fit is difficult due to the bulk of data and the use of improper priors. The DIC can be applied in complex hierarchical model settings like this. Zhu and Carlin (2000) found that the DIC performed reasonably well in their analysis although they were unable to obtain a satisfactory MCMC estimate of the variance of the DIC.

Green and Richardson (2002) apply a hierarchical model of the spatial heterogeneity of the rare count data arising in disease mapping by proposing a hidden discrete-state Markov random field model generated by an underlying finite-mixture model which allows spatial dependence. They also use the DIC as a model selection tool for this complex model.

Chapter 2

Bayesian Analysis of Finite Mixture Models

In this chapter we give an introduction to finite mixture models and their applications. We consider some of the issues involved in statistical inference for this type of model and, in particular, we shall show how variational methods, that have become popular in some of the neural computing/machine learning literature, can be used to determine a suitable number of components for a mixture model in the case of a mixture of Gaussian distributions. When this approach is taken, it turns out that, if one starts off with a large number of components, superfluous components are eliminated as the method converges to a solution, thereby leading to an automatic choice of model complexity. When the method is applied to simulated data-sets, results suggest that the method is able to recover the correct number of components.

Furthermore, we show how the DIC can be extended to this class of model via a variational approximation.

2.1 Finite Mixture Models

Finite mixtures have featured in statistical modelling throughout the past century. Mixture distributions provide a computationally convenient and flexible way of modelling complex probability distributions not well represented by the standard parametric models. The mixture density is made up of a linear combination of K, say, simpler component densities. This is an appropriate type of model when

one has experimental observations which are known to be grouped into K classes, where each component class is appropriately modelled by some parametric density and is weighted by the relative frequency of that class in the population. There is a vast array of literature available on the subject of finite mixtures particularly since computers became available to researchers. A comprehensive treatment of the subject is given by Titterington et al. (1985). The monograph by Everitt and Hand (1981) is also dedicated to the subject of finite mixture distributions. McLachlan and Peel (2000) is a recent text on the subject.

Finite mixture densities with K components, for an observation y_i, are of the form (Titterington et al. (1985))

$$p(y_i|\phi, \rho) = \sum_{j=1}^{K} \rho_j f(y_i|\phi_j), \tag{2.1}$$

where $f(\cdot|\phi)$ denotes a parametric family model and ρ_j is the missing weight associated with the j^{th} component. In most of the examples that one will encounter, all of the component densities will have the same parametric form, as they do in the examples we shall consider, but it is worth noting that this is not strictly necessary.

With mixture data, the component which gave rise to any particular observation is unknown and so mixture models can be interpreted as incomplete-data or missing-data models. If, for each observation, y_i, we introduce an imaginary indicator variable, z_i, which identifies the component our observation arose from, then since these indicators are unknown or missing to us, we have a missing-data model. If this set of indicator variables were available to us, parameter estimation would be straightforward.

Data sets which are suitably represented by mixture models arise naturally in a variety of settings. Common applications of mixture models include problems in medical diagnosis and biological applications. In a medical setting, patients may be considered as belonging to a particular class determined by the disease or condition from which they are suffering, but the class to which they belong may be unknown and the aim of inference would be to predict the appropriate classification. Biological studies often involve categorising organisms into recognisable species; in this case the mixture components would represent species groups. Recently, mixture models have become popular in machine learning where they are

used in unsupervised learning problems, and this has sparked interest in inference for mixture models among the machine learning community. In addition to these direct applications to data sets, mixture models also find a use in the development of statistical methodology; see Titterington et al. (1985) for an illustration.

Due to the complexity of mixture models, and the fact that formulae for model parameters generally cannot be written down explicitly, early progress in mixture model research was slowed by the lack of computational resources. In recent years, the availability of such resources has allowed rapid expansion of investigation in this area and various new inference methods have been created.

The main problems of inference for these models are estimation of the number of components in the model and estimation of the component parameters. Often, with problems involving mixture models, the exact number of components present is unknown, and to estimate this assumptions are often made about what parametric form these components might take, or, if there is sufficient data available then this parametric form may be known. Estimation of the number of component densities is not straightforward. The usual hypothesis test framework of testing hypotheses of alternative potential numbers of components cannot be easily applied here. For instance, asymptotic assumptions underlying tests such as the generalised likelihood ratio test based on the chi-squared approximation are not satisfied. There have been attempts to apply modifications of such tests, but the problem of estimating a suitable number of components still remains open. Recent approaches making use of MCMC schemes which are capable of comparing models with different numbers of components were developed, such as the reversible jump Markov Chain Monte Carlo (MCMC) method of Richardson and Green (1997) or the birth-death MCMC method, based on a continuous-time Markov birth-death process, of Stephens (2000).

To estimate the unknown component parameters of the mixture model, various methods have been applied, for example, graphical methods, the method of moments, maximum likelihood estimation, minimum Chi-squared (see chapter 4 of Titterington et al. (1985) for a description of all of these) and, increasingly popularly, Bayesian approaches, particularly using MCMC schemes; we elaborate on this below. The historic paper by Pearson (1894) used the method of moments for inference on finite mixture models, and, despite some drawbacks, this was the main method of analysis used until computers became widely available (Titterington et al. (1985)). After that, maximum likelihood estimation became

popular, principally interpreting the mixture models as incomplete data models and making use of the Expectation-Maximisation (EM) algorithm of Dempster, Laird and Rubin (1977) to obtain maximum likelihood estimates of the mixing weights, means and variances not available in closed form. The EM algorithm involves two main stages, the expectation step and the maximisation step. Initially observations are allocated to components and parameters are estimated as if the data were complete. In the first step the expected value of the complete data log-likelihood at these values is calculated. In the maximisation step the expectation found in the first step is maximised and parameter values which maximise it are used as the approximate maximising parameter values in the first step of the next iteration. Iterations alternate between the two steps until the algorithm converges to a solution. The EM algorithm is attractive due to the ease of implementation and its monotonicity. The monotonicity property means that the observed data likelihood never decreases as the iterations of the algorithm progress. Each new observed data likelihood will always be greater than or equal to the previous estimate. This means that the algorithm will converge monotonically to some value; of course convergence to the global maximiser is not guaranteed. A drawback of this method is that the convergence rate is very slow. Many modifications to the basic EM algorithm have been made over the years in an attempt to improve performance; see for example Celeux and Diebolt (1985,1989).

The most notable advances in mixture model analysis in recent years have made use of a Bayesian approach and it is with the Bayesian approach that we are primarily concerned. Bayesian inference involves updating prior to posterior information through the relationship that the posterior is proportional to the prior times the likelihood. The problem with this paradigm for finite mixtures is that the posteriors which arise involve large numbers of terms, and this makes calculation of quantities such as posterior means infeasible if the mixture model has more than one or two unknown parameters. To circumvent this computational problem, most recent progress has made use of MCMC sampling methods which also interpret the mixture as an incomplete-data model (see, for example, Robert and Diebolt (1994), Casella et al. (2002)). These MCMC methods have become commonplace in Bayesian analysis, but MCMC schemes which are capable of performing model comparison between models of varying dimensions, such as those mentioned above, were introduced much more recently.

Green (1995) introduced the "reversible jump" methodology which constructs a Markov chain with a stationary distribution which is equal to the joint posterior distribution over the model and the parameters. The algorithm occasionally proposes "jumps" between different possible models with a rejection rate which ensures the desired stationary distribution is retained. It is hoped that the algorithm will adequately explore the various potential models. However, in practice it can be difficult to manage this. Richardson and Green (1997) applied this reversible jump move to perform Bayesian analysis of mixture models. They analysed three of the real data sets we use in section 2.8.3, where we compare their results with the ones we obtain. Stephens (2000) described an alternative MCMC method for estimating the number of components of a mixture model which is based on a continuous-time Markov birth-death process rather than reversible jump moves. MCMC methods of this kind had previously been used for Bayesian analysis of point process model parameters. To adapt this idea to mixture models, Stephens (2000) views the mixture parameters as a marked point process with each point representing a mixture component. His MCMC scheme permits changes to the number of components by permitting the "birth" of new components and the "death" of some existing ones, taking place in continuous time. The relative birth and death rates determine the stationary distribution of the chain. There is a prior on the births so that they occur at a constant rate and the death rate is high for components which are not useful in explaining the data and low for components that are useful. So, rather than accepting or rejecting jumps as in Richardson and Green (1997), Stephens (2000) has good and bad births with bad ones being removed through the death process. Stephens (2000) states that his method seems to involve computational time comparable to that involved in the method of Richardson and Green (1997), in the case of mixtures of univariate Normals. However Stephen's (2000) method does not require the calculation of a complicated Jacobian and, with his method, it is straightforward to alter the algorithm to consider a different parametric model for the mixture components.

A drawback of these iterative MCMC methods is that they can be time-consuming. In addition to this, it can also be difficult to assess when the sampler has reached convergence. The variational technique we describe is a non-iterative deterministic alternative to the MCMC methods. Variational methods can also be used to compare competing models with differing numbers of components while si-

multaneously estimating model parameters. The attractiveness of the variational approximation is the fast computational time and the ease of implementation.

2.2 Variational Methods for Analysis of Finite Mixture Distributions in Machine Learning

Variational methods for inference about mixture models have been appearing in the machine learning literature over the last decade. Waterhouse et al. (1996) proposed estimating the parameters and hyperparameters of a mixture model by using a Bayesian framework based on the variational approximation. This was presented as an alternative to the maximum likelihood approach to parameter estimation in artificial neural networks which tends to over-fit the model. This paper does not consider the idea of using the variational framework for model selection.

Attias (1999) extends the variational Bayes technique to perform model selection as well as estimating parameters by introducing a prior over the model structure. This results in a variational posterior distribution over the model structure. For mixture models this leads to a posterior distribution over the number of components in the model. For application of the algorithm to mixtures of Gaussians, Attias (1999) assigns non-informative priors to the model parameters and the number of components is given a uniform prior based on a prescribed maximum number of components. Attias (1999) uses the log posterior over the number of components which arises from application of the algorithm to identify a suitable number of components, the optimal number being the peak of this posterior. However, Attias (1999) states that, for this model set-up, if the number of observations assigned to a component is one or less, the posterior mean of the mixing weight of that component is zero, effectively indicating that that component is unnecessary and eliminating it. Attias (1999) suggests that, in this way, the variational algorithm avoids the problem of singularities which can arise with the EM algorithm. The problem is that such a component may become centred at a single observation point resulting in zero variance, which leads to an infinite likelihood and the incorrect model having a larger likelihood than a correct one.

Both Waterhouse et al. (1996) and Attias (1999) emphasise the connection between the EM algorithm and the variational Bayes algorithm. Variational

Bayes is an EM-like algorithm, the expectation step of EM corresponding to finding the expected value of the posterior of the component indicator variables in variational Bayes. The maximisation step of EM relates to estimating the model parameters in variational Bayes by maximising the lower bound on the marginal log-likelihood.

Corduneanu and Bishop (2001) also apply the variational learning technique to the analysis of a finite mixture of Gaussians. They consider the variational approach to estimating the number of components as well as estimating component parameters. They take an approach which involves optimising the mixing co-efficients using type-2 maximum likelihood and marginalising out the model parameters using variational methods which leads to automatic recovery of the number of components. They have a fixed maximum potential number of components. They employ an EM-like procedure in which they alternately maximise the lower bound on the marginal log-likelihood with respect to the mixing weight co-efficients and then update the expected values of the model parameters and hidden variables. Corduneanu and Bishop (2001) find that optimising the mixing co-efficients using type-2 maximum likelihood causes the mixing weights of unwanted components to go to zero. As their algorithm progresses, if Gaussians with similar parameters are fitting the same part of the data, they become unbalanced, in terms of the expectations of the mixing co-efficients, until one dominates the rest which means the others can be removed. Corduneanu and Bishop (2001) remove components when the expectations of the mixing co-efficients are less than 10^{-5}. Corduneanu and Bishop (2001) found that starting their program with initial means which were equal or too similar made differentiation between components during the optimisation stage difficult and led to slow convergence and removal of too many components. To address this problem, the authors use K-means clustering to set the initial means. They assign large initial covariance matrices to the components, which they opine is enough to avoid local maxima.

Ueda and Ghahramani (2002) state that variational learning algorithms can become trapped in poor local optima near initial values. With a view to simultaneously optimising the parameters of a mixture model and automatically selecting the number of components whilst avoiding becoming trapped in poor local optima, Ueda and Ghahramani (2002) present a Variational Bayes Split and Merge Expectation-Maximisation (Variational Bayes SMEM) algorithm, following on from their SMEM for mixture models developed within the maximum

likelihood approach (Ueda et al. (2000)). The maximum likelihood framework used in Ueda et al. (2000) limits the algorithm to fitting a model with a fixed number of components, since the maximum likelihood value tends to increase with model complexity meaning that the SMEM cannot find the optimal model structure. Ueda and Ghahramani (2002) define prior distributions over the model parameters, hyperparameters, hidden variables and the model complexity (i.e. the number of components). The Variational Bayes SMEM algorithm first performs the conventional variational Bayes method for a model with a given number of components, and then, for this fitted model, the algorithm seeks to maximise the lower bound on the log marginal likelihood by performing either a split of components, a merge of components or a split and merge simultaneously. In this way the algorithm searches for the optimal number of components whilst trying to avoid local maxima. The variational posterior for this new model is then generated and if the lower bound is improved this proposal is accepted, otherwise it is rejected. This is repeated until the lower bound is no longer improved. The technique involves a greedy search strategy in that at each stage it tries to find a better local maximum for the objective function and so convergence to a global maximum cannot be guaranteed, but at each stage the value of the objective function is improved and so a better local maximum is obtained. The authors were successful in applying this method to real and simulated data sets. Ueda and Ghahramani (2002) do not report that unwanted components are automatically removed through the application of their variational Bayes.

In our implementation of the variational method, we also observe the component elimination property noted by Attias (1999) and Corduneanu and Bishop (2001), although our model hierarchy is different from that used in these papers. We did not find it necessary to use a clustering method to choose the initial estimates of the component means as was done in the paper by Corduneanu and Bishop (2001) nor did we encounter any problems from initialising our algorithm with equal component means set to zero.

2.3 Mixture Models Interpreted as Missing-Data Models

In the context of a random sample of mixture data, we can interpret the model as a missing data model by introducing a set of missing binary indicator variables $\{z_{ij}\}$, $(i = 1, ..., n, j = 1, ..., K)$ to describe which component gave rise to a particular observation. The $\{z_{ij}\}$ are defined so that if observation y_i is from component m, say, then

$$\begin{aligned} z_{ij} = \quad & 1 \quad \text{if} \quad j = m \\ & 0 \quad \text{if} \quad j \neq m. \end{aligned}$$

This leads to a model of the form

$$p(y|\theta) = \prod_{i=1}^{n} \{\sum_{z_i} p(y_i, z_i|\theta)\},$$

where n denotes the sample size and $z = \{z_i\}$ denotes missing data. The parameters θ include mixing weights and parameters of the component densities. It is well known that, with mixture data, posterior densities $p(\theta|y)$ are complicated and exact evaluation of posterior expectations is not practicable. We shall deal with this difficulty by using a factorised variational approximation for $p(\theta, z|y)$ in order to calculate p_D and to find an expression for $\tilde{\theta}$, the latter to be inserted into the exact formula for $p(y|\tilde{\theta})$.

2.4 DIC for Mixture Models

This section examines how the DIC can be applied to mixture models. The most convenient forms for the DIC and p_D, for our purposes, are those given in Chapter 1:

$$\text{DIC} = 2p_D - 2\log p(y|\tilde{\theta})$$

and

$$p_D = \mathbf{E}_{\theta|y}\{-2\log p(y|\theta)\} + 2\log p(y|\tilde{\theta}).$$

For mixture data, it is straightforward to calculate $p(y|\theta)$ but evaluation of $\mathrm{E}_{\theta|y}$ is difficult. Ideally $\tilde{\theta}$ will be a summary parameter from $p(\theta|y)$. We take $\tilde{\theta}$ to be the posterior mean.

The following section explains how the difficulty in this case can be dealt with using a factorised variational approximation for $p(\theta, z|y)$ in order to calculate p_D and to find an expression for $\tilde{\theta}$. The DIC value can then be obtained by substituting the exact formula for $p(y|\tilde{\theta})$.

2.5 Mixture of K Univariate Gaussian Distributions

Consider a mixture of K univariate Gaussian distributions with unknown means, variances and mixing weights. The mixture model is of the form

$$p(y_i|\theta) = \sum_{j=1}^{K} \rho_j N(y_i; \mu_j, {\tau_j}^{-1}), \quad \text{for} \quad i = 1, ..., n$$

where τ_i is the precision and is equal to $\frac{1}{{\sigma_j}^2}$.

We consider a data set made up of observations $y_1, ..., y_n$ which are assumed to have been drawn independently from the mixture distribution. By the definition given in section 2.3, for each observation y_i, z_{ij} is defined such that $z_{ij} = 1$ if y_i comes from component j and $z_{ij} = 0$ otherwise. By this definition $\sum_{j=1}^{K} z_{ij} = 1$, for each i, and we can write

$$\begin{aligned} p(y, z|\theta) &= \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j N(y_i; \mu_j, {\tau_j}^{-1})\}^{z_{ij}} \\ &= \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j \sqrt{\frac{\tau_j}{2\pi}} e^{-\frac{\tau_j}{2}(y_i - \mu_j)^2}\}^{z_{ij}}. \end{aligned}$$

2.5.1 The Variational Approach

Assigning the Prior Distributions

We now assign prior distributions to the parameters of θ.

The mixing weight coefficients are given a Dirichlet distribution

$$\rho = (\rho_1, ..., \rho_K) \sim \text{Dir}({\alpha_1}^{(0)}, ..., {\alpha_K}^{(0)}).$$

Conditional on the precisions, the means are assigned independent univariate Normal conjugate priors, so that

$$p(\mu|\tau) = \prod_{j=1}^{K} N(\mu_j; {m_j}^{(0)}, ({\beta_j}^{(0)}\tau_j)^{-1}).$$

The precisions are given independent Gamma prior distributions, so that

$$p(\tau) = \prod_{j=1}^{K} Ga(\tau_j|\frac{1}{2}{\gamma_j}^{(0)}, \frac{1}{2}{\delta_j}^{(0)}).$$

We then have

$$p(\theta) = p(\rho)p(\mu|\tau)p(\tau)$$

$$\propto \prod_{j=1}^{K} {\rho_j}^{{\alpha_j}^{(o)}-1} \prod_{j=1}^{K} \sqrt{\frac{{\beta_j}^{(o)}\tau_j}{2\pi}} e^{-\frac{{\beta_j}^{(o)}\tau_j}{2}(\mu_j - {m_j}^{(o)})^2} \prod_{j=1}^{K} \tau_j^{\frac{1}{2}{\gamma_j}^{(0)}-1} e^{-\frac{1}{2}{\delta_j}^{(0)}\tau_j}.$$

Thus, the resulting joint distribution of all of the random variables is

$$p(y, z, \theta) \propto \prod_{j=1}^{K} {\rho_j}^{{\alpha_j}^{(o)}-1+\sum_{i=1}^{n} z_{ij}} \prod_{j=1}^{K} [\sqrt{\tau_j}^{(1+\sum_{i=1}^{n} z_{ij})} \tau_j^{\frac{1}{2}{\gamma_j}^{(0)}-1} \exp\{-\frac{\tau_j}{2}\sum_{i=1}^{n} z_{ij}(y_i - \mu_j)^2\}$$

$$\times \exp\{-\frac{{\beta_j}^{(0)}\tau_j}{2}(\mu_j - m_j^{(0)})^2 - \frac{1}{2}{\delta_j}^{(0)}\tau_j\}].$$

For the variational approximation to $p(z, \theta|y)$ we take $q(\theta, z)$ to have the factorised form

$$q(\theta, z) = q_\theta(\theta)q_z(z).$$

Forms of the Variational Posterior Distributions

For details of the calculation of these formulae see Appendix C.1. The optimal $q_\rho(\rho)$ and $q_j(\mu_j, \tau_j)$ have the forms

$$q_\rho(\rho) \propto \prod_{j=1}^{K} \rho_j^{\alpha_j - 1},$$

$$q_j(\mu_j, \tau_j) = q_j(\mu_j|\tau_j) q_j(\tau_j),$$

where

$$q_j(\mu_j|\tau_j) \propto \sqrt{\beta_j \tau_j} \exp\{-\frac{\beta_j \tau_j}{2}(\mu_j - m_j)^2\}$$

$$q_j(\tau_j) \propto \tau_j^{\frac{1}{2}\gamma_j - 1} \exp\{-\frac{1}{2}\delta_j \tau_j\}.$$

For the $q_{z_i}(z_i)$ we have, for each $i = 1, ..., n$ and each $j = 1, ..., K$,

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q \log \rho_j + \frac{1}{2}\mathbf{E}_q \log \tau_j\} \exp\{-\frac{1}{2\beta_j}\} \exp\{-\frac{1}{2}\mathbf{E}_q[\tau_j (y_i - m_j)^2]\}$$

$$= \exp\{\Psi(\alpha_j) - \Psi(\sum_{j\cdot} \alpha_{j\cdot}) + \frac{1}{2}\Psi(\frac{1}{2}\gamma_j) - \frac{1}{2\beta_j} - \frac{\gamma_j}{2\delta_j}(y_i - m_j)^2\} \times \frac{2}{\delta_j},$$

where $\Psi(\cdot)$ is the digamma function,

$$\Psi(\alpha) = \frac{\frac{\partial}{\partial \alpha}\Gamma(\alpha)}{\Gamma(\alpha)} = \frac{\partial}{\partial \alpha} \ln \Gamma(\alpha).$$

The $q_{z_i}(z_i = j)$ are normalised to sum to 1 over j for each i.

For q's going with the parameters we have

$$q_\rho(\rho) \propto \prod_{j=1}^{K} \rho_j^{\alpha_j{}^{(0)} + \sum_i q_{z_i}(z_i = j) - 1}$$

$$q_j(\mu_j, \tau_j) \propto \tau_j^{\frac{1}{2}(1 + \sum_{i=1}^{n} q_{z_i}(z_i = j) + \gamma_j{}^{(0)} - 2)} \exp\{-\frac{\tau_j}{2} \sum_{i=1}^{n} q_{z_i}(z_i = j)(y_i - \mu_j)^2\}$$

$$\times \exp\{-\frac{\beta_j{}^{(0)} \tau_j}{2}(\mu_j - m_j^{(0)})^2 - \frac{1}{2}\delta_j{}^{(0)} \tau_j\}].$$

The posterior distributions are therefore

$$\rho \sim Dir(\alpha_1, ..., \alpha_k)$$

$$\mu_i|\tau_j \sim N(m_j, \frac{1}{\beta_j \tau_j})$$

$$\tau_i \sim Ga(\frac{1}{2}\gamma_j, \frac{1}{2}\delta_j)$$

If we denote $q_{z_i}(z_i = j)$ by q_{ij}, hyperparameters are given by

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\beta_j = \beta_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\gamma_j = \gamma_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$m_j = \frac{\beta_j{}^{(0)} m_j{}^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}}$$

$$\delta_j = \delta_j{}^{0)} + \sum_{i=1}^{n} q_{ij} y_i{}^2 + \beta_j{}^{(0)} m_j{}^{(0)2} - \beta_j m_j{}^2.$$

$$\begin{aligned} q_{ij} &= \frac{\exp\{\Psi(\alpha_j) - \Psi(\sum_j \alpha_{j\cdot}) + \frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j} - \frac{\gamma_j}{2\delta_j}(y_i - m_j)^2\}}{s_i} \\ &= \frac{\varphi_{ij}}{s_i}, \end{aligned}$$

say, where $s_i = \sum_{j=1}^{K} \varphi_{ij}$.

Obtaining Formulae for p_D and the DIC

Now we derive formulae to calculate p_D and DIC (see Appendix C.2 for details). We have

$$\log p(y|\tilde{\theta}) = \sum_{i=1}^{n} \log[\sum_{j=1}^{K} \tilde{\rho}_j \sqrt{\frac{\tilde{\tau}_j}{2\pi}} \exp\{-\frac{\tilde{\tau}_j}{2}(y_i - \tilde{\mu}_j)^2\}],$$

where we use

$$\tilde{\rho}_j = \frac{\alpha_j}{\sum_{j=1}^{K} \alpha_j}$$

$$\tilde{\mu}_j = m_j$$

$$\tilde{\tau}_j = \frac{\gamma_j}{\delta_j}.$$

In this case we obtain

$$p_D \approx -2 \int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta + 2 \log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$\begin{aligned} = & - 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\{\Psi(\alpha_j) - \Psi(\alpha_.) + \frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j}\}] \\ & + 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij}) \log(\frac{\alpha_j}{\alpha_.}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij}) \log(\frac{\gamma_j}{\delta_j})], \end{aligned}$$

where

$$\alpha_. = \alpha_1 + ... + \alpha_K.$$

We can then obtain a value for the DIC through the formula

$$\text{DIC} = 2p_D - 2 \log p(y|\tilde{\theta}).$$

2.6 Multivariate Case

The method applied to the mixtures of univariate Normals can easily be extended to the multivariate case. In this case we have

$$p(y, z|\theta) = \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j N_d(y_i; \mu_j, T_j^{-1})\}^{z_{ij}}.$$

where N_d denotes the multivariate Normal density with dimensionality d and T_j denotes the j^{th} precision matrix, equal to the inverse of the j^{th} covariance matrix.

2.6.1 The Variational Approach

Assigning the Prior Distributions

As with the univariate case, the mixing weight coefficients are given a Dirichlet prior distribution

$$p(\rho) = Dir(\rho|{\alpha_1}^{(0)}, ..., {\alpha_K}^{(0)}).$$

The means are assigned the multivariate Normal conjugate prior, conditional on the covariance matrices, so that

$$p(\mu|T) = \prod_{j=1}^{K} N_d(\mu_j; {m_j}^{(0)}, ({\beta_j}^{(0)} T_j)^{-1}),$$

where

$$\mu = (\mu_1, ..., \mu_K)$$

$$T = (T_1, ..., T_K).$$

The precision matrices are given independent Wishart prior distributions,

$$p(T) = \prod_{j=1}^{K} W(T_j|{v_j}^{(0)}, {\Sigma_j}^{(0)}).$$

The complete prior is then

$$p(\theta) = p(\rho)p(\mu|T)p(T)$$

$$\propto \prod_{j=1}^{K} {\rho_j}^{{\alpha_j}^{(o)}-1} \prod_{j=1}^{K} |{\beta_j}^{(o)} T_j|^{\frac{1}{2}} \exp\{-\frac{1}{2}(\mu_j - {m_j}^{(o)})^T {\beta_j}^{(o)} T_j (\mu_j - {m_j}^{(o)})\}$$

$$\times \prod_{j=1}^{K} \frac{|T_j|^{\frac{v_j{}^{(0)}-d-1}{2}} \exp\{-\frac{1}{2}\mathrm{tr}(\Sigma^{(0)}T_j)\}}{2^{\frac{v_j{}^{(0)}d}{2}} \pi^{\frac{d(d-1)}{4}} |\Sigma_j{}^{(0)}|^{\frac{-v_j{}^{(0)}}{2}} \prod_{s=1}^{d} \Gamma[\frac{1}{2}(v_j{}^{(0)}+1-s)]}.$$

Thus, the resulting joint distribution of all of the variables is

$$p(y,z,\theta) \propto \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(o)}+\sum_{i=1}^{n} z_{ij}-1} \prod_{j=1}^{K} |\beta_j{}^{(o)}T_j|^{\frac{1}{2}} \exp\{-\frac{1}{2}(\mu_j - m_j{}^{(o)})^T \beta_j{}^{(o)}T_j(\mu_j - m_j{}^{(o)})\}$$

$$\times \prod_{j=1}^{K} |T_j|^{\frac{\sum_{i=1}^{n} z_{ij}}{2}} \exp\{-\frac{1}{2}\sum_{i=1}^{n} z_{ij}(y_i-\mu_j)^T T_j(y_i-\mu_j)\}$$

$$\times \prod_{j=1}^{K} \frac{|T_j|^{\frac{v_j{}^{(0)}-d-1}{2}} \exp\{-\frac{1}{2}\mathrm{tr}(\Sigma^{(0)}T_j)\}}{2^{\frac{v_j{}^{(0)}d}{2}} |\Sigma_j{}^{(0)}|^{-\frac{v_j{}^{(0)}}{2}} \prod_{s=1}^{d} \Gamma[\frac{1}{2}(v_j{}^{(0)}+1-s)]}$$

For the variational approximation to $p(z,\theta|y)$ take $q(\theta,z)$ to have the factorised form

$$q(\theta,z) = q_\theta(\theta)q_z(z).$$

Form of the Variational Posterior Distributions

Details of the calculations are given in Appendix D.1. The posterior distributions which maximise the right hand side of (1.3) are then

$$\rho \sim Dir(\alpha_1,...,\alpha_k)$$

$$\mu_j|T_j \sim N_d(m_j,(\beta_j T_j)^{-1})$$

$$T_j \sim W(v_j,\Sigma_j),$$

with hyperparameters given by

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\beta_j = \beta_j^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$m_j = \frac{\beta_j^{(0)} m_j^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j}$$

$$\Sigma_j = \Sigma_j^{(0)} + \sum_{i=1}^{n} q_{ij} y_i y_i^T + \beta_j^{(0)} m_j^{(0)} m_j^{(0)T} - \beta_j m_j m_j^T$$

$$v_j = v_j^{(0)} + \sum_{i=1}^{n} q_{ij}$$

with

$$\begin{aligned} q_{ij} &= \frac{\exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log T_j] - \frac{1}{2}\mathrm{tr}(\mathbf{E}_q[T_j](y_i - m_j)(y_i - m_j)^T + \frac{1}{\beta_j}\mathbf{I}_d)\}}{s_i} \\ &= \frac{\varphi_{ij}}{s_i}, \end{aligned}$$

say, where $s_i = \sum_{j=1}^{K} \varphi_{ij}$.

The expectations are given by

$$\mathbf{E}_q[\mu_j] = m_j$$

$$\mathbf{E}_q[T_j] = v_j \Sigma_j^{-1}$$

$$\mathbf{E}_q[\ln |T_j|] = \sum_{s=1}^{d} \Psi(\frac{v_j + 1 - s}{2}) + d \ln (2) - \ln |\Sigma|$$

$$\mathbf{E}_q[\ln(\rho_j)] = \Psi(\hat{\alpha}_j) - \Psi(\hat{\alpha}_\cdot).$$

Obtaining Formulae for p_D and the DIC

Now we derive formulae for calculating p_D and DIC (see Appendix D.2). We have

$$\log p(y|\tilde{\theta}) = \sum_{i=1}^{n} \log[\sum_{j=1}^{K} \tilde{\rho}_j \frac{|\tilde{T}_j|^{\frac{1}{2}}}{2\pi^{\frac{d}{2}}} \exp\{-\frac{1}{2}(y_i - \tilde{\mu}_j)^T \tilde{T}_j (y_i - \tilde{\mu}_j)\}],$$

where we use

$$\tilde{\rho}_j = \frac{\alpha_j}{\sum_{j=1}^{K} \alpha_j}$$

$$\tilde{\mu}_j = m_j$$

$$\tilde{T}_j = v_j {\Sigma_j}^{-1}.$$

p_D is approximated as

$$p_D \approx -2 \int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta + 2 \log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$= -2 \left[\sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \{\mathrm{E}_q[\log \rho_j] + \frac{1}{2} \log \mathrm{E}_q[|T_j|] - \frac{1}{2\beta_j}\}\right]$$
$$+ 2 \left[\sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \log(\frac{\alpha_j}{\alpha_.}) + \frac{1}{2} \sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \log |v_j {\Sigma_j}^{-1}|\right].$$

We can then obtain a value for the DIC through the formula

$$\mathrm{DIC} = 2p_D - 2 \log p(y|\tilde{\theta}).$$

2.7 Practical Implementation

Our variational method and calculation of the DIC and p_D values is implemented using a program which is run in R. Code has been written to deal with both one-dimensional and two-dimensional data sets and can be initialised to start with any number of maximum potential components.

The user must specify the initial number of components, K, to start with, and the data set to be analysed. The user's input data must contain the observed data and a list of indices ranging from $1, ..., K$ initially allocating the observations to one of the K components. In our examples, we allocated roughly equal numbers

of the observations to each of the K components. No particular method was used for doing so but we found that the initial allocation did not seem to affect results. The initial allocation is just to start the program off and as the algorithm cycles through its iterations, the observations find their own places.

At the initialisation stage, a user-specified value is given for the weight that is to be assigned to each observation indicator variable (the q_{ij}'s). These initial values for the q_{ij}'s were chosen to give a slightly higher weighting to the initial allocation to components just to start the algorithm running. For example, suppose that the initial number of components chosen by the user is 5 and that the initial weighting is 0.3. If observation i has initially been assigned to component 1, then

$$q_{i1} = 0.3$$

and

$$q_{ij} = \frac{1 - 0.3}{4} \quad \text{for} \quad j \neq 1.$$

In most cases, the results obtained were the same for all values of the initial weight but occasionally it led to slight differences. When this was found to be the case, the DIC value was used to choose between models.

The user has the option to specify initial values for the sufficient statistics or, alternatively, defaults which specify broad priors are available. All of our examples use these broad priors.

As the program runs, the resulting q_{ij}'s are nonnegative and sum to 1 over j for each i. They therefore form a set of predictive probabilities for the indicator variables for the data. The sum of the q_{ij}'s over the i time points for each component provides an estimate of the number of observations that are being allocated to each component, we can think of this as a weighting for each component. The cutoff value determines at which point a component is no longer deemed to be part of the solution. The default value we use for this is 1 and this was the value used in all examples given. This means that a component is not considered useful if less than one observation is assigned to it. When one component's weight falls below this cutoff value it is removed from consideration and the program continues with one fewer component.

At each iteration of the code, the DIC and p_D values are computed and the updated weights for each component are obtained. The program runs until it converges and the solution it finds will have a number of components which

is less than or equal to the number the user started with. This means that components which were considered to be superfluous are removed as the program cycles through its iterations.

To summarise, application of the method leads to an automatic choice of model:

- The algorithm is initialised with a number of components larger than one would expect to find.
- If two or more Gaussians with similar parameters seem to be representing the same component of the data then one Gaussian will dominate the others causing their weightings to go to zero.
- When a component's weighting becomes sufficiently small, taken to be less than one observation in our approach, the component is removed from consideration and the algorithm continues with the remaining components until it converges to a solution.
- At each step the DIC value and p_D value are computed. In our results, these decrease as the algorithm converges so that the model chosen by comparing the DIC values corresponds to the model chosen by the variational method.

2.8 Performance of the Method on Simulated and Real Data Sets

2.8.1 Results of Analysis of Simulated Data from Mixtures of Multivariate Gaussians

We first consider multivariate data-sets simulated from the three models analysed by Corduneanu and Bishop (2001). As mentioned previously, Corduneanu and Bishop (2001) take an approach based on optimising the mixing co-efficients using type-2 maximum likelihood and marginalise out the model parameters using variational methods which leads to automatic recovery of the number of components. Their approach is based on a different prior for the component means than that used in our approach. They assign a Gaussian prior, with zero mean, and a covariance matrix, proportional to the identity matrix, chosen to give a broad

prior over the component means. Corduneanu and Bishop's (2001) prior for the component means is not conditional on the precision matrix of the component as ours is. Another difference in their prior structure is that they assign a discrete distribution to the latent variables, conditioned on the mixing co-efficients. No prior is assigned to mixing co-efficients and the joint distribution over all the random variables is conditioned on them. The means and covariances of the true models the data were generated from and the fitted variational posteriors for each data-set are displayed in Tables 2.1, 2.2 and 2.3. Data-set 1 comprises 600 observations, data-set 2 comprises 900 observations and data-set 3 comprises 400 observations. In each case the correct number of components is automatically found by our method and it is clear that the method obtains good posterior estimates of the component parameters. All three data-sets are generated from models with equal mixing weights and we find good estimates of these also (the variational posterior estimates of the mixing weights for data-set 1 were 0.20,0.20,0.20,0.23,0.17, for data-set 2 they were 0.31,0.37,0.32 and for data-set 3 they were 0.34,0.30,0.36). We applied our method to these data-sets, initialising the program with 7 components (a maximum above the number of components we knew to be present), and in each case our method automatically recovered the correct number of components for the model. However, with our approach it turned out not to be necessary to use clustering methods to assign the initial means, as was done by Corduneanu and Bishop(2001). In our approach, the means were assigned independent bivariate Gaussian priors, conditional on the precision matrices, and the initial means were all set to zero. The parameter $\beta^{(0)}$ was chosen to be 0.05 to give a broad prior over the mean. The precision matrices were assigned a Wishart prior and the initial values for the degrees of freedom and the scale matrix were taken to be 2 and $[0, 0; 0, 0]$ respectively. The mixing weights were given a Dirichlet prior with the initial α's set to 0. These choices lead to improper priors for the precision matrices and the mixing weights. Figures 2.1-2.3 show the final model fitted using the variational method (dashed line) and the true distribution from which the data were generated (solid line) and in each case the method has returned a close fit to the true model. The ellipses, corresponding to each fitted component, are plotted using the variational posterior estimate of the mean and the variational posterior estimate of the covariance matrices. Each ellipse defines an area of probability content equal to 0.95 for the corresponding Gaussian distribution.

Table 2.1: True and Fitted Means and Covariances for Data-Set 1

	True Distribution		Variational Posterior	
Component	Mean	Covariance	Mean	Covariance
1	$[0, 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$[0.01, 0.03]$	$\begin{bmatrix} 1.08 & 0.05 \\ 0.05 & 1.19 \end{bmatrix}$
2	$[3, -3]$	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	$[3.19, -2.84]$	$\begin{bmatrix} 0.95 & 0.54 \\ 0.54 & 1.1 \end{bmatrix}$
3	$[3, 3]$	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$	$[3.04, 2.98]$	$\begin{bmatrix} 1.1 & -0.61 \\ -0.61 & 0.81 \end{bmatrix}$
4	$[-3, 3]$	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	$[-2.97, 2.88]$	$\begin{bmatrix} 1.07 & 0.66 \\ 0.66 & 1.17 \end{bmatrix}$
5	$[-3, -3]$	$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$	$[-2.78, -3.15]$	$\begin{bmatrix} 0.87 & -0.46 \\ -0.46 & 0.95 \end{bmatrix}$

Table 2.2: True and Fitted Means and Covariances for Data-Set 2

	True Distribution		Variational Posterior	
Component	Mean	Covariance	Mean	Covariance
1	$[0, -2]$	$\begin{bmatrix} 2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$[0.03, -2.02]$	$\begin{bmatrix} 2.28 & -0.01 \\ -0.01 & 0.24 \end{bmatrix}$
2	$[0, 0]$	$\begin{bmatrix} 2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$[0.00, -0.01]$	$\begin{bmatrix} 2.15 & -0.02 \\ -0.02 & 0.20 \end{bmatrix}$
3	$[0, 2]$	$\begin{bmatrix} 2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$[-0.13, 2.02]$	$\begin{bmatrix} 2.22 & 0.01 \\ 0.01 & 0.18 \end{bmatrix}$

Table 2.3: True and Fitted Means and Covariances for Data-Set 3

	True Distribution		Variational Posterior	
Component	Mean	Covariance	Mean	Covariance
1	$[0, 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}$	$[0.06, -0.03]$	$\begin{bmatrix} 0.96 & 0.07 \\ 0.07 & 0.18 \end{bmatrix}$
2	$[0, 0]$	$\begin{bmatrix} 0.02 & -0.08 \\ -0.08 & 1.5 \end{bmatrix}$	$[0.01, -0.01]$	$\begin{bmatrix} 0.01 & -0.08 \\ -0.08 & 1.4 \end{bmatrix}$
3	$[0, 0]$	$\begin{bmatrix} 0.5 & 0.4 \\ 0.4 & 0.5 \end{bmatrix}$	$[0.02, 0.07]$	$\begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.57 \end{bmatrix}$

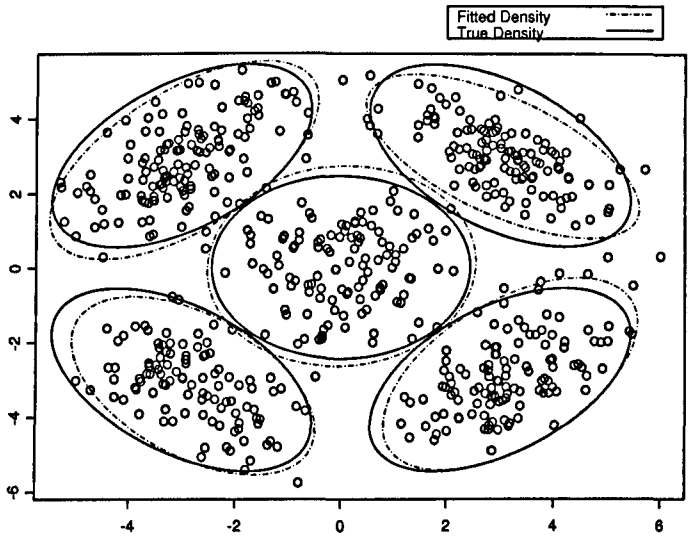

Figure 2.1: Fitted model and true distribution for data-set 1

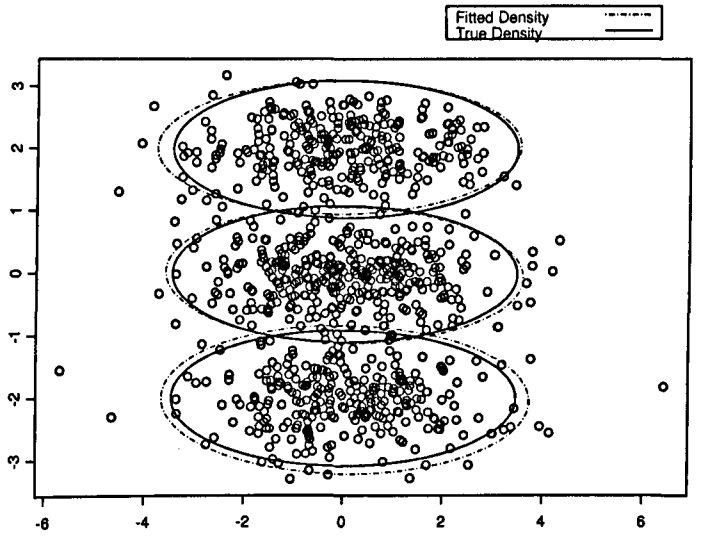

Figure 2.2: Fitted model and true distribution for data-set 2

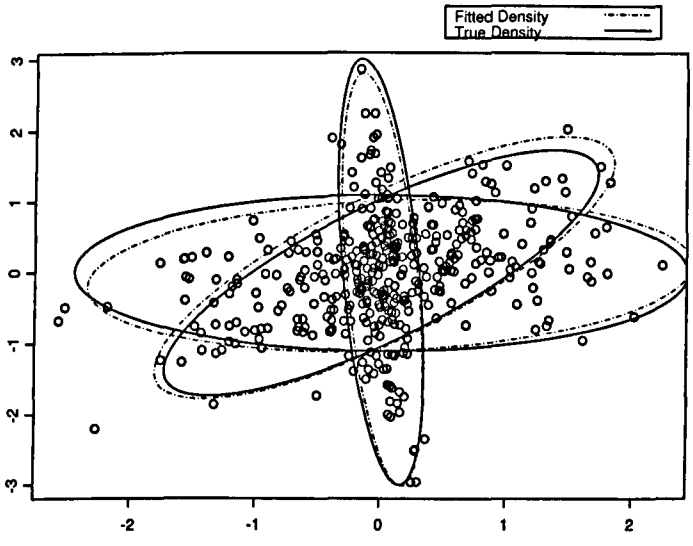

Figure 2.3: Fitted model and true distribution for data-set 3

Table 2.4 shows the DIC and p_D values obtained for each data-set for number of components ranging from 2 to 7, to allow comparison of the DIC values one would obtain in each case. It was not possible to force the program to converge with a number of components higher than that automatically selected because if superfluous components are not removed the algorithm cannot progress. To obtain an estimate of what these DIC's might be, we have reported "indicative" DIC's (highlighted by *) which were calculated as the algorithm was converging to its solution, these are not what would be obtained for fixed K. For example, the DIC for $K = 7$ components corresponds to the last DIC output by the program before the 7^{th} component was dropped. The table clearly contains scenarios in which the algorithm is initially implemented with a number of components which is smaller than the correct number. In general, if we begin with fewer components than we would reasonably expect to discover, or fewer than the number automatically selected, the algorithm does converge to a solution with this number of components, but one can see that the DIC value is higher, reflecting the 'incorrect' choice. In this way, DIC values are useful for validating the model selected using the variational method.

