Monte Carlo and approximate inference approaches for some latent variable models

Jason Wyse

The thesis is submitted to University College Dublin in fulfilment of the requirements for the degree of Doctor of Philosophy

School of Mathematical Sciences, University College Dublin

Head of School: Dr. Mícheál Ó Searcóid

Principal Supervisor: Prof. Nial Friel

Doctoral Studies Panel Membership:

Dr. Claire Gormley

Prof. Brendan Murphy

November 2010
# Contents

1 Introduction
   1.1 Latent variable models .............................................. 1
   1.2 Bayesian analysis of latent variable models ..................... 2
   1.3 Some examples of latent variable models and their estimation ... 4
      1.3.1 Finite mixture models ........................................ 4
      1.3.2 Changepoint models .......................................... 5
      1.3.3 Stochastic volatility models ................................. 7
      1.3.4 Hidden Markov models ....................................... 8
   1.4 The issue of model selection ...................................... 12
      1.4.1 Marginal likelihood from the Gibbs output .................. 13
      1.4.2 Approximating marginal likelihood using power posteriors . 15
      1.4.3 Reversible jump MCMC ....................................... 16
      1.4.4 A case study in model selection: Gaussian finite mixtures with
           an unknown number of components ............................ 18
   1.5 Thesis outline .................................................... 24

2 Simulation based Bayesian analysis of multiple changepoint models 26
   2.1 Introduction ....................................................... 26
   2.2 Changepoint models ............................................... 28
   2.3 Collapsing changepoint models ................................... 29
   2.4 Sampling changepoints ........................................... 31
   2.5 Prior specification .............................................. 32
2.6 Analysis by filtering recursions ........................................... 33
2.7 Poisson data: coal mining disasters ...................................... 35
2.8 Streakiness in sports ............................................................ 36
2.9 Gaussian changepoint models ............................................... 40
2.10 Discussion ............................................................................ 45

3 Collapsed sampling for Bayesian Latent Block models .......... 48
  3.1 Introduction ......................................................................... 48
  3.2 Models .................................................................................. 50
    3.2.1 Latent block models ....................................................... 51
    3.2.2 Bayesian latent block models ......................................... 53
  3.3 MCMC sampling of clusterings ............................................. 58
    3.3.1 MCMC moves ............................................................... 58
    3.3.2 Form of reversible jump sampler ................................... 62
    3.3.3 Label switching ........................................................... 63
    3.3.4 Summarizing MCMC output ......................................... 65
  3.4 Simulated data ...................................................................... 66
  3.5 Congressional voting in US senate ....................................... 69
  3.6 Microarray experiments ......................................................... 73
  3.7 Discussion ............................................................................ 75

4 Changepoint modelling using Gaussian Markov random fields . 80
  4.1 Introduction ......................................................................... 80
  4.2 Hierarchical GMRF models .................................................. 82
    4.2.1 General model definition .............................................. 82
    4.2.2 Integrated nested Laplace approximations .................... 84
    4.2.3 Gaussian approximations ............................................. 85
    4.2.4 Inference for the field and the hyperparameters .......... 87
    4.2.5 The R-INLA package ................................................... 88
  4.3 Changepoint models ............................................................. 88
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>Recursively computing the posterior</td>
<td>89</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Choice of changepoint prior and computational cost</td>
<td>89</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Posterior of any changepoint</td>
<td>90</td>
</tr>
<tr>
<td>4.4</td>
<td>Approximate changepoint inference using INLAs</td>
<td>91</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Reduced filtering recursions for changepoints</td>
<td>92</td>
</tr>
<tr>
<td>4.5</td>
<td>CGH studies of Coriel cell lines</td>
<td>99</td>
</tr>
<tr>
<td>4.6</td>
<td>Coal mining disasters</td>
<td>104</td>
</tr>
<tr>
<td>4.7</td>
<td>Well-log data</td>
<td>109</td>
</tr>
<tr>
<td>4.8</td>
<td>Stochastic volatility data</td>
<td>114</td>
</tr>
<tr>
<td>4.9</td>
<td>Discussion</td>
<td>117</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions and further work</td>
<td>121</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions</td>
<td>121</td>
</tr>
<tr>
<td>5.2</td>
<td>Further work</td>
<td>122</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Collapsed sampling approaches for hidden Markov models</td>
<td>122</td>
</tr>
<tr>
<td>5.2.2</td>
<td>A full assessment of the approximation error in RFRs</td>
<td>125</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Software for R</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Appendix</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>131</td>
</tr>
</tbody>
</table>
Abstract

Latent variable models are ubiquitous in statistical applications and encapsulate a very rich collection of statistical models for which complex dependencies in the data can be modeled via complicated dependencies in unobserved, latent or hidden variables. Examples of such models include finite mixture models; hidden Markov models; Markov random fields, to name but a few.

There are two main aims of this thesis. The first of these is to improve upon current approaches for Monte Carlo inference in two classes of latent variable models; changepoint models with an unknown number of changepoints and latent block models with an unknown number of components. Both of these models have widespread use in the applied sciences. The second aim is to provide approximate inference procedures for changepoint models with an unknown number of changepoints where conditional independence assumptions in the observed data model are relaxed.

The overall outcomes of this thesis are the following. The first main outcome is that it is now possible to perform Bayesian inference for models, which are traditionally classed as variable dimension, within a fixed dimensional framework. This leads to a much more straightforward and statistically efficient approach for multiple changepoint models and latent block models, both of which are otherwise inferentially difficult. The outcome of the second aim provides a new approximate inferential approach for statistically analysing multiple changepoint models which incorporate more complicated and realistic dependencies which were not hitherto possible.
Statement of Original Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Jason Wyse

Dated: November 24, 2010
Collaborations

The work contained in Chapters 2 and 3 of this thesis was undertaken jointly with Prof. Nial Friel, School of Mathematical Sciences, University College Dublin.

The work contained in Chapter 4 of this thesis was undertaken jointly with Prof. Nial Friel and Prof. Håvard Rue, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.
Acknowledgments

Firstly, I would like to thank my supervisor Prof. Nial Friel. I thank him for giving me the opportunity to do research under his guidance and for sharing some of his broad and thorough knowledge of mathematics and statistics with me. Most of all, I thank him for his patience and understanding during my time working with him and for allowing me to do research freely.

I express great gratitude to Dr. Adrian Dunne. I will be always be grateful for his support in the early days. I thank him also for imparting some of his second to none knowledge of applied statistics to me. His shrewdness and instinct with data is astounding; to have even a fraction of either would be more than I could wish for in my future career.

I thank Prof. Håvard Rue who has collaborated on some of this thesis. His interest in my work, his enthusiasm, and his promptness to answer (silly!) questions are very much appreciated.

There are many others I would like to thank. However, brevity is the soul of wit. I thank Prof. Brendan Murphy for his advice and feedback on some of the work in this thesis. I also say thank you to Prof. John Connolly, Dr. Claire Gormley, Ms. Marie Doyle and Dr. Nuria Garcia.

Academic acknowledgments aside, I would like to turn attention to my colleagues and friends. The inevitability of a zero-point-zero-zero research fund balance means that many have come and gone through L533 during my extended stay there. It was always comforting to see a friendly face and share the good and bad times. My days there are my happiest so far and will be fondly remembered. I think of Fiona, Úna,
Clare, Caroline, Adrian, Marie, Erik, Eleni, Zee, Damien, Gift, Triona, Arthur and others too. Of course running out of funding isn’t the only inevitability in this life, something harshly reaffirmed with the recent passing of our friend and L533 colleague Cian. Cian’s strength in the face of illness is beyond comprehension and admirable beyond words. For what little it may be worth Cian, I dedicate this thesis to you and I hope you are happier and freer wherever you are. Outside of L533, I’d like to thank Aidan for many laughs and welcome distractions over the past four years, and overall, for being a great friend.

Finally, I thank my family for their support. Especially, I thank my mother, who heard all my PhD woes at length and listened patiently and always comforted me with her great cooking! I thank my brother Conor, who provided lodgings as a charitable donation to my academic career. I also thank Ronan and his wife Jean for offering a holiday retreat in their home in Ennis on many occasions. If anyone was keeping a tab, I’ll never pay it back with all the worry and sacrifices made on my account over the years!