Comparing the DIC's and considering the model with the lowest value to be the most suitable also indicates the correct number of components for all three data-sets, so that the two methods of selection are in agreement for these examples. In general we have found that there is agreement between models selected by the variational scheme and the DIC. Looking at Table 2.4, one can see that the DIC values calculated at each stage of the iterations are decreasing as the variational scheme throws out components and converges to a solution. Solutions with fewer components than that selected by the variational approach also have higher DIC's. We found this pattern repeated with other examples we considered.

Table 2.4: DIC and (p_D) values for the three simulated data-sets

Components	Data-set 1	Data-set 2	Data-set 3
7	5187* (29.32)	6336* (25.96)	1711* (29.20)
6	5186* (26.11)	6339* (21.97)	1689* (26.72)
5	5184 (23.84)	6333* (18.08)	1703* (20.55)
4	5468 (18.87)	6331* (15.41)	1696* (15.62)
3	5577 (13.94)	6329 (13.96)	1691 (13.9)
2	5752 (8.98)	6533 (8.97)	1703 (8.9)

2.8.2 Results of Analysis of Simulated Data from Mixtures of Univariate Gaussians

To further investigate performance of the method, the program was used to analyse several simulated univariate data sets. The first such example is a simulation of 150 values from a Normal distribution with mean 0 and standard deviation 1. The result obtained for this, starting with 7 components, is given in Table 2.5. The program automatically finds the correct number of components and good estimates of the mean and standard deviation.

Table 2.5: Results for Simulation from N(0,1)

No. of Components Fitted	Mean	Standard Deviation	DIC	p_D
1	-0.078	1.008	428	1.99

Figure 2.4 shows a kernel plot of the simulated data used in the one component example. Superimposed is a plot of the exact density from which the data were generated, and a plot of the density which was fitted. This kernel plot, and all other kernel plots displayed in this thesis, were produced using the sm library for S-Plus which accompanies the book by Bowman and Azzalini (1997).

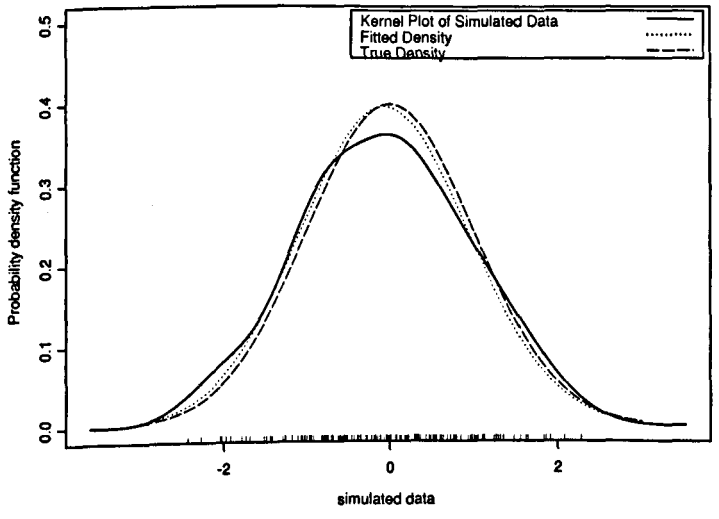

Figure 2.4: Simulated Values from N(0,1)

The next example is taken from the paper by Celeux et al. (2006). Celeux et

al. (2006) consider using different versions of the DIC for model comparison in the case of latent variable models. The underlying distribution is the 4-component Normal mixture

$$0.288N(0, 0.2) + 0.26N(-1.5, 0.5) + 0.171N(2.2, 3.4) + 0.281N(3.3, 0.5),$$

and the sample size is 146.

When started with 7 components, the program automatically recovered 4 components with means and standard deviations given in Table 2.6. The variational method finds the correct number of components and reasonably good estimates of means and variances. The DIC value for this was 599 and the p_D value was 10.83. The DIC for this model was the lowest and so the DIC also selects a 4 component mixture.

In the Celeux et al. (2006) analysis of this simulated data set, only two of the forms of the DIC they use select the correct number of components. However, these particular DIC's have negative p_D values which is not satisfactory.

Table 2.6: Results for Simulation from Mixture of 4 Normals.

Component	Mean	Standard Deviation	Mixing Weight
1	0.005	0.147	0.251
2	-1.49	0.44	0.206
3	1.36	3.3	0.296
4	3.38	0.54	0.247

Figure 2.5 displays a kernel plot of the simulated data used in the four component mixture examples. Superimposed is a plot of the exact density from which the data were generated and the density which was fitted.

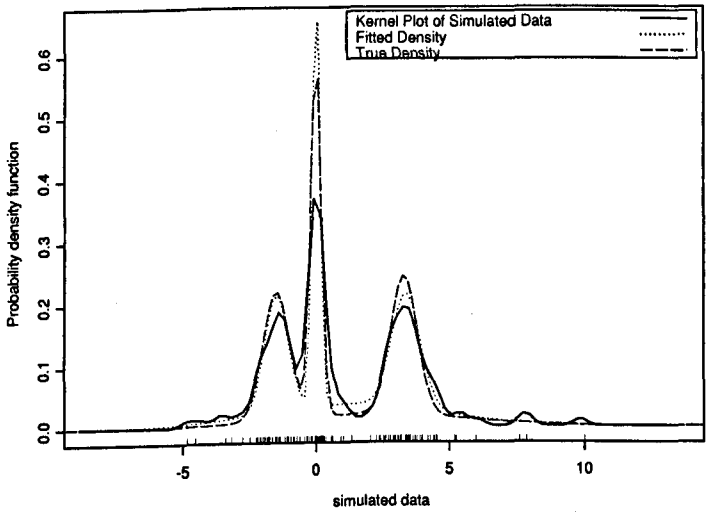

Figure 2.5: Results for Simulation from Mixture of 4 Normals

The following examples are simulated from mixtures of 2 Normal distributions with means +/- δ and standard deviation 1, the first component having weight ρ and the second having weight $1-\rho$. In each case the sample size is 150. These mixtures are of the form

$$\rho N(-\delta, 1) + (1-\rho)N(\delta, 1).$$

The sample size is not very large and, for certain combinations of δ and ρ, the resulting mixture will be unimodal making it very difficult to distinguish between components. For examples where the method can converge to more than one solution depending on how the program is initialised, the DIC value is used to select the model. Table 2.7 displays the results for the different combinations. Figures 2.6 to 2.11 show the simulated data plotted with the true densities and the fitted densities.

The analysis of the first and fourth mixtures selects 1 component in the mixture rather than two; however the two mixtures are not well separated as one can see from Figures 2.6 and 2.9. The method chooses 3 components for the fifth mixture. In the remaining cases the method recovers the correct number of components.

Table 2.7: Results for Simulation from Mixtures of 2 Normals.

ρ	δ	Components Fitted	Means	Standard Deviations	Mixing Weights
0.25	0.5	1	0.33	1.12	1
0.25	1	2	-2.36 0.8	0.45 1.15	0.07 0.93
0.25	2	2	-1.97 2.18	0.84 0.95	0.29 0.71
0.5	0.5	1	0.07	1.12	1
0.5	1	3	-1.53 -0.14 1.12	0.93 0.17 0.79	0.39 0.1 0.51
0.5	2	2	-2.11 1.77	0.9 1.14	0.47 0.53

Table 2.8: DIC and p_D Values for Simulated Mixtures of 2 Normals.

ρ	δ	DIC	p_D
0.25	0.5	464	1.99
0.25	1	524	4.87
0.25	2	584	4.95
0.5	0.5	466	1.99
0.5	1	546	4.94
0.5	2	630	4.96

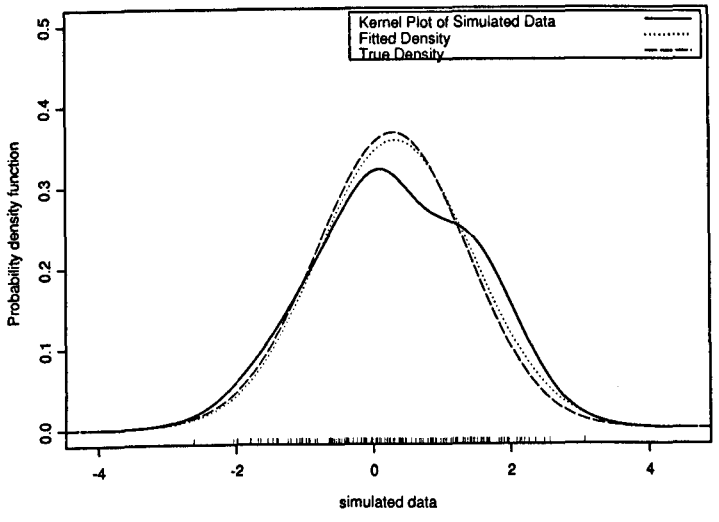

Figure 2.6: Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 0.5$

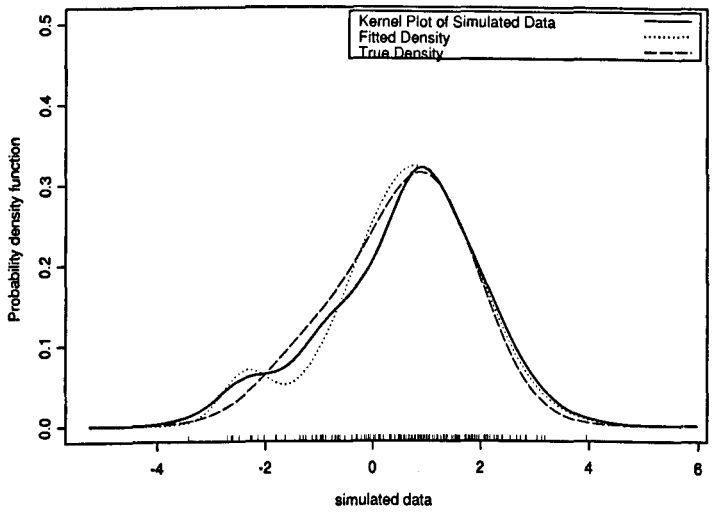

Figure 2.7: Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 1$

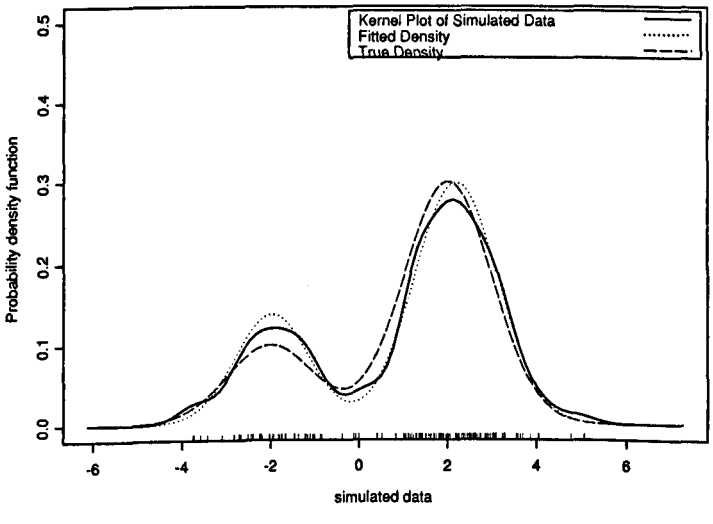

Figure 2.8: Simulated Values from Mixture of 2 Normals with $\rho = 0.25$ and $\delta = 2$

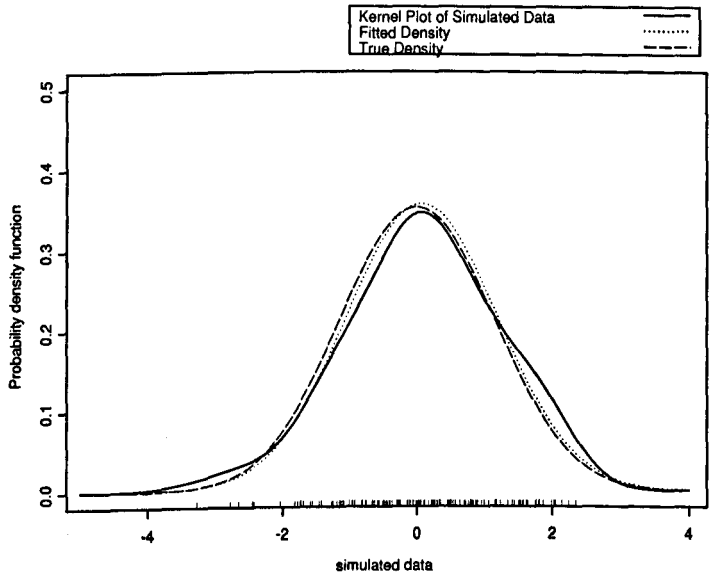

Figure 2.9: Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 0.5$

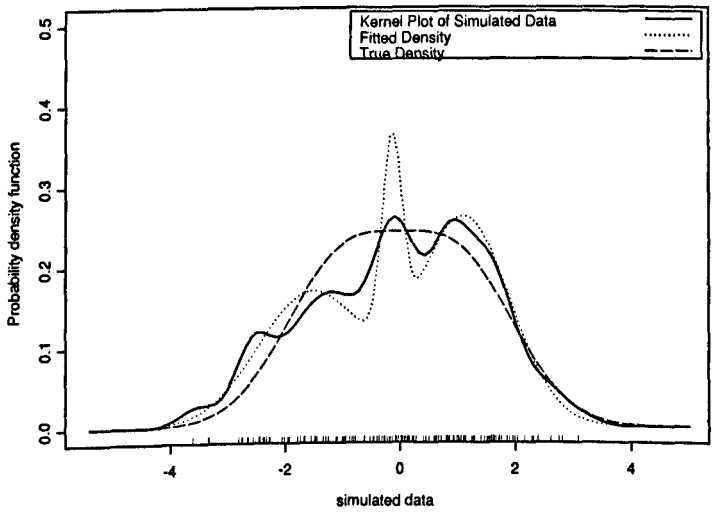

Figure 2.10: Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 1$

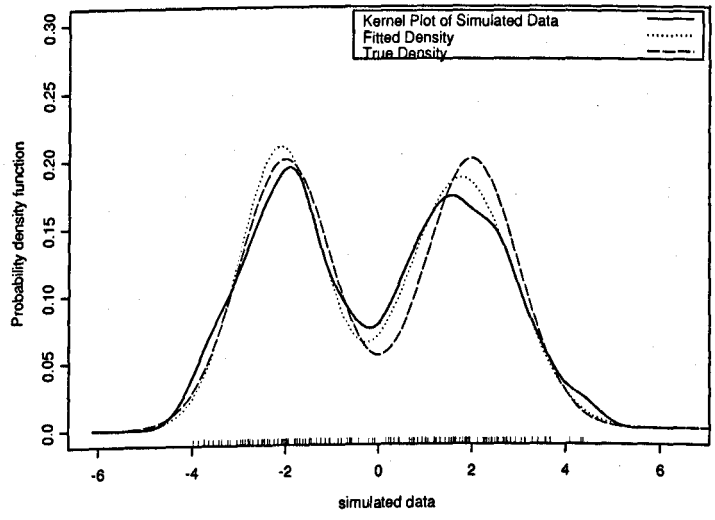

Figure 2.11: Simulated Values from Mixture of 2 Normals with $\rho = 0.5$ and $\delta = 2$

2.8.3 Analysis of Real Data Sets

The real data are the familiar examples of the galaxy, acidity and enzyme data analysed by Richardson and Green (1997) and Corduneanu and Bishop (2001), among others. A copy of these data sets is given in Appendix E.1.

Galaxy Data

This data set comprises the velocities (in 10^3 km/s) of 82 distant galaxies, diverging from our own galaxy. The observations come from six well-separated conic regions of the corona Borealis. Multimodality of the velocities is of interest as it might suggest the presence of superclusters of galaxies which are surrounded by large voids (since distance of a galaxy is proportional to the measured velocity), each mode being interpreted as a cluster moving away at a particular speed (more detail is given in Roeder (1990)). The original data set was analysed by Postman et al. (1986) and contained 83 observations but when it was analysed by Roeder (1990) one of the observations was removed. It is this data set which was subsequently analysed under different mixture models by several authors including Richardson and Green (1997) and Stephens (2000). We analyse this same data set to allow comparison of results.

Acidity Data

This data is from an acid neutralising capacity (ANC) index measured in a sample of 155 lakes in North-central Wisconsin, United States. Acidification is an environmental problem and identifying different subpopulations of lake (e.g. at risk lakes, not at risk lakes) can be useful in determining which lake characteristics, if any, can be used to predict higher acidification. This data set was previously analysed as a mixture of Gaussian distributions on the log scale by Crawford et al. (1992).

Enzyme Data

This data set concerns the distribution of enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances, among a group of 245 unrelated individuals. The study was undertaken to validate caffeine as a probe drug to establish the genetic status of rapid metabolisers and slow metabolisers and to use such subgroups as a marker of genetic polymorphism in

the general population. This data set was analysed by Bechtel et al. (1993), who fitted a mixture of 2 skewed distributions using maximum likelihood techniques.

We display the results we obtained for each of these data sets when our program was started with 7 initial components. We chose 7 as the maximum number of components as it seemed unlikely that we would fit any more components than this to any of these data sets. Upon convergence, the method finds 3, 2 and 4 components for these data sets, respectively. The variational posterior means of all parameters are given in Table 2.9 and the DIC and p_D values are displayed in Table 2.10. Figures 2.12-2.14 show plots of kernel-based density estimates based on the actual data, together with the Gaussian mixture based on the estimates given in Table 2.9. The kernel plots were produced using a kernel smoothing function with constant bandwidth. For some data-sets, there will occasionally be convergence to another solution for certain values of the initial starting weights given to the components. However, it is important to note that these alternative solutions have higher DIC values. Also, we found that the occurrence of alternative solutions seems to become less frequent as the initial number of components increases and eventually one will obtain the same answer for any initialisation. So far we have considered what happens if the program begins with a number of components which seems larger than the number we would expect to have. Alternatively, if one begins with a number of components which is less than the number selected via the variational approach then the program will still converge with this number of components. The result will have a DIC value which is higher, reflecting this. The DIC's given below in Table 2.10 correspond to the lowest that one could possibly obtain by starting with any number of components for each data set.

The treatment of the enzyme data by Richardson and Green (1997) produced similar results to our method. Their method, which produced a set of probabilities associated with different numbers of components, favoured a choice of between 3 and 5 components for the data, the highest posterior probability being for 4 components which is the same as the number of components selected by our method. Their analysis of the galaxy and acidity data sets favoured a slightly higher number of components than was selected here. They estimated there to be between 5 and 7 components for the galaxy data, the highest probability being for 6 components. For the acidity data their posterior distribution estimated between 3 and 5 components, with 3 having the highest probability. Corduneanu

and Bishop (2001) also analysed these three data sets with results similar to those of Richardson and Green (1997).

Celeux et al. (2006) analyse the galaxy data-set with several versions of the DIC. Each version indicates that there are 3 components in the mixture which corresponds to our selection. Stephens (2000) analyses the galaxy data by fitting a mixture of t densities and a mixture of Normal densities, and the posterior over the number of components selects 4 and 3 components for each fit, respectively. This is also similar to our result.

Table 2.9: Number of components fitted and posterior estimates of means, variances and mixing weights for the three real data sets

Data	Components	Means	Variances	Mixing Weights
Galaxy	3	9.64 21.35 31.58	0.6589 4.8875 23.31	0.085 0.872 0.043
Acidity	2	4.32 6.23	0.144 0.304	0.59 0.41
Enzyme	4	0.16 0.31 1.05 1.49	0.003 0.003 0.034 0.282	0.48 0.13 0.17 0.22

Table 2.10: DIC and p_D values for the three real data sets

Data	DIC	p_D
Galaxy	430	7.51
Acidity	380	4.96
Enzyme	104	10.88

The figures below show a plot of a kernel-based density estimate based on the actual data. Superimposed upon this is the exact density fitted using the program which was constructed using the posterior means and variances fitted to the parameters and the mixing weights which were assigned to each component. These plots were produced using a kernel smoothing function with constant bandwidth.

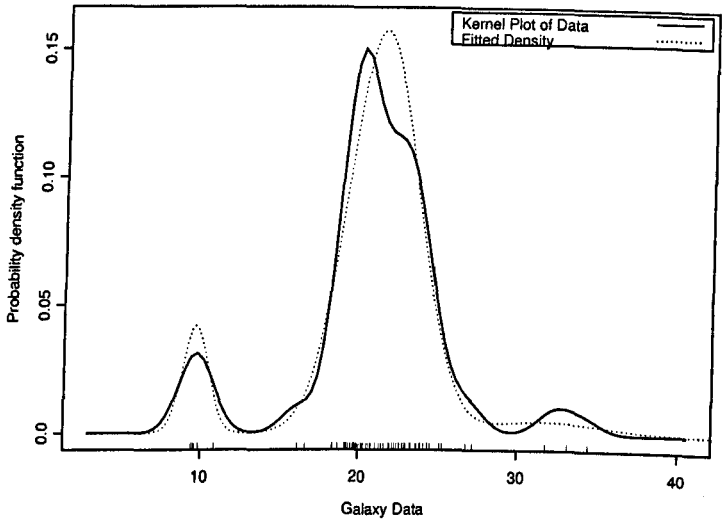

Figure 2.12: Galaxy Data

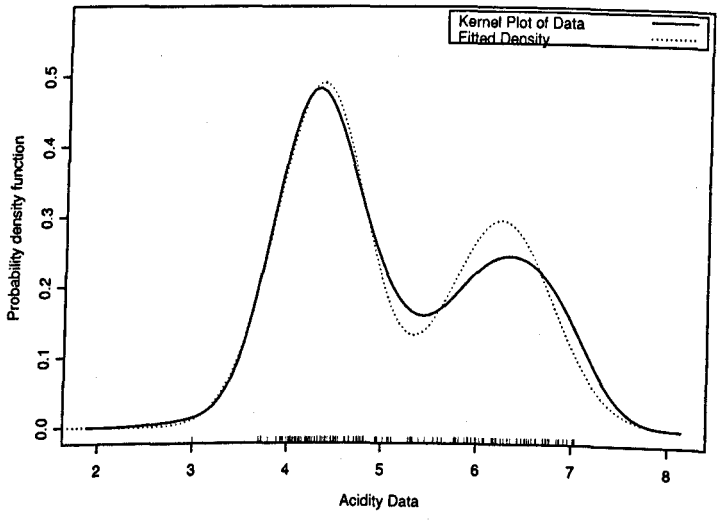

Figure 2.13: Acidity Data

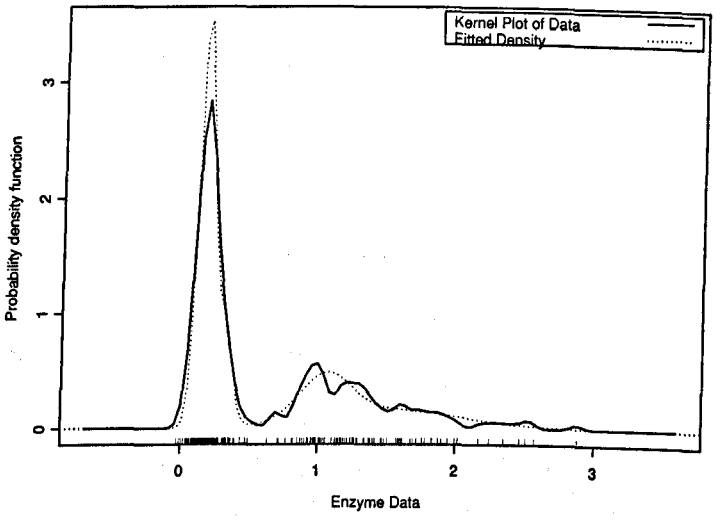

Figure 2.14: Enzyme Data

2.9 Conclusions

We have shown in this chapter how application of variational methods to model selection in the case of mixtures of Gaussian distributions leads to an automatic choice of model. For the simulated data sets considered, our variational scheme has found good estimates of the number of components and posterior estimates of components parameters. For the real data sets, we obtained results which seem to be reasonable fits to the observed data and which are comparable with models fitted via other methods.

We have also shown how variational techniques can be used to extend the DIC framework to include the comparison of mixture models. Furthermore, the models indicated as being most suitable according to the DIC values obtained by this approximating method correspond to the models automatically selected through the variational approach, and it therefore seems feasible that DIC values could be used to validate the model selection.

The variational method is computationally efficient. For example, convergence of our method and calculation of p_D and DIC values for the galaxy data analysed in section 2.8.3 took 4.49 seconds to run in R on a Windows NT Intel P4 2GHz workstation. Analysis of the Enzyme data set took 62.6 seconds. This compares favourably with the computational expense of MCMC based methods. For instance, Richardson and Green's (1997) analysis involves 200 000 MCMC sweeps for the enzyme data set set and they report that their program makes around 160 sweeps per second on a SUN SPARC 4 workstation. The running time involved would be around 1250 seconds (about 21 minutes). Stephens (2000) reports a time of between 150 and 250 seconds to fit different mixture models to the galaxy data and making 20 000 iterations using a Sun Ultrasparc 200 workstation, 1997. The longest example to run of those we considered was the analysis of the 900 simulated data points from a mixture of 5 bivariate Gaussians. This took 1487 seconds which is approximately 24 minutes. So, even the longest running time for our variational method was fairly short which demonstrates how fast results can be obtained using this approach.

In this chapter we have demonstrated how the variational approximation method is straightforward to implement, produces good results and is computationally efficient for the analysis of finite mixtures of Gaussians. We have considered other methods of analysis, in particular MCMC methods which are

difficult to implement and monitor and are more time consuming, and so we have shown that the variational approach is a practical and useful alternative to MCMC analysis of finite mixture data.

Chapter 3

Application of the Variational Approach to Mixtures of Exponential Family Models

We now consider how the variational approach can be taken with mixtures of other exponential family distributions. In particular, we consider mixtures of Poisson and Exponential models but first we show the general results for any member of the exponential family in the one-parameter case. In each case we will consider a data set $y_1, ..., y_n$ which we assume has been drawn independently from the relevant mixture model with parameters ϕ, and, as before, we introduce a set of missing binary indicator variables $\{z_{ij}\}$, $(i = 1, ..., n, j = 1, ..., K)$ to describe which component gave rise to a particular observation. The $\{z_{ij}\}$ are defined so that if observation y_i is from component m, say, then

$$\begin{aligned} z_{ij} &= 1 \quad \text{if} \quad j = m \\ &= 0 \quad \text{if} \quad j \neq m. \end{aligned}$$

Mixtures of densities which are members of the exponential family have the general complete-data form

$$p(y, z|\phi, \rho) = \prod_{i=1}^{n} \prod_{j=1}^{K} \rho_j{}^{z_{ij}} \left[s(y_i) t(\phi_j) \exp\{a(y_i) b(\phi_j)\} \right]^{z_{ij}} .$$

where $b(\phi_j)$ is called the natural parameter and $a(y_i)$, $s(y_i)$ and $t(\phi_j)$ are functions

which define the exponential family. The conjugate prior for the parameters will have the form

$$p(\phi, \rho|\eta, \nu) \propto \prod_{j=1}^{K} \rho_j^{\alpha_j^{(0)}-1} \prod_{j=1}^{K} w(\eta_j^{(0)}, \nu_j^{(0)}) t(\phi_j)^{\eta_j^{(0)}} \exp\{\nu_j^{(0)} b(\phi_j)\}.$$

Therefore,

$$\begin{aligned} p(y, z, \phi, \rho|\eta, \nu) \propto & \prod_{i=1}^{n} \prod_{j=1}^{K} \rho_j^{z_{ij}} [s(y_i) t(\phi_j) \exp\{a(y_i) b(\phi_j)\}]^{z_{ij}} \\ & \times \prod_{j=1}^{K} \rho_j^{\alpha_j^{(0)}-1} \prod_{j=1}^{K} w(\eta_j^{(0)}, \nu_j^{(0)}) t(\phi_j)^{\eta_j^{(0)}} \exp\{\nu_j^{(0)} b(\phi_j)\}. \end{aligned}$$

The variational approximation for the posterior is defined by the factorisation $q(z, \theta) = q(z, \phi, \rho) = \prod_{i=1}^{n} q_{z_i}(z_i) q_\phi(\phi) q_\rho(\rho)$, and in fact $q_\phi(\phi) = \prod_{j=1}^{K} q_{\phi_j}(\phi_j)$. We now derive the form of the variational posterior for z. We find the form of the variational posterior, $q_{z_i}(z_i = j)$, by maximising the lower bound on the marginal log-likelihood.

$$\begin{aligned} \sum_z & \int \{\prod_{i=1}^{n} q_{z_i}(z_i)\} q_\phi(\phi) q_\rho(\rho) \log \frac{p(\phi, \rho) \prod_{i=1}^{n} p(y_i, z_i|\phi, \rho)}{\{\prod_{i=1}^{n} q_{z_i}(z_i)\} q_\phi(\phi) q_\rho(\rho)} d\phi d\rho \\ = & \sum_j \int q_{z_i}(z_i = j) q_\phi(\phi) q_\rho(\rho) \log \frac{p(y_i, z_i = j|\phi, \rho)}{q_{z_i}(z_i = j)} d\phi d\rho \\ & + \text{ terms independent of } q_{z_i} \\ = & \sum_j q_{z_i}(z_i = j) \{ \int q_\phi(\phi) q_\rho(\rho) \log p(y_i, z_i = j|\phi, \rho) d\phi d\rho - \log q_{z_i}(z_i = j) \} \\ & + \text{ terms independent of } q_{z_i} \\ = & \sum_j q_{z_i}(z_i = j) \log [\frac{\exp \int q_\phi(\phi) q_\rho(\rho) \log p(y_i, z_i = j|\phi, \rho) d\phi d\rho}{q_{z_i}(z_i = j)}] \\ & + \text{ terms independent of } q_{z_i}. \end{aligned}$$

Therefore,

$$q_{z_i}(z_i = j) \propto \exp\{\int q_\phi(\phi) q_\rho(\rho) \log p(y_i, z_i = j|\phi, \rho) d\phi d\rho\}.$$

Substituting the general forms for the mixture density and the prior density gives us

$$\begin{aligned} q_{z_i}(z_i = j) &\propto \exp\{\int q_\phi(\phi) q_\rho(\rho) \left[\log \rho_j + \log t(\phi_j) + a(y_i) b(\phi_j)\right] d\phi_j d\rho\} \\ &= \exp\{\mathbf{E}_q[\log \rho_j] + \mathbf{E}_q[\log t(\phi_j)] + a(y_i)\mathbf{E}_q[b(\phi_j)]\}. \end{aligned}$$

In a similar way, we obtain the form of the variational approximation to the posteriors for the mixing weights, ρ, and model parameters, ϕ_j, by focusing on relevant parts of the lower bound,

$$\sum_z \int \{\prod_{i=1}^n q_{z_i}(z_i)\} q_\phi(\phi) q_\rho(\rho) \log \frac{p(\phi, \rho) \prod_{i=1}^n p(y_i, z_i | \phi, \rho)}{\{\prod_{i=1}^n q_{z_i}(z_i)\} q_\phi(\phi) q_\rho(\rho)} d\phi dp.$$

It turns out that

$$\begin{aligned} q_\rho(\rho) &\propto \prod_{j=1}^K \rho_j^{\alpha_j^{(0)} + \sum_{i=1}^n q_{ij} - 1} \\ &= \prod_{j=1}^K \rho_j^{\alpha_j - 1}, \end{aligned}$$

where $q_{ij} = q_{z_i}(z_i = j)$ and

$$\alpha_j = \alpha_j^{(0)} + \sum_{i=1}^n q_{ij}.$$

The variational posterior for the parameter ϕ_j is of the form

$$\begin{aligned} q_{\phi_j}(\phi_j) &\propto t(\phi_j)^{\sum_{i=1}^n \mathbf{E}_{q_{z_i}}[z_{ij}]} \exp[\sum_{i=1}^n \mathbf{E}_{q_{z_i}}[z_{ij}] a(y_i) b(\phi_j)] t(\phi_j)^{\eta_j^{(0)}} \exp[\nu_j^{(0)} b(\phi_j)] \\ &= t(\phi_j)^{\sum_{i=1}^n q_{ij} + \eta_j^{(0)}} \exp\left[\left\{\sum_{i=1}^n q_{ij} a(y_i) + \nu_j^{(0)}\right\} b(\phi_j)\right] \\ &= t(\phi_j)^{\eta_j} \exp\left[\nu_j b(\phi_j)\right], \end{aligned}$$

and

$$\eta_j = \eta_j^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\nu_j = \nu_j^{(0)} + \sum_{i=1}^{n} q_{ij} a(y_i).$$

It would be straightforward to extend these formulae to the multidimensional case by treating ϕ as a vector rather than a scalar.

We now look at the variational approach for two examples of exponential family distributions, namely, the Poisson distribution and the Exponential distribution.

3.1 The Poisson Distribution

In this case, we have a model of the form

$$\begin{aligned} p(y, z|\theta) &= \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j \mathrm{Po}(y_i; \phi_j)\}^{z_{ij}} \\ &= \prod_{i=1}^{n} \prod_{j=1}^{K} \left\{ \rho_j \exp(-\phi_j) \frac{{\phi_j}^{y_i}}{y_i!} \right\}^{z_{ij}}. \end{aligned}$$

Assigning the Prior Distributions

The mixing weights are given a Dirichlet prior distribution,

$$p(\rho) = \mathrm{Dir}(\rho; {\alpha_1}^{(0)}, ..., {\alpha_K}^{(0)}).$$

The means are given independent Gamma conjugate prior distributions, with hyperparameters $\{{\gamma_j}^{(0)}\}$ and $\{{\beta_j}^{(0)}\}$, so that

$$p(\phi) = \prod_{j=1}^{K} \mathrm{Ga}(\phi_j; {\gamma_j}^{(0)}, {\beta_j}^{(0)}),$$

which gives,

$$
\begin{aligned}
p(\theta) &= p(\rho)p(\phi) \\
&\propto \prod_{j=1}^{K} \rho_j^{\alpha_j{}^{(0)}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}-1} \exp(-\beta_j{}^{(0)}\phi_j)
\end{aligned}
$$

and

$$
\begin{aligned}
p(y,z,\theta) &= p(y,z|\theta)p(\theta) \\
&\propto \prod_{i=1}^{n}\prod_{j=1}^{K} \rho_j{}^{z_{ij}} \exp(-\phi_j z_{ij}) \phi_j{}^{y_i z_{ij}} \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}-1} \exp(-\beta_j{}^{(0)}\phi_j) \\
&= \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)}+\sum_{j=1}^{K} z_{ij}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}+\sum_{j=1}^{K} y_i z_{ij}-1} \exp[-\phi_j(\beta_j{}^{(0)} + \sum_{j=1}^{K} z_{ij})]
\end{aligned}
$$

Form of the Variational Posterior Distributions

Clearly, the variational posteriors will turn out to have the conjugate forms (see Appendix F.1 for details):

$$
\begin{aligned}
q_\rho(\rho) &= \mathrm{Dir}(\rho; \alpha_1, ..., \alpha_K) \\
&\propto \prod_{j=1}^{K} \rho_j{}^{\alpha_j - 1}
\end{aligned}
$$

$$
\begin{aligned}
q_\phi(\phi) &= \prod_{j=1}^{K} \mathrm{Ga}(\phi_j; \gamma_j, \beta_j) \\
&\propto \prod_{j=1}^{K} \phi_j{}^{\gamma_j - 1} \exp(-\phi_j \beta_j)
\end{aligned}
$$

with,

$$
\alpha_j = \alpha_j{}^{(0)} + \sum_{j=1}^{K} q_{ij}
$$

$$\gamma_j = \gamma_j^{(0)} + \sum_{j=1}^{K} y_i q_{ij}$$

and

$$\beta_j = \beta_j^{(0)} + \sum_{j=1}^{K} q_{ij}.$$

In addition,

$$\begin{aligned} q_{ij} = q_{z_i}(z_i = j) &= \frac{\exp\{\mathbf{E}_q[\log \rho_j] - \mathbf{E}_q[\phi_j] + y_i \mathbf{E}_q[\log \phi_j]\}}{s_i} \\ &= \frac{\varphi_{ij}}{s_i} \end{aligned}$$

say, where $s_i = \sum_{j=1}^{K} \varphi_{ij}$.

$$\mathbf{E}_q[\phi_j] = \frac{\gamma_j}{\beta_j}$$

and $\mathbf{E}_q[\log \rho_j]$ and $\mathbf{E}_q[\log \phi_j]$ can be evaluated using the digamma function.

3.2 The Exponential Distribution

Here, we have a model of the form

$$\begin{aligned} p(y, z|\theta) &= \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j \mathrm{Ex}(y_i; \phi_j)\}^{z_{ij}} \\ &= \prod_{i=1}^{n} \prod_{j=1}^{K} \{\rho_j \phi_j \exp(-\phi_j y_i)\}^{z_{ij}} \end{aligned}$$

Assigning the Prior Distributions

The mixing weights are given a Dirichlet prior distribution,

$$p(\rho) = \mathrm{Dir}(\rho; \alpha_1^{(0)}, ..., \alpha_K^{(0)}).$$

The exponential rates are given independent Gamma conjugate prior distributions, with hyperparameters $\{\gamma_j{}^{(0)}\}$ and $\{\beta_j{}^{(0)}\}$, so that

$$p(\phi) = \prod_{j=1}^{K} \mathrm{Ga}(\phi_j; \gamma_j{}^{(0)}, \beta_j{}^{(0)}),$$

giving

$$\begin{aligned} p(\theta) &= p(\rho)p(\phi) \\ &\propto \prod_{j=1}^{K} \rho_j^{\alpha_j{}^{(0)}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}-1} \exp(-\beta_j{}^{(0)}\phi_j) \end{aligned}$$

and

$$\begin{aligned} p(y,z,\theta) &= p(y,z|\theta)p(\theta) \\ &\propto \prod_{i=1}^{n}\prod_{j=1}^{K} \rho_j{}^{z_{ij}} \phi_j{}^{z_{ij}} \exp(-\phi_j y_i z_{ij}) \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}-1} \exp(-\beta_j{}^{(0)}\phi_j) \\ &= \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)}+\sum_{j=1}^{K} z_{ij}-1} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)}+\sum_{j=1}^{K} z_{ij}-1} \exp[-\phi_j(\beta_j{}^{(0)} + \sum_{j=1}^{K} y_i z_{ij})]. \end{aligned}$$

Form of the Variational Posterior Distributions

Again, the variational posteriors will turn out to have the conjugate forms (see Appendix F.2 for details):

$$\begin{aligned} q_\rho(\rho) &= \mathrm{Dir}(\rho; \alpha_1, ..., \alpha_K) \\ &\propto \prod_{j=1}^{K} \rho_j^{\alpha_j-1} \end{aligned}$$

$$
\begin{aligned}
q_\phi(\phi) &= \prod_{j=1}^{K} \mathrm{Ga}(\phi_j; \gamma_j, \beta_j) \\
&\propto \prod_{j=1}^{K} {\phi_j}^{\gamma_j - 1} \exp(-\phi_j \beta_j)
\end{aligned}
$$

with,

$$
\alpha_j = {\alpha_j}^{(0)} + \sum_{j=1}^{K} q_{ij}
$$

$$
\gamma_j = {\gamma_j}^{(0)} + \sum_{j=1}^{K} q_{ij}
$$

and

$$
\beta_j = {\beta_j}^{(0)} + \sum_{j=1}^{K} y_i q_{ij}.
$$

In addition,

$$
\begin{aligned}
q_{ij} = q_{z_i}(z_i = j) &= \frac{\exp\{\mathbf{E}_q[\log \rho_j] - \mathbf{E}_q[\log \phi_j] - y_i \mathbf{E}_q[\phi_j]\}}{s_i} \\
&= \frac{\phi_{ij}}{s_i}
\end{aligned}
$$

say, where $s_i = \sum_{j=1}^{K} \phi_{ij}$.

$$
\mathbf{E}_q[\phi_j] = \frac{\gamma_j}{\beta_j}
$$

and $\mathbf{E}_q[\log \rho_j]$ and $\mathbf{E}_q[\log \phi_j]$ can be evaluated using the digamma function.