THANK YOU ALL
FOR MY DEAR FRIEND
AND FELLOW STATISTICIAN,
Publications and supplementary material

Publications

The material in this thesis is already archived in e-prints and has been submitted or accepted to peer reviewed journals. This work has been co-authored by Prof. Nial Friel and Prof. Håvard Rue.


Supplementary materials

Code written in C implementing the examples from some of the papers above, and the corresponding examples in this thesis is available from www.ucd.ie/statdept/jwyse.
Chapter 1

Introduction

1.1 Latent variable models

Latent variable models are frequently used by data analysts. The scope of such models is so large that it is regular or accepted practice to use a latent variable model in many situations. The idea of these models is simple. Latent variables are unobserved but will have some clear instrumental purpose in performing inference on the observed data. To take two common examples, the latent variables could classify members of different groups or give the true intensity of an underlying signal which is corrupted by noise. An observed data point is assumed to arise from a distribution that conditions on the value of a latent variable and the latent variables are assumed to be jointly distributed following some known parametric density. The data point is termed the observed data and the latent variable is often called the hidden or unobserved data.

To be more precise, let \( y = (y_1, \ldots, y_n) \) be some vector of observed data values and let \( z = (z_1, \ldots, z_n) \) be the latent variables. Then the model may be specified as a hierarchical model with two levels; the first of these is for the observed data

\[
y_i \sim_{iid} \pi(y|z_i, \theta) \quad i = 1, \ldots, n,
\]

and the second is for the unobserved data

\[
z \sim \pi(z|\phi).
\]
Figure 1.1: Structure of a latent variable model. The observed data $y$ depends on the unobserved data $z$ and parameters $\theta$.

The vectors $\theta$ and $\phi$, usually assumed unknown, are the parameters of the parametric distributions assumed for the observed and unobserved data. Figure 1.1 illustrates the structure of a latent variable model. As mentioned, the latent variables will usually have a clear purpose or interpretation. Knowing their values may give additional information or make patterns in the observable data more transparent. Often, inferring their values will be one of the sole purposes of an analysis.

### 1.2 Bayesian analysis of latent variable models

A Bayesian analysis is particularly well suited to hierarchical models. This is because hierarchical models may be viewed as an ordered collection of conditional probabilities i.e. the probability of the data conditional on specific parameter values; the probability of specific parameter values conditional on the value of some hyperparameters. Inference for the unknown parameters is carried out using their posterior distributions, which are obtained by conditioning on the observed data. In the case of latent variable models, one may condition on the data to get the posterior distributions of the latent variables and the parameters. Independent prior distributions may be introduced for $\theta$ and $\phi$ to
give the un-normalized hierarchical posterior

\[ \pi(z, \theta, \phi \mid y) \propto \pi(y \mid z, \theta)\pi(z \mid \phi)\pi(\theta)\pi(\phi). \]

The normalizing constant of this posterior, given by

\[ \pi(y) = \int \pi(y \mid z, \theta)\pi(z \mid \phi)\pi(\theta)\pi(\phi) \, dz \, d\theta \, d\phi, \quad (1.1) \]

is not available in most cases. This is due to one or both of the following reasons. The first is that the integrands have an intractable algebraic form which requires intense numerical methods for evaluation. The second is that in the case of discrete latent variables \( z \), the support of the posterior distribution is simply too large to either fully or efficiently evaluate in order to compute the normalizing constant.

Intractability of the normalizing constant \((1.1)\) means that the marginal posterior distributions

\[
\begin{align*}
\pi(z_i \mid y) &= \int \pi(z, \theta, \phi \mid y) \, dz_{-i} \, d\theta \, d\phi, \quad i = 1 \ldots, n \\
\pi(\theta \mid y) &= \int \pi(z, \theta, \phi \mid y) \, dz \, d\phi \\
\pi(\phi \mid y) &= \int \pi(z, \theta, \phi \mid y) \, dz \, d\theta
\end{align*}
\]

are unavailable. Here, \( z_{-i} \) denotes \( z \) with the \( i^{th} \) entry removed. If the latent variables have an explicit interpretation, such as group membership, the marginal posterior distributions of the \( z_i \) will be of particular interest. One way to overcome the obstacle of intractable normalizing constants is to use Markov chain Monte Carlo (MCMC) methods to give simulation exact representations of the relevant posterior marginal distributions. It is assumed the reader has a working knowledge of MCMC techniques. For a comprehensive treatment of MCMC methodology see Robert & Casella (2004). It is not always necessary to resort to MCMC; there are some difficult cases where efficient deterministic computational methods exist. This will be seen in the next section which aims to give a review of some well known and widely used latent variable models, and techniques for their estimation.
1.3 Some examples of latent variable models and their estimation

1.3.1 Finite mixture models

A finite mixture model assumes an observable data point $y$ arises from any one of a number of distributions. All of these distributions have the same parametric density, $\pi(y|\theta)$ but $\theta$ is different in each. The candidate distributions are called the components of the mixture density. Assuming there are $K$ possible components, the density of $y$ is

$$
\pi(y|K, \omega, \Theta) = \sum_{k=1}^{K} \omega_k \pi(y|\theta_k),
$$

where $\Theta$ denotes the collection $\{\theta_k\}_{k=1}^{K}$ and $\omega = (\omega_1, \ldots, \omega_K)$ are the mixture weights. The mixture weights sum to one and give the marginal probability that $y$ arises from the corresponding component.

Viewing data $y = (y_1, \ldots, y_n)$ as independent draws from the above model, the likelihood for the unknown parameters $\omega$ and $\Theta$ is

$$
\pi(y|K, \omega, \Theta) = \prod_{i=1}^{n} \left( \sum_{k=1}^{K} \omega_k \pi(y_i|\theta_k) \right).
$$

Taking a standard Dirichlet($\alpha_1, \ldots, \alpha_K$) prior on the mixture weights and prior $\pi(\theta_k|\xi_k)$ on the component parameters, the joint posterior distribution of $\Theta$ and $\omega$ is

$$
\pi(\Theta, \omega|K, y) \propto \prod_{k=1}^{K} \omega_k^{\alpha_k-1} \prod_{k=1}^{K} \pi(\theta_k|\xi_k) \prod_{i=1}^{n} \left( \sum_{k=1}^{K} \omega_k \pi(y_i|\theta_k) \right)
$$

with $\xi_k$ the hyperparameters for the prior on $\theta_k$.

Evaluating the normalizing constant of (1.2) or the posterior densities of $\Theta$ and $\omega$, would require summing over all allocations of the $n$ data points to the $K$ component densities. Computing these $K^n$ likelihoods will not be feasible. Usually, when mixture models are employed for distinguishing heterogeneous subgroups in data, interest will focus on high probability allocations of the data points to the components. This allows separation of the data into homogeneous groups.
The introduction of latent variables can greatly improve upon the intractability issues that (1.2) suffers from. Let $z = (z_1, \ldots, z_n)$ be a vector of labels, one for each data point. Each $z_i$ takes a value from $\{1, \ldots, K\}$ and indicates the component which $y_i$ belongs to. Inference is then possible for the $z_i$ through Gibbs sampling (see for example Diebolt & Robert (1994)):

1. Draw $\Theta$ and $\omega$ from their full conditionals;

$$
\theta_k \sim \pi(\theta_k | K, y, z, \xi_k), \quad k = 1, \ldots, K;
$$

$$
\omega \sim \text{Dirichlet}(n_1 + \alpha_1, \ldots, n_K + \alpha_K),
$$

where $n_k$ is the number of $z_i$ that currently have value $k$. If a direct draw from the full conditional of $\theta_k$ is not possible, use a Metropolis-Hastings move.

2. Draw each element of $z$ from its full conditional

$$
\pi(z_i | K, y, z_{-i}, \Theta, \omega) = \text{Multinomial} \left( \frac{\omega_1 \pi(y_i | \theta_1)}{\sum_{k=1}^{K} \omega_k \pi(y_i | \theta_k)}, \ldots, \frac{\omega_K \pi(y_i | \theta_K)}{\sum_{k=1}^{K} \omega_k \pi(y_i | \theta_k)} \right),
$$

for $i = 1, \ldots, n$.

The Gibbs sampling/Metropolis-Hastings iterates can be used to estimate the posterior distributions of the $z_i$, $\Theta$ and $\omega$, albeit in the absence of label switching which may need to be corrected for (see Section 3.3.3 for more details).