Chapter 4

Bayesian Analysis of Hidden Markov Models

In this chapter we will extend the ideas of the previous chapters to apply them to Bayesian inference for hidden Markov models. So far we have dealt with finite mixture distributions, where the missing indicator variables for each observation are considered to be independent. This model structure can be thought of as the 'standard' one for mixture models but extensions to this are possible. If instead we assume that the missing indicator variables that determine which component gave rise to a particular observation are not independent, but are governed by a stationary Markov chain, this results in the hidden Markov chain structure, most commonly referred to as a hidden Markov model (HMM) structure. This model structure is suitable for modelling data that varies over time and can be thought of as having been generated by a process which switches between different phases or states at different times, such as speech or stock market data. These states are the components of the Markov mixture model and the particular sequence of states which gave rise to an observation set is unobserved i.e. the states are 'hidden'. The extension to the hidden Markov structure presents new challenges for Bayesian inference and here we discuss these together with possible solutions.

4.1 An Introduction to Hidden Markov Models

4.1.1 Origins and Applications of Hidden Markov Modelling

The theory of hidden Markov modelling was initially introduced in the 1960s by Leonard E. Baum and Ted Petrie (1966), who considered inference for stationary, ergodic, finite state Markov chains where observations could only take values in a finite set. These ideas were developed further by Baum, Petrie, and their colleagues at the Institute for Defense Analyses, in the following years (Baum and Egon (1967), Baum and Sell (1968), Petrie (1969), Baum et al. (1970), Baum (1972)). Baum et al. (1970) made a significant contribution to the progress in inference for HMM's by introducing an early version of the EM algorithm for maximum likelihood estimation as well as presenting the well known forward backward procedure (also known as the Baum-Welch procedure) which removes the computational difficulties attached to calculating the likelihood and obtaining estimates of parameters for HMM's. Baum and Petrie proved the consistency and asymptotic normality of the maximum likelihood estimator for these finite set HMM's. Later this was extended by Leroux (1992), who established the conditions under which the maximum likelihood estimator is consistent for general HMM's, and Bickel et al. (1998), who established asymptotic normality of the maximum likelihood estimator for general HMM's.

HMM's were quickly applied to speech recognition, see Baker (1975) and Jelinek et al. (1975), for example. Since then, there has been a lot of interest in using HMM's for automatic speech recognition and nowadays practically all speech recognition systems use HMM's. There is a wealth of material available on this subject. Ferguson (1980), Bahl et al. (1983) and Juang and Rabiner (1991), are some examples, and the tutorial by Rabiner (1989) provides an excellent introduction to hidden Markov models, discusses practical implementation and considers application to speech recognition. These studies on speech recognition popularised the theory of HMM's and these models have since been applied to a wide range of applications where data can be grouped into components or states and there is a time dependency between observations. Examples include biometrics problems such as gene sequencing (Churchill (1995) and Boys et al. (2000), for example), econometrics (Chib (1996), for instance) and finance (see

Rydén et al. (1998)). MacDonald and Zucchini (1997) is a recent text on the subject of hidden Markov modelling.

4.1.2 The Characteristics of a Hidden Markov Model

A hidden Markov model (HMM) is a stochastic process generated by a stationary Markov chain whose state sequence cannot be directly observed. A Markov model assumes that a system can be in one of K states, $1, ..., K$, at a given time point i ($i = 1, 2, ...$ are evenly spaced time points). At each time point the system changes to a different state or stays in the same state. A discrete first-order Markov model has the property that the probability of occupying a state, z_i, at time i, depends only on the state occupied at the previous time point i.e.

$$p(z_i = j_3 | z_{i-1} = j_2, z_{i-2} = j_1, ...) = p(z_i = j_3 | z_{i-1} = j_2), \quad 1 \leq j_1, j_2, j_3 \leq K.$$

It is possible to define higher-order Markov models, although these are encountered less frequently, but these shall not be considered here. To define a HMM, we begin with an initial state probability distribution,

$$\pi_j = p(z_1 = j), \quad 1 \leq j \leq K.$$

This is the probability that the Markov chain is in a particular state at the first time point. Then, the probability of moving from one state to another is characterised by a transition matrix

$$\pi = \{\pi_{j_1 j_2}\},$$

where

$$\pi_{j_1 j_2} = p(z_i = j_2 | z_{i-1} = j_1), \quad 1 \leq j_1, j_2 \leq K,$$

$\pi_{j_1 j_2} \geq 0$ and $\sum_{j_2=1}^{K} \pi_{j_1 j_2} = 1$, for each j_1. The transition probabilities represent the probability that the system is in state j_2 at time i given that it was in state j_1 at time $i-1$. As mentioned above, we assume that the Markov chain is stationary (time invariant); in other words, the state transition probabilities are independent of the actual time point at which the transition takes place.

In a HMM, the state sequence for observations is 'hidden', and instead what we observe is a probabilistic function of the state. In other words we have 'noisy' observations. There is a density function associated with each state and the probabilities attached to each observation are known as emission probabilities. However, despite the fact that we do not observe the states, for many real world applications, different states, or sets of states, have a physical interpretation attached to them. For instance, in speech recognition, the hidden states might be different words or phonemes and, in gene sequencing, hidden states could correspond to segments of interest within a DNA sequence. For instance, Boys et al. (2000) use HMM's to identify homogeneous segments within DNA sequences. This is of interest as these segments may have a functional role, and locating them allows researchers to study them and gain more insight into their purpose. The Bayesian HMM approach taken by Boys et al. (2000), allows the incorporation of prior knowledge about significant aspects of the sequence. The observations are the DNA bases A,C,T and G which make up the sequence, and the hidden states, in this application, are the homogeneous segment types. The approach taken by Boys et al. (2000) uses Gibbs sampling with data augmentation. This method involves repeatedly simulating a sequence of segmentation types, conditional on the parameters, then simulating the parameters, conditional on the segmentation sequence. Simulating the segmentation sequence makes use of a forward backward iterative scheme. Their analysis shows that HMM's are an effective tool for modelling DNA regions and locating homogeneous segments.

If the observations can be thought of as discrete symbols from a finite set, then a discrete density can be used within each state, but in many applications observations are continuous outputs and so a continuous observation density is more suitable. Throughout this chapter we will assume Gaussian observation densities with unknown means and variances:

$$p_j(y_i|z_i = j) \sim N(y_i; \mu_j, \tau_j^{-1}).$$

4.1.3 Problems of Inference for Hidden Markov Models and their Solutions

Given a set of observations from a hidden Markov model there are three main problems of inference (these are described in detail in the tutorial by Rabiner

(1989)):

1. calculating the likelihood of an observed sequence given a certain HMM
2. finding the best state sequence to account for the observed data
3. finding the best estimates of the model parameters to account for the observed data.

We are primarily concerned with problems 1 and 3. Problem 2 corresponds to finding the 'hidden' part of the model i.e. the 'correct' state sequence. However, of course, there is no 'correct' sequence to find, so instead we would look for an optimal sequence. This may be of interest to discover more about the model structure or to find the best state sequence for a continuous speech processing problem, for example. This problem is usually solved using the Viterbi algorithm (Viterbi (1967)) which finds, via recursion, the whole state sequence with the maximum likelihood. However, we will not consider this aspect of inference any further.

Problem 3, which corresponds to optimising the model parameters to find the best description of how our observation sequence was generated, often referred to as 'training' the model, is an important problem as it allows us to build the best models to represent real phenomena. It is also the most difficult inference problem associated with HMM's.

Calculation of the likelihood is not as straightforward for HMM's as it was in the case of finite mixtures. Essentially we want to compute the marginal probability of observing a given set of data, marginalising over all state sequences. The 'straightforward' way to do this would be to consider every possible state sequence for the n time points. Unfortunately this would involve of the order of $2n \times K^n$ calculations, which is clearly computationally infeasible. Fortunately there is an algorithm known as the forward backward algorithm (Baum et al. (1970)) which provides an efficient way of performing these calculations. The forward backward algorithm leads to an expectation-maximization algorithm that can be used to find the unknown parameters of a hidden Markov model. It computes the maximum likelihood estimates for the parameters of an HMM, given a set of observations and so it also provides a solution to problem 3. In fact, evaluation of the likelihood only involves the forward part of the algorithm; the likelihood is obtained by summing over the final forward variable but problem

3 can be solved using both the forward and backward parts. Details are provided later.

The algorithm for solving problem 3 has two steps: based on some initial estimates, the first involves calculating the forward probability and the backward probability for each state, and the second determines the expected frequencies of the paired transitions and emissions. These are obtained by weighting the observed transitions and emissions by the probabilities specified in the current model. The new expected frequencies then provide new estimates, and iterations continue until there is no improvement. The method is guaranteed to converge to at least a local maximum, and estimates of the transition probabilities and parameter values can be obtained.

We apply the variational method to perform model selection (i.e. select the number of hidden states) and optimise model parameters at the same time, making use of the forward backward procedure to obtain estimates of the marginal posterior probabilities of the indicator variables. The forward backward algorithm is described in Appendix G.2. In addition, we will extend the DIC model selection criterion to this case, using the forward variables to obtain the likelihood.

4.2 Inference for Hidden Markov Models with an Unknown Number of States

Much of the original research done on HMM's was based on the premise that the number of hidden states was fixed. Also, more recently, Robert et al. (1993), Chib (1996) and Robert and Titterington (1998), for instance, consider Bayesian inference for HMM's with a fixed number of states. We are interested in the problem of inference when the number of hidden states, K, is unknown. This is still an open issue in HMM inference. Classical approaches to determining a suitable number of states include likelihood ratio tests and penalised likelihood methods such as the AIC. As with mixture models, for HMM's the underlying assumptions of the likelihood ratio test are not satisfied. Finding the limiting distribution of the likelihood ratio requires a simulation based approximation. Rydén et al. (1998) use a parametric bootstrap approximation to the limiting distribution of the likelihood ratio but this approach is severely limited by the computational intensity involved. Because of this drawback, Rydén et al. (1998)

were only able to use it to test models with number of states $K = 1$ against $K = 2$ and $K = 2$ against $K = 3$. Testing of a higher number of states would be too time consuming with this method.

Robert et al. (2000) approach the problem of determining a suitable number of states using the RJMCMC technique of Green (1995). Their method provides a posterior distribution over the possible number of states for the model up to some maximum value. This extends the ideas used in Richardson and Green (1997) for finding the number of components in a mixture model. In a similar manner, Robert et al. (2000) estimate a suitable number of states using trans-dimensional moves which either split a state in two or merge two together, these moves being accepted based on acceptance criteria. In applications, the acceptance rates for the moves to split and merge states were low, and so it would be desirable to increase these. However, this method is attractive computationally and it simultaneously performs model selection and parameter estimation. In this way, Robert et al. (2000) present a Bayesian alternative to the classical approaches for selecting the number of states.

MacKay (1997) was the first to propose applying variational methods to HMM's. He showed how variational methods could be applied to HMM's with discrete observations, assuming Dirichlet priors over the model parameters. MacKay (1997) shows that the algorithm is a modified version of the forward backward/Baum-Welch algorithm. In a short paper written in 2001, MacKay discusses some of the problems with using variational methods for mixture models or HMM's (which he notes were observed by Zhoubin Ghahramani in studying MacKay (1997)), namely that of the component or state removal effect, which he refers to as model pruning. As mentioned in Chapter 2, this phenomenon has previously been noted by other authors in the context of mixtures (Attias (1999) and Corduneanu and Bishop (2001)). MacKay expresses concern about the appropriateness of this automatic removal of extra degrees of freedom as it prevents the user from fitting the model they wish to fit and obtaining error estimates.

Despite the lack of understanding of this phenomenon, variational methods are beginning to be applied to HMM's. For instance, Lee et al. (2003) propose a variational learning algorithm for HMM's applied to continuous speech processing and conclude that variational methods have potential in this area.

Variational methods have been shown to be successful in other applications but their full potential in HMM analysis is yet to be explored.

4.3 The Variational Approach to Inference about Hidden Markov Models with an Unknown Number of States

Here we consider the application of the variational technique to a HMM with continuous observations, an unknown number of states and hidden state sequence having Gaussian noise with unknown means, variances and transition probabilities. Suppose we have n observations, corresponding to n time points, i.e. data $y_i : i = 1, ..., n$ and K states. We define our initial state probability as

$$\pi_j = p(z_1 = j),$$

which is the probability that the first state is state j. In fact, we fix our first state by setting $z_1 = 1$. The other states in the sequence are not fixed and the probability of moving from one state to another is given by the transition probabilities, stored in the transition matrix

$$\pi = \{\pi_{j_1 j_2}\}, \quad 1 \leq j_1, j_2 \leq K$$

where

$$\pi_{j_1 j_2} = p(z_{i+1} = j_2 | z_i = j_1).$$

Then, the probability of observing y_i at time point i, given that the system is in state j, is given by

$$p(y_i | z_i = j) = p_j(y_i | \phi_j),$$

where the $\{\phi_j\}$ are the parameters within the j^{th} noise model. These probabilities are often called the emission probabilities. The model parameters are given by

$$\theta = (\pi, \phi)$$

with

$$\phi = \{\phi_j\}.$$

The prior densities are assumed to satisfy

$$p(\pi, \phi) = p(\pi)p(\phi).$$

Then, the joint density of all of the variables is

$$p(y, z, \theta) = \prod_{i=1}^{n} \prod_{j=1}^{K} \{p_j(y_i|\phi_j)\}^{z_{ij}} \prod_{i=1}^{n-1} \prod_{j_1} \prod_{j_2} (\pi_{j_1 j_2})^{z_{ij_1} z_{i+1 j_2}} p(\phi)p(\pi),$$

where z_{ij} indicates which state the chain is in for a given observation and is equal to the Kronecker delta, i.e.

$$\begin{aligned} z_{ij} = \quad & 1 \quad , \quad \text{if} \quad i = j \\ & 0 \quad , \quad \text{if} \quad i \neq j. \end{aligned}$$

Therefore,

$$\int \int q(z, \theta) \log\{\frac{p(y, z, \theta)}{q(z, \theta)}\} dz d\theta$$

is of the form

$$\sum_{\{z\}} \int_{\theta} q(z, \theta) \log[\frac{\prod_{i=1}^{n} \prod_{j=1}^{K} \{p_j(y_i|\phi_j)\}^{z_{ij}} \prod_{i=1}^{n-1} \prod_{j_1} \prod_{j_2} (\pi_{j_1 j_2})^{z_{ij_1} z_{i+1 j_2}} p(\phi)p(\pi)}{q(z, \theta)}] dz d\theta,$$

where we assume that $q(z, \theta) = q_z(z)q_\theta(\theta)$. The ϕ_j's are distinct and we assume prior independence, so that

$$p(\phi) = \prod_{j=1}^{K} p_j(\phi_j)$$

We also assume prior independence among the rows of the transition matrix, and therefore $q_\theta(\theta)$ takes the form

$$q_\theta(\theta) = \prod_{j=1}^{K} q_{\phi_j}(\phi_j) \prod_{j_1} q_{j_1}(\pi_{j_1}),$$

where

$$\pi_{j_1} = \{\pi_{j_1 j_2} : j_2 = 1, ..., K\}.$$

If $p_j(y_i|\phi_j)$ represents an exponential family model and $p_j(\phi_j)$ is taken to be from an appropriate conjugate family then q_{ϕ_j} belongs to the ϕ_j conjugate family as do the q_{j_1}.

Prior to performing this analysis, we first considered the case where $q_z(z)$ takes a fully factorised form, but results for this were not satisfactory, largely because of course the hidden states are not independent.

4.4 Model Specification

Assigning the Prior Distributions

For each j_1, we assign an independent Dirichlet prior for $\{\pi_{j_1 j_2} : j_2 = 1, ..., K\}$, so that

$$p(\pi) = \prod_{j_1} \text{Dir}(\pi_{j_1} | \{\alpha_{j_1 j_2}{}^{(0)}\}).$$

We assign univariate Normals with unknown means and variances to the $p_j(y_i|\phi_j)$, the emission probabilities. Therefore,

$$p_j(y_i|\phi_j) \sim N(y_i; \mu_j, \tau_j{}^{-1})$$

where τ_j is the precision and is equal to $\frac{1}{\sigma_j{}^2}$.

As with the case of a mixture of univariate Normals, the means are assigned independent univariate Normal conjugate priors, conditional on the precisions. The precisions themselves are assigned independent Gamma prior distributions so that

$$p(\mu|\tau) = \prod_{j=1}^{K} N(\mu_j; m_j{}^{(0)}, (\beta_j{}^{(0)}\tau_j)^{-1})$$

and

$$p(\tau) = \prod_{j=1}^{K} Ga(\tau_j | \frac{1}{2}\gamma_j{}^{(0)}, \frac{1}{2}\delta_j{}^{(0)}).$$

Then

$$\sum_{\{z\}} \int q(z,\theta) \log\{\frac{p(y,z,\theta)}{q(z,\theta)}\} d\theta$$

will have the form

$$\sum_{\{z\}} \int_{\pi} q_z(z) q(\pi) q_\phi(\phi) \log \left[\frac{\begin{array}{c} \prod_{i=1}^{n} \prod_{j=1}^{K} \{\sqrt{\frac{\tau_j}{2\pi}} \exp\{-\frac{\tau_j}{2}(y_i - \mu_j)^2\}\}^{z_{ij}} \\ \times \prod_{i=1}^{n-1} \prod_{j_1} \prod_{j_2} (\pi_{j_1 j_2})^{z_{ij_1} z_{i+1 j_2}} \\ \times \prod_{j_1} \frac{1}{D(\alpha_{j_1}{}^{(0)})} \prod_{j_2} (\pi_{j_1 j_2})^{\alpha_{j_1 j_2}{}^{(0)} - 1} \end{array}}{q_z(z) q(\pi) q_\phi(\phi)} \right] d\pi d\phi.$$

Form of the Variational Posterior Distributions

As in the case of a mixture of univariate Gaussians, the variational posteriors for the model parameters turn out to have the following forms:

$$q_{j_1}(\pi_{j_1}) \sim \mathrm{Dir}(\pi_{j_1} | \{\alpha_{j_1 j_2}\}),$$

where

$$\alpha_{j_1 j_2} = \alpha_{j_1 j_2}{}^{(0)} + \sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2);$$

$$q(\mu_j | \tau_j) \sim N(m_j, \frac{1}{\beta_j \tau_j})$$

and

$$q(\tau_j) \sim Ga(\frac{1}{2}\gamma_j, \frac{1}{2}\delta_j),$$

with hyperparameters given by

$$\beta_j = \beta_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\gamma_j = \gamma^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\delta_j = \delta^{(0)} + \sum_{i=1}^{n} q_{ij} {y_i}^2 + {\beta_j}^{(0)} {m_j}^{(0)2} - \beta_j m_j^2$$

$$m_j = \frac{{\beta_j}^{(0)} {m_j}^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j},$$

where $q_{ij} = q_z(z_i = j)$.

The variational posterior for $q_z(z)$ will have the form

$$q_z(z) \propto \prod_i \prod_j {b_{ij}^*}^{z_{ij}} \prod_i \prod_{j_1} \prod_{j_2} {a_{j_1 j_2}^*}^{z_{ij_1} z_{i+1 j_2}},$$

for certain $\{a_{j_1 j_2}^*\}$ and $\{b_{ij}^*\}$ (see Appendix G.1). This is the form of a conditional distribution of the states of a hidden Markov Chain, given the observed data. From this we need the marginal probabilities

$$q_z(z_i = j)$$

$$q_z(z_i = j_1, z_{i+1} = j_2).$$

These can be obtained by the forward-backward algorithm (see Appendix G.2), based on a^* and b^* quantities given by

$$a_{j_1 j_2}^* = \exp\{\mathbf{E}_q[\log \pi_{j_1 j_2}]\} = \exp\{\Psi(\alpha_{j_1 j_2}) - \Psi(\alpha_{j_1 \cdot})\},$$

$$b_{ij}^* = \exp\{\mathbf{E}_q[\log p_j(y_i|\phi_j)]\},$$

where

$$\mathbf{E}_q[\log p_j(y_i|\phi_j)] = \frac{1}{2}\Psi(\frac{1}{2}\gamma_j) - \frac{1}{2}\log\frac{\delta_j}{2} - \frac{1}{2}(\frac{\gamma_j}{\delta_j})(y_i - m_j)^2 - \frac{1}{2\beta_j}.$$

Here $a_{j_1 j_2}^*$ is an estimate of the probability of transition from state j_1 to j_2 and b_{ij}^* is an estimate of the emission probability density given that the system is in state j at time point i. One can obtain $q_z(z_i = j)$ and $q_z(z_i = j, z_{i+1} = j_2)$ from

the following formulae based on the forward and backward variables, which we denote by fvar and bvar, respectively:

$$\begin{aligned} q_z(z_i = j) = p(z_i = j_1|y_1, ..., y_n) &\propto \text{fvar}_i(j_1)\text{bvar}_i(j_1) \\ &= \frac{\text{fvar}_i(j_1)\text{bvar}_i(j_1)}{\sum_{j_2}\text{fvar}_i(j_2)\text{bvar}_i(j_2)} \end{aligned}$$

$$\begin{aligned} q_z(z_i = j_1, z_{i+1} = j_2) &\propto \text{fvar}_i(j_1)a^*_{j_1j_2}b^*_{i+1j}\text{bvar}_{i+1}(j_2) \\ &= \frac{\text{fvar}_i(j_1)a^*_{j_1j_2}b^*_{i+1j_2}\text{bvar}_{i+1}(j_2)}{\sum_{j_1}\sum_{j_2}\text{fvar}_i(j_1)a^*_{j_1j_2}b^*_{i+1j_2}\text{bvar}_{i+1}(j_2)}. \end{aligned}$$

Obtaining Formulae for p_D and the DIC

Our variational approximation to p_D is

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$\begin{aligned} = \; - \; 2\Bigg[\sum_{j_1}\sum_{j_2}\{\sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2)\}\{\Psi(\alpha_{j_1,j_2}) - \Psi(\alpha_{j_1\cdot})\} \\ + \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\left\{\frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j}\right\}\Bigg] \\ + \; 2\Bigg[\sum_{j_1}\sum_{j_2}\{\sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2)\}\log(\frac{\alpha_{j_1j_2}}{\sum_{j_2}\alpha_{j_1j_2}}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\gamma_j}{\delta_j})\Bigg]. \end{aligned}$$

To find the DIC value we use the usual formula

$$\text{DIC} = 2p_D - 2\log p(y|\tilde{\theta}),$$

in which $p(y|\tilde{\theta})$ can be found using the forward algorithm:

$$p(y|\tilde{\theta}) = \sum_{j=1}^{K}\text{fvar}_n(j).$$

Note that in practical applications it is often necessary to work with normalised versions of the forward and backward variables in order to avoid numerical problems when the data set is reasonably large (generally more than 100 observations). To see why this issue arises, consider that the forward variable is given by the sum of a large number of product terms. Each term in the product is smaller than one, often very much smaller, so that, as the number of observations increases, each term of the forward variable tends exponentially to zero. This means that, if the number of observations is large enough, values arising in the calculation of the forward variable will be beyond the precision range of the computer performing the calculations. For this reason, the forward and backward variables were normalised in our implementation; the details are given in Appendix G.2.

4.5 Practical Implementation

Our variational method and calculation of the DIC and p_D values is implemented using a program which is run in R. The code deals with one-dimensional data sets and can be initialised to start with any number of maximum potential states.

The user must specify the initial number of states, K, to start with, and the observed data set. As with the mixtures program, the user's input data must contain the observed data and a list of indices ranging from $1, ..., K$ initially allocating the observations to one of the K states. In our examples, we allocated roughly equal numbers of the observations to each of the K states. No particular method was used for doing so, but we found that the initial allocation did not seem to affect results. The initial allocation starts the program off and as the algorithm cycles through its iterations, the observations find their own places. At the initialisation stage, a user-specified value is given for the weight that is to be assigned to each observation indicator variable (the q_{ij}'s). In the same way as for the mixtures case, these initial values for the q_{ij}'s were chosen to give a slightly higher weighting to the initial allocation to states to start the algorithm running. Initial estimates for the $q_z(z_i = j_1, z_{i+1} = j_2)$'s were obtained from the initial q_{ij}'s by setting $q_z(z_i = j_1, z_{i+1} = j_2) = q_{ij_1} \times q_{i+1j_2}$. In most cases, the results obtained were the same for all values of the initial weight but occasionally it led to slight differences. As with the finite mixtures example, when this was found to be the

case, the DIC value was used to choose between models. The user has the option to specify initial values for the sufficient statistics or, alternatively, defaults which specify broad priors are available. All of our examples use these broad priors.

As the program runs, the resulting q_{ij}'s are nonnegative and they sum to 1 over j for each i. They therefore form a set of predictive probabilities for the indicator variables for the data. The sum of the q_{ij}'s over the i time points for each state provides an estimate of the number of times the system is in that state, and we can think of this as a weighting for each state. The cutoff value determines at which point a state is no longer deemed to be part of the solution. The default value we use for this is 1 and this was the value used in all examples given. This means that a state is not considered necessary if less than one observation is assigned to it. When a state's weight falls below this cutoff value, it is removed from consideration and the program continues with one fewer component. We found that, in implementing this method, the weightings of extra states tended towards zero more quickly than extra components did in the mixtures case, and so it was necessary to remove extra components simultaneously. We also found that it was often necessary to place a lower bound of 10^{-22} on values of a^* to ensure convergence. For our simulated examples, this was necessary when initialising the algorithm with too many states. We describe this and the effect on results in more detail when we consider some examples.

At each iteration of the code, the DIC and p_D values are computed and the updated allocation weights for each state are obtained. The program runs until it converges and the solution it finds will have a number of states which is less than or equal to the number the user started with. This means that states which were considered to be superfluous are removed as the program cycles through its iterations.

We summarise the method:

- Iteration initialised with a larger number of hidden states than one would expect to find.
- Placing a lower bound on a^* values ensures that the program converges for all initial numbers of states.
- In some cases, the weightings of one state will dominate those of others causing the latter's weightings to tend towards zero.

- When a state's weighting becomes sufficiently small it is removed from consideration and the algorithm continues with the remaining states.
- At each step the DIC and p_D values are computed.
- The algorithm eventually finds a solution with a number of states less than or equal to the number one started with.
- Applying variational methods to the learning of a HMM with Gaussian noise leads to an automatic choice of model complexity.
- Solutions with fewer states than that selected can be obtained by starting the algorithm with fewer states and the resulting DIC value can be compared with that obtained with more states.

4.6 Performance of the Method on Simulated and Real Data Sets

4.6.1 Application to Simulated Examples

A Well-Separated Example

We begin by considering a well-separated simulated example to investigate performance of the method and explore the automatic feature of the variational approximation when it is applied to HMM's. We simulated 800 data points from a 2-state hidden Markov model with transition matrix given by

$$\pi = \begin{bmatrix} 0.2 & 0.8 \\ 0.7 & 0.3 \end{bmatrix},$$

with Gaussian noise distributions with means 2,-2 and standard deviations of 0.5,0.5 respectively. The variational program was initialised with a number of states ranging from 1 to 15 and the resulting posterior estimates were extremely close to the true values. The results are reported in Table 4.1. However, there were some convergence issues that we had to consider in order to obtain these results.

We found that extra states were being eliminated as the program converged, as we had observed in the finite mixtures case. We originally designed the program

to remove only one component at a time, as we had done in the finite mixtures case. However, the program failed to converge for several numbers of starting states, generally the higher ones. This was because the q_{ij} values of two or more states were tending towards 0 simultaneously and as they were not removed from consideration quickly enough the program could not continue. Since the example is well-separated, superfluous states were quickly identified which led to the weightings of several states tending towards 0 at the same time. It is interesting to note that weightings for unnecessary states appear to tend to zero more quickly in the HMM case than we observed with our finite mixture examples, for which it was sufficient to remove unwanted components one at a time when the initial number of components was of a reasonable size. We modified the program to simultaneously remove more than one state in the same iteration which eliminated this convergence problem.

We then encountered another, more significant, convergence problem. When we initialised the program with an initial number of states larger than the true number (i.e. more than 2 states), the posterior estimates of some of the transition probabilities (a^* values) became so small that they exceeded the precision range of the computer performing the calculations. This could be because states which are unnecessary are still not being removed quickly enough. For this reason, it was necessary to impose a lower bound on the a^* values to allow the program to continue when the initial number of states was higher than the true number. The results given in Table 4.1 correspond to the 'best' solution found for each number of initial components with various initial weights for the q_{ij}, where the solution with the lowest DIC value is considered to be the 'best'.

One can see from Table 4.1 that, for this example, in most cases this lower bound allowed us to find a close estimate of the true solution, but, even in this well-separated example, it is possible to find alternative solutions to the ones reported in Table 4.1 when the initial value for the weights assigned to the q_{ij} is altered, even for the same number of starting states. However, these alternative solutions occurred infrequently. Some examples are given in Table 4.2 and we can see that some states are representing the same part of the data and are clearly redundant. We can also see, from the presence of the lower bound values (10^{-22}) in the estimated transition matrices, that we have only been able to obtain these solutions via the use of a lower bound. It also appears that, in these instances, the algorithm was unable to identify and remove these extra components. This is

slightly concerning as it suggests that this lower bound can cause the algorithm to fit a model which it would not have selected otherwise, and for real data sets this might be misleading. On the other hand, there may be cases where the variational algorithm is removing too many components.

It would be preferable if the solution were not being artificially altered since, as well as preventing any transition from having zero probability, in some cases it may be leading to a solution with more states than are truly present, with the program becoming trapped in these states which we have allowed to remain in consideration through the imposition of the lower bound. One could suggest that the need for a lower bound indicates that more states should be removed, but it would be difficult to justify forcibly removing these extra states or to find some other criterion for reducing the complexity of the model. It would be interesting to investigate this phenomenon further in an attempt to discover why in some cases superfluous states are not eliminated when they are in others. Perhaps the answer to this lies in the initial values assigned to the posterior estimates of the pairwise marginal probabilities $q_z(z_i = j_1, z_{i+1} = j_2)$.

Using the lower bound to force the algorithm to find a solution for all initial numbers of states also allows us to get some idea of what the DIC would be for these cases, although there is some slight variation in the resulting values of p_D and DIC for the same solution when reached by starting with a different number of states. Another point is that the lower bound on a^* values ensures that p_D, and hence the DIC, can be calculated, since p_D is not defined when any of the a^* values are exactly zero (since values of a^* equal to zero lead to values of zero for some $\alpha_{j_1 j_2}$'s and this means that the digamma function required in the calculation of p_D cannot be evaluated). Exact zeroes are not problematic in the variational approximation.

Figure 4.1 displays a kernel plot of the raw data with the true density and the fitted 2 state density, found by the variational method, superimposed.

Table 4.3 shows the p_D and DIC values corresponding to the aforementioned alternative solutions for this data set. Note that the DIC values are higher for these solutions than for the solution having 2 states and the DIC appears to be useful in identifying the best solution available for the alternative outcomes.

Table 4.1: Results for the 800-observation data set with a lower bound on a^*

No. of Initial States	No. of States Found	Estimated Posterior Means	Estimated Posterior st. dev.	Estimated Posterior Weights	Estimated Posterior Transition Probabilities	p_D	DIC
15	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
14	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
13	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
12	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
11	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
10	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
9	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
8	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
7	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
6	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
5	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
4	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.65	600
3	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
2	2	-2.03,1.98	0.48,0.51	0.45,0.55	0.19 0.81 0.67 0.32	5.99	600
1	1	0.15	2.05	1	[1]	1.99	1958

Table 4.2: Some other possible results obtained by using different initial weights for the q_{ij}'s

No. of Initial States	No. of States Found	Estimated Posterior Means	Estimated Posterior st. dev.	Estimated Posterior Weights	Estimated Posterior Transition Probabilities
11	3	−2.5	0.34	0.001	10^{-22} 10^{-22} 1
		−2.02	0.48	0.454	10^{-22} 0.19 0.81
		1.97	0.51	0.545	10^{-22} 0.67 0.33
8	4	−2.03	0.48	0.46	0.19 0.42 0.28 0.11
		1.81	0.47	0.30	0.55 0.37 0.08 10^{-22}
		2.25	0.47	0.19	0.78 10^{-22} 0.21 10^{-22}
		1.90	0.23	0.05	1 10^{-22} 10^{-22} 10^{-22}

Table 4.3: p_D and DIC values corresponding to other possible results given in Table 4.2

No. of Initial States	p_D	DIC
10	28.64	642
8	25.96	636

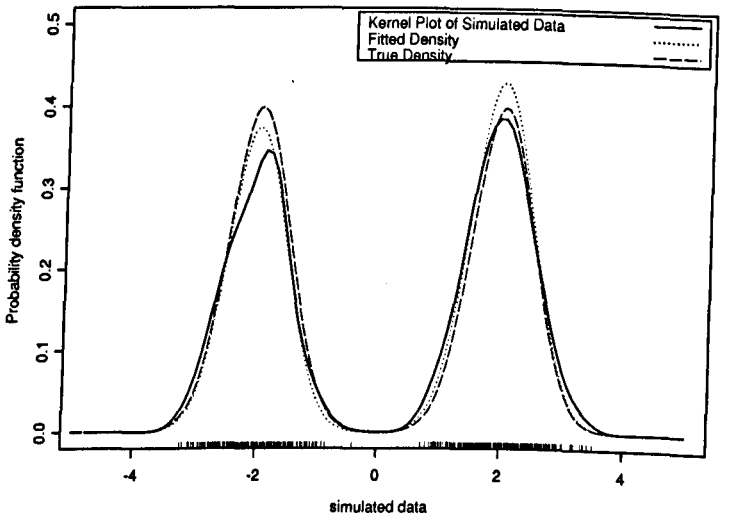

Figure 4.1: Results from 800-observation data initialised with number of states ranging from 2 to 15 resulting in a 2-state model with parameter estimates given in Table 4.1

A 4-State Example

Next we analysed a simulated data set, comprising 500 observations, generated from a 4-state hidden Markov chain with transition matrix

$$\pi = \begin{bmatrix} 0.2 & 0.2 & 0.3 & 0.3 \\ 0.3 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.3 & 0.2 \\ 0.3 & 0.3 & 0.2 & 0.2 \end{bmatrix}$$

and Gaussian noise distributions with means -1.5, 0, 1.5 and 3, and equal standard deviations of 0.25.

The results presented here are obtained by placing a lower bound on the values of a^*. Again, for this example, when we did not enforce a lower bound, the algorithm would only converge to a solution when initialised with the true number of states (4), or less. The results given also correspond to the 'best' solution found for each number of initial states, where the solution with the lowest DIC value is considered to be the 'best'. Table 4.4 reports the variational estimates of the posterior means, standard deviations and weights. Table 4.5 shows the relevant DIC and p_D values.

For this example, using the lower bound, we were able to recover a 4 state solution with good posterior estimates for initial number of states ranging from 4-6. For 7 initial states, the best solution we obtained had 5 states and so the lower bound has led to a solution with too many states. However, two of the states have noise models that are close together and one of the estimated posterior weights is small. Again, the DIC appears to be useful in indicating the appropriateness of the fitted model.

Figure 4.2 shows a kernel plot of the raw data with the fitted model and true generating distribution superimposed.

Table 4.4: Results for the 500-observation data set with a lower bound on a^*

No. of Initial States	No. of States Found	Estimated Posterior Means	Estimated Posterior st. dev.	Estimated Posterior Weights
7	5	−1.47	0.26	0.25
		−0.01	0.24	0.22
		0.04	0.22	0.05
		1.5	0.23	0.24
		3	0.27	0.24
6	4	−1.47	0.26	0.25
		−0.005	0.24	0.27
		1.5	0.23	0.24
		3	0.27	0.24
5	4	−1.47	0.26	0.25
		−0.005	0.24	0.27
		1.5	0.23	0.24
		3	0.27	0.24
4	4	−1.47	0.26	0.25
		−0.005	0.24	0.27
		1.5	0.23	0.24
		3	0.27	0.24
3	3	−1.5	0.23	0.22
		0.68	0.97	0.56
		3	0.24	0.22
2	2	−0.67	0.85	0.53
		2.24	0.85	0.47
1	1	0.7	1.68	1

Table 4.5: DIC and p_D values for the 500-observation data set corresponding to the solutions presented in Table 4.4

No. of Initial States	No. of States Selected	p_D	DIC
7	5	35.09	540
6	4	20.03	513
5	4	20.03	513
4	4	20.03	513
3	3	12.00	729
2	2	5.99	936
1	1	1.99	1025

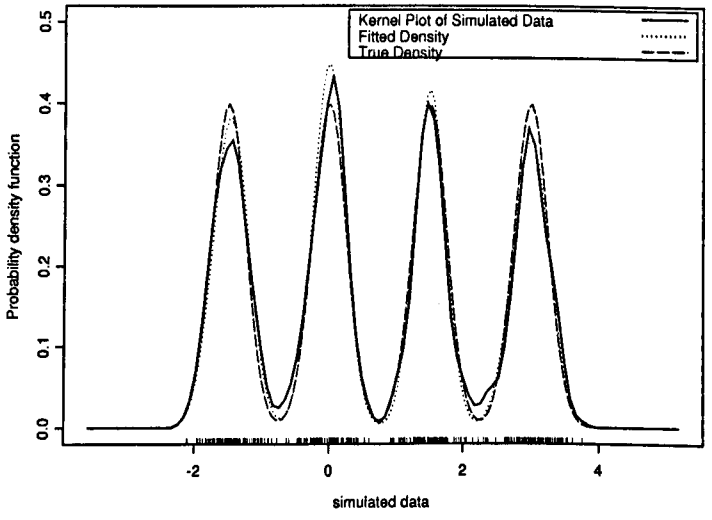

Figure 4.2: Results from 500-observation data initialised with number of states ranging from 4 to 6 resulting in a 4 state model with parameter estimates given in Table 4.4

4.6.2 Application to Real Data Sets

The three data sets used in this section were analysed by Robert et al. (2000) using RJMCMC and have also previously been analysed by other authors. We analyse these data sets using our variational approach and compare our results with other treatments of the data.

Daily Returns Data

The first data set is an extract from the Standard and Poors 500 stock index consisting of 1700 observations of daily returns from the 1950s. It was previously analysed by Rydén et al. (1998) and was the data referred to as subseries E in their paper.

Wind Data

This data comprises a series of 500 hourly wind velocity measurements taken in Athens in January 1990. This data has had a small uniform disturbance added to avoid exact zeroes. This data was also analysed by Francq and Roussignol (1997) (without the added disturbance).

Geomagnetic Data

The third observation set is made up of 2700 residuals from a fit of an autoregressive moving average model to a planetary geomagnetic activity index. Francq and Roussignol (1997) also analysed this data set.

Running the program on the wind data set with 7 initial states and placing a lower bound on the values of a^* resulted in the selection of a posterior solution with 5 states using the variational method. However, the variational algorithm failed to converge for initial numbers of states which were larger than 2 without this lower bound on a^*. The DIC and p_D values associated with other initial numbers of states are given in Table 4.7 and from this we can see that the DIC selects 2 or 3 states for this data. The variational posterior for the transition probability matrix obtained by starting with 7 initial states was

$$\begin{bmatrix} 10^{-22} & 0.84 & 0.14 & 10^{-22} & 10^{-22} \\ 10^{-22} & 0.89 & 0.085 & 10^{-22} & 0.013 \\ 0.12 & 10^{-22} & 0.46 & 0.4 & 10^{-22} \\ 0.092 & 10^{-22} & 10^{-22} & 0.9 & 10^{-22} \\ 10^{-22} & 0.046 & 10^{-22} & 10^{-22} & 0.95 \end{bmatrix}.$$

It is clear that in many cases the estimated transition probability has been set to 10^{-22} and so we have artificially ensured that the algorithm converges. This need for a lower bound could be due to the presence of too many states, but it is unclear why these were not removed along with the other two that were, if indeed they were superfluous. Lack of separation could be a contributing factor. The posterior estimates of some of the parameters from the solutions obtained, when starting with 7 states, are given below in Table 4.6. Figure 4.3 shows a Kernel plot of the data with the density fitted by the algorithm superimposed.

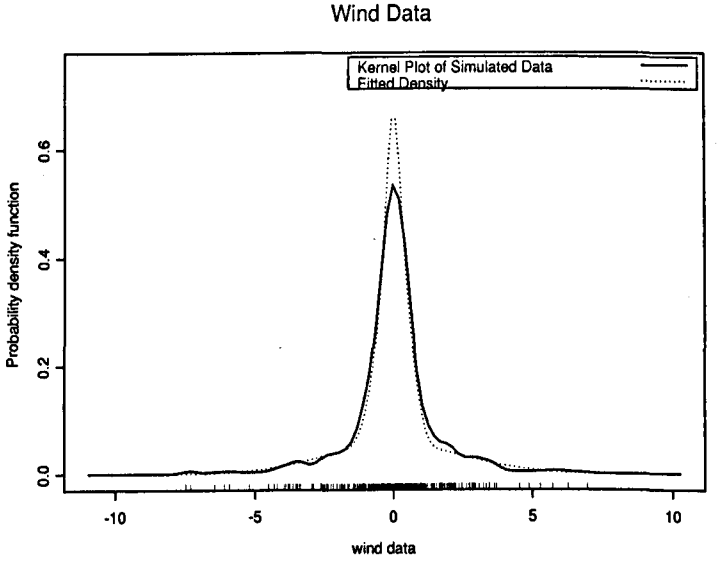

Figure 4.3: Wind data starting with 7 initial states, terminating with 5 states

Table 4.6: Results for the Wind Data starting algorithm with 7 states

Estimated posterior means	-0.81	-0.06	0.008	0.02	0.61
Estimated posterior standard deviations	0.36	0.42	2.48	0.23	0.33
Estimated posterior weights	0.04	0.417	0.34	0.118	0.08

Table 4.7: DIC and p_D values and number of states selected for the Wind Data with different initial numbers of components

Initial No. of States	No. of States Found	DIC	p_D
15	5	510	43.94
7	5	507	43.06
6	6	537	65.72
5	5	506	43.73
4	4	492	27.95
3	3	467	14.06
2	2	476	6.00

The DIC favours 3 states for this data-set, but based on the fact that the variational algorithm only converges for solutions with a number of states less than or equal to 2, from our analysis, the most appropriate solution appears to be the 2-state solution with variational posterior transition matrix

$$\begin{bmatrix} 0.955 & 0.045 \\ 0.074 & 0.926 \end{bmatrix}$$

and variational posterior estimates given in Table 4.8. The fitted density is plotted in Figure 4.4.

Table 4.8: Results for the Wind Data starting algorithm with 2 states

Estimated posterior means	-0.015	0.0069
Estimated posterior standard deviations	0.43	2.37
Estimated posterior weights	0.62	0.38

The method used by Robert et al.(2000), selected a 3-state model, and Francq and Roussignol's (1997) analysis selected a 2-state model for the wind data set. The posterior estimates we obtain for the transition probabilities and the state density standard deviations for our 2 state model resemble those found in the analysis by Francq and Roussignol (1997). Francq and Roussignol (1997) estimated that $\pi_{12} = 0.048$, $\pi_{21} = 0.079$ and $\sigma_1 = 0.437$, $\sigma_2 = 2.393$. They suggest that their two state model can only distinguish between periods having fairly constant gusts of wind and periods with many gust of wind.

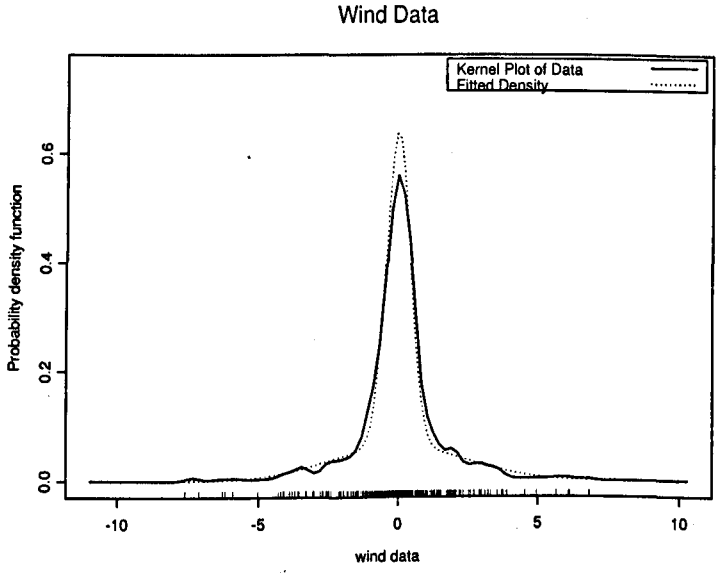

Figure 4.4: 2-state model fitted to the wind data when the algorithm was initialised with 2 states

For the geomagnetic data, the number of states selected, DIC and p_D values associated with different numbers of starting states, as obtained from the program with a lower bound on a^* values (posterior transition probability estimates), is reported in Table 4.9. Without this lower bound, the algorithm does not converge if initialised with more than 2 states. The DIC favours 4 states.

Table 4.9: DIC and p_D values and number of states selected for the geomagnetic data with different initial numbers of components

Initial No. of States	No. of States Found	DIC	p_D
15	7	7325	84.44
7	7	7268	84.98
6	6	7236	63.66
5	5	7200	41.99
4	4	7183	23.99
3	3	7203	12.11
2	2	7217	5.99

Since the variational algorithm will not converge to a solution with more than 2 states without the lower bound on a^*, it seems most appropriate to consider the 2-state solution with variational posterior transition matrix

$$\begin{bmatrix} 0.982 & 0.018 \\ 0.187 & 0.813 \end{bmatrix}$$

and variational posterior estimates given in Table 4.10. The fitted density is plotted in Figure 4.5.

Table 4.10: Results for the geomagnetic data, starting the algorithm with 2 states

Estimated posterior means	-0.209	1.769
Estimated posterior standard deviations	1.997	5.408
Estimated posterior weights	0.911	0.089

This analysis by Robert et al.(2000) selected a 3-state model for this data. Francq and Roussignol's (1997) analysis selected a 2-state model for the geomagnetic data set. The posterior estimates we obtain for the transition probabilities and the state density standard deviations for our 2-state model are similar to those found by Francq and Roussignol (1997). Francq and Roussignol (1997) estimated that $\pi_{12} = 0.014$, $\pi_{21} = 0.16$ and $\sigma_1 = 2.034$, $\sigma_2 = 5.840$. Francq and Roussignol (1997) suggest that this two-state model corresponds to tumultuous and quiet states, the tumultuous state being the one with the highest variability. Since their model visits tumultuous states less frequently than it does quiet states, and spends less time in them, they propose that these tumultuous states might correspond to geomagnetic storms.

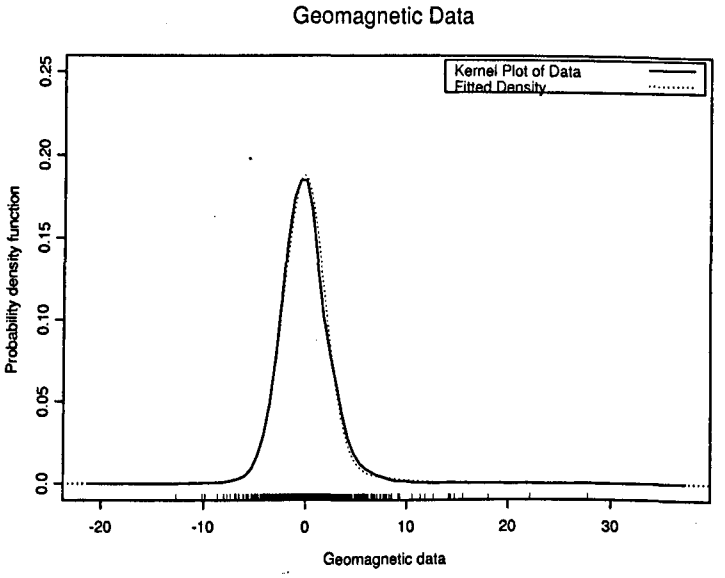

Figure 4.5: 2-state model fitted to the geomagnetic data when the algorithm was initialised with 2 states

For the daily returns data set, Table 4.11 summarises the results for different initial numbers of states, obtained by imposing a lower bound on the a^* values.

Table 4.11: DIC and p_D values and number of states selected for the daily returns data with different initial numbers of components

Initial No. of States	No. of States Found	DIC	p_D
15	7	-15488	90.91
7	5	-15576	43.99
6	6	-15540	64.94
5	5	-15572	43.98
4	4	-15594	27.99
3	3	-15580	13.99
2	2	-15563	6.00

For the daily returns data, the variational algorithm only converges for models with an initial number of states less than or equal to 2 if we do not impose a lower bound on posterior transition probability estimates. The DIC's are negative for this data set, but comparing these, the lowest DIC is for 4 states. Based on these results, we fitted the 2-state solution with variational posterior transition matrix

$$\begin{bmatrix} 0.96 & 0.04 \\ 0.07 & 0.93 \end{bmatrix}$$

and variational posterior estimates given in Table 4.12. The fitted density is plotted in Figure 4.6.

Table 4.12: Results for the daily returns data starting algorithm with 2 states

Estimated posterior means	0.00084	-0.00145
Estimated posterior standard deviations	0.00453	0.00898
Estimated posterior weights	0.63	0.37

Robert et al.(2000)'s analysis favoured 2 or 3 states in a model for this data and Rydén et al. (1998) selected a 2-state model. In the analysis by Robert et al.(2000), the estimated transition probabilities for the 2-state model were $\pi_{12} = 0.044$ and $\pi_{21} = 0.083$, and the estimated posterior standard deviations were $\sigma_1 = 0.0046$ and $\sigma_2 = 0.0093$. These were close to the estimates found

by Rydén et al. (1998); their estimates for the transition probabilities were $\pi_{12} = 0.037$ and $\pi_{21} = 0.069$, and their estimated posterior standard deviations were $\sigma_1 = 0.0046$ and $\sigma_2 = 0.0092$. Our estimated transition probabilities and posterior standard deviations are similar to both sets of estimates.

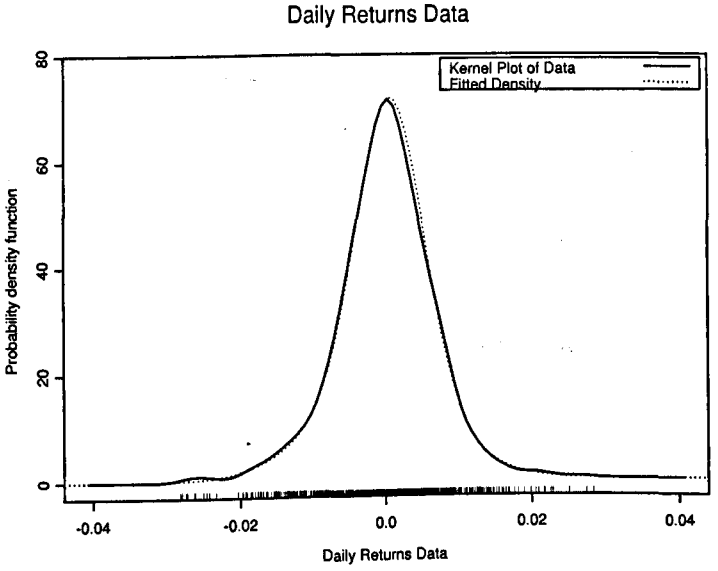

Figure 4.6: 2-state model fitted to the daily returns data when the algorithm was initialised with 2 states

To summarise, we give a brief comparison of results for these data sets found through different analyses:

- Geomagnetic Data :
  - Variational method - 2 states
  - DIC - 4 states
  - Analysis by Robert et al.(2000) - 3 states
  - Analysis by Francq and Roussignol (1997) - 2 states
- Daily Returns Data :
  - Variational method - 2 states
  - DIC - 4 states
  - Analysis by Robert et al.(2000) - 2 or 3 states
  - Analysis by Rydén et al. (1998) - 2 states
- Wind Data :
  - Variational method - 2 states
  - DIC - 3 states
  - Analysis by Robert et al.(2000) - 3 states
  - Analysis by Francq and Roussignol (1997) - 2 states (no preprocessing of data)

Modelling the Real Data Sets as Mixture Models, Ignoring the Time Dependency

We also analysed the 3 real data sets as mixture data, using the variational approach from Chapter 2. For the wind data set, initialising the algorithm with 15 components resulted in a solution with 7 components. All but 2 of the fitted components had small weights attached to them. So, for the wind data sets, ignoring the dependency between observations and treating them as identically and independently distributed, resulted in a more complex model than that which was obtained by modelling the dependency. We also initialised the mixture program with two components to allow a comparison of the fit with that obtained from the HMM analysis. Results are given in Table 4.13 and the fitted density is plotted

Table 4.13: Results from mixture analysis of the wind data starting algorithm with 2 components

Estimated posterior means	0.0049	-0.028
Estimated posterior standard deviations	0.47	2.45
Estimated posterior weights	0.64	0.36

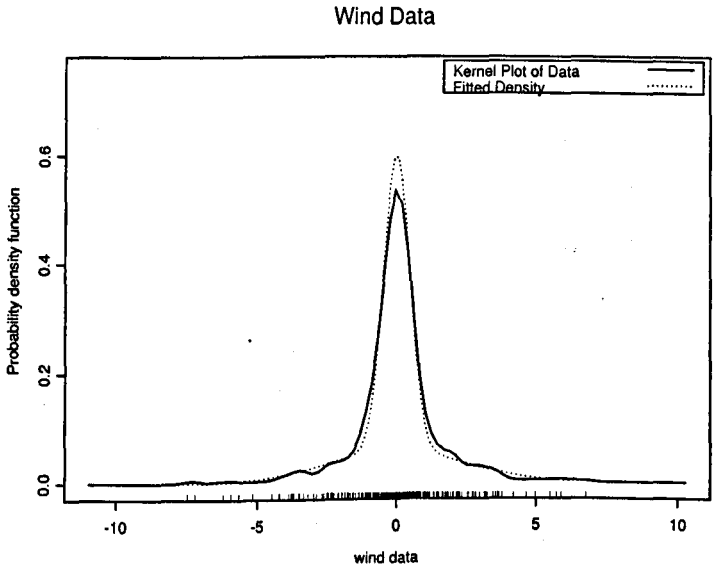

Figure 4.7: Density fitted to the wind data by mixture analysis

in Figure 4.7. There are similarities between these posterior estimates and those obtained from the HMM analysis (Table 4.8).

Interestingly, the DIC value for the 2 component model (DIC=1571.98) was higher than that for the 7 component model obtained by initialising with 15 states (DIC=1487.43), so going by the DIC, the more complex model has more support.

For the geomagnetic data, initialising the algorithm with 15 components resulted in a solution with 5 components. This is more complex than the result we obtained in the HMM analysis without imposing a lower bound on a^*. To compare results with the HMM analysis, the results found by fitting a mixture model with two components are reported in Table 4.14. The fitted density is shown in Figure 4.8.

Table 4.14: Results from mixture analysis of the geomagnetic data starting algorithm with 2 components

Estimated posterior means	-0.232	2.658
Estimated posterior standard deviations	2.02	5.69
Estimated posterior weights	0.93	0.07

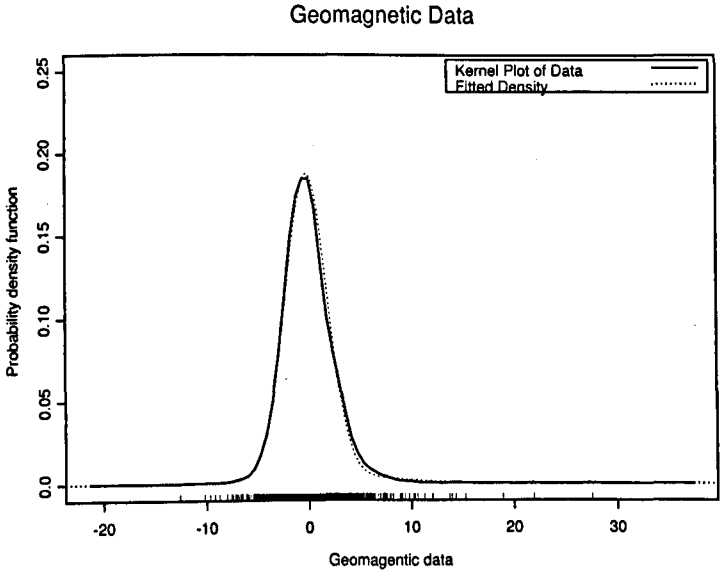

Figure 4.8: Density fitted to the geomagnetic data by mixture analysis

The posterior estimates presented above are similar to those found in the HMM analysis (Table 4.10). Again, the DIC value for the 2-state model (DIC=12289.25) was higher than that for the 5-state model (DIC=12261.81) and so the DIC favours the more complex model.

For the daily returns data, initialising the algorithm with 15 components resulted in a solution with 2 components; results are presented in Table 4.15 and the fitted density is plotted in Figure 4.9. In this case we select a model with the same complexity as that found by the HMM analysis with no lower bound on a^*.

Table 4.15: Results from mixture analysis of the daily returns data starting algorithm with 15 components

Estimated posterior means	0.00058	-0.00138
Estimated posterior standard deviations	0.00457	0.00978
Estimated posterior weights	0.7	0.3

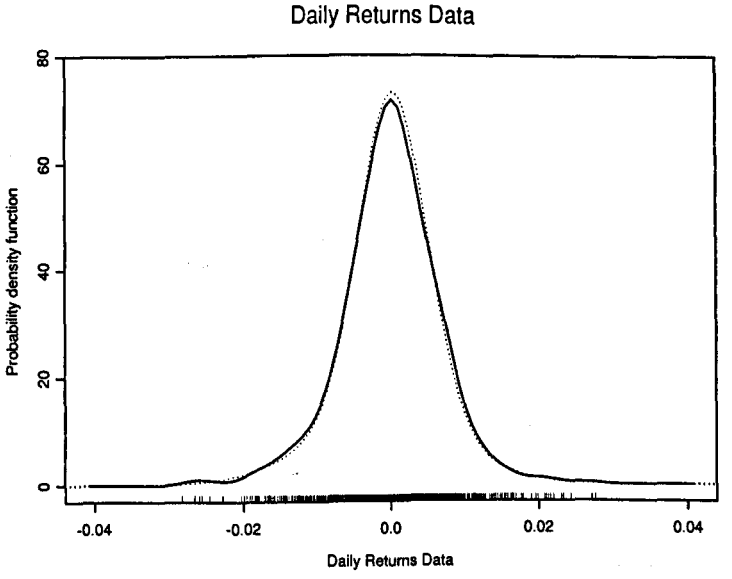

Figure 4.9: Density fitted to the daily returns data by mixture analysis

Again, these posterior estimates are similar to those obtained in the HMM analysis (Table 4.12).

In the case of the wind and geomagnetic data sets, the mixture analysis resulted in a more complex model than the HMM analysis did. In these instances, adding model complexity has led to a smaller number of components. However, for the daily returns data, the level of complexity found was the same. In all cases there were similarities between the posterior estimates found by both types of analysis.

4.7 Conclusions

As with the mixture case, applying variational methods in the case of a hidden Markov model with Gaussian noise leads to the removal of extra components. Solutions with fewer states than the number selected can be obtained by starting the algorithm with fewer states. The variational approximation also enables the calculation of DIC values which can be used to choose between competing models.

A difference between the finite mixture and HMM case was that weightings for unnecessary states appeared to tend to zero more quickly and so unwanted states had to be simultaneously removed.

Another difference is that it was necessary to impose a lower bound on the a^* values, which are the posterior estimates of the transition probabilities, to allow the algorithm to converge when initiated with larger numbers of initial states than were present in our simulated examples. In some cases this allowed us to find a close estimate of the true solution but in others this seemed to be leading to a solution with more states than were present, with the algorithm becoming trapped in these states which we had allowed to remain in consideration through the imposition of the lower bound. It was not necessary to impose a lower bound when we began with the correct number of states, or a smaller number than was actually present, in our simulated examples. This implies that the need for the lower bound could be indicative of the fact that there are too many states being considered, but if these are not removed by the method, via small weightings, then it is difficult to justify forcibly making extra removals based on bounded a^* values. Perhaps some other method of identifying superfluous states could be introduced, and it might also be worthwhile investigating the affect of the initial assignment of observations to states. These points merit further investigation.

The DIC can also be used to select a model. However, in the HMM's case,

there is less agreement between the model selected by the DIC and that selected using the variational method than we observed in the finite mixtures case.

We have also compared results obtained here with those we find by treating the data as if it had come from a mixture distribution and ignoring the extra dependencies. With the real data, in two cases the mixture analysis resulted in a model with higher complexity than the HMM analysis did. However, for the other data set, the level of complexity found was the same for both the mixture and the HMM set up. So, in two instances, adding model complexity led to a smaller number of components but there is insufficient evidence to apply this rule in general. For all 3 data sets the posterior estimates found by both types of analysis showed similarities.

Again, for this type of model, we found the variational method to be computationally efficient. For example, convergence to a variational solution and calculation of p_D and DIC values for the 2-state 800 observation data set analysed in section 4.6.1, initialising the algorithm with 15 states and imposing a lower bound on a^* values, took 367.58 seconds (around 6 minutes) to run in R on a Windows NT Intel P4 2GHz workstation.

Variational methods for the analysis of HMM's have potential, but there is much scope for further investigation into the state-removal phenomenon which occurs in the implementation.

Chapter 5

Hidden Markov Random Fields and Image Analysis

In previous chapters we have studied mixture models with different dependency structures. In all cases we have interpreted the mixture data as incomplete data, the missing information being the indicator variables or labels for the class of each observation. We considered finite mixture models, in which the indicator variables for each observation are independent. This might be thought of as the standard mixture dependency set-up. We then went beyond this structure and allowed the indicator variables to correspond to a stationary Markov chain, thereby leading to a hidden Markov chain or hidden Markov model. The next level of complexity is to have the indicator variables being realisations from a Markov random field with a two-dimensional index set. This leads to the hidden Markov random field model which we will study in this chapter. For more detail on the dependency structures of such models see Titterington (1990).

In this chapter we review the application of hidden Markov random fields to Bayesian digital image analysis. In addition, we attempt to extend the variational framework and the DIC to the spatial setting of analysis of an image represented by a hidden Markov random field. The empirical findings presented here are somewhat exploratory in nature and a fuller assessment would require further detailed study.

5.1 Introduction and Background

The Markov random field (MRF) concept dates back to Dobrushin (1968) although models of this kind were in fact already in use in statistical physics long before then. MRFs are spatial models whose spatial locations or sites generally follow some sort of lattice structure. Each site on the lattice has a set of neighbouring sites and the attractiveness of such models lies in the fact that the conditional probability at each site is dependent only upon the values of its neighbours. The MRF concept represents one method of extending the one-dimensional Markovian property to a more general setting like a two-dimensional spatial problem.

The MRF structure is of interest in a variety of research applications where there is spatial information and an interest in representing the spatial association between data. As encountered in previous chapters, there are also situations where we have missing data in the sense that the labels indicating the state to which a given observation belongs are unknown. In this setting, the hidden Markov random field (HMRF) formulation is often an appropriate representation. For example, HMRFs have been used in areas such as disease mapping, where interest lies in modelling any potential spatial dependency between regions or countries which are geographically close to another or which are related in some other way. See, for example, Green and Richardson (2002) for an example of HMRFs in disease mapping or Besag and Higdon (1999) for an application to agricultural field experiments.

Hidden Markov Random Fields and Image Analysis

The application area on which we shall focus is the area of image analysis and restoration. We can think of an image as a two-dimensional area subdivided into a rectangular lattice of sites which we call pixels. An image will typically have at least $256 \times 256 = 2^{16}$ pixels (so that they present a large data problem) each taking one of a finite set of possible intensities (or levels of brightness). For instance, an image can be obtained from a video camera where the light from a scene passes through the lens and on to a matrix of sensors. Each sensor records the rate at which photons strike it while the image is being captured. There are other ways of recording an image (e.g. scanners, photographic camera) but all of these result in the production of a finite matrix of brightness. Brightness

levels are often called grey levels and frequently these are represented by integers ranging from 0 to $K-1$. Black is usually considered to be the lowest intensity, 0, and white the highest, $K-1$. Grey scales generally use 256 different levels. This is sufficient to represent an image, as humans cannot distinguish between more levels than this on a video display. As well as binary images made up of black and white (or any other two colours of) pixels and grey scale images, we may have colour images which comprise the intensities of the colours red, green and blue.

There are occasions when an image can become corrupted, either through distortion which occurs during the actual imaging process or by degradation at a later stage. Image noise is the term used for anything included in an image which was not originally meant to appear; it can be summed up as the visible results of an error (or electronic interference) in the final image from the imaging apparatus. Images affected by noise are called noisy images. Thermal noise, for instance, is the most common and is always present in digital imaging systems. Thermal noise is caused by the irregularity of heat-generated electron fluctuations in the resistive parts of the physical apparatus used to record an image. It adds a snow-like appearance to the image, and hence thermal noise on digital television is commonly termed snow-noise. Uncorrelated Gaussian noise with mean 0 and variance σ^2 is called white Gaussian noise and thermal noise is in fact fairly well modelled by the Gaussian distribution (although strictly speaking the Gaussian model is a little unrealistic since it allows negative values on a grey scale). In this chapter we will concentrate on the statistical analysis of images with added white Gaussian noise. When an image has become degraded or affected by noise, rather than observing the actual value (or state) of each pixel, we have a noisy realisation of it and from this we have to reconstruct the original image. It is possible to imagine that there exists an allocation vector indicating the true value of each pixel, but of course these variables are hidden, leading us to the domain of the HMRF. Image analysis problems arise from various sources such as automated computer object recognition, astronomy, digital imaging software and numerous others. Satellite imaging also gives rise to image data. However, this is an example of a spatial problem where there might also be interest in incorporating a temporal element to the model as well as a spatial one; spatial-temporal problems will not be considered here.

Besag (1974,1975) was the first to propose general methods of statistical in-

ference for MRFs, overcoming the complications for such models by using the Hammersley-Clifford theorem (see section 5.2.1). This ground-breaking work was influential on later developments in the area.

MRF models were first introduced into the study of statistical image analysis by Geman and Geman (1984) who proposed a Bayesian approach to the problem of image restoration. They drew parallels between images and lattice-like systems arising in statistical physics where the Gibbs distribution (see section 5.2.1) is used as a model. Noting the equivalence of the Gibbs distribution and Markov random fields, as revealed by the Hammersley-Clifford theorem, they applied a Markov random field image model. They could then recover an original image from its degraded version by using a stochastic relaxation and annealing restoration algorithm to find maximum a posteriori (MAP) estimates of the original image given the degraded one. The algorithm generates a sequence of images that converges to the MAP estimates. The sequence is generated through local changes in pixel intensities and other variations in the scene, these changes being made randomly so as to avoid becoming trapped in local maxima. In statistical physics, the Gibbs distribution involves a temperature parameter representing the temperature in a physical system. This is a global control parameter and local conditional distributions are dependent upon it. The restoration algorithm is initialised at high temperatures and the temperature is then decreased as the algorithm continues. At the higher temperatures, many of the changes in intensities and boundary elements will decrease the posterior density whereas, at lower temperatures, these changes are likely to increase the posterior density. In this way Geman and Geman's (1984) algorithm simulates the chemical procedure called annealing which forces chemical systems to their low energy and highly regular states. The low energy states of the Gibbs posterior distribution are the MAP estimates of the true image given the corrupted version. This paper explicitly introduced the well known Gibbs Sampler which can be thought of as a special case of the Metropolis-Hastings algorithm. Geman and Geman's (1984) algorithm was very computationally demanding. This work, which was influenced by statistical physics, introduced new ideas to image analysis methodology.

Besag (1986) also dealt with the topic of MRFs in digital image analysis and proposed the method of Iterated Conditional Modes(ICM) for image restoration.

5.2 Markov Random Fields and the Hammersley-Clifford Theorem

In this section we set out the relevant theory of Markov random fields. We give the definition of a MRF and explore its equivalence to the Gibbs distribution as proven by the Hammersley-Clifford Theorem. We also explore the representation of a MRF through the well known Ising and Potts models from statistical physics.

5.2.1 Markov Random Fields

Pixels, Neighbours and Cliques

Consider a grid or lattice with regularly spaced sites $i = 1, ..., n$; these sites correspond to pixels in our image. Let $\mathcal{Y}$ denote the finite space of observed pixel states $Y = (Y_1, ..., Y_n)$, where the $\{y_i\}$ can take values in $\{1, ..., K\}$. The process Y is called a stochastic or random field if $p(y) > 0$ for all $y \in \mathcal{Y}$.

Before we can define a Markov random field we first have to describe the concept of a neighbourhood system. A collection of subsets $\delta = (\delta_i : i \in 1, ..., n)$ of the space of sites is called a neighbourhood system if

(i) $i \notin \delta_i$

(ii) $i \in \delta_j \iff j \in \delta_i$.

The sites $j \in \delta_i$ are called neighbours of i. We also use the notation $i \sim j$ to indicate that i and j are neighbours of one another.

A subset of the space of sites, or pixels, is called a clique if any two sites in that subset are always neighbours. We let C denote the set of all cliques on our set of pixels under the neighbourhood system δ.

In this thesis we consider first-order neighbourhood systems. The first-order neighbours of a pixel are the pixel above, below, to the left and to the right as shown in Figure 5.1. Naturally pixels on any edge of the image will not have four neighbours. It is common practice to use some method which gives edge pixels four neighbours to tie in with the other pixels for calculations. In this research, the two neighbours of each edge pixel are used twice to make four neighbours. This can be thought of as reflecting the edges of the image in a mirror and using the reflected pixels as neighbours. Another common approach is the toroidal

design in which one imagines the image is wrapped around a cylinder (vertically and horizontally) so that every pixel has four neighbours. Figure 5.2 shows the cliques for a first order neighbourhood system.

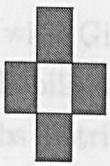

Figure 5.1: First-Order Neighbours

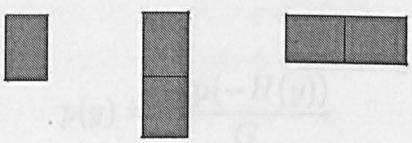

Figure 5.2: Cliques for a First-Order Neighbourhood System

Definition of a Markov Random Field

We give the definition of a Markov random field as established by Dobrushin (1968). The process Y is defined to be a Markov random field if, for all $y \in \mathcal{Y}$,

(i) $p(y) > 0$

(ii) $p(Y_i = y_i | Y_j = y_j, i \neq j) = p(Y_i = y_i | Y_j = y_j, j \in \delta_i)$.

So, in relation to our image model, condition **(i)** ensures that every possible configuration (in terms of grey levels or colours) of our original image prior is possible. Of course condition **(i)** is the condition which the process has to meet in order to be a random field. Condition **(ii)** requires that, in order for the process to be a Markov random field, the conditional probability at a given pixel depends only upon the values of neighbouring pixels.

A drawback of MRFs is that their definition does not lead to a natural way of writing down their distribution. This limited early research in MRFs as it was unclear how the joint probability distribution should be evaluated or how local conditional probabilities could be defined in a way which led to a valid joint

probability distribution. A significant breakthrough with regard to this problem was made by Besag (1974) who detailed a probabilistic structure for MRFs which was made possible through the Hammersley-Clifford Theorem which is described below. The progress made in this area is described in more detail in section 5.3. The theorem identifies MRFs with Gibbs distributions, which is useful in the specification of conditional probabilities, and so usually MRFs are defined through their representation as Gibbs distributions.

Gibbs Distributions

Gibbs distributions originate in statistical physics where there was interest in discovering properties of large systems from local characteristics. Gibbsian models are of the form

$$p(y) = \frac{\exp(-H(y))}{G}$$

where H is a real-valued function, $H : Y \mapsto \Re, y \mapsto H(y)$. In physics terminology, H is referred to as the energy function of $p(y)$ and the normalising constant, G, is called the partition function. The energy function is specified in terms of potentials, with

$$H(y) = \sum_A V_A(y),$$

where V_A is called the potential corresponding to A and A is a subset of the sites. The potential, V_A, is such that

(i) $V_\emptyset = 0$

(ii) $V_A(x) = V_A(y)$ if $Y_A(x) = Y_A(y)$.

If, for a given neighbourhood system, δ, $V_A(y) = 0$ whenever A is not a clique, then V is called a neighbour potential with respect to δ. This case is of particular interest to us. The energy function is then specified in terms of potentials for each clique in the neighbourhood system. For a clique $c \in C$ we let $V_c(y)$ be its potential and then define the energy function as

$$H_V(y) = \sum_{c \in C} V_c(y).$$

Probability functions of a Gibbsian form are always strictly positive and so they are always random fields. Every random field can be expressed in Gibbsian form.

The Hammersley-Clifford Theorem

The Hammersley-Clifford theorem essentially states that the process Y is a Markov random field if and only if its corresponding probability density function is a Gibbs distribution. This result makes it possible to write down the form of the joint probability distribution for any given MRF.

The Hammersley-Clifford theorem was first presented in a paper by its authors in 1971. However Hammersley and Clifford decided not to publish the work as they felt it had been superseded by subsequent research and their original proof remained unpublished for a number of years. A proof of a special case of the theorem was given though in the seminal paper by Besag (1974) in which novel methods of inference were developed for MRFs based on the Hammersley-Clifford result. For more detail on the developments of the theorem and the reasons why it was not published by its authors, the reader is referred to Hammersley's discussion of Besag (1974).

5.2.2 The Ising Model and the Potts Model

We now consider two well-studied examples of Gibbsian models that are often used to represent images: the Ising model and its generalisation, the Potts model.

The Ising Model

We shall consider the HMRF representation of an image in more detail. Consider a rectangular lattice where observations y_i are pixels or sites on this lattice. We initially restrict our attention to a binary image where pixels can only take the values -1 or +1 corresponding to black and white (values 0 and 1 are more commonly used to represent black and white in the computing literature). Such an image is often well-represented locally by the well-known Ising model which originated in statistical physics and has been studied extensively since the 1920s. It is named after the German physicist Ernst Ising who used it to try to explain theoretical properties of ferromagnets (although the model was in fact postulated by Ising's doctoral supervisor Wilhelm Lenz in 1920). Magnetic materials are

essentially made up of atoms called magnetic dipoles (i.e. they have North and South poles). In atomic material, magnetic dipoles are caused by the spinning of the atomic particles, and are often called spins. The idea behind the Ising model is that the magnetisation of a magnetic material comprises the sum of the magnetic dipole positions of the numerous spins of the atoms within it. The model proposes that the atoms form a lattice of any geometry and at each site on this lattice there is a magnetic spin or dipole which can take the value +1 or -1. These states correspond to the physical interpretation of up- and down-pointing spins of magnitude 1. Spins cannot move around the lattice, but they can flip between up and down states. If there is an equal number of up and down states then the magnetisation will be zero; it will be non-zero if there is a majority of either state. The total system energy for a given spin configuration is proportional to the products of spins of adjacent sites, reflecting the interaction between spins on neighbouring sites which is present in real magnetic material. The Ising model is simplistic but it displays phenomena typical of more complex models and so it has been extensively researched by physicists. For further detail on the role of the Ising model in statistical physics the reader is directed to the book by Newman and Barkema (1999).

The idea of the Ising model translates in a natural way to the binary image setting, if we represent the two colours by +/-1 and assume that nearby pixels are likely to have similar colour values. Suppose we have data (or observed pixel values) y, model parameters θ and hidden variables z corresponding to the states such that $z_i \epsilon \{-1, +1\}$. Then the Ising model is of the form

$$p(z|\theta) = p(z|\beta) = \frac{\exp\{\beta \sum_{i \sim j} z_i z_j\}}{G(\beta)},$$

where $i \sim j$ means that i, j are neighbours and $G(\beta)$ is a normalising constant which is not usually computable.

The Potts Model

The natural extension of the Ising model to more than two states is given by the Potts model. In the Potts model the spin on each site can take more than two discrete values. For instance, a K-state Potts model is one in which each site can take values in states $1, \ldots, K$. In a remote sensing problem, the states might correspond to land-use types, for example.

For $K = 2$ states, the Potts model is equivalent to the Ising model up to an additive constant. Further, for larger values of K, the Potts model behaves in a similar way to the Ising model.

5.3 Estimating the Parameters of a Hidden Markov Random Field

The Intractability of the Normalising Constant

These models provide an appealing representation of images and other spatial problems. Unfortunately the difficulty with these models lies in the fact that the normalising constant, also called the partition function, cannot be evaluated unless the observation set is very small since it involves such a large number of terms. This means that there is no straightforward method for finding likelihoods or the normalising constant for the prior distribution of the HMRF and these difficulties have limited their use. The forward-backward algorithm which made such computations feasible in the HMM case is no longer available to us here. This presents a great obstacle in analyses utilising these representations and consequently the investigation of normalising constants is an area of active research. Some recent research has included a method by Pettitt, Friel and Reeves (2003) which does not require simulation. A toroidal view of the lattice is taken i.e. it is thought of as having been wrapped around a cylinder (so that the first and last columns are alongside one another) and then each column on the cylinder is taken to have two neighbouring columns, namely the columns on either side of it. A matrix approach is then taken to find the normalising constant.

There is also interest in this subject among the machine learning community. For instance, Murray and Ghahramani (2004) suggest many different ways of trying to overcome the normalising constant intractability issue. See section 5.9 for a discussion of this work. They consider several possible combinations of existing methods and highlight the fact that clearly there are many possibilities yet to be explored.

The Pseudo-likelihood Approach

The approach which is taken in this thesis involves replacing the intractable likelihood, at appropriate stages, by the pseudo-likelihood proposed by Besag

(1974,1975). This follows the approach taken by Rydén and Titterington (1998). They proposed analysing HMMs and HMRFs using versions of the Gibbs sampler which were made obtainable by using the pseudo-likelihood in the simulation step for the parameters of the Markov system. The pseudo-likelihood is given by

$$\begin{aligned} p_{PL}(z|\beta) &= \prod_{i=1}^{n} p(z_i|z_{i'},\beta), \quad \text{where} \quad z_{i'} = \{z_j : j \neq i\} \\ &= \prod_{i=1}^{n} p(z_i|z_{\delta_i},\beta), \end{aligned}$$

where z_{δ_i} denotes the z-values of the pixels which are neighbours of pixel i. The normalising constants for the factors of the pseudo-likelihood are trivial and so the unobtainable quantity has been replaced by something more amenable.

Qian and Titterington (1989) refer to this form of the pseudo-likelihood as the point pseudo-likelihood as it only involves a single site or point and its neighbours. They suggest that the point pseudo-likelihood could be generalised by partitioning the observation sites $\{1,\ldots,n\}$ into R parts where R is a number smaller than the number of sites. One could then define the pseudo-likelihood by taking the product over the conditional probability of each partition "block" given the neighbouring partition "blocks", i.e. a "block" pseudo-likelihood.

Besag's work on lattice data, which led to the proposal of the pseudo-likelihood, was motivated by the desire to find a non-degenerate conditional probability formulation of a spatial stochastic process. Bartlett (1955,1967,1968) defined the conditional probability distribution of a realisation from a rectangular lattice, given all other observations, as depending only upon the four nearest sites (or nearest neighbours) to the point in question. Besag (1974) notes the appeal of Bartlett's interpretation of conditional probability as it is computationally feasible and for binary data it leads to the formulation of the Ising model. Besag (1974) also points out that there are drawbacks to this interpretation; some of the problematic issues were considered in Besag (1972). However the practical advantages associated with this conditional probability structure encouraged Besag to pursue this type of model. Besag (1974) takes the approach of allowing the conditional distribution of a particular observation to depend on more sites

than just the 4 nearest ones and building a hierarchical modelling structure which extends the theory of 1st and higher-order Markov chains to the spatial domain. This was possible through the Hammersley-Clifford Theorem.

The Hammersley-Clifford Theorem essentially states that, in order to find the conditional probability of any observation, given all the others, we only have to know about the neighbouring observations of that site. Besag (1974) investigated coding techniques to estimate parameters of lattice schemes, but these were found to be lacking in efficiency and so Besag (1975) proposed the concept of pseudo-likelihood in an attempt to improve point estimates. The pseudo-likelihood technique is more efficient as it uses information on all sites of the lattice and does not require coding.