### 1.3.2 Changepoint models

Changepoint models are widely used in bioinformatics, econometrics and signal processing and will be the subject of two of the subsequent chapters in this thesis. Consider time ordered data $y = (y_1, \ldots, y_n)$. Time could be fictitious here and only refers to some natural ordering of the data. A segment is a number of consecutive data points $y_{i:j} = (y_i, \ldots, y_j)$ with $i < j$. Under a $K$ changepoint model, the data is assumed to split into $K + 1$ segments defined by the time points $\tau_1, \ldots, \tau_K$ such that data in
different segments independently follow different specifications of the same parametric model. For convenience, define \( \tau_0 = 0 \) and \( \tau_{K+1} = n \). The likelihood is

\[
\pi(y|K, \tau_1, \ldots, \tau_K, \Theta) = \prod_{k=1}^{K+1} \prod_{i = \tau_{k-1} + 1}^{\tau_k} \pi(y_i|\theta_k).
\]

Again, priors \( \pi(\theta_k|\xi_k), k = 1, \ldots, K + 1 \) may be introduced on the segment parameters. A prior on the changepoints is also introduced, \( \pi(\tau_1, \ldots, \tau_K|K, \phi) \), where \( \phi \) is a vector of hyperparameters. Here the latent variables are the locations of the changepoints. This can be made more explicit, as in Chapter 2. Knowing the changepoint positions is the main purpose of the analysis, so that the data may be split into contiguous consistent sequences.

The joint posterior distribution of \( \Theta \) and the changepoints is

\[
\pi(\tau_1, \ldots, \tau_K, \Theta|K, y) \propto \pi(\tau_1, \ldots, \tau_K|\Theta) \prod_{k=1}^{K+1} \pi(\theta_k|\xi_k) \prod_{k=1}^{K+1} \prod_{i = \tau_{k-1} + 1}^{\tau_k} \pi(y_i|\theta_k).
\]

In the case of one changepoint and small \( n \), this model can be manageable analytically or with minimal recourse to numerical techniques. See for example Ferreira (1975) or Smith (1975). If this is not the case, the normalizing constant will have \( \binom{n-1}{K} \) terms, and quickly becomes intractable. Note that there are \( n - 1 \) possible times where a changepoint may occur, since there may not be a change at the last time point.

Again, a Gibbs scheme may be employed to estimate posterior marginal densities. A classic approach was proposed by Carlin, Gelfand & Smith (1992) in the case of models with one changepoint. For more than one changepoint, a general scheme may be applied as follows:

1. Draw \( \Theta \) from its full conditional

\[
\theta_k \sim \pi(\theta_k|K, y, \tau_{k-1}, \tau_k, \xi_k), \quad k = 1, \ldots, K + 1.
\]

If a direct draw is not possible, use a Metropolis-Hastings move to update \( \theta_k \).

2. Draw \( \tau_k \) from its full conditional

\[
\pi(\tau_k = t|K, y, \tau_{(-k)}, \theta_{(-k)}, \theta_k) \propto \pi(y_{\tau_{k-1}+1:t}|\theta_{(-k)}) \pi(y_{t+1:K}|\theta_k) \\
\times \pi(\tau_1, \ldots, \tau_{k-1}, t, \tau_{k+1}, \ldots, \tau_K).
\]
for $t = \tau_{k-1} + 1, \ldots, \tau_k - 1$, where $\tau^{(-k)} = (\tau_1, \ldots, \tau_{k-1}, \tau_{k+1}, \ldots, \tau_K)$.

Gibbs schemes like this are standard practice for estimation in changepoint models. However, as will be investigated in Chapters 2 and 4, there are efficient methods for computing the normalizing constant and carrying out exact inference for some multiple changepoint models.

### 1.3.3 Stochastic volatility models

Stochastic volatility models are used in financial applications and are an example of continuous latent variable models (see Chib, Nardari & Shephard (2002)). To account for fluctuating variability in financial time series, this model assumes that the logarithm of the variance of a Gaussian process follows (usually) an autoregressive model of order one (AR(1)) with some intercept. Here, denote the latent variables by $x = (x_1, \ldots, x_n)$.

The stochastic volatility model may be specified as

$$y_i \sim N \left(0, \beta^2 e^{x_i} \right), \quad i = 1, \ldots, n.$$  

The log variance of any $y_i$ is given by

$$2 \log \beta + x_i,$$

where $x$ follows the AR(1) process,

$$x_i = \phi x_{i-1} + \varepsilon_i, \quad i = 2, \ldots, n.$$  

Here, the errors $\varepsilon_i \sim N(0, \sigma^2)$. The definition of the AR(1) process is completed by assuming it is well defined,

$$x_1 \sim N \left(0, \sigma^2/(1 - \phi^2) \right).$$

These models have been the focus of much study, and are almost a benchmark for testing new MCMC techniques. See Celeux, Marin & Robert (2006) for a comparison of some Monte Carlo techniques and Kitagawa & Sato (2001) for an approach based
on sequential Monte Carlo methods. A recent advance has been made in the analysis of such models through the Riemann manifold Hamiltonian Monte Carlo approach of Girolami & Calderhead (2011).

Taking independent priors for the parameters $\beta$, $\phi$ and $\sigma$, their joint posterior with $x$ may be written

$$
\pi(x, \beta, \phi, \sigma | y) \propto \pi(\phi) \pi(\beta) \pi(\sigma) \times \sigma^{-n} \beta^{-n} \sqrt{1 - \phi^2} \times \exp\left\{-\frac{1 - \phi^2}{2\sigma^2} x_i^2 - \frac{1}{2\sigma^2} \sum_{i=2}^n (x_i - \phi x_{i-1})^2 - \frac{1}{2} \sum_{i=1}^n x_i \right\} \times \exp\left\{-\frac{1}{2\beta^2} \sum_{i=1}^n y_i^2 e^{-x_i} \right\}.
$$

For Monte Carlo inference, Gibbs updates are possible for $\beta$ and $\sigma$ using conjugate priors, but $\phi$ will require a Metropolis-Hastings update. Clearly the posterior is highly intractable with respect to marginalizing out even a subset of the latent variables $x$. These can be updated using Metropolis-Hastings on their full conditionals,

$$
\pi(x_i | y, x_{-i}, \beta, \phi, \sigma) \propto \exp\left\{-\frac{1 + \phi^2}{2\sigma^2} \left( x_i - \phi(x_{i-1} + x_{i+1}) + \sigma^2 \right)^2 - \frac{1}{2\beta^2} y_i^2 e^{-x_i} \right\},
$$

$i = 1, \ldots, n$. Proposals for these updates could be difficult to tune given the complex form of the full conditional. Another thing to consider is that there will be $n$ updates relating to $x$ in each sweep of a MCMC algorithm.

The approach of Girolami & Calderhead (2011) improves greatly on this, by providing an automatically adaptive Monte Carlo method. The idea is to introduce an auxiliary vector $p$ of dimension $n$ which acts as the momentum of a mechanical system described by $x$. The Hamiltonian of this (imaginary) mechanical system is the sum of the potential (negative log posterior plus normalizing term) and kinetic (quadratic form involving the momentum vector) energies,

$$
H(x, p) = -\log \pi(x, \beta, \phi, \sigma | y) + \frac{1}{2} \log\{(2\pi)^n |G(x)|\} + \frac{1}{2} p^T G(x)^{-1} p,
$$

where $G(x)$, referred to as the metric tensor, is the Fisher information matrix of $x$, a sparse tri-diagonal matrix in the case of an AR(1) process (see Section 4.2.1 for
more details). An update for the latent volatilities is then achieved by generating \( \mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{G}(\mathbf{x})) \) and integrating Hamilton’s equations,

\[
\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial H}{\partial \mathbf{p}} \quad \text{and} \quad \frac{\partial \mathbf{p}}{\partial t} = -\frac{\partial H}{\partial \mathbf{x}}
\]

numerically, with respect to the (fictitious) time \( t \), to get an approximate solution flow along an isocontour of \( H(\mathbf{x}, \mathbf{p}) \), so that the system energy is preserved. The numerical integration is performed using a generalized Leapfrog scheme (see Girolami & Calderhead (2011) for more details). This allows for efficient transitions since the metric tensor adapts for the local geometric features of the posterior. Suppose after the integration, the proposed mechanical system has configuration \((\mathbf{x}^*, \mathbf{p}^*)\). This is accepted as the next state in the Markov chain with probability

\[
\min (1, \exp \{ -H(\mathbf{x}^*, \mathbf{p}^*) + H(\mathbf{x}, \mathbf{p}) \}) .
\]

Two huge advantages to this are the high acceptance rate of updates (greater than 90% in some cases) and that all the latent variables are updated at once.

Another recent advance in estimation of continuous latent variable models in general is the integrated nested Laplace approximation approach of Rue, Martino & Chopin (2009). This topic will be discussed in Chapter 4.

### 1.3.4 Hidden Markov models

In a hidden Markov model, the latent variable follows a finite state Markov chain. The value of the latent variable determines the assumed generating distribution for the observed data. These models are widely used in speech recognition and bioinformatics applications. To be clearer about the model,

\[
y_i \sim \pi(y_i | K, z_i, \theta_{z_i})
\]

where \( z_i \) represents one of the \( K \) Markov chain states, and is related to \( z_{i-1} \) by the \( K \times K \) transition matrix \( \mathbf{R} = (r_{st}) \). It is useful in this setting to refer to \( i \) as a time due to the ordering of the states. In the following, the dependence on the number of
states $K$ is suppressed, but is to be understood. The likelihood of the observed data for a given value of $\Theta$ and $R$ is

$$
\pi(y|\Theta, R) = \sum_{z_1} \pi(z_1) \pi(y_1|z_1, \theta_{z_1}) \prod_{i=2}^{n} r_{z_{i-1} z_i} \pi(y_i|z_i, \theta_{z_i}).
$$

(1.3)

where $\pi(z)$ is the initial distribution of the hidden Markov chain and $Z$ denotes the space of all possible state trajectories. At first glance, this likelihood may appear intractable, due to the sum over all elements of $Z$. However, it is possible to compute it relatively efficiently without recourse to Monte Carlo techniques. Notice that (1.3) can be decomposed as follows:

$$
\pi(y|\Theta, R) = \sum_{z_1} \pi(z_1) \pi(y_1|z_1, \theta_{z_1}) \sum_{z_2} r_{z_1 z_2} \pi(y_2|z_2, \theta_{z_2})
$$

$$
\cdots \sum_{z_i} \sum_{z_{i-1} z_i} r_{z_{i-1} z_i} \pi(y_i|z_i, \theta_{z_i}) \cdots \sum_{z_n} r_{z_{n-1} z_n} \pi(y_n|z_n, \theta_{z_n}).
$$

(1.4)

Define the quantity $F_i(s)$ as the probability of the data $y_{1:i}$ and the hidden chain being in state $s$ at time $i$ i.e. $z_i = s$. Then

$$
F_i(s) = \pi(y_{1:i}, z_i = s|\Theta, R)
$$

$$
= \pi(y_i|z_i = s, \theta_s) \sum_{z_{i-1}} r_{z_{i-1} z_i} \pi(y_{1:i-1}, z_{i-1}|\Theta, R)
$$

$$
= \pi(y_i|z_i = s, \theta_s) \sum_{z_{i-1}} r_{z_{i-1} z_i} F_{i-1}(z_{i-1}),
$$

and it can be seen that $F_i(s)$ would constitute one term in the sum over $z_i$ in (1.4). Thus, the likelihood can be computed recursively, forwards in time, and

$$
\pi(y|\Theta, R) = \sum_{s=1}^{K} F_n(s).
$$

It is also possible to define a recursion backwards in time conditioning on what has been learned using the forward recursion. This time, let $L_i(s)$ be the probability of
\( z_i = s \) conditioning on all of the data \( \mathbf{y} = \mathbf{y}_{1:n} \),

\[
L_i(s) = \pi(z_i = s|\mathbf{y}, \Theta, \mathbf{R}) \\
= \sum_{t=1}^{K} \pi(z_i = s, z_{i+1} = t|\mathbf{y}, \Theta, \mathbf{R}) \\
= \sum_{t=1}^{K} \pi(z_i = s|z_{i+1} = t, \mathbf{y}, \Theta, \mathbf{R}) \pi(z_{i+1} = t|\mathbf{y}, \Theta, \mathbf{R}) \\
= \sum_{t=1}^{K} \pi(z_i = s|z_{i+1} = t, \mathbf{y}_{1:i+1}, \Theta, \mathbf{R}) L_{i+1}(t).
\]

This can be simplified further by noting

\[
\pi(z_i = s|z_{i+1} = t, \mathbf{y}_{1:i+1}, \Theta, \mathbf{R}) = \frac{\pi(z_i = s, z_{i+1} = t, \mathbf{y}_{1:i+1}|\Theta, \mathbf{R})}{\pi(z_{i+1} = t, \mathbf{y}_{1:i+1}|\Theta, \mathbf{R})} \\
= \frac{\pi(y_{i+1}|z_{i+1} = t, \theta_t) r_{st} F_i(s)}{F_{i+1}(t)},
\]

so that

\[
L_i(s) = \sum_{t=1}^{K} \frac{\pi(y_{i+1}|z_{i+1} = t, \theta_t) r_{st} F_i(s)}{F_{i+1}(t)} L_{i+1}(t).
\]

These recursions together are termed the forward-backward algorithm and were developed by Baum, Petrie, Soules & Weiss (1970). Scott (2002) has given an MCMC scheme for hidden Markov models using these recursions to update \( z \). The principle works on backwards simulation. Draw \( z_n \) from \( \pi(z_n|\mathbf{y}, \Theta, \mathbf{R}) \). Conditioning on its value, draw \( z_{n-1} \) using

\[
\pi(z_{n-1} = s|\mathbf{y}, z_n, \Theta, \mathbf{R}) \propto \frac{\pi(y_n|z_n, \theta_{z_n}) r_{sz_n} F_{n-1}(s)}{F_n(z_n)} L_n(z_n)
\]

and so on. Then update \( \Theta \) and \( \mathbf{R} \). It was shown by Scott (2002), that such an algorithm was more efficient (in the sense of lower autocorrelation between MCMC iterates) than a MCMC algorithm that updates each \( z_i \) from its full conditional;

\[
\pi(z_i|\mathbf{y}, \mathbf{z}, \Theta, \mathbf{R}) \propto r_{z_i z_{i+1}} r_{z_{i+1} z_{i+2}} \pi(y_i|z_i, \theta_{z_i}).
\]

This more efficient algorithm comes at a price. Computing the forward recursions and performing the backward simulation in each sweep of the MC algorithm may be costly.
The forward recursions can be seen to be $O(nK^2)$ in time (at each time point sum $K$ terms where each depends on $K$ terms) and there is an $O(nK)$ storage requirement since they are needed for the backward simulation step. Also, the backward simulation step is $O(nK)$ in time ($K$ probability evaluations at each time point).

Recursive approaches have been developed for more difficult, higher order Markov dependence structures such as that in Friel & Rue (2007) who use forward-backward algorithm computations to perform simulation-free inference for general factorizable models and apply it to autologistic models defined on a rectangular lattice. The idea of recursive computing of posterior distributions and the forward-backward algorithm will be explored further in the context of changepoint models in Chapters 2 and 4.

### 1.4 The issue of model selection

The finite mixture model, changepoint model and hidden Markov model examples in the previous section each concerned situations where $K$ the number of components, changepoints, or states, respectively, are known. However, $K$ will rarely be known in practice and it will often be desired to include it in the collection of parameters to be inferred. For the mixture model example a reasonable question would be; how many groups are in the data?

In an ideal situation with two competing models $M_0$ and $M_1$ it would be possible to compute the Bayes factor to compare the models,

$$ B_{01} = \frac{\pi(y|M_0)}{\pi(y|M_1)} $$

where

$$ \pi(y|M_j) = \int \pi(\theta_j|y,M_j) d\theta_j, \quad j = 0, 1, $$

is the marginal likelihood of the data under the given model. However, the models in question here will generally not be amenable to these analytic calculations due to the reasons outlined in Section 1.2.
To put the problem in a wider setting, suppose there are $M$ models, $M_1, \ldots, M_M$, that are thought to be feasible for the data. The task here is to estimate the posterior model probability (PMP) of model $j$,

$$
\pi(M_j | y) = \frac{\pi(y | M_j) \pi(M_j)}{\sum_{m=1}^{M} \pi(y | M_m) \pi(M_m)}
$$

to a good approximation, so that the possible models can be compared, a posteriori. The prior model probabilities are $\pi(M_j)$ and their specification will depend on the application in hand. Here, the $M_j$ could correspond to multiple changepoint models, each with a different number of changepoints, or hidden Markov models with different numbers of hidden states.