5.4 Hidden Binary Markov Random Field

Here we consider a variational approximation for a Hidden Binary Markov Random Field modelled by the Ising Model with independent Gaussian noise as described above in section 5.2.2. Recall that

$$p(z|\theta) = p(z|\beta) = \frac{\exp\{\beta \sum_{i\sim j} z_i z_j\}}{G(\beta)}$$

The joint probability distribution over y, z and θ will be

$$\begin{aligned} p(y,z,\theta) &= \prod_{i=1}^{n} p(y_i|z_i,\phi)p(z|\beta)\prod_{l=1}^{2} p(\phi_l)p(\beta) \\ &= \prod_{i=1}^{n}\{p(y_i|\phi_1)\}^{\frac{1-z_i}{2}}\{p(y_i|\phi_2)\}^{\frac{1+z_i}{2}} p(z|\beta)\prod_{l=1}^{2} p(\phi_l)p(\beta). \end{aligned}$$

Here the ϕ_l are parameters within the l^{th} noise model and $\frac{1-z_i}{2}$ and $\frac{1+z_i}{2}$ act as indicator variables for the state the i^{th} observation is from, since $\frac{1-z_i}{2} = 0$ when $z_i = 1$ and $\frac{1+z_i}{2} = 0$ when $z_i = -1$. $p(\phi_l)$ and $p(\beta)$ are the prior distributions for ϕ and β. Altogether, $\theta = \{\{\phi_l\}, \beta\}$

So, the lower bound on $p(y)$, that we wish to maximise, will have the form

$$\sum_{\{z\}} \int_\theta q(z,\theta) \log[\frac{\prod_{i=1}^n \{p(y_i|\phi_1)\}^{\frac{1-z_i}{2}} \{p(y_i|\phi_2)\}^{\frac{1+z_i}{2}} p(z|\beta) \prod_{l=1}^2 p(\phi_l) p(\beta)}{q(z,\theta)}] d\theta.$$

The preferred assumption is that

$$q(z,\theta) = q_z(z) q_\theta(\theta).$$

The optimal variational approximation $q(\theta)$ would then have the form

$$q_\theta(\theta) = q_\beta(\beta) \prod_{l=1}^2 q_j(\phi_l).$$

If the $p(\phi_j)$ are conjugate priors then the q_j will have simple forms. $q_\beta(\beta)$ will be problematic because of the difficulty in computing $G(\beta)$. The function $q_z(z)$ will also be difficult to deal with if it is not simplified further; therefore, another assumption is required.

We will consider the simplest proposal for $q_z(z)$, namely

$$q_z(z) = \prod_{i=1}^n q_{z_i}(z_i),$$

a mean-field like approximation. (Clearly this is a drastic assumption, but without some such simplification it is not clear how to proceed.) Then the optimal $q_{z_i}(z_i)$ maximises

$$\sum_{\{z_i\}} q_{z_i}(z_i) \int \sum_{\{z_i'\}} \prod_{j \neq i} q_{z_j}(z_j) q(\theta) \log \left[\frac{p(y_i|\phi_1)^{\frac{(1-z_i)}{2}} p(y_i|\phi_2)^{\frac{1+z_i}{2}} \exp\{\beta z_i \sum_{j \epsilon \delta_i} z_j\}}{q_{z_i}(z_i)} \right] d\theta$$

$$= \sum_{\{z_i\}} q_{z_i}(z_i) \log \left[\frac{\{\exp \mathbf{E}_{\phi_1} \log p(y_i|\phi_1)\}^{\frac{(1-z_i)}{2}} \{\exp \mathbf{E}_{\phi_2} \log p(y_i|\phi_2)\}^{\frac{1+z_i}{2}} \times \exp\{\mathbf{E}_\beta(\beta) z_i \sum_{j \epsilon \delta_i} \mathbf{E}(z_j)\}}{q_{z_i}(z_i)} \right],$$

where δ_j denotes the j that are neighbours of i. Thus,

$$q_{z_i}(z_i = -1) = \frac{\exp[\mathbf{E}_{\phi_1}\{\log p(y_i|\phi_1)\} - \mathbf{E}_\beta(\beta)\sum_{j\epsilon\delta_i}\mathbf{E}(z_j)]}{s_i}$$
$$q_{z_i}(z_i = +1) = \frac{\exp[\mathbf{E}_{\phi_2}\{\log p(y_i|\phi_2)\} + \mathbf{E}_\beta(\beta)\sum_{j\epsilon\delta_i}\mathbf{E}(z_j)]}{s_i},$$

where s_i is a normalising constant that ensures that

$$q_{z_i}(z_i = -1) + q_{z_i}(z_i = +1) = 1.$$

Of course,

$$\mathbf{E}(z_j) = q_{z_j}(z_j = +1) - q_{z_j}(z_j = -1),$$

showing that the $\{q_{z_i}(z_i)\}$ are all interlinked.

There is still the problem of deciding what to do about $q_\beta(\beta)$. Its exact solution is the maximiser of

$$\int q_\beta(\beta)\log\{\frac{\frac{\exp\{\beta\sum_{i\sim j}\mathbf{E}_q(z_iz_j)\}p(\beta)}{G(\beta)}}{q_\beta(\beta)}\}d\beta.$$

Thus,

$$q_\beta(\beta) \propto \frac{\exp\{\beta\sum_{i\sim j}\mathbf{E}_q(z_iz_j)\}p(\beta)}{G(\beta)},$$

with

$$\mathbf{E}_q(z_iz_j) = \mathbf{E}_q(z_i)\mathbf{E}_q(z_j),$$

if the above factorised approximation to $q_z(z)$ is used.

The fact that we cannot calculate $G(\beta)$ causes a problem, as does the need to normalise $q_\beta(\beta)$. Furthermore $\mathbf{E}_\beta(\beta)$ is required for the calculation of $q_z(z)$.

One possibility is to replace $p(z|\beta)$ by the pseudo-likelihood at this stage, i.e.

$$
\begin{aligned}
p_{PL}(z|\beta) &= \prod_{i=1}^{n} p(z_i|z_{\delta_i},\beta) \\
&\propto \prod_{i=1}^{n} e^{\beta z_i(\sum_{j\epsilon\delta_i} z_j)} \\
&= \prod_{i=1}^{n} \frac{e^{\beta z_i(\sum_{j\epsilon\delta_i} z_j)}}{e^{-\beta(\sum_{j\epsilon\delta_i} z_j)} + e^{\beta(\sum_{j\epsilon\delta_i} z_j)}}.
\end{aligned}
$$

This then gives

$$
\begin{aligned}
q_\beta(\beta) &\propto \frac{\exp\{2\beta \sum_{i\sim j} \mathbf{E}(q(z_i z_j))\} p(\beta)}{\exp[\sum_i \mathbf{E} \log(e^{\beta(\sum_{j\epsilon\delta_i} z_j)} + e^{-\beta(\sum_{j\epsilon\delta_i} z_j)})]} \\
&= \frac{\exp\{2\beta \sum_{i\sim j} \mathbf{E}(q(z_i z_j))\} p(\beta)}{2^n \exp[\sum_i \mathbf{E}_{q_z} \log\cosh(e^{\beta(\sum_{j\epsilon\delta_i} z_j)})]}.
\end{aligned}
$$

This is not very complicated for small neighbourhoods. Also, there may be other approximations for the denominator.

5.5 Hidden K-State Markov Random Field

In this section we extend our ideas to a Hidden Markov Random Field with K states modelled by the Potts Model with independent Gaussian noise. More detail on the derivation of formulae used in this section is given in Appendix H.1. If the number of states is $K = 2$ then the Potts model corresponds to the Ising model. Suppose we have data y, model parameters θ and hidden variables z corresponding to the states $\{1, \ldots, K\}$. Then

$$
\begin{aligned}
p(y, z, \theta) &= p(y|z,\theta)p(z|\theta)p(\theta) \\
&= \prod_{i=1}^{n} p(y_i|z_i,\phi) p(z|\beta) \{\prod_{l=1}^{K} p(\phi_l)\} p(\beta)
\end{aligned}
$$

the ϕ_l are parameters within the l^{th} noise model and, if $z_i = (z_{i1}, ..., z_{iK})$ (i.e. the $\{z_{il}\}$ are indicator variables for the state of a particular observation), then

$$p(y,z,\theta) = \left[\prod_{i=1}^{n}\prod_{l=1}^{K}\{p(y_i|\phi_l)\}^{z_{il}}\right] p(z|\beta)\{\prod_{l=1}^{K} p(\phi_l)\}p(\beta).$$

We assume that the l^{th} noise model is $N(\mu_l, \tau_l^{-1})$, where τ_l is the precision ($\tau_l^{-1} = \sigma_l^2$).

Assigning the Prior Distributions

We assign independent Gaussian priors to the means, conditional on the precisions, so that

$$p(\mu|\tau) = \prod_{l=1}^{K} N(\mu_l; m_l^{(0)}, (\lambda_l^{(0)}\tau_l)^{-1}).$$

The precisions are given independent Gamma prior distributions:

$$p(\tau) = \sum_{l=1}^{K} Ga(\tau_l; \frac{1}{2}\gamma_l^{(0)}, \frac{1}{2}\xi_l^{(0)}).$$

Form of the Variational Posterior Distributions

We wish to maximise

$$\Delta(q,p) = \sum_{\{z\}}\int_{\theta} q(z,\theta)\log\left\{\frac{p(y,z,\theta)}{q(z,\theta)}\right\}d\theta.$$

Assume that

$$q(z,\theta) = q_z(z)\left\{\prod_{l=1}^{K} q_l(\phi_l)\right\} q_\beta(\beta).$$

Then $\Delta(q,p) =$

$$\sum_{\{z\}}\int_{\theta} q_z(z)\left\{\prod_{l=1}^{K} q_l(\phi_l)\right\} q_\beta(\beta)\log\left[\frac{\left[\prod_{i=1}^{n}\prod_{l=1}^{K}\{p(y_i|\phi_l)\}^{z_{il}}\right] p(z|\beta)\{\prod_{l=1}^{K} p(\phi_l)\}p(\beta)}{q_z(z)\left\{\prod_{l=1}^{K} q_l(\phi_l)\right\} q_\beta(\beta)}\right] d\theta \tag{5.1}$$

Then $q_l(\phi_l)$ optimises

$$\int_{\phi_l} q_l(\phi_l) \log \left[\frac{\exp\{\sum_{i=1}^n \mathbf{E}_{z_i} z_{il} \log p(y_i|\phi_l)\} p(\phi_l)}{q_l(\phi_l)} \right] d\phi_l$$

i.e.

$$q_l(\phi_l) \propto \prod_{i=1}^n \left\{ p(y_i|\phi_l)^{\mathbf{E}_z z_{il}} \right\} p(\phi_l),$$

where

$$\begin{aligned} \mathbf{E}_z z_{il} &= P_{q_z}(i^{th}\text{data point is in class } l) \\ &= q_{il}, \end{aligned}$$

according to the notation used in previous chapters. Thus,

$$q_l(\phi_l) \propto \prod_{i=1}^n \{ p(y_i|\phi_l)^{q_{il}} \} p(\phi_l). \tag{5.2}$$

This results in variational posteriors of the form

$$q(\mu_l|\tau_l) = N(\mu_l; m_l, (\lambda_l \tau_l)^{-1})$$
$$q(\tau_l) = Ga(\tau_l; \frac{1}{2}\lambda_l, \frac{1}{2}\xi_l),$$

with hyperparameters given by

$$\lambda_l = \lambda_l^{(0)} + \sum_{i=1}^n q_{il}$$

$$\gamma_l = \gamma_l^{(0)} + \sum_{i=1}^n q_{il}$$

$$m_l = \frac{\lambda_l^{(0)} m_l^{(0)} + \sum_{i=1}^n q_{il} y_i}{\lambda_l}$$

$$\xi_l = \xi_l^{(0)} + \sum_{i=1}^n q_{il} y_i^2 + \lambda_l^{(0)} {m_l^{(0)}}^2 - \lambda_l m_l^2.$$

Also, $q_\beta(\beta)$ optimises

$$\int_\beta q_\beta(\beta) \log \left[\frac{\exp\{\mathbf{E}_z \log p(z|\beta)\} p(\beta)}{q_\beta(\beta)} \right] d\beta$$

i.e.

$$q_\beta(\beta) \propto \exp\{\mathbf{E}_z \log p(z|\beta)\}\, p(\beta). \tag{5.3}$$

Finally, $q_z(z)$ optimises

$$\sum_{\{z\}} q_z(z) \log \left[\frac{\exp\{\sum_{i=1}^n \sum_{l=1}^K z_{il} \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) + \mathbf{E}_\beta \log p(z|\beta)\}}{q_z(z)} \right]. \tag{5.4}$$

We are unable to optimise (5.3) and (5.4) explicitly. We encounter difficulties because of the complexity of $p(z|\beta)$. In Section 5.5.1 we describe a way of circumventing the problem of optimising (5.4) which involves assuming a fully factorised form for $q_z(z)$. We also require $\mathbf{E}_\beta(\beta)$ in order to evaluate (5.4). Section 5.5.2 describes how (5.3) can be optimised using the pseudo-likelihood in place of the true likelihood function and Section 5.5.3 describes an approximation for $\mathbf{E}_\beta(\beta)$.

5.5.1 Optimisation of $q_z(z)$

Formula (5.4) cannot be optimised explicitly because $\log p(z|\beta)$ is a fairly complicated function of z. As in the binary case, the simplest proposal is to assume that

$$q_z(z) = \prod_{i=1}^n q_{z_i}(z_i).$$

Then q_{z_i} optimises

$$\sum_{\{z_i\}} q_{z_i}(z_i) \log \left[\frac{\exp\{\sum_{l=1}^K z_{il} \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) + \mathbf{E}_\beta \mathbf{E}_{z_{i'}} \log p_i(z_i|\beta)\}}{q_{z_i}(z_i)} \right], \tag{5.5}$$

where $z_{i'}$ represents all z_j's except for z_i.

We now have to consider the form of $\log p_i(z_i|\beta)$, which is the part of $\log p(z|\beta)$ that depends on z_i. Consider an isotropic pairwise-association Markov random field, for which

$$p(z|\beta) = \frac{\exp\{\beta \sum_{i\sim j} \delta(z_i, z_j)\}}{G(\beta)}.$$

This is the Potts model, where $i \sim j$ means that i, j are neighbours and

$$\begin{aligned} \delta(z_i, z_j) &= 1, \text{if} \quad {z_i}^T z_j = 1 \\ &= -1, \text{otherwise, i.e. if} \quad {z_i}^T z_j = 0. \end{aligned}$$

Here

$${z_i}^T z_j = \sum_{l=1}^{K} z_{il} z_{jl},$$

i.e.

$$\begin{aligned} \delta(z_i, z_j) &= 2{z_i}^T z_j - 1 \\ &= 2\sum_{l=1}^{K} z_{il} z_{jl} - 1 \end{aligned}$$

i.e.

$$p(z|\beta) = \frac{\exp\{2\beta \sum_{i\sim j} \sum_{l=1}^{K} z_{il} z_{jl}\}}{G^*(\beta)}$$

where $G^*(\beta) = G(\beta)e^{\beta}$, so that $G^*(\beta)$ absorbs the extra constant, $e^{-\beta}$, from the numerator.

$$\log p(z|\beta) = 2\beta \sum_{i\sim j} \sum_{l=1}^{K} z_{il} z_{jl} - \log G^*(\beta).$$

The quantity required to go in (5.5) as $\mathbf{E}_\beta \mathbf{E}_{z_i} \log p_i(z_i|\beta)$ is

$$\mathbf{E}_\beta \left\{ 2\beta \sum_{l=1}^{K} z_{il} \sum_{j\epsilon\delta_i} \mathbf{E}_{z_j} z_{jl} \right\} = 2\mathbf{E}_\beta(\beta) \sum_{l=1}^{K} z_{il} (\sum_{j\epsilon\delta_i} q_{jl}).$$

Thus, in (5.5), we have

$$\sum_{\{z_i\}} q_{z_i}(z_i) \log \left[\frac{\exp\{\sum_{l=1}^{K} z_{il}(\mathbf{E}_{\phi_l} \log p(y_i|\phi_l) + 2\mathbf{E}_\beta(\beta) \sum_{j \epsilon \delta_i} q_{jl})\}}{q_{z_i}(z_i)} \right],$$

which is optimised by

$$q_{il} \propto \exp \left\{ \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) + 2\mathbf{E}_\beta(\beta) \sum_{j \epsilon \delta_i} q_{jl} \right\}, \quad l = 1, ..., K, \tag{5.6}$$

normalised so that $\sum_{l=1}^{K} q_{il} = 1$. In the above, apart from an additive constant

$$\mathbf{E}_{\phi_l}[\log p(y_i|\phi_l)] = \frac{1}{2}\mathbf{E}_{\phi_l}[\log|\tau_l|] - \frac{1}{2}\mathbf{E}_{\phi_l}[\tau_l](y_i - m_l)^2 - \frac{1}{2\xi_l},$$

with expectations given by

$$\mathbf{E}_{\phi_l}[\log|\tau_l|] = \Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2})$$

$$\mathbf{E}_{\phi_l}[\tau_l] = \frac{\gamma_l}{\xi_l}.$$

Therefore, given the q_{il}'s, the $q_l(\phi_l)$'s can be updated through (5.2). Given $\mathbf{E}_\beta(\beta)$, the $\{q_{il}\}$ can be calculated/updated through (5.6).

5.5.2 Optimisation of $q_\beta(\beta)$

We now have to deal with β. The optimum q_β optimises

$$\int_\beta q_\beta(\beta) \log \left[\frac{\exp\{\mathbf{E}_z \log p(z|\beta)\} p(\beta)}{q_\beta(\beta)} \right] d\beta \tag{5.7}$$

i.e.

$$q_\beta(\beta) \propto \exp\{\mathbf{E}_z \log p(z|\beta)\} p(\beta)$$

with

$$\log p(z|\beta) = 2\beta \sum_{i \sim j} \sum_{l=1}^{K} z_{il} z_{jl} - \log G^*(\beta),$$

so that

$$\mathbf{E}\log p(z|\beta) = 2\beta \sum_{i\sim j} \sum_{l=1}^{K} \mathbf{E}_{z_i z_j}(z_{il} z_{jl}) - \log G^*(\beta).$$

If we assume a factorised $q_z(z)$ then

$$\mathbf{E}\log p(z|\beta) = 2\beta \sum_{i\sim j} \sum_{l=1}^{K} q_{il} q_{jl} - \log G^*(\beta),$$

so that

$$q_\beta(\beta) \propto \frac{\exp\{2\beta \sum_{i\sim j} \sum_{l=1}^{K} q_{il} q_{jl}\} p(\beta)}{G^*(\beta)}.$$

In principle $q_\beta(\beta)$ can be updated from the q_{il}'s. However, this is not practicable in practice and so we require a further approximation.

A Pseudo-Likelihood Approach

In the spirit of Rydén and Titterington (1998), our approach is to use the pseudo-likelihood (Besag (1974,1975)) and to replace $p(z|\beta)$ by

$$p_{PL}(z|\beta) := \prod_{i=1}^{n} p(z_i|z_{i'}, \beta)$$

i.e. by

$$\prod_{i=1}^{n} p(z_i|z_{\delta_i}, \beta).$$

Here

$$p(z|\beta) \propto \exp\left\{2\beta \sum_{i\sim j} \sum_{l=1}^{K} z_{il} z_{jl}\right\}$$

and

$$p(z_i|z_{i'}, \beta) \propto \exp\left\{2\beta \sum_{l=1}^{K} z_{il} \sum_{j\in\delta_i} z_{jl}\right\}.$$

$$p(z_i|z_{i'},\beta) = \frac{\exp\left\{2\beta\sum_{l=1}^{K} z_{il}\sum_{j\epsilon\delta_i} z_{jl}\right\}}{\sum_{z_i}\exp\left\{2\beta\sum_{l=1}^{K} z_{il}\sum_{j\epsilon\delta_i} z_{jl}\right\}} = \frac{\exp\left\{2\beta\sum_{l=1}^{K} z_{il}\sum_{j\epsilon\delta_i} z_{jl}\right\}}{\sum_{l=1}^{K}\exp\left\{2\beta\sum_{j\epsilon\delta_i} z_{jl}\right\}}.$$

To see this note that possible values for z_i are

$$\begin{pmatrix}1\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}, \begin{pmatrix}0\\1\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}, \ldots, \begin{pmatrix}0\\0\\\cdot\\\cdot\\\cdot\\1\end{pmatrix}.$$

Thus

$$\tilde{p}(z|\beta) = \prod_{i=1}^{n} \frac{\exp\left\{2\beta\sum_{l=1}^{K} z_{il}\sum_{j\epsilon\delta_i} z_{jl}\right\}}{\sum_{l=1}^{K}\exp\left\{2\beta\sum_{j\epsilon\delta_i} z_{jl}\right\}}.$$

If we replace $p(z|\beta)$ by $\tilde{p}(z|\beta)$ in (5.7), then the optimum $q_\beta(\beta)$ optimises

$$\int_\beta q_\beta(\beta)\log\left[\frac{\exp\{\mathbf{E}_z\log\tilde{p}(z|\beta)\}\,p(\beta)}{q_\beta(\beta)}\right]d(\beta),$$

giving

$$q_\beta(\beta) \propto \exp\{\mathbf{E}_z\log\tilde{p}(z|\beta)\}\,p(\beta). \tag{5.8}$$

However,

$$\begin{aligned}\log\tilde{p}(z|\beta) &= 2\beta\sum_{i=1}^{n}\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} z_{il}z_{jl} - \sum_{i=1}^{n}\log\left\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} z_{jl})\right\}\\ &= 4\beta\sum_{i\sim j}\sum_{l=1}^{K} z_{il}z_{jl} - \sum_{i=1}^{n}\log\left\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} z_{jl})\right\},\end{aligned}$$

and thus, assuming a factorised form for $q_z(z)$, we have

$$\mathbf{E}_z \log \tilde{p}(z|\beta) = 4\beta \sum_{i\sim j} \sum_{l=1}^{K} q_{il} q_{jl} - \sum_{i=1}^{n} \mathbf{E}_{z_{\delta_i}} \left[\log \left\{ \sum_{l=1}^{K} \exp(2\beta \sum_{j \epsilon \delta_i} z_{jl}) \right\} \right]$$

$$\Longrightarrow \exp\{\mathbf{E}_z \log \tilde{p}(z|\beta)\} = \frac{\exp\{4\beta \sum_{i\sim j} \sum_{l=1}^{K} q_{il} q_{jl}\}}{\exp(\sum_{i=1}^{n} \mathbf{E}_{z_{\delta_i}} [\log\{\sum_{l=1}^{K} \exp(2\beta \sum_{j \epsilon \delta_i} z_{jl})\}])}. \quad (5.9)$$

In principle this can be used in (5.8), but, in the case of a first-order hidden Markov random field, each $\mathbf{E}_{z_{\delta_i}}$ in the denominator contains K^4 terms, making computation impractical.

Here we suggest tackling this obstacle by approximating the denominator of (5.9) by

$$\exp\left[\sum_{i=1}^{n} [\log\{\sum_{l=1}^{K} \exp(2\beta \sum_{j \epsilon \delta_i} q_{jl})\}] \right] = \prod_{i=1}^{n} \left\{ \sum_{l=1}^{K} \exp(2\beta \sum_{j \epsilon \delta_i} q_{jl}) \right\}, \quad (5.10)$$

so that

$$q_\beta(\beta) \propto \prod_{i=1}^{n} \frac{\exp\{2\beta \sum_{j \epsilon \delta_i} \sum_{l=1}^{K} q_{il} q_{jl}\} p(\beta)}{\{\sum_{l=1}^{K} \exp(2\beta \sum_{j \epsilon \delta_i} q_{jl})\}}. \quad (5.11)$$

We have to choose a prior for β but there is no conjugate set-up for this. We shall use an improper prior for β, i.e. we will set $p(\beta) =$ constant.

There is still the problem of calculating $\mathbf{E}_\beta(\beta)$, for use in (5.6). We have

$$q_\beta(\beta) = CQ(\beta), \quad (5.12)$$

where $Q(\beta)$ is the right-hand side of (5.11) and C is a normalising constant. The difficulty here lies in calculating C. A possible solution is to approximate $q_\beta(\beta)$ and consequently the associated expectation.

5.5.3 Approximating the Expected Value of β

We wish to approximate $q_\beta(\beta)$ to obtain an approximation to $\mathrm{E}_\beta(\beta)$. Taking the log of (5.12) gives

$$\begin{aligned}\log q_\beta(\beta) &= \log(CQ(\beta)) \\ &= \log C + \log Q(\beta) \\ &\approx \text{constant} - \frac{1}{2\hat{\sigma}_\beta^2}(\beta - \hat{\beta})^2.\end{aligned}$$

i.e. We are approximating to $\log Q(\beta)$ by a quadratic. This type of approximation seemed reasonable looking at plots of $\log q_\beta(\beta)$. Then, we have,

$$q_\beta(\beta) \propto \exp\{-\frac{1}{2\hat{\sigma}_\beta^2}(\beta - \hat{\beta})^2\}$$

and so, approximately,

$$\beta \sim N(\hat{\beta}, \hat{\sigma}_\beta^2).$$

Estimates of $\hat{\beta}$ and $\hat{\sigma}_\beta^2$ can be found via the method of least squares estimation. This, of course, requires calculation of the value of $Q(\beta)$ for various values of β, we used values of β ranging from -1.5 to 1.5 with increments of 0.05. The prior for β was taken as being Uniform and so $p(\beta)$ was constant in the calculation.

However, when we actually implemented the method, using simulated data sets, the estimate of $\hat{\beta}$ obtained via the least squares method tended to be larger than the true value. In fact, it turned out that a better approximation to $\hat{\beta}$ was obtained by taking the mode of $\log Q(\beta)$ and as this led to better results, we used this as the estimate of $\hat{\beta}$ instead.

5.5.4 Obtaining the DIC and p_D Values

The formula for p_D is given by

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}.$$

and so in this case the form of p_D is

$$p_D = -\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\{\Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2}) - \frac{1}{\lambda_l}\} + \hat{\sigma}_\beta^2 F''(\hat{\beta}) + 2\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\log(\frac{\gamma_l}{\xi_l})$$
$$-2\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\} \tag{5.13}$$

See Appendix H.2 for further detail, including the definition of $F''(\hat{\beta})$. The DIC can then be obtained through the usual formula,

$$\text{DIC} = 2p_D - 2\log p(y|\tilde{\theta}).$$

However the difficulty with this formula is that the likelihood,

$$\begin{aligned} p(y|\theta) &= \sum_z p(y|z)p(z) \\ &= \sum_z \prod_{i=1}^{n} p(y_i|z_i)p(z), \end{aligned}$$

would be too computationally intensive to compute. Even use of the pseudo-likelihood, which gives

$$p(y|\theta) = \sum_z \prod_{i=1}^{n} p(y_i|z_i) \prod_{i=1}^{n} p(z_i|z_{\delta_i}),$$

would not simplify computation sufficiently. Instead, we propose using a mean-field approximation for the lower bound of the likelihood.

Mean-Field Approximation

Mean-field methods are a computationally efficient way of approximating intractable posterior probabilities. The simplest form of this approximation (known as simple mean-field approximation), which we adopt, involves using a completely factorised approximating function for the distribution of interest. The motivation behind the method is that, in a large lattice, each site is affected by interactions with numerous others, so that each individual influence is small and the total influence is approximately additive. Intuitively then, each site should be roughly characterised by its mean value. Each mean value is known only through its re-

lation to every other mean value, meaning that one can obtain coupled equations for the mean values which can be solved iteratively. Mean-field methodology can be developed from more than one perspective. The view we shall take is the variational one. This involves using the KL divergence as a measure of the quality of our approximation. Through Jensen's inequality we can obtain a lower bound on our likelihood which is maximised by minimising the KL divergence. See Appendix H.3 for more detail. An overview of mean-field theory and applications is given by Opper and Saad (2001).

Here we consider the mean-field approximation in the case of binary images. The variational lower bound on the likelihood is

$$\log p(y|\theta) \geq \sum_{\{z\}} \prod_{i}^{n} q_{z_i}(z_i) \log \frac{p(y|z)p(z)}{\prod_i^n q_{z_i}(z_i)}.$$

Note that our approximating distribution $q(z)$ is fully factorised. Our mean values are related through the following set of nonlinear equations

$$m_i = \frac{p(y_i|z_i=+1)e^{\beta \sum_{j\in\delta_i} m_j} - p(y_i|z_i=-1)e^{-\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i=+1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i=-1)e^{-\beta \sum_{j\in\delta_i} m_j}}.$$

This set of equations for the m_i can be solved iteratively. When we have the values of the m_i we can calculate

$$q_{z_i}(z_i=-1) = \frac{p(y_i|z_i=-1)e^{-\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i=+1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i=-1)e^{-\beta \sum_{j\in\delta_i} m_j}}$$

$$q_{z_i}(z_i=+1) = \frac{p(y_i|z_i=+1)e^{\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i=+1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i=-1)e^{-\beta \sum_{j\in\delta_i} m_j}}.$$

Then the approximate lower bound for $p(y|\theta)$ is given by

$$\sum_{i=1}^{n}\left[\sum_{\{z_i\}} q_{z_i}(z_i)\log\left\{\frac{p(y_i|z_i)}{q_{z_i}(z_i)}\right\}\right] + 2\beta\sum_{i\sim j} m_i m_j - \sum_{i=1}^{n}\sum_{\{z_j:j\in\delta_i\}}\prod_{j\in\delta_i} q_j(z_j)\log(e^{\beta\sum_{j\in\delta_i} z_j} + e^{-\beta\sum_{j\in\delta_i}}$$

This is our estimated likelihood, into which we substitute $\tilde{\theta}$ in order to calculate

the DIC.

5.6 Practical Implementation

We implemented the variational method and calculation of the DIC and p_D values in R. The code was written to deal with HMRF's having any number of states but in our examples we focused on application to binary HMRF's. Note that we have only approximated the DIC in the binary case.

As with the previous programs, the user must specify the initial number of states, K, and the observed data. The user's input data must contain the observed data and a list of indices ranging from $1, ..., K$ initially allocating the observations to one of the K states. In our examples, we allocated roughly equal numbers of the observations to each of the K states, not using any particular method. The initial allocation starts the algorithm and as iterations proceed, the observations find their own places. At the initialisation stage, a user-specified value is given for the weight that is to be assigned to each observation indicator variable (the q_{il}'s). In the same way as for the mixture and HMM programs, these initial values for the q_{il}'s were chosen to give a slightly higher weighting to the initial allocation to states to start the algorithm. The user has the option to specify initial values for the sufficient statistics or, alternatively, defaults which specify broad priors are available. Unless otherwise indicated, all of our examples use these broad priors.

The estimated q_{il}'s, obtained as the algorithm converges, are nonnegative and are normalised so that they sum to 1 over l for each i. They therefore form a set of predictive probabilities for the indicator variables for the data. The sum of the $q_z(z_i = l)$ over the n observations for each state provides an estimate of the number of observations that are being allocated to each state and can be thought of as a weighting for each state.

At each iteration of the code, the DIC and p_D values are computed and the updated weights for each component are obtained.

In our examples, we only considered data sets simulated from a binary HMRF, but we did try initialising the algorithm with more than 2 states, removing from consideration any state with less than 1 observation assigned to it, as we did with the previous models. We did occasionally observe the state removal phenomenon for certain initial values for the q_{il}'s, but in most experiments, extra states were

not removed.

5.7 Simulating a Binary Image Using Gibbs Sampling

5.7.1 The Gibbs Sampler (or Alternating Conditional Sampling) for Sampling from a Posterior Distribution

Gibbs sampling is the simplest Markov Chain Monte Carlo (MCMC) algorithm. The idea of MCMC algorithms is to sample from a Markov chain with a stationary distribution which is the target distribution of interest. It has been found to be useful in many multidimensional problems. Suppose, for instance, that it is of interest to sample from $p(\theta|y)$, the posterior density of a set of parameters θ, given data y. The algorithm is defined in terms of subvectors of θ. Suppose first of all that the parameter θ has been divided into d subvectors or components $\theta = (\theta_1, ..., \theta_d)$.

At each iteration step of the Gibbs sampler we cycle through the subvectors of θ, drawing each subset conditional on the value of the remaining subvectors. So, each iteration, t, involves d steps.

At each iteration, an ordering of the d subvectors is chosen and each θ_j^t is sampled from the conditional distribution given all the other subvectors of θ, i.e.

$$p(\theta_j|{\theta_{j'}}^{t-1}, y),$$

where ${\theta_{j'}}^{t-1}$ represents all components of θ except for θ_j, at their current values. Thus,

$${\theta_{j'}}^{t-1} = (\theta_1^t, ..., {\theta_{j-1}}^t, {\theta_{j+1}}^{t-1}, ..., {\theta_d}^{t-1}).$$

For most standard problems it is generally possible to sample directly from most or all of the conditional posterior distributions of the parameters. Under certain conditions, the sampler ultimately provides realisations from $p(\theta|y)$.

As mentioned previously, the Gibbs sampler can be considered a special case of the Metropolis-Hastings algorithm (due to Hastings (1970)) which generalises the Metropolis algorithm (Metropolis et al. (1953)).

5.7.2 Using a Gibbs Sampler to Sample from a Binary Markov Random Field

The Ising model for observations has the form

$$p(z|\beta) = \frac{\exp\{\beta \sum_{i\sim j} z_i z_j\}}{G(\beta)},$$

where $z_i \in \{-1, +1\}$ and $G(\beta)$ is the normalising function. We have

$$p(z_i|z_{i'}) = \frac{e^{\beta z_i \sum_{j\epsilon\delta_i} z_j}}{e^{\beta \sum_{j\epsilon\delta_i} z_j} + e^{-\beta \sum_{j\epsilon\delta_i} z_j}}.$$

We can simulate from the Binary HMRF as follows:

- Find the neighbours of each point in the image.
- Randomly assign points on the grid to be +1/-1 initially. This provides an initial estimate for our image Z given by

$$Z^{(0)} = (z_1^{(0)}, ..., z_n^{(0)}).$$

- We can then use the Gibbs sampler to set points in the simulated image as +1/-1. We begin by dividing Z into n subvectors corresponding to the n pixels in the image. As described in the previous section, in each iteration, we cycle through each of the n pixels, drawing each one conditional on the current values of the other remaining ones. I.e. for $m = 1, 2, ...$, generate

$$z_1^{(m)} \text{ from } p(z_1^{(m)}|z_2^{(m-1)}, ..., z_n^{(m-1)}),$$

...

$$z_i^{(m)} \text{ from } p(z_i^{(m)}|z_1^{(m)}, ..., z_{i-1}^{(m)}, z_{i+1}^{(m-1)}, ..., z_n^{(m-1)}),$$

...

$$z_n^{(m)} \text{ from } p(z_n^{(m)}|z_1^{(m)}, ..., z_{n-1}^{(m)}),$$

where the probability of a pixel being white, conditional on the other pixels, is given by

$$\begin{aligned} p(z_i = 1|z_{i'}) &= \frac{e^{\beta \sum_{j \epsilon \delta_i} z_j}}{e^{\beta \sum_{j \epsilon \delta_i} z_j} + e^{-\beta \sum_{j \epsilon \delta_i} z_j}} \\ &= 1 - p(z_i = -1|z_{i'}). \end{aligned}$$

- Once convergence has been reached, we can add Gaussian white noise to each pixel in our simulated image so that the noisy $\{y_i\}$ which we actually observe are given by

$$z_i + \text{ realisation from } N(0, \sigma^2).$$

The choice of β will affect how patchy the resulting image will be. Larger positive values of β encourage neighbouring pixels to be of the same colour so that increasing β leads to bigger sections of like coloured pixels in the image.

5.8 Results from the Analysis of Simulated Binary Images with Added White Gaussian Noise

We tested the algorithm on some data sets simulated from a binary hidden Markov random field. In each simulated image, the number of observations, or pixels, is 3136 and the image is of size 56×56. The true images are made up of black and white pixels, corresponding to the values -1 and +1, respectively. The images in this chapter were produced by a C++ application which read in the relevant image data and produced the noisy, true and recovered images. True images had values of -1 or +1 which were plotted as black and white pixels, respectively. In the noisy images, values were plotted according to a grey-scale. In the 2-state recovered images, the i^{th} pixel is labelled white if, in the variational posterior solution, $q(z_i = +1) > \frac{1}{2}$, and black otherwise.

5.8.1 Data generated from an Ising Model with $\beta = 0.45$

Simulated data sets 1-3 are generated by adding white Gaussian noise to a simulated image based on an Ising model with $\beta = 0.45$. The true image is that which is shown in Figure 5.3 and the value of β has led to an image which is made up of fairly large patches of like-coloured pixels.

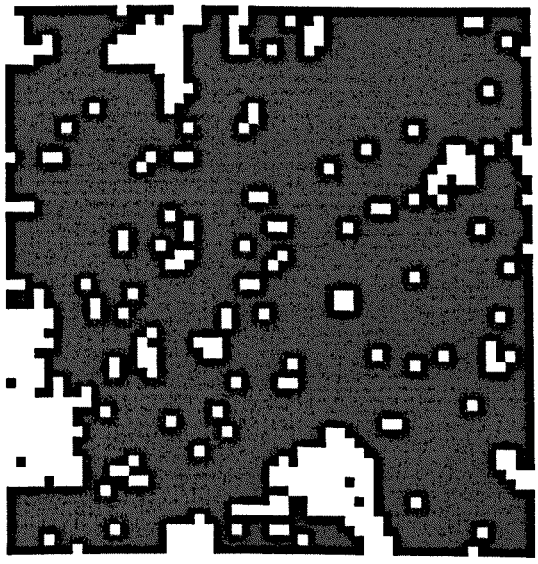

Figure 5.3: True image used in simulated data-sets 1-3

Initialising the Algorithm 2 States

We first analysed the data initialising the algorithm with 2 states, which is of course the correct number.

Simulated Data Set 1 was generated by adding Gaussian noise, with mean equal to 0 and a standard deviation of 0.25, to the true image shown above (Figure 5.3). The noisy image is shown in Figure 5.4. For Simulated Data Set 1, the HMRF variational program was initialised with 2 states and both initial means set to 0. Figure 5.5 shows the recovered image. A solution is reached in ten iterations of the program and variational estimates of the posterior means are -1.001 and 0.9998. Both estimates are very close to the true means. The expected posterior standard deviations are 0.25 and 0.25, which are equal to the true values. Our posterior estimate for β is 0.45, which again is equal to the true value of the parameter. 2701 (around 86%) pixels are labelled black and 435 (around 14%) are labelled white. The DIC is -31016 and p_D is -12659.61. All 3136 pixels are correctly labelled.

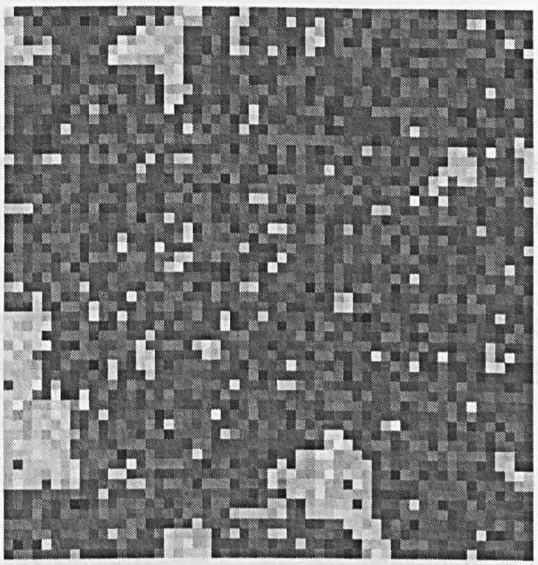

Figure 5.4: Noisy Image : Simulated Data Set 1

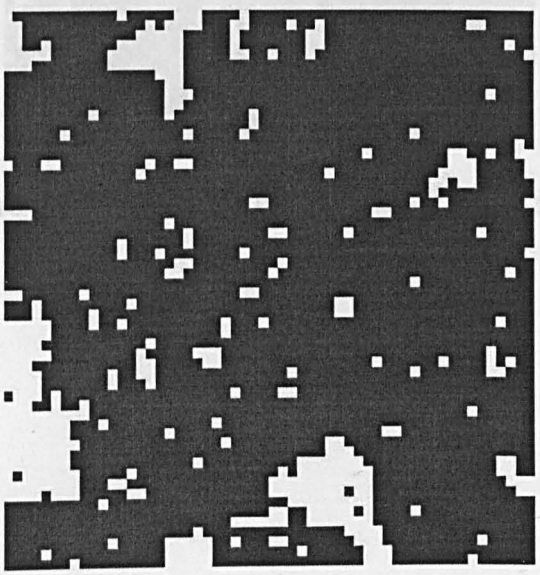

Figure 5.5: Recovered Image : Simulated Data Set 1

Simulated Data Set 2 is based on the same true distribution as Simulated Data Set 1, but more noise has been added (Figure 5.6). The added noise has mean equal to 0 and standard deviation equal to 0.5. When the HMRF program was run with 2 states and both initial means set to 0, we obtained the recovered image shown in Figure 5.7. Seventeen iterations are required before a solution is reached, at which the posterior means are -1.005 and 0.958, and the posterior standard deviations are 0.49 and 0.54. The posterior estimate for β is 0.45. Even with this extra noise, we have still obtained good estimates of the parameters. Again, around 86% of pixels are labelled as black and around 14% are labelled as white. The p_D value is -16886.94, and the DIC is -35262. This time, 3114 pixels, 99.3%, were labelled correctly.

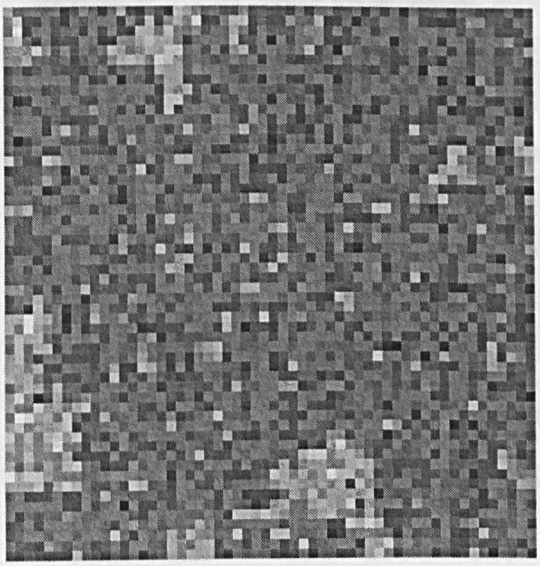

Figure 5.6: Noisy Image : Simulated Data Set 2

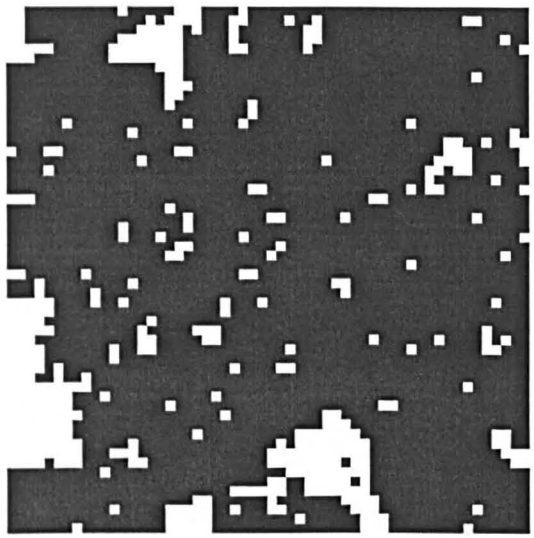

Figure 5.7: Recovered Image : Simulated Data Set 2

Simulated Data Set 3 is also based on the same true distribution as Simulated Data Set 1 but the added noise has mean equal to 0 and standard deviation equal to 1; see Figure 5.8. Again the HMRF program was run with 2 states and both initial means set to 0. The recovered image obtained is shown in Figure 5.9. A solution was found after 24 iterations. The posterior means were -1.072 and 0.937, and the posterior standard deviations were 0.98 and 0.96. This is the noisiest version of the image yet these posterior estimates are still reasonably close to the true values. In this case, the posterior estimate for β is 0.4 which is slightly lower than the true value. 83% of pixels were labelled as black and 17% were labelled as white. The p_D value was -18525.59 and DIC was -34620. This time 2970 pixels, 94.7%, were labelled correctly.

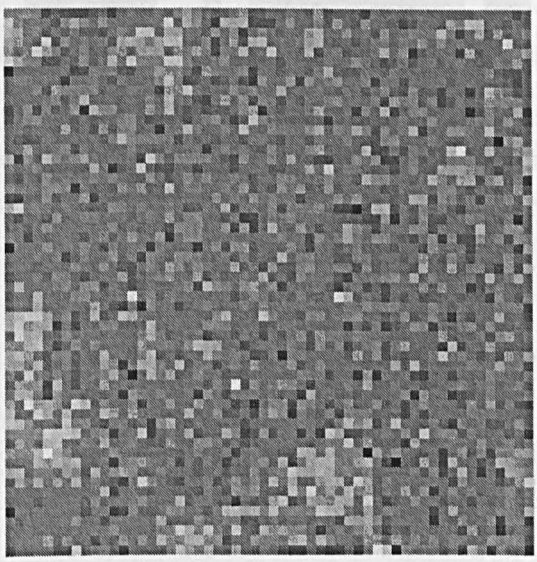

Figure 5.8: Noisy Image : Simulated Data Set 3

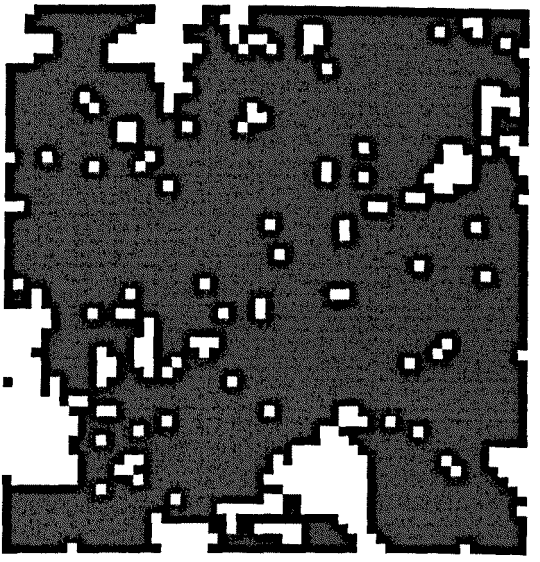

Figure 5.9: Recovered Image : Simulated Data Set 3

With the variational approximation we are able to obtain good posterior estimates of the parameters of a HMRF with added white Gaussian noise. We are also able to recover a close representation of the original image. When noise is low, we can accurately label 100% of the pixels, but clearly, as the added noise increases, the performance deteriorates, as we would expect. For all three data-sets, we also tried initialising the algorithm with means equal to -1 and +1, but this had no effect whatsoever on the results we obtained.

Initialising the Algorithm with More than 2 States

To investigate what would happen to the results if we initialised the algorithm with more than two states, we re-analysed data sets 1 to 3, this time initialising the algorithm with 4 states and initial means equal to 0. Most of the time, this lead to a variational solution with 4 states, but, for certain initial values of the weights for the q_{il}'s, the state removal phenomenon did occur. However, the resulting solutions when state removal had occurred, even when both extra states were removed, were not close to the true distribution for these examples. The algorithm also required a much larger number of iterations to converge a solution, whether states were removed or not, than were required when we initialised with 2 states.

For simulated data set 1, most initial values for the q_{il}'s led to a 4-state solution. The variational posterior means were 0.99, -0.99, -0.99 and -1.01, posterior standard deviations were 0.25, 0.27, 0.25 and 0.22, and the posterior weights for the states were 0.14, 0.23, 0.45 and 0.18. The extra states were not removed, but the posterior mean estimates were close to the true values of +/-1, and the posterior estimates of the standard deviations were all fairly close to the true values. However, 3 of the states seem to be representing the same part of the data. The weight appears to be fairly evenly spread amongst the states, but note that adding the weights assigned to states 2, 3 and 4 (which seem to correspond to black pixels) gives 0.86, and so we still have the same proportions of pixels in the categories black and white as we had in the 2-state solution. The estimate for β was 0.55, and is higher than the true value. So, we have obtained a solution with what appears to be 3 states corresponding to the same true state (-1), the posterior estimates of which are close to the true value and with weight spread evenly amongst the 3.

It was possible, for simulated data set 1, to obtain a 3-state solution. How-

ever, this only occurred for one configuration of the initial values for the q_{il}'s, a configuration which assigned an extremely low weight to the initial assignment of the observations to states. In this instance, the posterior estimates of the means were 0.99, -1.00 and -1.00, the variational posterior standard deviations were 0.25, 0.25 and 0.25, and the estimated weights were 0.14, 0.29 and 0.57. One extra state has been removed but the second and third states seem to represent the same component. If state 1 represents the white pixels, and states 2 and 3 represent the black pixels, then again we have the same proportions assigned to each as we did in the 2-state solution. The estimated value of β in this solution was 0.55.

For simulated data set 2, several initial values for the q_{il}'s led to the removal of one of the extra states, despite the fact that this data set had more added noise. However, the parameter estimates from the resulting 3-state solutions were not satisfactory.

For simulated data set 3, most initialisations of the q_{il}'s led to the removal of at least one of the extra states. This is surprising as it would seem more natural for states to be removed when noise is low and there is clearer separation in the data. Unfortunately, as with simulated data set 2, the resulting parameter estimates for this data set were not close to the true distribution, even when both extra states were removed.

From these results, it appears that, in the HMRF framework, the noisier the data, the more likely it is for extra states to be removed by the variational approximation algorithm. However, for noisy data, initialising the algorithm with more states than are truly present, prevents the method from obtaining satisfactory results.

5.8.2 Data generated from an Ising Model with $\beta = 0.6$

Simulated data sets 4 and 5 are generated by adding white Gaussian noise to a simulated image based on an Ising model with $\beta = 0.6$. The true image is that which is shown in Figure 5.10 and one can see that increasing the value of β to 0.6 has produced an image which is less patchy and is mainly one colour (black).

Initialising the Algorithm with 2 States

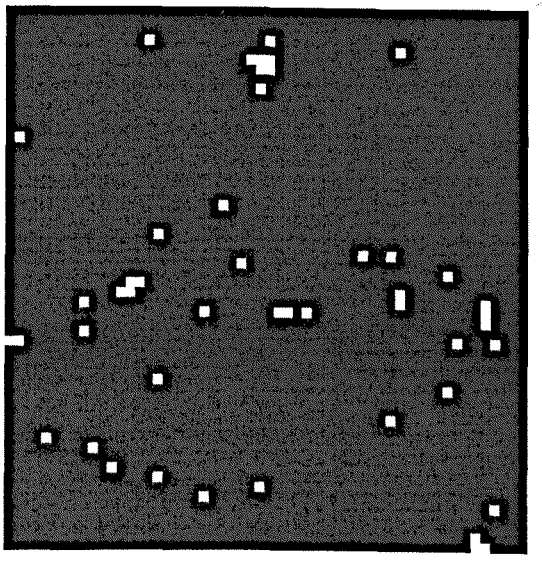

Figure 5.10: True image used in simulated data-sets 4 and 5

We analysed the data initialising the algorithm with the true number of states (2). As above, in the recovered images, the i^{th} pixel is labelled white if, in the variational posterior solution, $q(z_i = +1) > \frac{1}{2}$, and black otherwise.

Simulated Data Set 4 was generated by adding Gaussian noise, with mean equal to 0 and a standard deviation of 0.25, to the true image (Figure 5.10). The noisy image is shown in Figure 5.11. The variational algorithm correctly labelled 100% of the pixels and returned good estimates of the parameters in fourteen iterations of the algorithm. The variational posterior means were 1.01 and -0.99, the posterior standard deviations were 0.24 and 0.25 and the fitted weights were 0.02 and 0.98, i.e. 98% of the pixels were labelled as black. The value of p_D was -20739.84 and the DIC was -47944. The method also correctly estimated that the value of β was 0.6.

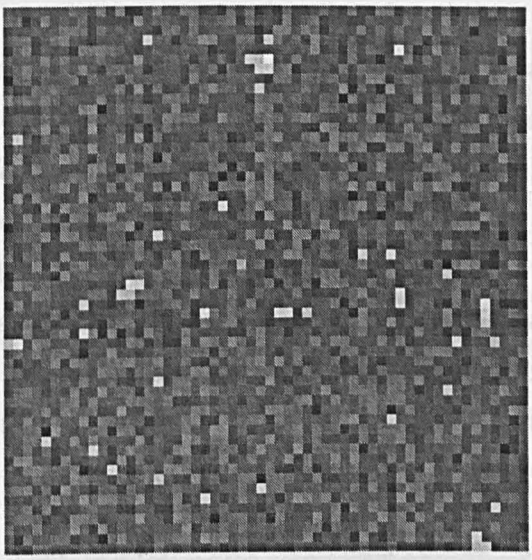

Figure 5.11: Noisy Image : Simulated Data Set 4

Simulated Data Set 5 had more noise added to the true image (Figure 5.10). The noisy image is shown in Figure 5.12. The added Gaussian noise had mean equal to 0 and a standard deviation of 0.5. With this higher level of noise, the algorithm did not obtain a good estimate of the true mean of 1. The resulting posterior estimates for the means were -0.52 and -1.03, for the standard deviations were 0.93 and 0.45, and for the weights were 0.11 and 0.89. Clearly, since so few observations truly had the value +1, this part of the data was not detected due to the noise. The estimated value of β was 0.35 which was far lower than the true value. The p_D value was -15964.20 and the DIC was -34142.

Initialising the means to have the values +/-1 had no effect on results with the broad priors that we have been using until now. We incorporated our prior knowledge about the states present in the data by setting the initial means to be +/-1 and increasing the value of $\lambda^{(0)}$ (which is a hyperparameter of the prior distribution on the means) to 1000 to place more importance on the initial values of the means. This forced the algorithm to identify the component of the data which relates to the less numerous white pixels. This led to variational posterior means of -1.00 and 1.00, standard deviations of 0.49 and 0.50, and weights of 0.985 and 0.015. These estimates are close to the true values. The estimate of β was 0.6 which was equal to the true value. 99.6% of the pixels were labelled correctly. For this solution the value of p_D was -25302.08 and the DIC was -52953.

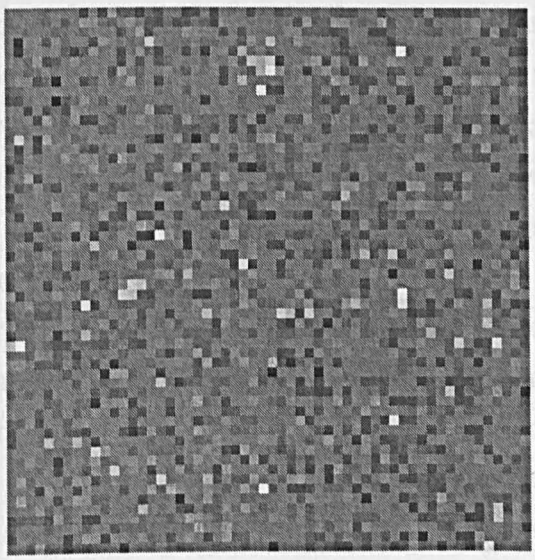

Figure 5.12: Noisy Image : Simulated Data Set 5

5.9 Other Interesting Applications of Hidden Markov Random Fields

HMRFs in Disease Mapping

The work by Green and Richardson (2002) provides an interesting example of the use of HMRFs in disease mapping. This application involves the investigation of whether or not there is any spatial link to disease risk, given data in the form of incident counts in a defined set of geographical regions. Green and Richardson (2002) provide a more flexible alternative to the models currently used in disease mapping. They build a hierarchical representation of the spatial heterogeneity of the rare count data by proposing a hidden discrete-state Markov random field model generated by an underlying finite-mixture model which allows spatial dependence. However, the main novelty of their approach is that the number of hidden states, or components of the mixture distribution, is not predefined and is estimated along with the model hyperparameters.

The main aim in studies of this nature is to make inference about the relative risk variable specific to each area in the study. Recent work in disease mapping commonly characterises the spatial dependence of the relative risk by parameters that are constant across the whole study region. Such global parameters can lead to oversmoothing and can prevent the detection of local discontinuities. Various ways of avoiding this problem have been investigated in the literature. Green and Richardson (2002) take the approach based on the idea of replacing the continuously-varying random field for the relative risk variable by a partition model having k different components and allocation variables indicating the component from which a particular observation comes. The resulting extra level of hierarchy increases model flexibility.

Green and Richardson (2002) use a Potts model for the allocation variables with an unknown number of states/components and unknown interaction strength. There is a prior distribution over the number of states and this, as well as the interaction strength, has to be estimated along with the hyperparameters. The allocation variables have a Markovian structure and, given the number of states, they follow a spatially correlated process. Since these are assumed to follow the Potts model, there is no explicit use of weights on components, as would be the case for mixture models.

The Potts spatial mixture is fitted using MCMC methods. Updating the number of states requires the use of a variable-dimension reversible jump move (Green (1995)). The method used in Green and Richardson (2002) follows that of Richardson and Green (1997) in that there is a random choice between merging two components into one component or splitting one component in two. However, one difference in the implementation is that here the reallocation of observations to the new set of components is not done independently for each observation but instead the reallocation of observations to the newly formed states is done while approximately respecting the spatial structure of the Potts model.

The normalising constant still presents a problem as it is required for updating the spatial interaction term and for the split and merge moves. Green and Richardson (2002) tackle this by using thermodynamic integration and then improving the resulting estimates via MCMC methods.

They also use the DIC to compare competing models fitted via this approach.

HMRFs in Machine Learning

There is active research into HMRF models within the machine learning community. For example, Murray and Ghahramani (2004) investigate various ways of overcoming the difficulty of normalising constant estimation in their paper on Bayesian learning in undirected graphical models. An undirected model is a model having a node for each variable and the edge connecting each pair of variables is undirected. The cliques are subgraphs which are fully connected. HMRFs come under this category.

The work in their paper is focused around the fully observed Boltzmann machine(BM) although the ideas should generalise to all undirected models. A BM is a Markov random field which defines a probability density function over a vector of binary observations. The energy function of the Boltzmann machine is similar to that of the Ising model.

Murray and Ghahramani (2004) consider the Metropolis Sampling Scheme and the Langevin Method. If one cannot compute the normalising constant then one cannot use the Metropolis Scheme since it is intrinsic to the method. Murray and Ghahramani explore ways of circumventing this problem such as approximating the normalising constant using a deterministic method leading to an approximate sampling method which will not converge to the true equilibrium distribution over the parameters. The authors remark that in the steps of the

Metropolis scheme this approximate value will be raised to the n^{th} power, which is perhaps problematic. Murray and Ghahramani also try to tackle the inefficiency of the Metropolis sampler for continuous spaces using the Langevin Method. In the Langevin Method, parameters are updated at every step, with no rejections, according to a rule which performs gradient descent. This too is only an approximation. They also suggest taking some of the existing learning algorithms and plugging them into stochastic-dynamics MCMC methods to perform Bayesian learning. Using these ideas, several approximate samplers are derived, in the case of a BM.

The first sampler is the Naive mean field approach. Using Jensen's inequality and a mean-field algorithm, they obtain a local maximum of the log of the normalising constant; this is then used in the Metropolis scheme leading to a mean-field Metropolis scheme.

The tree-structured variational approximation uses Jensen's inequality to obtain a tighter bound on the normalising constant than that provided by the naive mean-field method. Constraining to the set of all tree-structured distributions, a lower bound for the normalising constant can be found and then it can be used in the Metropolis algorithm to define the tree Metropolis algorithm.

Following the idea of Bethe approximation, which provides an approximation for the normalising constant, Murray and Ghahramani (2004) derive a loopy Metropolis algorithm. This entails running belief propagation on each proposal and then using Bethe free energy to approximate the acceptance probability.

A brief Langevin sampler is defined by using brief sampling to compute the expectations required for the Langevin method. The brief sampling results in low-variance but biased estimates of the required values, but the computational ease of this form of sampling makes it an attractive option. It is pointed out that one could perform the Langevin method using exact sampling by employing coupling from the past to obtain unbiased estimates of the required expectations. Variance could be lowered by re-using random numbers.

The authors found that the pseudo-likelihood approach was not suitable for the models used in their paper and so it was not investigated.

The paper also considers extension of these ideas to models with hidden variables. This is straightforward for the samplers based on approximations to the normalising constant or quantities related to it.

Murray and Ghahramani (2004) found that the mean-field and tree-based

variational Metropolis algorithms gave very poor results even for straightforward problems. The loopy Metropolis algorithm based on Bethe energy performed reasonably well on the artificial systems but performed poorly on their main application. The brief Langevin method performed reasonably well on larger systems where the others performed poorly.

Murray and Ghahramani (2004) suggest many different ways of overcoming the intractability issue for general undirected models using combinations of existing methods and they point out that there are many other possibilities to be explored.

5.10 Conclusions

In this chapter, we have reviewed the theory of HMRF's and discussed some of the important progress made in the area. We have also attempted to extend the variational approximate method and the DIC to this complex type of model. In doing so, it has been necessary to make some simplifying assumptions in order to make calculations possible. Three main approximations introduced for this purpose were that:

- we assumed a fully factorised form for $q_z(z)$
- we used the pseudo-likelihood in the formula for $q_\beta(\beta)$
- we approximate the denominator of the exponent of the expected value of the pseudo-likelihood (5.9).

Each of these approximations is a potential source of substantial difficulty. However, as there is no exact way to perform these calculations, approximation was necessary.

Despite these drastic assumptions, for our simulated data sets, we were often able to obtain good posterior estimates of model parameters using the variational method. When the noise added to the image was low, we could recover the original image with 100% accuracy. Naturally, as the noise level increases, the performance deteriorates. However, in some cases we can incorporate prior knowledge about the true distribution to obtain better estimates of the true parameters even when there is a lot of noise in our data.

We only considered application of our method to binary HMRF's and the next step for this problem is to consider data generated by a HMRF with more than 2 states. We did try initialising the algorithm with more than 2 states. In this case, we occasionally observed the state removal phenomenon that had occurred when we applied variational methods to finite mixture models and to hidden Markov models. However, for our simulated examples, starting the algorithm with extra states, even when they were subsequently removed during the iterations, led to unsatisfactory estimates of the model parameters. Although we observed the phenomenon of certain states being rejected by the method, as we did with the other types of model, the outcome for the HMRF case was much less useful. It might be worthwhile investigating how the results obtained by starting with more states than are truly present might be influenced by the choice of values for parameters in the prior distributions. It is worth noting that the cut-off value which determines the number of observations a state should have allocated to it before it is removed from the model will affect the final number of states in the model. We remove a state when less than 1 observation is allocated to it, but naturally if we increased this cut-off value (to 10 observations, for example) then in some cases this would result in a final fitted model with fewer components.

It has to be said that this investigation of the case of HMRFs can only be regarded as exploratory. The assumptions described above are really quite severe, especially the assumption of a factorised form for $q_z(z)$. As commented upon in Section 4.3, results based on this assumption for the hidden Markov model case were disappointing, so it can hardly be expected that the assumption will always be useful for HMRFs. However, it seemed hard to proceed without making some such assumption.

Chapter 6

Conclusions and Possible Areas for Future Research

We have reviewed the theory of finite mixture models and their extension to the hidden Markov model and hidden Markov random field structure. We have also described some of the difficulties associated with inference for such models and discussed some of the approaches that can be taken to overcome them. In particular, we have explored the variational approximation method for Bayesian inference and shown how it can be applied to the aforementioned models. In addition, we have extended Spiegelhalter et al.'s (2002) Deviance Information Criteria for model selection to each of these scenarios.

Applying the variational method, we have been able to obtain good posterior fits to simulated data sets for each of the models we considered. The method was also extremely time-efficient and, for these reasons, we consider it to be a viable alternative to MCMC methods and to have a great deal of potential for practical application.

In the case of finite mixtures of Gaussians, and HMM's with Gaussian observation densities, we observed the phenomenon of superfluous component/state removal, which occurs in the application of the variational method. In our simulated examples, we were able to show that, in many cases, this feature of the variational method leads to the recovery of the true number of components in the model. This feature may be viewed by some as a disadvantage, since it is unclear exactly why this occurs and it takes control of the fitted model complexity away from the user (see the discussion by Mackay (2001) for example). We take the

view that this is a very interesting feature of the method and deserves further exploration. Naturally, some theoretical understanding of this result and how it can be affected would be desirable.

We also used the DIC in conjunction with the variational technique in making a final decision on the most suitable model for each data set and, in many cases, the model selected by the DIC was the same as that selected using the variational approach.

We also observed the state removal phenomenon in the HMRF setting, but unfortunately, for our simulated examples, this did not lead to the automatic recovery of a close approximation of our true model as it had with the other types of model. Further investigation into the effect of including prior information would be worthwhile. We only considered binary HMRF's in our examples and clearly a next step would be to look at multiple-state examples, which, in the context of image analysis, might correspond to colour images. However, it seems likely that the approximations we have made so far are too crude, and more work is required to refine them.

As a next stage of progress, it would be interesting to extend the model hierarchy to include a temporal, as well as a spatial, dependence and investigate how variational methods might be applied to this case. Another consideration is models involving discrete as well as continuous variables.

In many practical application areas of statistics, researchers are interested in studying the effect of covariates in their models. It would be interesting to consider how these might be incorporated into the variational framework. Mixtures, for instance, are often applied to medical studies and it would seem reasonable to include a covariate term. In the HMRF setting, it would be useful to include such a term for application to areas such as disease mapping where there are covariates to be considered.

It would also seem feasible to consider combining variational methodology with MCMC schemes. This might have the effect of reducing the computational time involved in implementing MCMC methods, and of course it would be interesting to discover whether this would lead to the removal of components or states in the variational MCMC scheme.

In this thesis we have explored to some extent the potential for application of variational methods in statistics, but clearly there is still much scope for further investigation.

Appendix A

A.1 Reformulation of the DIC

Spiegelhalter et al.'s (2002) complexity measure, p_D, is based on a deviance,

$$D(\theta) = -2\log p(y|\theta) + 2\log f(y).$$

p_D is taken as the difference between the posterior mean of the deviance and the deviance at the posterior means of relevant parameters. $f(y)$ is a standardising term which is a function of the data alone and so it does not affect model comparison. p_D is a measure of the effective number of parameters in a model,

$$p_D = \overline{D(\theta)} - D(\tilde{\theta}).$$

To measure the fit of the model, the posterior mean deviance, $\overline{D(\theta)}$, is used. Then the deviance information criterion, or DIC, is formed by adding p_D and $\overline{D(\theta)}$:

$$\text{DIC} = \overline{D(\theta)} + p_D.$$

We can rewrite this as

$$\begin{aligned} \text{DIC} &= \overline{D(\theta)} + p_D \\ &= 2p_D + D(\tilde{\theta}) \\ &= 2p_D - 2\log p(y|\tilde{\theta}), \end{aligned}$$

since $f(y)$ can be assumed to be equal to 1.

Appendix B

B.1 'Monotonicity' of Variational Bayes

Define

$$\urcorner(q_\theta, q_z, y) = \int\int q_\theta(\theta) q_z(z) \log\left\{\frac{p(y,z,\theta)}{q_\theta(\theta) q_z(z)}\right\} d\theta dz$$

Suppose at stage 't' in the iteration, we have $q_\theta^{(t)}$, $q_z^{(t)}$.
Then

$$q_\theta^{(t+1)} = \text{argmax}_{q_\theta} \urcorner(q_\theta, q_z^{(t)}, y),$$

so that, in particular,

$$\urcorner(q_\theta{}^{(t+1)}, q_z^{(t)}, y) \geq \urcorner(q_\theta{}^{(t)}, q_z^{(t)}, y). \tag{B.1}$$

Next we obtain

$$q_z^{(t+1)} = \text{argmax}_{q_z} \urcorner(q_\theta^{(t+1)}, q_z, y),$$

so that, in particular,

$$\urcorner(q_\theta{}^{(t+1)}, q_z^{(t+1)}, y) \geq \urcorner(q_\theta{}^{(t+1)}, q_z^{(t)}, y). \tag{B.2}$$

Combining (B.1) and (B.2) we have

$$\urcorner(q_\theta{}^{(t+1)}, q_z^{(t+1)}, y) \geq \urcorner(q_\theta{}^{(t)}, q_z^{(t)}, y).$$

Note that monotonicity holds if $q_\theta{}^{(t+1)}$ and $q_z^{(t+1)}$ are any q_θ and q_z that achieve

(B.1) and (B.2).

Appendix C

In this appendix we describe how one can derive the form of the variational posterior for the model parameters and the missing indicator variables as well as approximate the DIC.

C.1 Finding the Hyperparameters of the Variational Posterior for a Mixture of Univariate Gaussian Distributions

The goal is to maximise the marginal likelihood (1.3) (which corresponds to minimising the Kullback-Leibler divergence). (1.3) is given by

$$\int \sum_{\{z\}} q(\theta, z) \log \frac{p(y, z, \theta)}{q(\theta, z)} d\theta$$

We assume that $q(\theta, z)$ factorises over the model parameters θ and the missing variables z so that $q(\theta, z) = q(\theta)q(z)$.

To obtain the form of the variational posterior for the univariate mixture of Gaussians, we begin by considering the form of $p(y, z, \theta)$. In the univariate case the joint p.d.f. is given by

$$p(y, z, \theta) \propto \prod_{j=1}^{K} {\rho_j}^{\alpha_j{}^{(o)} - 1 + \sum_{i=1}^{n} z_{ij}} \prod_{j=1}^{K} [\sqrt{\tau_j}^{(1 + \sum_{i=1}^{n} z_{ij})} {\tau_j}^{\frac{1}{2}\gamma_j{}^{(0)} - 1} \exp\{-\frac{\tau_j}{2} \sum_{i=1}^{n} z_{ij}(y_i - \mu_j)^2\}$$

$$\times \exp\{-\frac{\beta_j{}^{(0)}\tau_j}{2}(\mu_j - m_j^{(0)})^2 - \frac{1}{2}\delta_j{}^{(0)}\tau_j\}].$$

For the variational approximation, we take $q(\theta, z)$ to have the factorised form

$$q(\theta, z) = \{\prod_{i=1}^{n} q_{z_i}(z_i)\} q_\rho(\rho) \{\prod_{j=1}^{K} q_j(\mu_j, \tau_j)\},$$

over the parameters in the model and the missing variables.

Now the variational posteriors can be found by focusing on maximising the relevant parts of (1.3). We derive the forms for the model parameters ρ, μ and τ in this way. Similarly, we obtain the posterior for the missing variables z.

Finding the posterior distribution for ρ

We consider only the parts of the joint distribution that involve ρ. We have

$$\int \int q_\theta(\theta) q_z(z) \log\{\frac{p(y, \theta, \rho)}{q_\theta(\theta)}\} d\theta dz$$

$$= \int q(\mu, \tau) q_z(z) q_\rho(\rho) \log\{\frac{\prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)} + \sum_{i=1}^{n} z_{ij} - 1}}{q_\rho(\rho)}\} d\theta dz + \text{terms not involving } \rho$$

$$= \int q_\rho(\rho) q_z(z) \sum_{j=1}^{K} \{(\alpha_j{}^{(0)} + \sum_{i=1}^{n} z_{ij} - 1) \log \rho_j\} d\rho dz - \int q_\rho(\rho) \log(q_\rho(\rho)) d\rho$$

$$= \int q_\rho(\rho) \log[\frac{\prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{z_i}(z_i = j) - 1}}{q_\rho(\rho)}].$$

Thus,

$$\rho_j \sim Dir(\rho | \alpha_1, ..., \alpha_K)$$

where

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{ij},$$

in which q_{ij} denotes $q_{z_i}(z_i = j)$. Therefore,

$$q_\rho(\rho) \propto \prod_{j=1}^{K} \rho_j{}^{\alpha_j - 1}.$$

Finding the posterior for z_{ij}

$$\sum_z \int \{\prod_{i=1}^n q_{z_i}(z_i)\} q_\theta(\theta) \log \frac{p(\theta)\prod_{i=1}^n p(y_i, z_i|\theta)}{\{\prod_{i=1}^n q_{z_i}(z_i)\} q_\theta(\theta)} d\theta$$

$$= \sum_{z_i} \int q_{z_i}(z_i) q_\theta(\theta) \log \frac{p(y_i, z_i|\theta)}{q_{z_i}(z_i)} d\theta$$

$$= \sum_{z_i} q_{z_i}(z_i) \{\int q_\theta(\theta) \log p(y_i, z_i|\theta) d\theta - \log q_{z_i}(z_i)\} + \text{terms independent of } q_{z_i}$$

$$= \sum_j q_{z_i}(z_i = j) \{\int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta - \log q_{z_i}(z_i = j)\}$$

$$= \sum_j q_{z_i}(z_i = j) \log [\frac{\exp \int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta}{q_{z_i}(z_i = j)}].$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta\}$$

with

$$p(y_i, z_i = j|\theta) = \rho_j \sqrt{\frac{\tau_j}{2\pi}} e^{-\frac{\tau_j}{2}(y_i - \mu_j)^2}$$

$$\log p(y_i, z_i = j|\theta) = \frac{1}{2} \log |\tau_j| + \log \rho_j - \frac{1}{2} \tau_j (y_i - \mu_j)^2.$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q \log \rho_j + \frac{1}{2} \mathbf{E}_q \log |\tau_j| - \frac{1}{2} \mathbf{E}_q [\tau_j (y_i - \mu_j)^2]\},$$

where $\mathbf{E}_q$ denotes expectation with respect to q. Now,

$$\begin{aligned} \mathbf{E}_q[(y_i - \mu_j)^2|\tau_j] &= \mathbf{E}_q[(y_i - m_j + m_j - \mu_j)^2|\tau_j] \\ &= (y_i - m_j)^2 + \mathbf{E}_q[(m_j - \mu_j)^2|\tau_j] \\ &= (y_i - m_j)^2 + \frac{1}{\beta_j \tau_j} \end{aligned}$$

and so

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log|\tau_j|] - \frac{1}{2}\mathbf{E}_q[\tau_j((y_i - m_j)^2 + \frac{1}{\beta_j \tau_j})]\}$$

$$= \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log|\tau_j|] - \frac{1}{2}\mathbf{E}_q[\tau_j](y_i - m_j)^2 - \frac{1}{2\beta_j}\}$$

The expected values in the above are given by

$$\mathbf{E}_q[\log \rho_j] = \Psi(\alpha_j) - \Psi(\sum_{j\cdot} \alpha_{j\cdot})$$

$$\mathbf{E}_q[\log|\tau_j|] = \Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}$$

$$\mathbf{E}_q[\tau_j] = \frac{\gamma_j}{\delta_j}.$$

Finding the posterior for $\mu|\tau$ and τ.

We concentrate on the parts of the joint distribution that involve j:

$$\int q(\mu_j, \tau_j) \left[\sum_{i=1}^{n} q_{ij}\{\frac{1}{2}\log\tau_j - \frac{\tau_j}{2}(y_i - \mu_j)^2\} - \frac{1}{2}\beta_j{}^{(0)}\tau_j(\mu_j - m_j{}^{(0)})^2 \right.$$

$$\left. + \frac{\gamma_j{}^{(0)} - 1}{2}\log|\tau_j| - \frac{1}{2}\delta_j{}^{(0)}\tau_j\right]$$

$$= \int q(\mu_j, \tau_j)\log(\exp[\sum_{i=1}^{n} q_{ij}\{\frac{1}{2}\log\tau_j - \frac{\tau_j}{2}(y_i - \mu_j)^2\} - \frac{1}{2}\beta_j{}^{(0)}\tau_j(\mu_j - m_j{}^{(0)})^2$$

$$+\frac{\gamma_j{}^{(0)} - 1}{2}\log\tau_j - \frac{1}{2}\delta_j{}^{(0)}\tau_j])$$

$$= \int q(\mu_j, \tau_j)\log(\frac{1}{2}\tau_j - \exp(\frac{\tau_j}{2}(\mu_j - m_j)^2) \times \tau_j{}^{\frac{\gamma_j - 1}{2}} \exp(-\frac{1}{2}\delta_j\tau_j))$$

with the hyperparameters derived below.

i.e.

$$q(\mu_j|\tau_j) \sim N(m_j, \frac{1}{\beta_j \tau_j})$$

$$q(\tau_j) \sim Ga(\frac{1}{2}\gamma_j, \frac{1}{2}\delta_j).$$

Finding the hyperparameters

$$\sum_{i=1}^{n} q_{ij}\tau_j(y_i - \mu_j)^2 + \beta_j{}^{(0)}\tau_j(\mu_j - m_j{}^{(0)})^2$$

$$= \sum_{i=1}^{n} q_{ij}\tau_j(y_i{}^2 - 2y_i\mu_j + \mu_j{}^2) + \beta_j{}^{(0)}\tau_j(\mu_j{}^2 - 2\mu_j m_j{}^{(0)} + m_j{}^{(0)2})$$

$$= (\sum_{i=1}^{n} q_{ij}\tau_j + \beta_j{}^{(0)}\tau_j)\mu_j{}^2 + \sum_{i=1}^{n} q_{ij}\tau_j y_i{}^2 - 2\sum_{i=1}^{n} q_{ij}\tau_j y_i\mu_j - 2\beta_j{}^{(0)}\tau_j\mu_j m_j{}^{(0)} + \beta_j{}^{(0)}\tau_j m_j{}^{(0)2}$$

$$= \beta_j\tau_j\mu_j{}^2 + \sum_{i=1}^{n} q_{ij}\tau_j y_i{}^2 - 2\sum_{i=1}^{n} q_{ij}\tau_j y_i\mu_j - 2\beta_j{}^{(0)}\tau_j\mu_j m_j{}^{(0)} + \beta_j{}^{(0)}\tau_j m_j{}^{(0)2}$$

$$(\text{putting } \beta_j = \beta_j{}^{(0)} + \textstyle\sum_{i=1}^{n} q_{ij})$$

$$= \beta_j\tau_j\mu_j{}^2 + \sum_{i=1}^{n} q_{ij}\tau_j y_i{}^2 - 2\sum_{i=1}^{n} q_{ij}(\beta_j\tau_j)\beta_j{}^{-1} y_i\mu_j - 2\beta_j{}^{(0)}(\beta_j\tau_j)\beta_j{}^{-1}\mu_j m_j{}^{(0)} + \beta_j{}^{(0)}\tau_j m_j{}^{(0)2}$$

$$= \sum_{i=1}^{n} q_{ij}\tau_j y_i{}^2 + \beta_j\tau_j(\mu_j - m_j)^2 + \beta_j{}^{(0)}\tau_j m_j{}^{(0)2} - \beta_j\tau_j m_j{}^2,$$

where

$$\begin{aligned} m_j &= \frac{\beta_j{}^{(0)} m_j{}^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}} \\ &= \frac{\beta_j{}^{(0)} m_j{}^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j} \end{aligned}$$

$$= \beta_j \tau_j (\mu_j - m_j)^2 + \tau_j (\sum_{i=1}^{n} q_{ij} {y_i}^2 + {\beta_j}^{(0)} {m_j}^{(0)2} - \beta_j {m_j}^2).$$

Collecting terms then gives

$$\gamma_j = {\gamma_j}^{(0)} + \sum_{i=1}^{n} q_{ij}$$

$$\delta_j = {\delta_j}^{0)} + \sum_{i=1}^{n} q_{ij} {y_i}^2 + {\beta_j}^{(0)} {m_j}^{(0)2} - \beta_j {m_j}^2.$$

C.2 Derivation of the Formulae for p_D and DIC for a Mixture of Univariate Gaussian Distributions

We have

$$p_D \approx -2 \int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta + 2 \log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}.$$

For the first term, we have

$$\int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta$$

$$= \int q_\theta(\theta) \log[\prod_{j=1}^{K} {\rho_j}^{\alpha_j - {\alpha_j}^{(0)}} \prod_{j=1}^{K} \exp\{-\frac{\tau_j}{2}(\beta_j(\mu_j - m_j)^2 - {\beta_j}^{(0)}(\mu_j - {m_j}^{(0)})^2)\}$$

$$\times \prod_{j=1}^{K} {\tau_j}^{\frac{1}{2}(\gamma_j - {\gamma_j}^{(0)})} \exp\{-\frac{1}{2}\tau_j(\delta_j - {\delta_j}^{(0)})\}] d\theta + \text{constant}$$

$$= \int q_\theta(\theta) [\sum_{j=1}^{K} (\alpha_j - {\alpha_j}^{(0)}) \log \rho_j + \frac{1}{2} \sum_{j=1}^{K} (\gamma_j - {\gamma_j}^{(0)}) \log \tau_j$$

$$-\frac{1}{2} \sum_{j=1}^{K} \tau_j \{\beta_j(\mu_j - m_j)^2 - {\beta_j}^{(0)}(\mu_j - {m_j}^{(0)})^2 + \delta_j - {\delta_j}^{(0)}\}] d\theta + \text{constant}$$

$$= \sum_{j=1}^{K}(\alpha_j - \alpha_j^{(0)})\mathrm{E}_q[\log \rho_j] + \frac{1}{2}\sum_{j=1}^{K}(\gamma_j - \gamma_j^{(0)})\mathrm{E}_q[\log \tau_j]$$

$$-\frac{1}{2}\sum_{j=1}^{K}\mathrm{E}_{\mu_j,\tau_j}[\tau_j\{\beta_j(\mu_j - m_j)^2 - \beta_j^{(0)}(\mu_j - m_j^{(0)})^2 + \delta_j - \delta_j^{(0)}\}] + \text{constant}$$

$$= \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\mathrm{E}_q[\log \rho_j] + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\mathrm{E}_q[\log \tau_j]$$

$$-\frac{1}{2}\sum_{j=1}^{K}\mathrm{E}_{\tau_j}[\tau_j\{\frac{1}{\tau_j} - \frac{\beta_j^{(0)}}{\beta_j\tau_j} - \beta_j^{(0)}(m_j - m_j^{(0)})^2 + \delta_j - \delta_j^{(0)}\}] + \text{constant}$$

$$= \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\{\Psi(\alpha_j) - \Psi(\alpha_\cdot) + \frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j}\}$$

$$+\frac{1}{2}\sum_{j=1}^{K}(\frac{\gamma_j}{\delta_j})\{\beta_j^{(0)}(m_j - m_j^{(0)})^2 + \delta_j^{(0)} - \delta_j\} + \text{constant}.$$

For the second term in p_D we have

$$\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\} = \sum_{j=1}^{K}(\alpha_j - \alpha_j^{(0)})\log\tilde{\rho}_j + \frac{1}{2}\sum_{j=1}^{K}(\gamma_j - \gamma_j^{(0)})\log\tilde{\tau}_j$$

$$-\frac{1}{2}\sum_{j=1}^{K}\tilde{\tau}_j\{\beta_j(\tilde{\mu}_j - m_j)^2 - \beta_j^{(0)}(\tilde{\mu}_j - m_j^{(0)})^2 + \delta_j - \delta_j^{(0)}\} + \text{constant}$$

$$= \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\alpha_j}{\sum_{j=1}^{K}\alpha_j}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\gamma_j}{\delta_j})$$

$$+\frac{1}{2}\sum_{j=1}^{K}(\frac{\gamma_j}{\delta_j})\{\beta_j^{(0)}(m_j - m_j^{(0)})^2 + \delta_j^{(0)} - \delta_j\} + \text{constant}.$$

The constants in terms 1 and 2 are the same and so they subtract out when p_D is calculated. These terms give us

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$= - 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\{\Psi(\alpha_j) - \Psi(\alpha_{j\cdot}) + \frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j}\}]$$
$$+ 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\alpha_j}{\alpha_\cdot}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\gamma_j}{\delta_j})].$$

To find the DIC value we use

$$\mathrm{DIC} = 2p_D - 2\log p(y|\tilde{\theta})$$

where

$$\log p(y|\tilde{\theta}) = \sum_{i=1}^{n}\log[\sum_{j=1}^{K}\tilde{\rho}_j\sqrt{\frac{\tilde{\tau}_j}{2\pi}}\exp\{-\frac{\tilde{\tau}_j}{2}(y_i - \tilde{\mu}_j)^2\}],$$

where we use

$$\tilde{\rho}_j = \frac{\alpha_j}{\sum_{j=1}^{K}\alpha_j}$$
$$\tilde{\mu}_j = m_j$$
$$\tilde{\tau}_j = \frac{\gamma_j}{\delta_j}.$$

Appendix D

D.1 Finding the Hyperparameters of the Variational Posterior for a Mixture of Multivariate Gaussian Distributions

In the multivariate case the joint p.d.f of all of the variables is given by

$$p(y,z,\theta) \propto \prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(o)}+\sum_{i=1}^{n} z_{ij}-1} \prod_{j=1}^{K} |\beta_j{}^{(o)} T_j|^{\frac{1}{2}} \exp\{-\frac{1}{2}(\mu_j - m_j{}^{(o)})\beta_j{}^{(o)} T_j (\mu_j - m_j{}^{(o)})\}$$

$$\times \prod_{j=1}^{K} |T_j|^{\frac{\sum_{i=1}^{n} z_{ij}}{2}} \exp\{-\frac{1}{2}\sum_{i=1}^{n} z_{ij}(y_i-\mu_j)^T T_j (y_i-\mu_j)\}^{\sum_{i=1}^{n} z_{ij}}$$

$$\times \prod_{j=1}^{K} \frac{|T_j|^{\frac{v_j{}^{(0)}-d-1}{2}} \exp\{-\frac{1}{2} tr(\Sigma^{(0)} T_j)\}}{2^{\frac{v_j{}^{(0)} d}{2}} |\Sigma_j{}^{(0)}|^{-\frac{v_j{}^{(0)}}{2}} \prod_{s=1}^{d} \Gamma[\frac{1}{2}(v_j{}^{(0)}+1-s)]}.$$

For the variational approximation to $p(z,\theta,y)$ we take $q(\theta,z)$ to have the factorised form

$$q(\theta,z) = \{\prod_{i=1}^{n} q_{z_i}(z_i)\} q_\rho(\rho) \{\prod_{j=1}^{K} q_j(\mu_j,\tau_j)\}.$$

Now the variational posterior distributions can be found in the usual way.