There have been many approaches developed to offer solutions to the model selection problem. Most of these utilize the output from a series of MCMC schemes in a novel way in order to estimate marginal likelihoods, while the reversible jump algorithm of Green (1995) attempts to solve the model selection and within model estimation problem simultaneously. The remainder of this section gives a review of some of these approaches, followed by a case study in model selection for Gaussian finite mixtures with an unknown number of components. The case study examines a relatively new MCMC sampler for mixture models due to Nobile & Fearnside (2007), and discusses how this sampler can simplify an analysis compared with a reversible jump algorithm, but still obtain answers to the essential questions when performing a mixture analysis. In what follows, the conditioning on the model $M_j$ is suppressed when there is no ambiguity.

1.4.1 Marginal likelihood from the Gibbs output

Chib (1995) proposes a method to estimate the marginal likelihood from the output of a Gibbs sampling scheme. The approach is based on a simple rearrangement of Bayes’ theorem.

$$
\pi(\theta | y) = \frac{\pi(y | \theta) \pi(\theta)}{\pi(y)}
$$
implies that

\[
\pi(y) = \frac{\pi(y|\theta)\pi(\theta)}{\pi(\theta|y)}
\]  

for any value of \( \theta \). For latent variable models,

\[
\pi(\theta|y) = \int \pi(\theta|y,z)\pi(z|y,\theta)\,dz.
\]

For simplicity of notation, \( \phi \) is absorbed into \( \theta \) for the remainder of this section. The rearrangement (1.5) implies that for an appropriate point \( \tilde{\theta} \),

\[
\log \pi(y) = \log \pi(y|\tilde{\theta}) + \log \pi(\tilde{\theta}) - \log \pi(\tilde{\theta}|y).
\]

Thus obtaining an estimate of the marginal likelihood reduces to obtaining an estimate of \( \pi(\tilde{\theta}|y) \). Choosing \( \tilde{\theta} \) to be at the posterior mode, or a point with high posterior support should improve the estimation.

Supposing the model is amenable to Gibbs sampling (for example Gaussian finite mixtures and hidden Markov models) and that the parameter \( \theta \) can be split into \( B \) blocks; \( \tilde{\theta} = (\tilde{\theta}^1, \ldots, \tilde{\theta}^B) \), by the law of total probability

\[
\pi(\tilde{\theta}|y) = \pi(\tilde{\theta}^1|y)\pi(\tilde{\theta}^2|y,\tilde{\theta}^1) \ldots \pi(\tilde{\theta}^B|y,\tilde{\theta}^1,\ldots,\tilde{\theta}^{B-1})
\]

where \( \pi(\tilde{\theta}^i|y,\tilde{\theta}^1,\ldots,\tilde{\theta}^{i-1}) \) is the reduced full conditional ordinate of \( \theta^i \).

Assume that each full conditional, \( \pi(\theta^i|y,\theta^{-i}) \) is fully known. Then, \( \pi(\tilde{\theta}^1|y) \) is obtained by performing Gibbs sampling on \( \theta \) and constructing a density estimate of \( \theta^1 \) by averaging over the sampled \( \theta^2, \ldots, \theta^B, z \) values

\[
\pi(\theta^1|y) = \frac{1}{N} \sum_{t=1}^{N} \pi(\theta^1|y,\theta^2_t,\ldots,\theta^B_t,z_t).
\]  

(1.6)

The value \( \tilde{\theta}^1 \) could be obtained by taking the mode of a kernel density estimate of \( \pi(\theta^1|y) \) using some values obtained from (1.6).

The remaining \( (j < B) \) ordinates are found in a similar way, by running the chain for a further \( N \) iterations and averaging the full conditional of \( \theta^j \) over sampled values for \( \theta^{j+1}, \ldots, \theta^B, z \) assuming \( \theta^1, \ldots, \theta^{j-1} \) are fixed at \( \tilde{\theta}^1, \ldots, \tilde{\theta}^{j-1} \):

\[
\hat{\pi}(\theta^j|y,\tilde{\theta}^1,\ldots,\tilde{\theta}^{j-1}) = \frac{1}{N} \sum_{t=1}^{N} \pi(\theta^j|y,\tilde{\theta}^1_t,\ldots,\tilde{\theta}^{j-1}_t,\theta^{j+1}_t,\ldots,\theta^B_t,z_t).
\]
The final ordinate is obtained by performing $N$ Gibbs sampling iterations on the full conditionals

$$\pi(\theta^B|y, \tilde{\theta}^1, \ldots, \tilde{\theta}^{B-1}, z)$$

and then summing over the $z$ iterates,

$$\pi(\theta^B|y, \tilde{\theta}^1, \ldots, \tilde{\theta}^{B-1}) = \frac{1}{N} \sum_{t=1}^{N} \pi(\theta^B|y, \tilde{\theta}^1, \ldots, \tilde{\theta}^{B-1}, z_t).$$

Chib (1995) shows how this approach can be implemented to compute approximate marginal likelihoods for each of a number of components in a finite mixture model, and the number of latent states in a hidden Markov model. One potential drawback of this method is that it requires repeated runs of a Gibbs sampling algorithm, where each is reliant on the previously chosen $\tilde{\theta}^j$’s. It would be imagined that choosing a poor $\tilde{\theta}^j$ somewhere along the sequence $j = 1, \ldots, B$ may lead to a poor approximation of the marginal likelihood. Chib & Jeliazkov (2001) give a way to estimate marginal likelihood from the Metropolis-Hastings output which has similarities to the method based on the Gibbs output.

### 1.4.2 Approximating marginal likelihood using power posteriors

Friel & Pettitt (2008) suggest an approach to estimating marginal likelihood based on the idea of thermodynamic integration. A power posterior is a representation of the posterior density which allows one to express the degree to which evidence from the data is incorporated into calculation of the posterior density. A parameter $t \in [0, 1]$ gives the degree of evidence incorporated, and may be interpreted as an inverse temperature parameter as used in simulated annealing (Robert & Casella 2004). The power posterior is

$$\pi_t(\theta, z|y) \propto \pi(y|\theta, z)^t \pi(z|\theta) \pi(\theta).$$
It is clear that \( t = 0 \) corresponds to prior assumptions (no evidence from the data) and \( t = 1 \) corresponds to full incorporation of the data. It can be shown that

\[
\log \pi(y) = \int_0^1 \mathbb{E}_{\theta, z|y,t}(\log \pi(y|\theta, z)) \, dt
\]

where the expectation in the integrand is taken with respect to \( \pi_t(\theta, z|y) \).

Friel & Pettitt (2008) point out that it is possible to estimate the integral in (1.7) using a grid of \( t \) values \( 0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = 1 \) and the trapezoidal rule,

\[
\log \pi(y) \approx \sum_{k=0}^{N-1} \frac{1}{2}(t_{k+1} - t_k) \{\mathbb{E}_{\theta, z|y,t_{k+1}}(\log \pi(y|\theta, z)) + \mathbb{E}_{\theta, z|y,t_k}(\log \pi(y|\theta, z))\}.
\]

The expectations \( \mathbb{E}_{\theta, z|y,t}(\log \pi(y|\theta, z)) \) may be estimated by sampling the power posterior at inverse temperature \( t \). A serial Monte Carlo or population Monte Carlo (PMC) approach are possible ways to do this.

The power posterior approach is particularly useful when using exponential family models, since the full conditionals for a Gibbs sampler will usually have convenient forms despite the presence of \( t \) as noted by Friel & Pettitt (2008). In this case, scaling of the full conditionals is automatic so it should be possible to quickly obtain representative samples from \( \pi_t(\theta, z|y) \).

This approach is more flexible than the approach of Chib (1995), since it does not require calculating the \( \tilde{\theta}^j \)s and there is no possibility of the marginal likelihood estimate being greatly affected by the way in which blocks of parameters are updated. Friel & Pettitt (2008) offer guidance in relation to how the \( t \) values may be chosen. The spacing between points could affect the estimate of \( \pi(y) \) through the trapezoidal rule approximation of the integral.

### 1.4.3 Reversible jump MCMC

Reversible jump MCMC (Green 1995) (RJMCMC) is a Monte Carlo algorithm which samples both model space and parameter space. What is meant by this is that samples are generated from the product space

\[
\mathcal{X} = \prod_l \{l\} \times \{\Theta_l|M_l\},
\]
where $\Theta_l$ is the parameter space associated with model $\mathcal{M}_l$. So each sample in the MCMC output has a model index along with a sample of parameters associated with that model. Put simply, RJMCMC is a generalization of the Metropolis-Hastings algorithm which allows switching between different models. Different models will have parameter spaces of different dimension and will possibly be non-nested which makes this task difficult.

When not switching between model spaces, RJMCMC is equivalent to standard MCMC samplers. Switching between two model spaces, say, $\{l\} \times \{\Theta_l|\mathcal{M}_l\}$ to $\{k\} \times \{\Theta_k|\mathcal{M}_k\}$ is more involved. Moving from $(l, \theta_l)$ to $(k, \theta_k)$ requires generating random numbers $u_l$ from $g_l(\cdot)$ and setting $(k, \theta_k) = f_{lk}(l, \theta_l, u_l)$ for some deterministic function $f_{lk}(\cdot)$. For the reverse move from $(k, \theta_k)$ to $(l, \theta_l)$ generate random numbers $u_k \sim g_k(\cdot)$ and set $(l, \theta_l) = f_{kl}(k, \theta_k, u_k)$. It is essential that $f_{lk}$ and $f_{kl}$ are bijections with invertible differentials. A necessary condition for this is the dimension matching condition, $\dim(\theta_l) + \dim(u_l) = \dim(\theta_k) + \dim(u_k)$. The acceptance probability of the move from $(l, \theta_l)$ to $(k, \theta_k)$ will be the minimum of 1 and

$$\frac{\pi(\theta_k|y, \mathcal{M}_k)\pi(\mathcal{M}_k)P(k, l)g_k(u_k)}{\pi(\theta_l|y, \mathcal{M}_l)\pi(\mathcal{M}_l)P(l, k)g_l(u_l)} \left| \frac{\partial(k, \theta_k, u_k)}{\partial(l, \theta_l, u_l)} \right|$$

where the determinant is of the Jacobian for the transformation from $(l, \theta_l, u_l)$ to $(k, \theta_k, u_k)$ and $P(k, l)$ is the probability of choosing a move to model $l$ from model $k$.

The PMP $\pi(\mathcal{M}_l|y)$ may be estimated from the RJMCMC output by

$$\pi(\mathcal{M}_l|y) \approx \frac{1}{N} \sum_{t=1}^{N} I(\text{model at iteration } t \text{ is } \mathcal{M}_l)$$

where $t$ indexes the $N$ (post burn-in) RJMCMC iterates and $I(\cdot)$ is the indicator function. Clearly the accuracy of the approximation will be affected by the convergence, mixing and general quality of the RJMCMC output. An added difficulty to tuning the proposal parameters in a RJMCMC algorithm is that one must also tune parameters for switching model moves. This is to ensure the entire space $\mathcal{X}$ is being represented in roughly the correct proportions. Poor tuning will not only lead to poor estimation
of the posterior marginal densities within a given model, but also to poor estimation of the PMPs. Some work has been done on tuning and the design of reversible jump proposals (Brooks, Giudici & Roberts 2003, Green 2003). Despite this, tuning proposal parameters is usually case specific in all but the most simple models. This is usually the main drawback in the implementation of RJ algorithms.

1.4.4 A case study in model selection: Gaussian finite mixtures with an unknown number of components

Assuming data \( y = (y_1, \ldots, y_n) \) as independent draws from the finite mixture model in Section 1.3.1 with \( K \) components, where the observed data density is Gaussian, the likelihood is

\[
\pi(y|K, \omega, \Theta) = \prod_{i=1}^{n} \left( \sum_{k=1}^{K} \omega_k \phi\left(y_i|\mu_k, \sigma_k^2\right) \right)
\]

where \( \phi(y|\mu, \sigma^2) \) is the density of a \( N(\mu, \sigma^2) \) distribution evaluated at \( y \). The likelihood for \( y \) may be written conditional on the values of the latent labels \( z = (z_1, \ldots, z_n) \) as

\[
\pi(y|K, z, \omega, \Theta) = \prod_{k=1}^{K} \prod_{i: z_i = k} \omega_k \phi\left(y_i|\mu_k, \sigma_k^2\right).
\]

Marginalizing \( z \) out of this expression by summing over all possible labellings of the data returns the expression for the likelihood above. Take independent priors \( \pi(\mu_k, \sigma_k^2), \) \( k = 1, \ldots, K \) and \( \omega \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K) \) to obtain the joint posterior of labels and parameters conditional on \( K \). Now, consider \( K \) as unknown and give it prior \( \pi(K) \). Then the joint posterior of interest on \( K, z, \omega \) and \( \Theta \) is

\[
\pi(K, z, \omega, \Theta|y) \propto \pi(K)\pi(\omega|K)\pi(z|K, \omega) \prod_{k=1}^{K} \pi(\mu_k, \sigma_k^2) \prod_{k=1}^{K} \prod_{i: z_i = k} \omega_k \phi\left(y_i|\mu_k, \sigma_k^2\right).
\]

If there can be at most \( \tilde{K} \) components, the number of potential ways in which the \( n \) data points can be labelled is \( \sum_{k=1}^{\tilde{K}} k^n \), hence this is a difficult model search task. Two ways to proceed with inference based on this posterior distribution are now compared. The first is due to Richardson & Green (1997) and is based on RJMCMC and the second, the allocation sampler, was proposed by Nobile & Fearnside (2007).
Reversible jump MCMC approach

Conditional on a particular number of components $K$, the parameters $z, \omega$ and $\Theta$ may be updated using standard Gibbs draws as outlined in Section 1.3.1. Reversible jump moves are needed when changing the value of $K$, that is, when adding or removing components. Richardson & Green (1997) propose two types of move in this regard. The first of these involves, splitting or combining some of the current components. In the split move, a component is selected and split into two separate components. The combine move is the reverse of the split move, where two components are selected at random and combined into one. Suppose components $k_1$ and $k_2$ are being combined into a new component $k^*$. All $y_i$ with label $k_1$ or $k_2$ are re-labelled $k^*$. The requirement of a bijection between the model spaces corresponding to $K$ and $K - 1$ components means that the parameters for the new component $k^*$ must be some invertible one-to-one function of the parameters of the current components $k_1, k_2$ (and possibly some additional random variables, but not here). Richardson & Green (1997) suggest matching the zeroth, first and second order moments of the new component $k^*$ with $k_1, k_2$. This is obtained by setting,

$$
\begin{align*}
\omega_{k^*} &= \omega_{k_1} + \omega_{k_2} \\
\omega_{k^*}\mu_{k^*} &= \omega_{k_1}\mu_{k_1} + \omega_{k_2}\mu_{k_2} \\
\omega_{k^*}(\mu_{k^*}^2 + \sigma_{k^*}^2) &= \omega_{k_1}(\mu_{k_1}^2 + \sigma_{k_1}^2) + \omega_{k_2}(\mu_{k_2}^2 + \sigma_{k_2}^2).
\end{align*}
$$

(1.8)

Notice that this combine move is deterministic, since there is no stochastic generation involved in either the reallocation of observations to $k^*$ or the construction of the new parameters $\omega_{k^*}, \mu_{k^*}, \sigma_{k^*}^2$. For the reverse of this move, the component $k^*$ is split into the components $k_1$ and $k_2$. Since three parameters were lost in the deterministic combine move (a weight, a mean and a variance), it will be necessary to simulate three random variables $u_1, u_2, u_3$ to go from component $k^*$ to $k_1$ and $k_2$. This is in order to maintain
detailed balance and reversibility. The inverse of (1.8) is obtained by setting

\[
\begin{align*}
\omega_{k_1} &= \omega_{k^*} u_1 & \omega_{k_2} &= \omega_{k^*} (1 - u_1) \\
\mu_{k_1} &= \mu_{k^*} - u_2 \sigma_{k^*} \sqrt{\omega_{k_2} / \omega_{k_1}} \\
\mu_{k_2} &= \mu_{k^*} + u_2 \sigma_{k^*} \sqrt{\omega_{k_1} / \omega_{k_2}} \\
\sigma_{k_1}^2 &= u_3 (1 - u_2^2) \sigma_{k^*}^2 \omega_{k^*} / \omega_{k_1} \\
\sigma_{k_2}^2 &= (1 - u_3) (1 - u_2^2) \sigma_{k^*}^2 \omega_{k^*} / \omega_{k_2}.
\end{align*}
\] (1.9)