Finding the posterior distribution for ρ

We consider only the parts of the joint distribution that involve ρ. We have

$$\int q_\theta(\theta) \log\{\frac{p(y,\theta,\rho)}{q_\theta(\theta)}\} d\theta$$

$$= \int q(\mu,T) q_z(z) q_\rho(\rho) \log\{\frac{\prod_{j=1}^K \rho_j{}^{\alpha_j{}^{(0)}+\sum_{i=1}^n z_{ij}-1}}{q_\rho(\rho)}\} + \text{terms not involving } \rho$$

$$= \int q_\rho(\rho) q_z(z) \sum_{j=1}^K \{(\alpha_j{}^{(0)} + \sum_{i=1}^n z_{ij} - 1) \log \rho_j\} d\rho dz - \int q_\rho(\rho) \log(q_\rho(\rho)) d\rho$$

$$= \int q_\rho(\rho) \log[\frac{\prod_{j=1}^K \rho_j{}^{\alpha_j{}^{(0)}+\sum_{i=1}^n q_{z_i}(z_i=j)-1}}{q_\rho(\rho)}].$$

Thus,

$$\rho_j \sim Dir(\rho|\alpha_1, ..., \alpha_K)$$

where

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^n q_{ij},$$

in which again q_{ij} denotes $q_{z_i}(z_i = j)$. Therefore,

$$q_\rho(\rho) \propto \prod_{j=1}^K \rho_j{}^{\alpha_j - 1}.$$

Finding the posterior for z_{ij}

$$\sum_z \int \{\prod_{i=1}^n q_{z_i}(z_i)\} q_\theta(\theta) \log \frac{p(\theta) \prod_{i=1}^n p(y_i, z_i|\theta)}{\{\prod_{i=1}^n q_{z_i}(z_i)\} q_\theta(\theta)} d\theta$$

$$= \sum_{z_i} \int q_{z_i}(z_i) q_\theta(\theta) \log \frac{p(y_i, z_i|\theta)}{q_{z_i}(z_i)} d\theta$$

$$= \sum_{z_i} q_{z_i}(z_i) \{\int q_\theta(\theta) \log p(y_i, z_i|\theta) d\theta - \log q_{z_i}(z_i)\} + \text{terms independent of } q_{z_i}$$

$$= \sum_j q_{z_i}(z_i = j)\{\int q_\theta(\theta) \log p(y_i, z_i = j) d\theta - \log q_{z_i}(z_i = j)\}$$

$$= \sum_j q_{z_i}(z_i = j) \log[\frac{\exp \int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta}{q_{z_i}(z_i = j)}].$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta\}$$

with

$$p(y_i, z_i = j|\theta) = \rho_j \frac{|T_j|^{\frac{1}{2}}}{2\pi^{\frac{d}{2}}} \exp\{-\frac{1}{2}(y_i - \mu_j)^T T_j (y_i - \mu_j)\}$$

$$\log p(y_i, z_i = j|\theta) = \frac{1}{2} \log |T_j| + \log \rho_j - \frac{1}{2}(y_i - \mu_j)^T T_j (y_i - \mu_j).$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q \log \rho_j + \frac{1}{2}\mathbf{E}_q \log |T_j| - \frac{1}{2}\mathbf{E}_q tr(T_j (y_i - \mu_j)(y_i - \mu_j)^T)\}$$

$$= \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log |T_j|]$$

$$-\frac{1}{2}\mathbf{E}_q tr(\mathbf{E}_q[T_j](y_i - m_j + m_j - \mu_j)(y_i - m_j + m_j - \mu_j)^T)\}$$

$$= \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log |T_j|] - \frac{1}{2} tr(\mathbf{E}_q[T_j](y_i - m_j)(y_i - m_j)^T$$

$$+\mathbf{E}_q[T_j]\mathbf{Cov}(\mu_j|T_j)) >\}$$

$$= \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log |T_j|] - \frac{1}{2} tr(\mathbf{E}_q[T_j](y_i - m_j)(y_i - m_j)^T$$

$$+\mathbf{E}_q[T_j \beta_j^{-1} T_j^{-1}])\}$$

$$= \exp\{\mathbf{E}_q[\log \rho_j] + \frac{1}{2}\mathbf{E}_q[\log|T_j|] - \frac{1}{2}tr(\mathbf{E}_q[T_j](y_i - m_j)(y_i - m_j)^T + \frac{1}{\beta_j}\mathbf{I}_d)\},$$

where $\mathbf{I}_d$ denotes the identity matrix of dimension d and

$$\mathbf{E}_q[\mu_j] = m_j$$

$$\mathbf{E}_q[T_j] = v_j \Sigma_j{}^{-1}$$

$$\mathbf{E}_q[\log|T_j|] = \sum_{s=1}^{d} \Psi(\frac{v_j + 1 - s}{2}) + d\log(2) - \log|\Sigma|$$

$$\mathbf{E}_q[\log(\rho_j)] = \Psi(\hat{\alpha}_j) - \Psi(\hat{\alpha}_\cdot).$$

Finding the posterior for $\mu|T$ and T.

For this we concentrate on the parts of the joint distribution that involve j. So we consider

$$\int q(\mu_j, T_j)[\sum_{i=1}^{n} q_{ij}\{\frac{1}{2}\log|T_j| - \frac{1}{2}(y_i - \mu_j)^T T_j(y_i - \mu_j)\} + \frac{1}{2}\log|\beta_j{}^{(0)}T_j|$$

$$-\frac{1}{2}(\mu_j - m_j{}^{(0)})^T \beta_j{}^{(0)} T_j(\mu_j - m_j{}^{(0)}) + \frac{v_j{}^{(0)} - d - 1}{2}\log|T_j| - \frac{1}{2}tr(\Sigma_j{}^{(0)}T_j)]$$

$$= \int q(\mu_j, T_j)\log(\exp[\sum_{i=1}^{n} q_{ij}\{\frac{1}{2}\log|T_j| - \frac{1}{2}(y_i - \mu_j)^T T_j(y_i - \mu_j)\} + \frac{1}{2}\log|\beta_j{}^{(0)}T_j|$$

$$-\frac{1}{2}(\mu_j - m_j{}^{(0)})^T \beta_j{}^{(0)} T_j(\mu_j - m_j{}^{(0)}) + \frac{v_j{}^{(0)} - d - 1}{2}\log|T_j| - \frac{1}{2}tr(\Sigma_j{}^{(0)}T_j)])$$

$$= \int q(\mu_j, T_j)\log[|T_j|^{\frac{1}{2}}\exp\{-\frac{1}{2}(\mu_j - m_j)^T \beta_j T_j(\mu_j - m_j)\}$$

$$\times|T_j|^{\frac{v_j - d - 1}{2}}\exp\{-\frac{1}{2}tr(\Sigma_j T_j)\}]$$

i.e.

$$q(\mu_j|T_j) \sim N_d(m_j, (\beta_j T_j)^{-1})$$

$$q(T_j) \sim W(v_j, \Sigma_j).$$

Finding the hyperparameters

$$\sum_{i=1}^{n} q_{ij}(y_i - \mu_j)^T T_j (y_i - \mu_j) + (\mu_j - m_j^{(0)})^T \beta_j^{(0)} T_j (\mu_j - m_j^{(0)})$$

$$= \sum_{i=1}^{n} q_{ij}(y_i^T T_j y_i - \mu_j^T T_j y_i - y_i^T T_j \mu_j + \mu_j^T T_j \mu_j) + (\mu_j^T \beta_j^{(0)} T_j \mu_j - m_j^{(0)T} \beta_j^{(0)} T_j \mu_j$$

$$-\mu_j^T \beta_j^{(0)} T_j m_j^{(0)} + m_j^{(0)T} \beta_j^{(0)} T_j m_j^{(0)})$$

$$= \mu_j^T \beta_j^{(0)} T_j \mu_j + \sum_{i=1}^{n} q_{ij} \mu_j^T T_j \mu_j + \sum_{i=1}^{n} q_{ij} y_i^T T_j y_i + m_j^{(0)T} \beta_j^{(0)} T_j m_j^{(0)}$$

$$- \sum_{i=1}^{n} q_{ij} \mu_j^T T_j y_i - \sum_{i=1}^{n} q_{ij} y_i^T T_j \mu_j - m_j^{(0)T} \beta_j^{(0)} T_j \mu_j - \mu_j^T \beta_j^{(0)} T_j m_j^{(0)}$$

$$= \mu_j^T \beta_j T_j \mu_j + \sum_{i=1}^{n} q_{ij} y_i^T T_j y_i + m_j^{(0)T} \beta_j^{(0)} T_j m_j^{(0)} - \sum_{i=1}^{n} q_{ij} \mu_j^T T_j y_i - \sum_{i=1}^{n} q_{ij} y_i^T T_j \mu_j$$

$$-m_j^{(0)T} \beta_j^{(0)} T_j \mu_j - \mu_j^T \beta_j^{(0)} T_j m_j^{(0)}$$

$$(\text{putting } \beta_j = \beta_j^{(0)} + \textstyle\sum_{i=1}^{n} q_{ij})$$

$$= tr(\sum_{i=1}^{n} q_{ij} y_i y_i^T T_j) + \mu_j^T \beta_j T_j \mu_j + m_j^{(0)T} \beta_j^{(0)} T_j m_j^{(0)} - \sum_{i=1}^{n} q_{ij} \mu_j^T (\beta_j T_j) \beta_j^{-1} y_i$$

$$-\beta_j^{-1} \sum_{i=1}^{n} q_{ij} y_i^T \beta_j T_j \mu_j - m_j^{(0)T} \beta_j^{(0)} (\beta_j T_j) \beta_j^{-1} \mu_j - \mu_j^T \beta_j^{(0)} (\beta_j T_j) \beta_j^{-1} m_j^{(0)}$$

$$= tr(\sum_{i=1}^{n} q_{ij} y_i {y_i}^T T_j) + (\mu_j - m_j)^T \beta_j T_j (\mu_j - m_j) + {m_j^{(0)}}^T \beta_j^{(0)} T_j m_j^{(0)} - {m_j}^T \beta_j T_j m_j$$

where

$$\begin{aligned} m_j &= \frac{\beta_j^{(0)} m_j^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j^{(0)} + \sum_{i=1}^{n} q_{ij}} \\ &= \frac{\beta_j^{(0)} m_j^{(0)} + \sum_{i=1}^{n} q_{ij} y_i}{\beta_j} \end{aligned}$$

$$= (\mu_j - m_j)^T \beta_j T_j (\mu_j - m_j) + tr(\{\sum_{i=1}^{n} q_{ij} y_i {y_i}^T + \beta_j^{(0)} m_j^{(0)} {m_j^{(0)}}^T - \beta_j m_j {m_j}^T\} T_j).$$

Collecting terms then gives

$$\Sigma_j = {\Sigma_j}^{(0)} + \sum_{i=1}^{n} q_{ij} y_i {y_i}^T + \beta_j^{(0)} m_j^{(0)} {m_j^{(0)}}^T - \beta_j m_j {m_j}^T$$

and

$$v_j = {v_j}^{(0)} + \sum_{i=1}^{n} q_{ij}.$$

D.2 Derivation of the Formulae for p_D and DIC for a Mixture of Multivariate Gaussian Distributions

We have

$$p_D \approx -2 \int q_\theta(\theta) \log\{\frac{q_\theta(\theta)}{p(\theta)}\} d\theta + 2 \log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}.$$

For the first term we have

$$\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta$$

$$= \int q_\theta(\theta)\log\left[\prod_{j=1}^{K}{\rho_j}^{\alpha_j-{\alpha_j}^{(0)}}\right.$$

$$\times\prod_{j=1}^{K}\exp\{-\frac{1}{2}((\mu_j-m_j)^T\beta_jT_j(\mu_j-m_j)-(\mu_j-{m_j}^{(0)})^T{\beta_j}^{(0)}T_j(\mu_j-{m_j}^{(0)}))\}$$

$$\left.\times\prod_{j=1}^{K}|T_j|^{\frac{1}{2}({v_j}-{v_j}^{(0)})}\exp[-\frac{1}{2}(\mathrm{tr}(\Sigma_jT_j)-\mathrm{tr}({\Sigma_j}^{(0)}T_j))\right]d\theta+\text{constant}$$

$$= \int q_\theta(\theta)\left[\sum_{j=1}^{K}(\alpha_j-{\alpha_j}^{(0)})\log\rho_j+\frac{1}{2}\sum_{j=1}^{K}({v_j}-{v_j}^{(0)})\log|T_j|\right.$$

$$-\frac{1}{2}\sum_{j=1}^{K}\Big\{(\mu_j-m_j)^T\beta_jT_j(\mu_j-m_j)-(\mu_j-{m_j}^{(0)})^T{\beta_j}^{(0)}T_j(\mu_j-{m_j}^{(0)})$$

$$\left.+\mathrm{tr}(\Sigma_jT_j)-\mathrm{tr}({\Sigma_j}^{(0)}T_j)\Big\}\right]d\theta+\text{constant}$$

$$=\sum_{j=1}^{K}(\alpha_j-{\alpha_j}^{(0)})\mathbf{E}_q[\log\rho_j]+\frac{1}{2}\sum_{j=1}^{K}({v_j}-{v_j}^{(0)})\mathbf{E}_q[\log|T_j|]$$

$$-\frac{1}{2}\sum_{j=1}^{K}\mathbf{E}_{\mu_j,T_j}\Big[(\mu_j-m_j)^T\beta_jT_j(\mu_j-m_j)-(\mu_j-{m_j}^{(0)})^T{\beta_j}^{(0)}T_j(\mu_j-{m_j}^{(0)})$$

$$+\mathrm{tr}(\{\Sigma_j-{\Sigma_j}^{(0)}\}T_j)\Big]+\text{constant}$$

$$=\sum_{j=1}^{K}(\sum_{i=1}^{n}q_{ij})\mathbf{E}_q[\log\rho_j]+\frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n}q_{ij})\log\mathbf{E}_q[|T_j|]$$

$$-\frac{1}{2}\sum_{j=1}^{K}\mathbf{E}_{T_j}\Big[\beta_jT_j(\beta_jT_j)^{-1}-{\beta_j}^{(0)}T_j(\beta_jT_j)^{-1}-(m_j-{m_j}^{(0)})^T{\beta_j}^{(0)}T_j(m_j-{m_j}^{(0)})$$

$$+\mathrm{tr}(\{\Sigma_j-{\Sigma_j}^{(0)}\}T_j)\Big]+\text{constant}$$

$$= \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})[\mathrm{E}_q[\log \rho_j] + \frac{1}{2}\log \mathrm{E}_q[|T_j|] - \frac{1}{2\beta_j}]$$

$$+\frac{1}{2}\sum_{j=1}^{K} \mathrm{tr}\{(\beta_j \mathbf{I}_d(m_j - m_j{}^{(0)})(m_j - m_j{}^{(0)})^T + \Sigma_j{}^{(0)} - \Sigma_j)\mathbf{E}_q[T_j]\} + \text{constant.}$$

For the second term in p_D we have

$$\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\} = \sum_{j=1}^{K}(\alpha_j - \alpha_j{}^{(0)})\log\tilde{\rho}_j + \frac{1}{2}\sum_{j=1}^{K}(v_j - v_j{}^{(0)})\log|\tilde{T}_j|$$

$$-\frac{1}{2}\sum_{j=1}^{K}\{(\tilde{\mu}_j - m_j)^T\beta_j\tilde{T}_j(\tilde{\mu}_j - m_j) - (\tilde{\mu}_j - m_j{}^{(0)})^T\beta_j{}^{(0)}\tilde{T}_j(\tilde{\mu}_j - m_j{}^{(0)})$$

$$+\mathrm{tr}(\Sigma_j\tilde{T}_j) - \mathrm{tr}(\Sigma_j{}^{(0)}\tilde{T}_j)\} + \text{constant}$$

$$= \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\alpha_j}{\sum_{j=1}^{K}\alpha_j}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log|v_j\Sigma_j{}^{-1}|$$

$$+\frac{1}{2}\sum_{j=1}^{K} \mathrm{tr}\{(\beta_j \mathbf{I}_d(m_j - m_j{}^{(0)})(m_j - m_j{}^{(0)})^T + \Sigma_j{}^{(0)} - \Sigma_j)\tilde{T}_j\} + \text{constant.}$$

The constants in terms 1 and 2 are the same and so they subtract out when p_D is calculated. These terms give us

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$\begin{aligned} = & - 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})[\mathrm{E}_q[\log \rho_j] + \frac{1}{2}\log \mathrm{E}_q[|T_j|] - \frac{1}{2\beta_j}]] \\ & + 2[\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\alpha_j}{\alpha_.}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log|v_j\Sigma_j{}^{-1}|]. \end{aligned}$$

To find the DIC value we use

$$\mathrm{DIC} = 2p_D - 2\log p(y|\tilde{\theta})$$

where

$$\log p(y|\tilde{\theta}) = \sum_{i=1}^{n} \log[\sum_{j=1}^{K} \tilde{\rho}_j \frac{|\tilde{T}_j|^{\frac{1}{2}}}{2\pi^{\frac{d}{2}}} \exp\{-\frac{1}{2}(y_i - \tilde{\mu}_j)^T \tilde{T}_j (y_i - \tilde{\mu}_j)\}].$$

Appendix E

E.1 Real Data Sets Used in Section 2.8.3

Galaxy Data

9.172 9.350 9.483 9.558 9.775 10.227 10.406 16.084 16.170 18.419
18.552 18.600 18.927 19.052 19.070 19.330 19.343 19.349 19.440
19.473 19.529 19.541 19.547 19.663 19.846 19.856 19.863 19.914
19.918 19.973 19.989 20.166 20.175 20.179 20.196 20.215 20.221
20.415 20.629 20.795 20.821 20.846 20.875 20.986 21.137 21.492
21.701 21.814 21.921 21.960 22.185 22.209 22.242 22.249 22.314
22.374 22.495 22.746 22.747 22.888 22.914 23.206 23.241 23.263
23.484 23.538 23.542 23.666 23.706 23.711 24.129 24.285 24.289
24.366 24.717 24.990 25.633 26.960 26.995 32.065 32.789 34.279

Enzyme Data

0.130 0.080 1.261 0.224 0.132 1.052 0.085 0.124 0.718 0.280 0.687
0.106 0.088 0.137 0.096 0.124 0.126 1.279 1.007 0.195 0.167 0.213
0.108 1.371 0.190 0.184 1.298 1.036 0.205 1.950 1.018 0.172 0.148
0.292 0.113 0.185 0.129 1.329 0.149 0.236 2.545 1.073 0.162 2.518
0.142 2.880 0.178 1.075 0.128 0.083 0.409 0.340 0.246 1.195 1.452
1.123 1.361 0.222 0.962 0.875 0.078 0.520 0.194 1.195 0.709 0.021

0.166 0.081 0.265 0.159 0.308 1.604 0.179 0.172 0.131 0.305 0.215
0.214 0.853 0.137 0.466 1.419 2.016 1.944 1.040 1.200 0.255 0.232
0.200 0.240 0.216 0.277 2.427 0.320 0.142 0.134 0.198 0.126 1.173
0.342 1.672 0.193 1.633 0.860 1.293 0.207 1.811 1.741 1.488 0.124
1.326 0.148 0.109 1.848 1.310 0.118 1.004 0.204 0.192 0.299 1.885
0.264 0.230 0.250 0.061 0.953 0.138 0.313 0.174 1.768 1.369 0.130
1.113 0.320 0.190 0.818 1.461 0.149 0.291 0.225 1.622 0.185 0.198
0.360 0.387 2.338 1.713 0.368 1.573 0.309 0.232 0.347 0.325 1.861
0.258 0.258 1.625 0.291 1.169 0.210 0.241 0.112 0.183 0.258 0.357
1.176 0.111 0.978 0.279 1.742 0.184 0.230 0.275 2.183 2.264 1.405
0.408 0.126 0.263 0.162 0.902 1.516 0.293 0.198 0.118 0.305 0.031
0.192 0.151 0.182 0.909 0.379 1.010 0.167 0.929 0.083 0.179 1.567
1.241 0.077 0.166 1.271 0.100 1.229 0.152 1.374 0.157 1.003 0.084
0.171 0.953 0.192 0.967 1.300 0.122 1.036 0.200 0.070 0.998 0.176
0.673 0.839 0.867 0.985 0.096 0.238 0.933 1.231 0.162 0.044 0.175
0.132 1.166 0.144 0.180 0.945 0.180 0.152 0.108 0.923 0.192 0.895
0.176 0.191 1.161

Acidity Data

2.928524 3.910021 3.732896 3.688879 3.822098 3.735286 4.143135
4.276666 3.931826 4.077537 4.779123 4.234107 4.276666 4.543295
6.467388 4.127134 3.977811 4.264087 4.007333 3.921973 5.384495
4.912655 4.046554 4.043051 4.406719 4.505350 3.931826 6.752270
6.928538 5.994460 4.248495 4.060443 4.727388 6.047372 4.082609
4.244200 4.890349 4.416428 5.743003 4.127134 5.489764 4.778283
5.249652 4.855929 4.128746 4.442651 4.025352 4.290459 4.593098
4.652054 4.178992 4.382027 5.569489 5.049856 4.188138 6.629363
4.647271 4.784989 4.348987 5.361292 4.574711 4.442651 6.120297
4.060443 4.143135 4.510860 6.049733 4.510860 4.406719 6.343880
4.430817 5.929589 5.973301 4.481872 4.301359 6.452680 4.204693
4.143135 6.603944 4.644391 5.863631 4.025352 5.717028 5.308268

6.267201 4.060443 5.017280 4.510860 5.834811 4.330733 4.007333
6.806829 5.257495 4.624973 4.781641 4.099332 7.044382 3.914021
4.330733 4.016383 5.572154 4.043051 4.843399 4.110874 4.454347
4.356709 6.154858 6.284321 6.978214 4.301359 5.929855 4.465908
6.035481 6.726473 7.105130 6.014937 4.882802 7.032095 4.518522
6.476665 6.125558 4.189655 5.323498 4.938065 6.313548 5.853925
6.278146 7.020191 5.023881 4.262680 6.725634 6.489205 5.743003
6.739337 6.466145 6.855409 5.120983 5.913773 6.516932 4.058717
6.213608 6.554218 6.155707 4.314818 6.662494 6.749931 6.100319
4.112512 6.946014 4.131961 6.234411 6.595781 6.683861 6.957973
4.497585

Appendix F

F.1 The Poisson Distribution

Finding the posterior for z_{ij}

$$q_{z_i}(z_i = j) \propto \exp\{\int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta\}$$

with

$$p(y_i, z_i = j|\theta) = \rho_j \exp\{-\phi_j\} {\phi_j}^{y_i},$$

and

$$\log p(y_i, z_i = j|\theta) = \log \rho_j - \phi_j + y_i \log(\phi_j).$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q \log \rho_j - \mathbf{E}_q \log \phi_j + y_i \mathbf{E}_q \log(\phi_j)\}.$$

Finding the posterior distribution for ρ

We consider only the parts of the joint distribution that involve ρ. We have

$$\sum_z \int q_\theta(\theta) q_z(z) \log\{\frac{p(y, z, \theta)}{q(\theta)}\} d\theta$$

$$= \sum_z \int q_\phi(\phi) q_z(z) q_\rho(\rho) \log\{\frac{\prod_{j=1}^K {\rho_j}^{\alpha_j{}^{(0)} + \sum_{i=1}^n z_{ij} - 1}}{q_\rho(\rho)}\} d\phi d\rho + \text{terms not involving } \rho$$

$$= \sum_z \int q_\rho(\rho) q_z(z) \sum_{j=1}^{K} \{(\alpha_j{}^{(0)} + \sum_{i=1}^{n} z_{ij} - 1) \log \rho_j\} d(\rho) d(z) - \int q_\rho(\rho) \log(q_\rho(\rho)) d(\rho)$$

$$= \int q_\rho(\rho) \log[\frac{\prod_{j=1}^{K} \rho_j{}^{\alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{z_i}(z_i = j) - 1}}{q_\rho(\rho)}].$$

Thus, the optimal $q_\rho(\rho)$ is

$$q_\rho(\rho) = \text{Dir}(\rho; \alpha_1, ..., \alpha_K)$$

where

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^{n} q_{ij}.$$

Thus,

$$q_\rho(\rho) \propto \prod_{j=1}^{K} \rho_j{}^{\alpha_j - 1}.$$

Finding the posterior distribution for ϕ

We consider the parts of the joint distribution which involve ϕ. We have

$$\sum_z \int q_\theta(\theta) \log\{\frac{p(y, z, \theta)}{q_\theta(\theta)}\} d\theta$$

$$\propto \sum_z \int q_\phi(\phi) q_z(z) \log\{\frac{\prod_{i=1}^{n} \prod_{j=1}^{K} \exp(-\phi_j z_{ij}) \phi_j{}^{y_i z_{ij}} \prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)} - 1} \exp(-\beta_j{}^{(0)} \phi_j)}{q_\phi(\phi)}\} d\phi$$

$$+ \quad \text{terms not involving } \phi$$

$$= \sum_z \int q_\phi(\phi) \log\{\frac{\prod_{j=1}^{K} \phi_j{}^{\gamma_j{}^{(0)} + \sum_{j=1}^{K} y_i z_{ij} - 1} \exp[-\phi_j(\beta_j{}^{(0)} + \sum_{j=1}^{K} z_{ij})]}{q_\phi(\phi)}\} d\phi$$

$$+ \quad \text{terms not involving } \phi$$

$$= \int q_\phi(\phi) \log\{\frac{\prod_{j=1}^{K} \phi_j^{\gamma_j - 1} \exp(-\phi_j \beta_j)}{q_\phi(\phi)}\} d\phi$$

$$+ \quad \text{terms not involving } \phi$$

say, putting

$$\gamma_j = \gamma_j^{(0)} + \sum_{j=1}^{K} y_i q_{ij}$$

and

$$\beta_j = \beta_j^{(0)} + \sum_{j=1}^{K} q_{ij}.$$

F.2 The Exponential Distribution

Finding the posterior for z_{ij}

As detailed above,

$$q_{z_i}(z_i = j) \propto \exp\{\int q_\theta(\theta) \log p(y_i, z_i = j|\theta) d\theta\}.$$

In the case of a mixture of Exponential distributions,

$$p(y_i, z_i = j|\theta) = \rho_j \phi_j \exp\{-\phi_j y_i\},$$

and

$$\log p(y_i, z_i = j|\theta) = \log \rho_j - \log \phi_j - \phi_j y_i.$$

Therefore

$$q_{z_i}(z_i = j) \propto \exp\{\mathbf{E}_q \log \rho_j - \mathbf{E}_q \log \phi_j + y_i \mathbf{E}_q \log(\phi_j)\}.$$

Finding the posterior distribution for ρ

We consider only the parts of the joint distribution that involve ρ. We have

$$\sum_z \int q_\theta(\theta) \log\{\frac{p(y, z, \theta)}{q_\theta(\theta)}\} d\theta$$

$$= \sum_z \int q_\phi(\phi) q_z(z) q_\rho(\rho) \log\{\frac{\prod_{j=1}^K \rho_j{}^{\alpha_j{}^{(0)} + \sum_{i=1}^n z_{ij} - 1}}{q_\rho(\rho)}\} d\phi + \text{terms not involving } \rho$$

$$= \sum_z \int q_\rho(\rho) q_z(z) \sum_{j=1}^K \{(\alpha_j{}^{(0)} + \sum_{i=1}^n z_{ij} - 1) \log \rho_j\} d(\rho) d(z) - \int q_\rho(\rho) \log(q_\rho(\rho))$$

$$= \int q_\rho(\rho) \log[\frac{\prod_{j=1}^K \rho_j{}^{\alpha_j{}^{(0)} + \sum_{i=1}^n q_{z_i}(z_i = j) - 1}}{q_\rho(\rho)}].$$

Thus, the optimal $q_\rho(\rho)$ is

$$q_\rho(\rho) = \text{Dir}(\rho; \alpha_1, ..., \alpha_K)$$

where

$$\alpha_j = \alpha_j{}^{(0)} + \sum_{i=1}^n q_{ij}$$

and letting q_{ij} denote $q_{z_i}(z_i = j)$. So

$$q(\rho) \propto \prod_{j=1}^K \rho_j{}^{\alpha_j - 1}.$$

Finding the posterior distribution for ϕ

We consider the parts of the joint distribution which involve ϕ. We have

$$\sum_z \int q_\theta(\theta) \log\{\frac{p(y, z, \theta)}{q_\theta(\theta)}\} d\theta$$

$$\propto \sum_z \int q_\phi(\phi) q_z(z) \log\{\frac{\prod_{i=1}^n \prod_{j=1}^K \phi_j{}^{z_{ij}} \exp(-\phi_j y_i z_{ij}) \prod_{j=1}^K \phi_j{}^{\gamma_j{}^{(0)} - 1} \exp(-\beta_j{}^{(0)} \phi_j)}{q_\phi(\phi)}\} d\phi dz$$

$$+ \quad \text{terms not involving } \phi$$

$$= \sum_z \int q_\phi(\phi) \log\{\frac{\prod_{j=1}^K \phi_j^{\gamma_j^{(0)}+\sum_{j=1}^K z_{ij}-1} \exp[-\phi_j(\beta_j^{(0)} + \sum_{j=1}^K y_i z_{ij})]}{q_\phi(\phi)}\} d\phi$$

$$+ \quad \text{terms not involving } \phi$$

$$= \int q_\phi(\phi) \log\{\frac{\prod_{j=1}^K \phi_j^{\gamma_j-1} \exp(-\phi_j \beta_j)}{q_\phi(\phi)}\} d\phi$$

$$+ \quad \text{terms not involving } \phi$$

say, putting

$$\gamma_j = \gamma_j^{(0)} + \sum_{j=1}^K q_{ij}$$

and

$$\beta_j = \beta_j^{(0)} + \sum_{j=1}^K y_i q_{ij}.$$

Appendix G

G.1 Finding the Form of the Variational Posterior for $q_z(z)$ in the case of a Hidden Markov Model with Gaussian Noise

We consider the parts of $\int\int q(z,\theta)\log\{\frac{p(y,z,\theta)}{q(z,\theta)}\}dzd\theta$ which involve $q_z(z)$:

$$\sum_{\{z\}}\int_\theta q_z(z)q_\theta(\theta)\log\frac{\prod_{i=1}^n\prod_{j=1}^K\{p_j(y_i|\phi_j)\}^{z_{ij}}\prod_{i=1}^{n-1}\prod_{j_1}\prod_{j_2}(\pi_{j_1j_2})^{z_{ij_1}z_{i+1j_2}}}{q_z(z)q_\theta(\theta)}d\theta$$

$$\begin{aligned}
&= \sum_{\{z\}}q_z(z)[\int q_\theta(\theta)\sum_{i=1}^{n-1}\sum_{j_1}\sum_{j_2}z_{ij_1}z_{i+1j_2}\log\pi_{j_1j_2}d\theta \\
&\quad + \int q_\theta(\theta)\sum_{i=1}^{n}\sum_{j=1}^{K}z_{ij}\log p_j(y_i|\phi_j)d\theta - \log q_z(z)] + \text{constant}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\{z\}}q_z(z)[\int q_\pi(\pi)\sum_{i=1}^{n-1}\sum_{j_1}\sum_{j_2}z_{ij_1}z_{i+1j_2}\log\pi_{j_1j_2}d\pi \\
&\quad + \sum_{i=1}^{n}\sum_{j=1}^{K}z_{ij}\int q_\phi(\phi_j)\log p_j(y_i|\phi_j)d\phi_j - \log q_z(z)] + \text{constant}
\end{aligned}$$

$$= \sum_{\{z\}} q_z(z)[\sum_{i=1}^{n-1}\sum_{j_1}\sum_{j_2} z_{ij_1} z_{i+1j_2} \mathbf{E}_q[\log \pi_{j_1 j_2}]$$
$$+ \sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij} \mathbf{E}_q[\log p_j(y_i|\phi_j)] - \log q_z(z)] + \text{constant}$$

$$= \sum_{\{z\}} q_z(z) \log \left(\frac{\prod_{i=1}^{n}\prod_{j=1}^{K} {b^*_{ij}}^{z_{ij}} \prod_{i=1}^{n-1}\prod_{j_1}\prod_{j_2} {a^*_{j_1 j_2}}^{z_{ij_1} z_{i+1j_2}}}{q_z(z)} \right) + \text{constant},$$

where

$$a^*{}_{j_1 j_2} = \exp\{\mathbf{E}_q[\log \pi_{j_1 j_2}]\} = \exp\{\Psi(\alpha_{j_1 j_2}) - \Psi(\alpha_{j_1 \cdot})\}$$

and

$$b^*_{ij} = \exp\{\mathbf{E}_q[\log p_j(y_i|\phi_j)]\}$$

in which

$$\mathbf{E}_q[\log p_j(y_i|\phi_j)] = \frac{1}{2}\Psi(\frac{1}{2}\gamma_j) - \frac{1}{2}\log\frac{\delta_j}{2} - \frac{1}{2}(\frac{\gamma_j}{\delta_j})(y_i - m_j)^2 - \frac{1}{2\beta_j}.$$

Thus, the optimal $q_z(z)$ is given by

$$q_z(z) \propto \prod_i \prod_j {b^*_{ij}}^{z_{ij}} \prod_i \prod_{j_1} \prod_{j_2} {a^*_{j_1 j_2}}^{z_{ij_1} z_{i+1j_2}}.$$

G.2 The Forward Backward Algorithm

The Forward Algorithm

Forward Algorithm : calculate the probability of being in state j at time i and the partial observation sequence up until time i given the model.
The forward variable is given by $\text{fvar}_i(j_1) = p(y_1, y_2, ..., y_i, z_i = j_1)$.

1. $\text{fvar}_1(j_1) = \pi_{j_1} p(y_1|z_1 = j_1)$ for j_1 such that $1 \leq j_1 \leq K$, and then

normalise such that $\sum_{j_1=1}^{K} \text{fvar}_1(j_1) = 1$, i.e. define

$$\widetilde{\text{fvar}}_1(j_1) = \frac{\text{fvar}_1(j_1)}{\sum_{j_1=1}^{K} \text{fvar}_1(j_1)}.$$

2. For $i = 1, ..., n-1$ and each j_2,

$$\text{fvar}^*_{i+1}(j_2) = \{\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2).$$

We then normalise once again, giving

$$\widetilde{\text{fvar}}_i(j_1) = \frac{\text{fvar}^*_i(j_1)}{\sum_{j_1=1}^{K} \text{fvar}^*_i(j_1)}.$$

3. We finally have

$$p(y_1, ..., y_n) = \sum_{j_1} \text{fvar}_n(j_1) = \frac{1}{c_n} \sum_{j_1=1}^{K} \widetilde{\text{fvar}}_n(j_1) = \frac{1}{c_n},$$

since $\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_n(j_1) = 1$ and where c_n is the normalising constant fvar is multiplied by at the n_{th} iteration.

We can calculate the n_{th} normalising constant, c_n, since one can obtain c_{i+1} from c_i in the following way;

$$\begin{aligned}
\widetilde{\text{fvar}}_{i+1}(j_2) &= \frac{\{\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2)}{\sum_{j_2=1}^{K} \{\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2)} \\
&= \frac{c_i \{\sum_{j_1=1}^{K} \text{fvar}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2)}{\sum_{j_2=1}^{K} \{\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2)} \\
&= \frac{c_i}{d_i} \text{fvar}_{i+1}(j_2)
\end{aligned}$$

where,

$$d_i = \sum_{j_2=1}^{K} \{\sum_{j_1=1}^{K} \widetilde{\text{fvar}}_i(j_1) p(z_{i+1} = j_2 | z_i = j_1)\} p(y_{i+1} | z_{i+1} = j_2).$$

Thus,

$$c_{i+1} = \frac{c_i}{d_i},$$

and c_n can be found recursively in this way. It may be more convenient to deal with the logarithms of these normalising constants, c_i, in which case we would have

$$\log c_{i+1} = \log c_i - \log d_i.$$

The Backward Algorithm

Backward Algorithm : works back from the last time, n.
The backward variable is given by $\text{bvar}_i(j_1) = p(y_{i+1}, y_{i+2}, ..., y_n | z_i = j_1)$, i.e. the probability of generating the last $n - i$ observations given state j at time i.

1. $\text{bvar}_n(j_1) = 1$, for all j_1, and we normalise such that $\sum_{j_1=1}^{K} \text{bvar}_n(j_1) = 1$, i.e.
$$\widetilde{\text{bvar}}_n(j_1) = \frac{\text{bvar}_n(j_1)}{\sum_{j_1=1}^{K} \text{bvar}_n(j_1)}.$$

2. For $i = n-1, n-2, ..., 1$,
$$\text{bvar}_i(j_1) = \sum_{j_2} p(z_{i+1} = j_2 | z_i = j_1) \text{bvar}_{i+1}(j_2) p(y_{i+1} | z_{i+1} = j_2).$$

We normalise again, giving

$$\widetilde{\text{bvar}}_i(j_1) = \frac{\text{bvar}_i(j_1)}{\sum_{j_1=1}^{K} \text{bvar}_i(j_1)}.$$

In the above algorithms, for $p(z_{i+1} = j_2 | z_i = j_1)$ we use the quantity $a^*_{j_1 j_2}$ and for $p(y_{i+1} | z_{i+1} = j_2)$ we use the quantity $b^*_{i+1 j_2}$.