Members of component $k^*$ are reallocated to $k_1$ or $k_2$ by drawing from their Gibbs full conditional distribution on these components,

\[
\pi(z_i = k | \cdots) \propto \frac{\omega_k}{\sigma_k} \exp \left\{ -\frac{1}{2\sigma_k^2} (y_i - \mu_k)^2 \right\}, \quad k = k_1, k_2.
\]

Such an elaborate move is necessary to satisfy the dimension matching requirements and is a perfect example of how difficult it can be to design sensible jump moves in RJMCMC applications. In the case of mixtures, an identifiability constraint is also required as a measure to avoid the label switching problem. For the RJMCMC algorithm here, Richardson & Green (1997) suggest the constraint $\mu_k < \mu_l$ for all $k < l$. This means that for the split move above, if after computing the quantities (1.9), $\mu_{\min(k_1,k_2)} > \mu_{\max(k_1,k_2)}$, then the move is rejected outright. Such moves do not follow intuitively, and are achievements and new inventions in themselves. In this regard, reversible jump techniques can be quite difficult to implement. It will rarely be obvious how to obtain dimension matching in a clever enough way, so that proposed points in the new subspace(s) will be in an area of high enough support. Proposing poor points leads to a poorly mixing RJMCMC algorithm.

The second type of move on the number of components considered by Richardson & Green (1997) is to add or remove an empty component. This is slightly more straightforward, but still requires close attention to parameters and dimensions to ensure these moves satisfy detailed balance and reversibility.
The allocation sampler

The allocation sampler of Nobile & Fearnside (2007) also has capacity to allow for uncertainty in $K$, but does not require such difficult jump dynamics as outlined for RJMCMC above. The idea of the allocation sampler is that supposing the labels $z$ and the number of components $K$ are really the items of interest (as they will be in many mixture applications, see for example Raftery (1996)), the parameters $\omega$ and $\mu_k, \sigma^2_k, k = 1, \ldots, K$ can be viewed as nuisance parameters. By assuming a fully conjugate prior for each of these, they may be analytically integrated out from the posterior, to leave a joint posterior for $K$ and $z$. To demonstrate this, consider the full posterior

$$
\pi(K, z, \omega, \Theta|y) \propto \pi(K) \pi(\omega|K) \prod_{k=1}^K \pi(\mu_k, \sigma^2_k) \prod_{i: z_i = k} \omega_k \phi(y_i|\mu_k, \sigma^2_k)
$$

where $n_k$ is the number of observations with label $k$. The joint posterior of $K$ and $z$ may be obtained by integrating out $\omega$ and $\Theta$ from this posterior;

$$
\pi(K, z|y) = \int \pi(K, z, \omega, \Theta|y) \, d\omega \, d\Theta.
$$

This can be obtained up to a constant of proportionality that does not depend on $K$ or $z$ by computing the necessary integrals.

$$
\pi(K, z|y) \propto \int \pi(K) \frac{\Gamma \left\{ \sum_{k=1}^K \alpha_k \right\}}{\prod_{k=1}^K \Gamma \{\alpha_k\}} \prod_{k=1}^K \omega_k^{n_k + \alpha_k - 1} \prod_{k=1}^K \pi(\mu_k, \sigma^2_k) \prod_{i: z_i = k} \phi(y_i|\mu_k, \sigma^2_k) \, d\omega \, d\Theta
$$
\[ \pi(K) \frac{\Gamma \left\{ \sum_{k=1}^{K} \alpha_k \right\} \prod_{k=1}^{K} \Gamma \left\{ n_k + \alpha_k \right\}}{\prod_{k=1}^{K} \Gamma \{ \alpha_k \} \Gamma \left\{ n + \sum_{k=1}^{K} \alpha_k \right\}} \times \prod_{k=1}^{K} \int \pi(\mu_k, \sigma^2_k) \prod_{i:z_i=k} \phi(y_i|\mu_k, \sigma^2_k) \, d\mu_k \, d\sigma^2_k. \]

This gives

\[ \pi(K, z|y) \propto \pi(K) \frac{\Gamma \left\{ \sum_{k=1}^{K} \alpha_k \right\} \prod_{k=1}^{K} \Gamma \left\{ n_k + \alpha_k \right\}}{\prod_{k=1}^{K} \Gamma \{ \alpha_k \} \Gamma \left\{ n + \sum_{k=1}^{K} \alpha_k \right\}} \prod_{k=1}^{K} M_k \]  

(1.10)

where \( M_k \) is

\[ M_k = \int \pi(\mu_k, \sigma^2_k) \prod_{i:z_i=k} \phi(y_i|\mu_k, \sigma^2_k) \, d\mu_k \, d\sigma^2_k. \]

It will be possible to calculate \( M_k \) explicitly because of the conjugacy assumptions. In this case, if the assumed priors are of the form \( \pi(\mu_k, \sigma^2_k) = \pi(\mu_k|\sigma^2_k) \pi(\sigma^2_k) \) where

\[ \mu_k|\sigma^2_k \sim N(\xi_k, \sigma^2_k/\eta_k) \]

\[ \sigma^2_k \sim \text{Gamma}(\gamma_k/2, \lambda_k/2), \]

then

\[ M_k = \pi^{-n_k/2} \lambda_k^{\gamma_k/2} \sqrt{\frac{\eta_k}{n_k + \eta_k}} \frac{\Gamma \{ (n_k + \gamma_k)/2 \}}{\Gamma \{ \gamma_k/2 \}} \times \left\{ ss_k + \eta_k \xi_k^2 - (n_k + \eta_k)^{-1} (s_k + \eta_k \xi_k)^2 + \lambda_k \right\}^{-(n_k + \gamma_k)/2} \]

where \( ss_k = \sum_{i:z_i=k} y_i^2 \) and \( s_k = \sum_{i:z_i=k} y_i \). The conditional prior of \( \mu_k \) on \( \sigma^2_k \) may seem restrictive, however, this is an intuitive prior in some sense, since this specifies that prior information on \( \mu_k \) is the same as that obtained from \( \eta_k \) data points. So \( \eta_k < 1 \) would correspond to a prior for \( \mu_k \) which is more diffuse than the data itself. For example, setting \( \eta_k = 1/r^2 \) for \( k = 1, \ldots, K \) would say that the standard deviation of group means equals \( r \) times the data standard deviation. Choosing a large value of \( r \) would imply prior belief that the component means are very far apart (\( r \) standard deviations), while small \( r \) implies that they are close together, \textit{a priori}.

The posterior (1.10) is amenable to Gibbs sampling for updates of the labels for a particular number of components \( K \). A similar situation will be discussed in detail.
in Chapter 3. To add or remove components, termed “eject and absorb” in Nobile & Fearnside (2007), is more straightforward than the RJ case. Suppose a component \( k_1 \) is ejecting a new component \( k_2 \). Then a draw \( p_E \) from a Beta\((a, a)\) distribution is made, and those observations labelled \( k_1 \) are moved to component \( k_2 \) with probability \( p_E \). For the reverse absorb move, all the data in component \( k_2 \) is just placed into component \( k_1 \). As noted in Nobile & Fearnside (2007), the parameter \( a \) can have a crucial effect on the performance of the allocation sampler, however, choosing its value so that empty components are proposed often appears to perform satisfactorily. The values of \( a \) can be computed before analysis and called when needed. Notice that in this case, the moves change \( K \), but there is no need to design the bijective function between model spaces with \( K \) and \( K - 1 \) components. This leads to a simplified algorithm implementation, which still obtains the results of interest, namely; to what group the observations belong; and how many groups there are. Also, an identifiability constraint on parameters is not required. Label switching, if it occurs, may be dealt with efficiently using an algorithm based on the square assignment problem (see Section 3.3.3 for more details).