G.3 Obtaining Formulae for p_D and DIC in the case of a Hidden Markov Model with Gaussian Noise

We have

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\},$$

where $p(\theta)$ is given by

$$\begin{aligned} p(\theta) &= p(\phi)p(\pi) \\ &= p(\mu|\tau)p(\tau)p(\pi). \end{aligned}$$

For the first term, we have

$$\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta$$

$$\begin{aligned} = \int q_\theta(\theta)\log\Bigg[&\prod_{j_1}\prod_{j_2}\pi_{j_1j_2}{}^{\alpha_{j_1j_2}-\alpha_{j_1j_2}{}^{(0)}} \\ &\times\prod_{j=1}^{K}\exp\{-\frac{\tau_j}{2}(\beta_j(\mu_j-m_j)^2-\beta_j{}^{(0)}(\mu_j-m_j{}^{(0)})^2)\} \\ &\times\prod_{j=1}^{K}\tau_j{}^{\frac{1}{2}(\gamma_j-\gamma_j{}^{(0)})}\exp[-\frac{1}{2}\tau_j(\delta_j-\delta_j{}^{(0)})]\Bigg]\,d\theta \end{aligned}$$

$$= \int q_\theta(\theta) \left[\sum_{j_1} \sum_{j_2} (\alpha_{j_1 j_2} - \alpha_{j_1 j_2}{}^{(0)}) \log \pi_{j_1 j_2} + \frac{1}{2} \sum_{j=1}^{K} (\gamma_j - \gamma_j{}^{(0)}) \log \tau_j \right.$$
$$\left. -\frac{1}{2} \sum_{j=1}^{K} \tau_j \{ \beta_j (\mu_j - m_j)^2 - \beta_j{}^{(0)} (\mu_j - m_j{}^{(0)})^2 + \delta_j - \delta_j{}^{(0)} \} \right] d\theta$$

$$= \sum_{j_1} \sum_{j_2} (\alpha_{j_1 j_2} - \alpha_{j_1 j_2}{}^{(0)}) \mathbf{E}_q[\log \pi_{j_1 j_2}] + \frac{1}{2} \sum_{j=1}^{K} (\gamma_j - \gamma_j{}^{(0)}) \mathbf{E}_q[\log \tau_j]$$
$$-\frac{1}{2} \sum_{j=1}^{K} \mathbf{E}_{\mu_j, \tau_j} [\tau_j \{ \beta_j (\mu_j - m_j)^2 - \beta_j{}^{(0)} (\mu_j - m_j{}^{(0)})^2 + \delta_j - \delta_j{}^{(0)} \}]$$

$$= \sum_{j_1} \sum_{j_2} \{ \sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2) \} \mathbf{E}_q[\log \pi_{j_1 j_2}] + \frac{1}{2} \sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \mathbf{E}_q[\log \tau_j]$$
$$-\frac{1}{2} \sum_{j=1}^{K} \mathbf{E}_{\tau_j} [\tau_j \{ \frac{1}{\tau_j} - \frac{\beta_j{}^{(0)}}{\beta_j \tau_j} - \beta_j{}^{(0)} (m_j - m_j{}^{(0)})^2 + \delta_j - \delta_j{}^{(0)} \}]$$

$$= \sum_{j_1} \sum_{j_2} \{ \sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2) \} \{ \Psi(\alpha_{j_1, j_2}) - \Psi(\alpha_{j_1 \cdot}) \}$$
$$+ \sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \{ \frac{1}{2} \{ \Psi(\frac{1}{2} \gamma_j) - \log \frac{\delta_j}{2} \} - \frac{1}{2 \beta_j} \}$$
$$+ \frac{1}{2} \sum_{j=1}^{K} (\frac{\gamma_j}{\delta_j}) \{ \beta_j{}^{(0)} (m_j - m_j{}^{(0)})^2 + \delta_j{}^{(0)} - \delta_j \} + \text{constant.}$$

The second term in the approximate p_D is given by

$$\log \{ \frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})} \} = \sum_{j_1} \sum_{j_2} (\alpha_{j_1 j_2} - \alpha_{j_1 j_2}{}^{(0)}) \log \tilde{\pi}_{j_1 j_2} + \frac{1}{2} \sum_{j=1}^{K} (\gamma_j - \gamma_j{}^{(0)}) \log \tilde{\tau}_j$$
$$-\frac{1}{2} \sum_{j=1}^{K} \tilde{\tau}_j \{ \beta_j (\tilde{\mu}_j - m_j)^2 - \beta_j{}^{(0)} (\tilde{\mu}_j - m_j{}^{(0)})^2 + \delta_j - \delta_j{}^{(0)} \} + \text{constant}$$

$$= \sum_{j_1} \sum_{j_2} \{ \sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2) \} \log (\frac{\alpha_{j_1 j_2}}{\sum_{j_2} \alpha_{j_1 j_2}}) + \frac{1}{2} \sum_{j=1}^{K} (\sum_{i=1}^{n} q_{ij}) \log (\frac{\gamma_j}{\delta_j})$$

$$+\frac{1}{2}\sum_{j=1}^{K}(\frac{\gamma_j}{\delta_j})\{\beta_j^{(0)}(m_j - m_j^{(0)})^2 + \delta_j^{(0)} - \delta_j\} + \text{constant}.$$

The constants in the two terms in the approximate p_D are the same, and so they subtract out when p_D is calculated, giving

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}$$

$$\begin{aligned} = \; - \; 2 \Bigg[& \sum_{j_1}\sum_{j_2}\{\sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2)\}\{\Psi(\alpha_{j_1,j_2}) - \Psi(\alpha_{j_1\cdot})\} \\ & + \sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\left\{\frac{1}{2}\{\Psi(\frac{1}{2}\gamma_j) - \log\frac{\delta_j}{2}\} - \frac{1}{2\beta_j}\right\}\Bigg] \\ + \; 2 \Bigg[& \sum_{j_1}\sum_{j_2}\{\sum_{i=1}^{n-1} q_z(z_i = j_1, z_{i+1} = j_2)\}\log(\frac{\alpha_{j_1 j_2}}{\sum_{j_2}\alpha_{j_1 j_2}}) + \frac{1}{2}\sum_{j=1}^{K}(\sum_{i=1}^{n} q_{ij})\log(\frac{\gamma_j}{\delta_j})\Bigg]. \end{aligned}$$

To find the DIC value we use

$$\text{DIC} = 2p_D - 2\log p(y|\tilde{\theta}),$$

in which $p(y|\tilde{\theta})$ can be found using the forward algorithm:

$$p(y|\tilde{\theta}) = \sum_{j=1}^{K} \text{fvar}_n(j).$$

Appendix H

H.1 Finding the Forms of the Hyperparameters for the Gaussian Noise Model of a Hidden Markov Random Field

We have

$$
\begin{aligned}
q_l(\phi_l) \;\propto\; & \prod_{i=1}^{n} \{p(y_i|\phi_l)^{q_{il}}\}\, p(\phi_l) \\
= \; & \prod_{i=1}^{n} \tau_l^{\frac{q_{il}}{2}} \exp\{-\frac{1}{2}\tau_l(y_i-\mu_l)^2\}^{q_{il}} (\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{-\frac{1}{2}\lambda_l^{(0)}\tau_l(\mu_l - m_l^{(0)})^2\} \\
& \qquad \times \tau_l^{\frac{1}{2}\gamma_l^{(0)}-1} \exp\{-\frac{1}{2}\xi_l^{0)}\tau_l\} \\
= \; & \tau_l^{\frac{\sum_{i=1}^{n} q_{il}}{2}} \exp\{-\frac{1}{2}\tau_l \sum_{i=1}^{n} q_{il}(y_i-\mu_l)^2\} (\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{-\frac{1}{2}\lambda_l^{(0)}\tau_l(\mu_l - m_l^{(0)})^2\} \\
& \qquad \times \tau_l^{\frac{1}{2}\gamma_l^{(0)}-1} \exp\{-\frac{1}{2}\xi_l^{0)}\tau_l\} \\
= \; & \tau_l^{\frac{1}{2}(\gamma_l^{(0)}+\sum_{i=1}^{n} q_{il})-1} \exp\{-\frac{1}{2}(\sum_{i=1}^{n} q_{il}\tau_l(y_i-\mu_l)^2 + \lambda_l^{(0)}\tau_l(\mu_l - m_l^{(0)})^2)\} \\
& \qquad \times (\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{-\frac{1}{2}\xi_l^{0)}\tau_l\} \\
= \; & \tau_l^{\frac{1}{2}\gamma_l-1} \exp\{-\frac{1}{2}(\sum_{i=1}^{n} q_{il}\tau_l(y_i-\mu_l)^2 + \lambda_l^{(0)}\tau_l(\mu_l - m_l^{(0)})^2)\} \\
& \qquad \times (\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{-\frac{1}{2}\xi_l^{0)}\tau_l\}
\end{aligned}
$$

if we put $\gamma_l = \gamma_l^{(0)} + \sum_{i=1}^n q_{il}$. As derived previously,

$$\sum_{i=1}^n q_{il}\tau_l(y_i - \mu_l)^2 + \lambda_l^{(0)}\tau_l(\mu_l - m_l^{(0)})^2$$

can be written as

$$\lambda_l\tau_l(\mu_l - m_l)^2 + \tau_l(\sum_{i=1}^n q_{il}y_i^2 + \lambda_l^{(0)}m_l^{(0)2} - \lambda_l m_l^2)$$

by putting

$$\lambda_l = \lambda_l^{(0)} + \sum_{i=1}^n q_{il}$$

$$m_l = \frac{\lambda_l^{(0)}m_l^{(0)} + \sum_{i=1}^n q_{il}y_i}{\lambda_l}.$$

We therefore have

$$(\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{\frac{1}{2}\lambda_l\tau_l(\mu_l - m_l)^2\} \times \tau_l^{\frac{1}{2}\gamma_l - 1} \exp\{\frac{1}{2}(\xi_l^{(0)}\tau_l + \tau_l(\sum_{i=1}^n q_{il}y_i^2 + \lambda_l^{(0)}m_l^{(0)2} - \lambda_l m_l^2))\}.$$

Putting

$$\xi_l = \xi_l^{(0)} + \sum_{i=1}^n q_{il}y_i^2 + \lambda_l^{(0)}m_l^{(0)2} - \lambda_l m_l^2$$

gives

$$(\lambda_l^{(0)}\tau_l)^{\frac{1}{2}} \exp\{\frac{1}{2}\lambda_l\tau_l(\mu_l - m_l)^2\} \times \tau_l^{\frac{1}{2}\gamma_l - 1} \exp\{\frac{1}{2}(\xi_l\tau_l)\}$$

$$\propto N(\mu_l; m_l, (\lambda_l\tau_l)^{-1}) \times Ga(\tau_l; \frac{1}{2}\lambda_l, \frac{1}{2}\xi_l),$$

and so

$$q(\mu_l|\tau_l) \sim N(\mu_l; m_l, (\lambda_l\tau_l)^{-1})$$

$$q(\tau_l) \sim Ga(\tau_l; \frac{1}{2}\lambda_l, \frac{1}{2}\xi_l).$$

The formula for updating q_{il} is of the form

$$q_{il} \propto \exp\left\{ \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) + 2\mathbf{E}_\beta(\beta) \sum_{j \epsilon \delta_i} q_{jl} \right\}, \quad l = 1, ..., K \qquad \text{(H.1)}$$

normalised so that $\sum_{l=1}^{K} q_{il} = 1$. In the above,

$$\begin{aligned} \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) &\propto \mathbf{E}_{\phi_l}[\log(\tau_l^{\frac{1}{2}} \exp\{-\frac{1}{2}\tau_l(y_i - \mu_l)^2\})] \\ &= \frac{1}{2}\mathbf{E}_q[\log|\tau_l|] - \frac{1}{2}\mathbf{E}_q[\tau_l(y_i - \mu_l)^2], \end{aligned}$$

with expectations given by

$$\mathbf{E}_q[\log|\tau_l|] = \Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2}),$$

$$\mathbf{E}_q[\tau_l] = \frac{\gamma_l}{\xi_l},$$

$$\begin{aligned} \mathbf{E}(y_i - \mu_l)^2|\tau_l &= \mathbf{E}(y_i - m_l + m_l - \mu_l)^2|\tau_l \\ &= (y_i - m_l)^2 + \mathbf{E}(m_l - \mu_l)^2|\tau_l \\ &= (y_i - m_l)^2 + \frac{1}{\xi_l \tau_l}, \end{aligned}$$

and so

$$\begin{aligned} \mathbf{E}_{\phi_l} \log p(y_i|\phi_l) &= \frac{1}{2}\mathbf{E}_q[\log|\tau_l|] - \frac{1}{2}\mathbf{E}_q[\tau_l((y_i - m_l)^2 + \frac{1}{\xi_l\tau_l})] \\ &= \frac{1}{2}\mathbf{E}_q[\log|\tau_l|] - \frac{1}{2}\mathbf{E}_q[\tau_l](y_i - m_l)^2 - \frac{1}{2\xi_l} \\ &= \frac{1}{2}(\Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2})) - \frac{1}{2}(\frac{\gamma_l}{\xi_l})(y_i - m_l)^2 - \frac{1}{2\xi_l}. \end{aligned}$$

H.2 Derivation of Formulae for p_D and DIC for a Hidden Markov Random Field with Gaussian Noise

In this case

$$p(\theta) = p(\phi)p(\beta) = p(\mu|\tau)p(\tau)p(\beta)$$

and so

$$q(\theta) = q(\mu|\tau)q(\tau)q(\beta) = \prod_{l=1}^{K}\{q(\mu_l|\tau_l)q(\tau_l)\}q(\beta).$$

The formula for p_D is given by

$$p_D \approx -2\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta + 2\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\}.$$

The first term is

$$\begin{aligned}
&\int q_\theta(\theta)\log\{\frac{q_\theta(\theta)}{p(\theta)}\}d\theta z \\
&= \int q_\theta(\theta)\log[\prod_{l=1}^{K}\exp\{-\frac{1}{2}\tau_l[\lambda_l(\mu_l-m_l)^2 - \lambda_l^{(0)}(\mu_l-m_l^{(0)})^2]\} \\
&\quad \times\prod_{l=1}^{K}\tau_l^{\frac{1}{2}(\gamma_l-\gamma_l^{(0)})}\exp[-\frac{1}{2}\tau_l(\xi_l-\xi_l^{(0)})] \\
&\quad \times\prod_{i=1}^{n}\frac{\exp\{2\beta\sum_{j\epsilon\delta_i}\sum_{l=1}^{K}q_{il}q_{jl}\}}{\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}}]d\theta + \text{constant} \\
&= \int q_\theta(\theta)[-\frac{1}{2}\sum_{l=1}^{K}\tau_l\{\lambda_l(\mu_l-m_l)^2 - \lambda_l^{(0)}(\mu_l-m_l^{(0)})^2 + \xi_l - \xi_l^{(0)}\} \\
&\quad +\frac{1}{2}\sum_{l=1}^{K}(\gamma_l-\gamma_l^{(0)})\log\tau_l \\
&\quad +2\beta\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K}q_{il}q_{jl}\} - \sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}]d\theta + \text{constant}
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2}\sum_{l=1}^{K}\mathbf{E}_{\mu_l,\tau_l}[\tau_l\{\lambda_l(\mu_l-m_l)^2-\lambda_l^{(0)}(\mu_l-m_l^{(0)})^2+\xi_l-\xi_l^{(0)}\}] \\
&\quad +\frac{1}{2}\sum_{l=1}^{K}\sum_{i=1}^{n}q_{il}\mathbf{E}_q[\log\tau_l]+2\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K}q_{il}q_{jl}\} \\
&\quad -\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}\right)+\text{constant} \\
&= -\frac{1}{2}\sum_{l=1}^{K}\mathbf{E}_{\tau_l}[\tau_l\{\frac{1}{\tau_l}-\frac{\lambda_l^{(0)}}{\lambda_l\tau_l}-\lambda_l^{(0)}(m_l-m_l^{(0)})^2+\xi_l-\xi_l^{(0)}\}] \\
&\quad +\frac{1}{2}\sum_{l=1}^{K}\sum_{i=1}^{n}q_{il}\mathbf{E}_q[\log\tau_l]+2\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K}q_{il}q_{jl}\} \\
&\quad -\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}\right)+\text{constant} \\
&= \frac{1}{2}\sum_{l=1}^{K}\mathbf{E}_q[\tau_l]\{\lambda_l^{(0)}(m_l-m_l^{(0)})^2+\xi_l^{(0)}-\xi_l\} \\
&\quad +\frac{1}{2}\sum_{l=1}^{K}\sum_{i=1}^{n}q_{il}\{\mathbf{E}_q[\log\tau_l]-\frac{1}{\lambda_l}\}+2\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K}q_{il}q_{jl}\} \\
&\quad -\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}\right)+\text{constant},
\end{aligned}
$$

where

$$\mathbf{E}_q[\log|\tau_l|]=\Psi(\frac{\gamma_l}{2})-\log(\frac{\xi_l}{2})$$

$$\mathbf{E}_q[\tau_l]=\frac{\gamma_l}{\xi_l}.$$

As explained in Section 5.5.3, $\mathbf{E}_q[\beta]$ is estimated as the mode of the function $\log Q(\beta)$. We will require another approximation for

$$\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i}q_{jl})\}\right).$$

We derive an approximation below.

The second term in p_D is given by

$$
\begin{aligned}
\log\{\frac{q_\theta(\tilde{\theta})}{p(\tilde{\theta})}\} &= -\frac{1}{2}\sum_{l=1}^{K}\tilde{\tau}[\lambda_l(\tilde{\mu}_l - m_l)^2 - \lambda_l{}^{(0)}(\tilde{\mu}_l - m_l^{(0)}) + \xi_l - \xi_l^{(0)}] \\
&\quad + \sum_{l=1}^{K}(\gamma_l - \gamma_l^{(0)})\log\tilde{\tau}_l + 2\tilde{\beta}\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} \\
&\quad - \sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\} + \text{constant} \\
&= -\frac{1}{2}\sum_{l=1}^{K}\tilde{\tau}[\lambda_l(\tilde{\mu}_l - m_l)^2 - \lambda_l{}^{(0)}(\tilde{\mu}_l - m_l^{(0)}) + \xi_l - \xi_l^{(0)}] \\
&\quad + \sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\log\tilde{\tau}_l + 2\tilde{\beta}\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} \\
&\quad - \sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\} + \text{constant},
\end{aligned}
$$

where

$$\tilde{\tau}_l = \frac{\gamma_l}{\xi_l}$$

and

$$\tilde{\mu}_l = m_l.$$

We take $\tilde{\beta}$ to be the expected value of β as obtained via the least squares approach. We therefore have

$$
\begin{aligned}
p_D &= -\sum_{l=1}^{K} \mathbf{E}_q[\tau_l]\{\lambda_l^{(0)}(m_l - m_l^{(0)})^2 + \xi_l^{(0)} - \xi_l\} - \sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\{\mathbf{E}_q[\log\tau_l] - \frac{1}{\lambda_l}\} \\
&\quad -4\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} + 2\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} q_{jl})\}\right) \\
&\quad -\sum_{l=1}^{K}\tilde{\tau}[\lambda_l(\tilde{\mu}_l - m_l)^2 - \lambda_l^{(0)}(\tilde{\mu}_l - m_l^{(0)}) + \xi_l - \xi_l^{(0)}] + 2\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\log\tilde{\tau}_l \\
&\quad +4\tilde{\beta}\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} - 2\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\} \\
&= -\sum_{l=1}^{K}(\frac{\gamma_l}{\xi_l})\{\lambda_l^{(0)}(m_l - m_l^{(0)})^2 + \xi_l^{(0)} - \xi_l\} - \sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\{\Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2}) - \frac{1}{\lambda_l}\} \\
&\quad -4\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} + 2\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} q_{jl})\}\right) \\
&\quad +\sum_{l=1}^{K}(\frac{\gamma_l}{\xi_l})[\lambda_l^{(0)}(m_l - m_l^{(0)}) + \xi_l^{(0)} - \xi_l] + 2\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\log(\frac{\gamma_l}{\xi_l}) \\
&\quad +4\tilde{\beta}\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} - 2\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\} \\
&= -\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\{\Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2}) - \frac{1}{\lambda_l}\} \\
&\quad -4\mathbf{E}_q[\beta]\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} + 2\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} q_{jl})\}\right) \\
&\quad +2\sum_{l=1}^{K}\sum_{i=1}^{n} q_{il}\log(\frac{\gamma_l}{\xi_l}) \\
&\quad +4\tilde{\beta}\sum_{i=1}^{n}\{\sum_{j\epsilon\delta_i}\sum_{l=1}^{K} q_{il}q_{jl}\} - 2\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\tilde{\beta}\sum_{j\epsilon\delta_i} q_{jl})\}.
\end{aligned}
$$

The constants in terms 1 and 2 cancel each other out in p_D and, since we are taking $\mathbf{E}_q[\beta] = \tilde{\beta}$, we will have further cancellation in the above formula.

In order to approximate $\mathbf{E}_\theta\left(\sum_{i=1}^{n}\log\{\sum_{l=1}^{K}\exp(2\beta\sum_{j\epsilon\delta_i} q_{jl})\}\right)$ we recall the approximation to $q_\beta(\beta)$ used in Section 5.5.3. We have

$$\begin{aligned}\log q_\beta(\beta) &= \log(CQ(\beta)) \\ &= \log C + \log Q(\beta) \\ &\approx \text{constant} - \frac{1}{2\hat{\sigma}_\beta^2}(\beta - \hat{\beta})^2.\end{aligned}$$

Then, approximately, after approximating to $\log Q(\beta)$ by a quadratic, we have

$$\beta \sim N(\hat{\beta}, \hat{\sigma}_\beta^2).$$

If we denote the function $\sum_{i=1}^n \log\{\sum_{l=1}^K \exp(2\beta \sum_{j\epsilon\delta_i} q_{jl})$, by $F(\beta)$ then we can expand to give

$$F(\beta) \approx F(\hat{\beta}) + (\beta - \hat{\beta})F'(\hat{\beta}) + \frac{1}{2}(\beta - \hat{\beta})^2 F''(\hat{\beta}),$$

which implies that

$$\mathrm{E}(F(\beta) - F(\hat{\beta})) \approx \frac{1}{2}\hat{\sigma}_\beta^2 F''(\hat{\beta}),$$

with

$$F''(\hat{\beta}) = \sum_{i=1}^n \left[-\frac{4(\sum_{l=1}^K (\sum_{j\epsilon\delta_i} q_{jl}) e^{2\beta \sum_{j\epsilon\delta_i} q_{jl}})^2}{(\sum_{l=1}^K e^{2\beta \sum_{j\epsilon\delta_i} q_{jl}})^2} + \frac{4\sum_{l=1}^K (\sum_{j\epsilon\delta_i} q_{jl})^2 e^{2\beta \sum_{j\epsilon\delta_i} q_{jl}}}{\sum_{l=1}^K e^{2\beta \sum_{j\epsilon\delta_i} q_{jl}}} \right].$$

So we can approximate $\mathbf{E}_\theta \left(\sum_{i=1}^n \log\{\sum_{l=1}^K \exp(2\beta \sum_{j\epsilon\delta_i} q_{jl})\} \right)$ by $\frac{1}{2}\hat{\sigma}_\beta^2 F''(\hat{\beta})$. Plugging this into our formula for p_D and cancelling out terms gives

$$\begin{aligned} p_D &= -\sum_{l=1}^K \sum_{i=1}^n q_{il}\{\Psi(\frac{\gamma_l}{2}) - \log(\frac{\xi_l}{2}) - \frac{1}{\lambda_l}\} + \hat{\sigma}_\beta^2 F''(\hat{\beta}) + 2\sum_{l=1}^K \sum_{i=1}^n q_{il} \log(\frac{\gamma_l}{\xi_l}) \\ &\quad -2\sum_{i=1}^n \log\{\sum_{l=1}^K \exp(2\tilde{\beta} \sum_{j\epsilon\delta_i} q_{jl})\} \end{aligned}$$

The DIC can then be obtained through the usual formula,

$$\mathrm{DIC} = 2p_D - 2\log p(y|\tilde{\theta}).$$

H.3 Mean-Field Approximation

The lower-bound on the log-likelihood is given by

$$\begin{aligned}\log p(y|\theta) &= \log\left\{\sum_z p(y,z)\right\} \\ &= \log\left\{\sum_z \left(\frac{q_z(z)p(y,z)}{q_z(z)}\right)\right\} \\ &\geq \sum_{\{z\}} q_z(z)\log\frac{p(y,z)}{q_z(z)}.\end{aligned}$$

Of course we would have equality if we were to take $q_z(z) = p(z|y)$ but in order to simplify calculations, the most straightforward approach is to take

$$q_z(z) = \prod_i^n q_{z_i}(z_i)$$

leading to

$$\log p(y|\theta) \geq \sum_{\{z\}}\prod_i^n q_{z_i}(z_i)\log\frac{p(y|z)p(z)}{\prod_i^n q_{z_i}(z_i)}. \tag{H.2}$$

We have

$$p(y|z) = \prod_{i=1}^n p(y_i|z_i),$$

and, for the Ising model,

$$\begin{aligned}p(z) &= \frac{1}{G(\beta)}\exp\left\{\beta\sum_{i\sim j} z_i z_j\right\} \\ &= \frac{1}{G(\beta)}\exp\left\{\beta z_i\sum_{j\in\delta_i} z_j + \text{terms not involving } z_i\right\}.\end{aligned}$$

If we concentrate on terms involving z_i, the right-hand side of (H.2) becomes

$$\begin{aligned}
\sum_{z_i,\{z_j:j\in\delta_i\}} q_{z_i}(z_i) \prod_{j\in\delta_i} q_{z_j}(z_j) \left[\log\left(\frac{p(y_i|z_i)}{q_{z_i}(z_i)}\right) + \beta z_i \sum_{j\in\delta_i} z_j\right] \\
= \sum_{\{z_i\}} q_{z_i}(z_i) \left[\log\left(\frac{p(y_i|z_i)}{q_{z_i}(z_i)}\right) + \beta z_i \sum_{j\in\delta_i} m_j\right] \\
= \sum_{\{z_i\}} q_{z_i}(z_i) \left[\log\left(\frac{p(y_i|z_i)e^{\beta z_i \sum_{j\in\delta_i} m_j}}{q_{z_i}(z_i)}\right)\right]
\end{aligned}$$

where $m_j = \mathbf{E}_{q_{z_j}}(z_j)$. Then optimising with respect to q_{z_i} gives

$$q_{z_i}(z_i) \propto p(y_i|z_i)e^{\beta z_i \sum_{j\in\delta_i} m_j}$$

Since z_i can take the value +1 or -1, we have

$$q_{z_i}(z_i = -1) = \frac{p(y_i|z_i = -1)e^{-\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i = +1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i = -1)e^{-\beta \sum_{j\in\delta_i} m_j}}$$

$$q_{z_i}(z_i = +1) = \frac{p(y_i|z_i = +1)e^{\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i = +1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i = -1)e^{-\beta \sum_{j\in\delta_i} m_j}}.$$

where the $\{m_i\}$ are given by

$$m_i = \frac{p(y_i|z_i = +1)e^{\beta \sum_{j\in\delta_i} m_j} - p(y_i|z_i = -1)e^{-\beta \sum_{j\in\delta_i} m_j}}{p(y_i|z_i = +1)e^{\beta \sum_{j\in\delta_i} m_j} + p(y_i|z_i = -1)e^{-\beta \sum_{j\in\delta_i} m_j}}$$

This set of nonlinear equations for the m_i can be solved iteratively. From this we will have obtained $q_{z_i}(z_i = +1)$ and $q_{z_i}(z_i = -1)$ for each pixel i. These can then be used to obtain a lower bound approximation for $p(y|\theta)$.

The right-hand side of (H.2) can be written

$$
\begin{aligned}
&\sum_{\{z\}} \prod_i^n q_{z_i}(z_i) \log \prod_i^n \frac{p(y_i|z_i)}{q_{z_i}(z_i)} + \sum_{\{z\}} \prod_i^n q_{z_i}(z_i) \log p(z) \\
&= \sum_i^n \sum_{\{z_i\}} q_{z_i}(z_i) \log \frac{p(y_i|z_i)}{q_{z_i}(z_i)} + \sum_{\{z\}} \prod_i^n q_{z_i}(z_i) \{\beta \sum_{i\sim j} z_i z_j - \log G(\beta)\} \\
&= \sum_i^n \left[\sum_{\{z_i\}} q_{z_i}(z_i) \log \frac{p(y_i|z_i)}{q_{z_i}(z_i)} \right] + \beta \sum_{i\sim j} m_i m_j - \log G(\beta).
\end{aligned}
$$

The term $\log G(\beta)$ is too complex for computational purposes, and we therefore require an approximation. We will use a pseudo-likelihood approximation again, for $\log p(z)$.

$$
p_{PL} = \prod_{i=1}^n \frac{e^{\beta z_i \sum_{j\in\delta_i} z_j}}{e^{\beta \sum_{j\in\delta_i} z_j} + e^{-\beta \sum_{j\in\delta_i} z_j}} = \frac{e^{2\beta \sum_{i\sim j} z_i z_j}}{\prod_{i=1}^n (e^{\beta \sum_{j\in\delta_i} z_j} + e^{-\beta \sum_{j\in\delta_i} z_j})}
$$

Using this approximation we obtain

$$
\begin{aligned}
&\sum_i^n \left[\sum_{\{z_i\}} q_{z_i}(z_i) \log \frac{p(y_i|z_i)}{q_{z_i}(z_i)} \right] \\
&\quad + \sum_z \prod_{i=1}^n q_{z_i}(z_i) \left\{ 2\beta \sum_{r\sim s} z_r z_s - \sum_r \log(e^{\beta \sum_{s\in\delta_r} z_s} + e^{-\beta \sum_{s\in\delta_r} z_s}) \right\} \\
&= \sum_i^n \left[\sum_{\{z_i\}} q_{z_i}(z_i) \log \frac{p(y_i|z_i)}{q_{z_i}(z_i)} \right] \\
&\quad + 2\beta \sum_{i\sim j} m_i m_j - \sum_{i=1}^n \sum_{z_j : j\in\delta_i} \prod_{j\in\delta_i} q_{z_j}(z_j) \log \left(e^{\beta \sum_{j\in\delta_i} z_j} + e^{-\beta \sum_{j\in\delta_i} z_j} \right).
\end{aligned}
$$

Then the approximate lower bound for $p(y|\theta)$ is given by

$$\sum_{i=1}^{n}\left[\sum_{\{z_i\}} q_{z_i}(z_i)\log\left\{\frac{p(y_i|z_i)}{q_{z_i}(z_i)}\right\}\right] + 2\beta\sum_{i\sim j} m_i m_j$$
$$- \sum_{i=1}^{n}\sum_{\{z_j : j\in\delta_i\}}\prod_{j\in\delta_i} q_{z_j}(z_j)\log(e^{\beta\sum_{j\in\delta_i} z_j} + e^{-\beta\sum_{j\in\delta_i} z_j}).$$

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings of the Second International Symposium on Information Theory* (B.N. Petrov and F. Csaki, eds.), pp.267-281 Budapest: Akademiai Kiado.

Andrieu, C., de Freitas, N., Doucet, A. and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, **50** 5-43.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. *Proceedings of Conference on Uncertainty in Artificial Intelligence* (UAI).

Bahl, L. R., Jelinek, F. and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **5** 179-190.

Baker, J.K. (1975). The dragon system - an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **23** 24-29.

Bartlett, M.S. (1955). *An Introduction to Stochastic Processes.* Cambridge University Press.

Bartlett, M.S. (1967). Inference and stochastic processes. *Journal of the Royal Statistical Society, Series A*, **130** 457-477.

Bartlett, M.S. (1968). A further note on nearest neighbour models. *Journal of the Royal Statistical Society, Series A*, **131** 579-580.

Baum, L.E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, **37** 1554-1563.

Baum, L.E. and Egon, J.A. (1967). An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. *Bulletin of the American Meteorological Society*, **73** 360-363.

Baum, L.E. and Sell, G.R. (1968). Growth functions for transformations on manifolds. *Pac. Journal of Mathematical Statistics*, **27** 211-227.

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41** 164-171.

Baum, L.E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, **3** 1-8.

Bechtel, Y.C., Bonaïti-Pellié, C., Poisson, N., Magnette, J. and Betchel, P.R. (1993). A population and family study of N-acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology and Therapeutics*, **54** 134-141.

Berg, A., Meyer, R. and Yu, J. (2004). Deviance information criterion for comparing stochastic volatility models. *Journal of Business and Economic Statistics*, **22** 107-119.

Besag, J. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society, Series B*, **34** 75-83.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series B*, **36** 192-236.

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, **24** 179-195.

Besag, J. (1986). On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society, Series B*, **48** 259-302.

Besag, J. and Higdon, D. (1999). Bayesian analysis of agricultural field experiments. *Journal of the Royal Statistical Society, Series B*, **61** 691-746.

Bickel, P. J., Ritov, Y. and Rydén, T. (1998). Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *Annals of Statistics*, **26** 1614-1635.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations.* Oxford Statistical Science Series, Oxford University Press, Oxford.

Boys, R.J., Henderson, D.A. and Wilkinson, D.J. (2000). Detecting homogeneous segments in DNA sequences by using hidden Markov models. *Journal of the Royal Statistical Society, Series C*, **49** 269-285.

Burnham, K.P. and Anderson, D.R. (1998). *Model Selection and Inference: A Practical Information-Theoretical Approach.* Springer, New York.

Casella, G., Mengersen, K.L., Robert, C.P. and Titterington, D.M. (2002). Perfect samplers for mixtures of distributions. *Journal of the Royal Statistical Society, Series B*, **64** 777-790.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, **2** 73-82.

Celeux, G. and Diebolt, J. (1989). Une version de type recuit simulé de l'algorithme EM. *Notes aux Comptes Rendus de l'Académie des Sciences*, **310** 119-124.

Celeux, G., Forbes, F., Robert, C. and Titterington, D.M. (2006). Deviance Information Criteria for missing data models. *Bayesian Analysis*, to appear.

Cheng, B. and Titterington, D.M. (1994). Neural networks: a review form a statistical perspective (with discussion). *Statistical Science*, **9** 2-54.

Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics*, **75** 79-97.

Churchill, G.A. (1995). Accurate restoration of DNA sequences (with discussion). In *Case Studies in Bayesian Statistics* (C. Gastonis, J.S. Hodges, R.E. Kass, and N.D. Singpurwalla, eds.), vol. 2 pp. 90-148. Springer, New York.

Corduneanu, A. and Bishop, C.M. (2001). Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics* (T. Jaakkola and T. Richardson, eds.), pp.27-34 Morgan Kaufmann.

Crawford, S.L., DeGroot, M.H., Kadane, J.B. and Small, M.J. (1992). Modeling lake chemistry distributions: approximate Bayesian methods for estimating a finite-mixture model. *Technometrics*, **34** 441-453.

Dempster, A.P. (1974). The direct use of likelihood for significance testing. In *Proceedings of Conference on Foundational Questions in Statistical Inference* (University of Aarhus), pp.335-352.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, **39** 1-38.

Diebolt, J. and Robert, C.P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, Series B*, **56** 363-375.

Dobrushin, R.L. (1968). The description of a random field by means of conditional probabilities and conditions of its regularity. *Theory of Probability and Its Applications* , **13** 197-224.

Doucet, A. de Freitas, N. and Gordon, N.J., eds (2001). *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, Berlin.

Everitt, B.S. and Hand, D.J. (1981). *Finite Mixture Distributions.* Chapman and Hall.

Ferguson, J. D. (1980). Hidden Markov analysis: An introduction. In *Hidden Markov Models for Speech* (J. D. Ferguson, ed.). pp. 8-15. Institute for Defense Analyses, Princeton, NJ.

Francq, C. and Roussignol, M. (1997). On white noise driven by hidden Markov chains. *Journal of Time Series Analysis*, **18** 553-578.

Gelfand, A. and Smith, A.F.M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85** 398-409.

Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B. (2003). *Bayesian Data Analysis*, Second Edition. Chapman and Hall/CRC.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6** 721-741.

Geyer, C. (1992). Practical Markov chain Monte Carlo (with discussion). *Statistical Science*, **7** 473-483.

Ghahramani, Z. (2001). An introduction to Hidden Markov models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, **15** 9-42.

Ghahramani, Z. and Beal, M. (2001). Propagation algorithms for variational learning. In *Advances in Neural Information Processing, Vol.13* (D.S. Touretzky, M.C. Mozer and M.E. Hasselmo, eds.). MIT Press, Cambridge, MA.

Gilks, W.R., Best, N.G. and Tan, K.K.C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Journal of Applied Statistics*, **44** 455-472.

Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (eds.) (1996). *Markov Chain Monte Carlo in Practice.* Chapman and Hall, London.

Gose, E. and Johnsonbaugh, R. (1996). *Pattern Recognition and Image Analysis.* Prentice Hall.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82** 711-732.

Green, P.J. and Richardson, S. (2002). Hidden Markov models and disease mapping. *Journal of the American Statistical Association*, **97** 1055-1070.

Han, C. and Carlin, B.P. (2001). Markov chain Monte Carlo methods for computing Bayes factors: a comparative review. *Journal of the American Statistical Association*, **96** 1122-1132.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57** 97-109.

Jelinek, F., Bahl, L.R. and Mercer, R.L. (1975). Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Trans. Information Theory*, **21** 250-256.

Jordan, M. I., Ghahramani, Z, Jaakkola, T. S. and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, **37** 183-233.

Jordan, M.I. (2004). Graphical models. *Statistical Science*, **19** 140-155.

Juang, B.H. and Rabiner, L.R. (1991). Hidden Markov models for speech recognition. *Technometrics*, **33** 251-272.

Kass, R.E. and Raftery, A.E. (1995). Bayes factors. *Journal of the American Statistical Association*, **90** 773-795.

Katz, R.W. (1981). On some criteria for estimating the order of a Markov chain. *Technometrics*, **23** 243-249.

Kay, J.W. and Titterington, D.M. (Eds.) (1999). *Statistics and Neural Networks: Recent Advances at the Interface.* Oxford University Press, Oxford.

Kullback, S. and Leibler, R.A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, **22** 79-86.

Lee, L., Attias, H. and Deng, L. (2003). Variational inference and learning for segmental switching state space models of hidden speech dynamics. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Hong Kong, Apr, 2003.*

Leroux, B. G. (1992). Maximum-likelihood estimation for hidden Markov models. *Stochastic Processes and their Applications*, **40** 127-143.

MacDonald, I.L. and Zucchini, W. (1997). *Hidden Markov Models and Other Models for Discrete-valued Time Series.* Chapman & Hall, London.

MacKay, D.J.C. (1997). Ensemble learning for hidden Markov models. Technical Report, Cavendish Laboratory, University of Cambridge.

MacKay, D.J.C. (2001). Local minima, symmetry-breaking, and model pruning in variational free energy minimization. Available from: http://www.inference.phy.cam.ac.uk/mackay/minima.pdf.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models.* Wiley Series in Probability and Statistics.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21** 1087-1092.

Murray, I. and Ghahramani, Z. (2004). Bayesian learning in undirected graphical models: approximate MCMC algorithms. In *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence (UAI, 2004)* .

Neal, R.M. (1996). *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics No. 118, Springer-Verlag, New York.

Newman, M.E.J. and Barkema, G.T. (1999). *Monte Carlo Methods in Statistical Physics.* Oxford University Press.

Opper, M. and Saad, D, eds. (2001). *Advanced Mean Field Methods.* MIT Press.

Pearson, K. (1894). Contribution to the mathematical theory of evolution. *Phil. Trans. Roy. Soc. Series A*, **185** 71-110.

Petrie, T. (1969). Probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, **40** 97-115.

Pettitt, A.N., Friel, N. and Reeves, R. (2003). Efficient calculation of the normalizing constant of the autologistic and related models on the cylinder and lattice. *Journal of the Royal Statistical Society, Series B*, **65** 235-247.

Postman, M., Huchra, J.P. and Geller, M.J. (1986). Probes of large-scale structure in the Corona Borealis region. *The Astronomical Journal*, **92** 1238-1247.

Qian, W. and Titterington, D. M. (1989). On the use of Gibbs Markov chain models in the analysis of image based on second-order pairwise interactive distributions. *Journal of Applied Statistics*, **16** 267-281.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77** 257-284.

Raftery, A.E. (1998). Bayes factors and BIC: Comment on Weakliem. *Technical Report no. 347*, Department of Statistics, University of Washington.

Richardson, S. and Green, P.J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B*, **59** 731-792.

Ripley, B.D. (1993). Statistical aspects of neural networks. In *Networks and Chaos-Statistical and Probabilistic Aspects* (Barndorff-Nielsen, J.L., Jensen, J.L. and Kendall, W.S., eds), pp. 40-123. Chapman and Hall, London.

Ripley, B.D. (1994). Neural networks and related methods for classification (with discussion). *Journal of the Royal Statistical Society, Series B*, **56** 409-456.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

Robert, C.P., Celeux, G. and Diebolt, J. (1993). Bayesian estimation of hidden Markov chains: a stochastic implementation. *Statistics and Probability Letters*, **16** 77-83.

Robert, C.P. and Titterington, D.M. (1998). Resampling schemes for hidden Markov models and their application for maximum likelihood estimation. *Statistical Computing*, **8** 145-158.

Robert, C.P., Rydén, T. and Titterington, D.M. (2000). Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society, Series B*, **62** 57-75.

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, **85** 617-624.

Rubin, D.B. (1988). Using the SIR algorithm to simulate posterior distributions. In *Bayesian Statistics 3* (J.M. Bernardo, M.H. DeGroot, D.V.Lindley and A.F.M. Smith, eds.). Cambridge, MA: Oxford University Press, 394-402 .

Rustagi, J.S. (1976). *Variational Methods in Statistics.* Academic Press.

Rydén, T., Teräsvirta, T. and Åsbrink, S. (1998). Stylized facts of daily return series and the hidden Markov model. *Journal of Applied Econometrics* **13** 217-244.

Rydén, T. and Titterington, D.M. (1998). Computational Bayesian analysis of hidden Markov models. *Journal of Computational and Graphical Statistics*, **7** 194-211.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6** 461-464.

Shibata, R. (1976). Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*, **63** 117-126.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P. and Van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, **64** 583-639.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components - an alternative to reversible jump methods. *The Annals of Statistics*, **28** 40-74.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, **22** 1701-1728.

Titterington, D.M., Smith, A.F.M. and Makov, U.E. (1985). *Statistical Analysis of Finite Mixture Distributions.* John Wiley and Sons.

Titterington, D.M. (1990). Some recent research in the analysis of mixture distributions. *Statistics*, **21** 619-641.

Titterington, D.M. (2004). Bayesian methods for neural networks and related models. *Statistical Science*, **19** 128-139.

Ueda, N., Nakano, R., Ghahramani, Z. and Hinton, G.E. (2000). SMEM algorithm for mixture models. *Neural Computation*, **12** 2109-2128.

Ueda, N. and Ghahramani, Z. (2002). Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, **15** 1223-1241.

Viterbi, A.J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, **13** 260-269.

Waterhouse, S., Mackay, D. and Robinson, T. (1996). Bayesian methods for mixtures of experts. *Advances in Neural Information Processing Systems* **8** (Touretzky, D.S. et al. eds). MIT Press.

Winkler, G. (1995). *Image Analysis, Random Fields and Dynamic Monte Carlo Methods.* Springer.

Zhu, L. and Carlin, B.P. (2000). Comparing hierarchical models for spatio-temporally misaligned data using the deviance information criterion. *Statistics in Medicine*, **19** 2265-2278.