**Conclusion**

When it is possible to implement, an approach along the lines of Nobile & Fearnside (2007) should be implemented over RJMCMC once inference for the model parameters is paramount i.e. in situations where the latent variables are the quantities that are of most interest, the prior conjugacy assumptions required are not overly restrictive for the application in hand and the necessary analytic marginalization of parameters is straightforward to perform. The approach is more intuitive than the design moves required for RJMCMC algorithms. There will also be computational savings since sampling of the model parameters is avoided. The need to run the algorithm until there is joint convergence (to the target distribution) of a larger number of sampled quantities is also improved upon. There should be quicker convergence to the target distribution, since the posterior support shrinks when nuisance parameters are integrated from the
model.

1.5 Thesis outline

In Chapter 2 a reversible jump free Monte Carlo sampler is proposed for sampling in multiple changepoint problems where the number of changepoints is unknown. This is motivated by the results of the case study above (Section 1.4.4). The MCMC changepoint sampler is compared with analysis of changepoint problems based on filtering recursions (Fearnhead 2006), which have become popular methods of late. Strengths and weaknesses of both approaches are discussed in the context of applications to real datasets. The datasets analyzed include a series of Tiger Woods’ championship win records, records of coal-mining disasters in the UK over 112 years, and values of the nuclear magnetic response of rock strata when drilling for oil. Each of these datasets requires a different distributional assumption for the data, demonstrating application of the sampler in three different situations. Overall, it is found that the MCMC sampler could be more robust than approaches based on filtering recursions in situations where little prior information is available. It is shown that the sampler can be made fully automatic for particular model assumptions.

Also motivated by the case study, Chapter 3 applies the same ideas in a much more challenging situation, that of latent block modelling. A Monte Carlo algorithm which searches simultaneously over model space and allocation space has not previously been applied for this particular class of models. The algorithm proposed searches probabilistically over the number of row and column components when blocking a data matrix. In some ways, the approach can be thought of as a generalization of the allocation sampler of Nobile & Fearnside (2007) to two directions, although not identical. The sampler is tested on some simulated data. Following this, the sampler is compared to a maximum likelihood approach based on Expectation-Maximization developed specifically for latent block models by Govaert & Nadif (2008). The comparison is made through application to a dataset recording voting records of members of the 98th US
congress on a series of important issues. The sampler is also applied to a microarray data set arising from experiments on the budding yeast *Saccharomyces Cerevisiae*.

Chapter 4 returns to the theme of multiple changepoint models. The focus of this chapter is on providing approximate methods for models where the assumption of conditional independence of data within segments is relaxed by assuming hierarchical Gaussian Markov random field models. This opens up the opportunity for many new changepoint models including; non-homogeneous poisson processes with data dependency over time and the possibility of large sudden changes in the intensity; auto-regressive dependency in normal data where the auto-regressive relationship may vary over segments; and stochastic volatility models where there can be changepoints in the latent volatilities over time. A way to perform approximate inference with a lower computational overhead than filtering recursions, called reduced filtering recursions (RFRs), is proposed. The new methods are applied to genomic cell line data while the coal mining disasters data and Well-log data are re-analyzed with these new more realistic models. The methods are also applied to simulated stochastic volatility data with changepoints.

The thesis concludes with a broad discussion of the outcomes of the work undertaken, and ideas of how this work could potentially be extended in new directions. These include a model selection algorithm for hidden Markov models with an unknown number of states, which offers a simple alternative to the reversible jump approach of Robert, Rydén & Titterington (2000).
Chapter 2

Simulation based Bayesian analysis of multiple changepoint models

2.1 Introduction

The range of applications of changepoint models is evident from the substantial volume of literature devoted to this problem in the econometrics, signal processing and bioinformatics literatures. A process generating data can often undergo changes over time such that one model will not be appropriate for all time periods. Here “time” refers to some natural sequential indexing of the data. Some examples are occurrences of coal mining disasters during the 18th and 19th century (Raftery & Akman 1986), DNA or protein composition analysis over base number (Liu & Lawrence 1999) and winning streaks in sports (Yang 2004).

Markov chain Monte Carlo (MCMC) techniques can be used to estimate models with a fixed number of changepoints. When the number of changepoints is unknown, inference is more challenging. Chib (1998) estimates a collection of changepoint models and compares these using Bayes factors estimated from the MCMC output. Green (1995) uses reversible jump MCMC (RJMCMC) to explore the number of changepoints in the coal mining disaster data. RJMCMC allows moves between models which satisfy detailed balance.
The use of alternatives to MCMC has grown in this area in recent years. Fearnhead (2006) uses filtering recursions to derive the posterior distribution of changepoints. This can be done for both a known and unknown number of changepoints. An advantage of this approach is that one can draw independent samples from the posterior. MCMC can only do this approximately at best. Extension to online analysis of changepoint models is also possible (Fearnhead & Liu 2007). However methods based on filtering recursions rely on strong prior information in most cases. This paper aims to offer an efficient MCMC alternative which can overcome strong reliance on prior assumptions as encountered in recursive computing approaches. The class of models considered is similar to Fearnhead (2006). For this reason it is possible that this could be used to give useful starting values for an analysis using filtering recursions.

Qualitatively, the work in this paper is similar in some aspects to work by Lavielle & Lebarbier (2001) and Punska, Andrieu, Doucet & Fitzgerald (2002) in terms of the class of models considered. The sampling aspect of the approach bears similarities to the samplers of Lavielle & Lebarbier (2001) and Girón, Moreno & Casella (2007). This paper extends these works to a broader range of data models and proposes a more efficient way of sampling changepoints. An aim is also to highlight possible shortcomings of alternatives to MCMC and how these could be overcome by using simulation approaches to inform choices for recursive computing approaches.

The remainder of the chapter is organised as follows. In Section 2.2 the type of changepoint model under consideration is presented. Section 2.3 reviews the reversible jump approach to changepoint estimation and discusses how this can be simplified into a fixed dimensional sampling scheme. Section 2.4 gives the moves to sample from the simpler fixed dimensional posterior. Prior specification is discussed in Section 2.5 and Section 2.6 reviews the filtering recursion approach to generating samples of changepoints. Performance of the sampler is validated by analyzing the coal mining disasters data in Section 2.7 while Sections 2.8 and 2.9 compare qualitative aspects of the simulation based sampler approach and filtering recursions approach using two real data examples. A brief discussion concludes the chapter.
2.2 Changepoint models

Consider the data $y_1: n = (y_1, \ldots, y_n)$ which is time ordered. Here $y_i$ is observed before $y_j$ if $i < j$. Time in this context can refer to any natural ordering of the data as it is observed. A changepoint occurs at time $t$ if $y_1, \ldots, y_t$ are generated differently to $y_{t+1}, \ldots, y_n$. Referring to $y_{s:r} (s < r)$ as a segment, this says that the segments $y_{1:t}$ and $y_{t+1:n}$ are heterogeneous between but homogeneous within. Parametric changepoint models assign a different parameter for each segment to account for this heterogeneity.

This paper considers multiple changepoints which will be denoted $\tau_1, \ldots, \tau_k$. These split the data into $k + 1$ segments. The likelihood for segment $j$ has parameter $\theta_j$. Conditional on a segmentation, the data within each segment is assumed independent. It is also assumed that the regime parameters $\theta_j$ are independent. The likelihood of the segmentation $\tau = (\tau_1, \ldots, \tau_k)$ is

$$\prod_{j=1}^{k+1} \prod_{i=\tau_{j-1}+1}^{\tau_j} \pi(y_i|\theta_j)$$

where for convenience $\tau_0 = 0, \tau_{k+1} = n$. Instead of using $\tau$, segmentations can be labelled with the binary latent vector $z = (z_1, \ldots, z_n)$ with $z_t = 1$ indicating a changepoint at time $t$ and $z_n = 0$. Independent priors are assumed for each member of $\theta = (\theta_1, \ldots, \theta_{k+1})$ with hyperparameter $\gamma$ and there is a prior for the changepoints with hyperparameter $\xi$, given by $\pi(z|k, \xi)$. The posterior may be written

$$\pi(z, \theta|y, k, \xi, \gamma) \propto \pi(z|k, \xi)\pi(\theta|k, \gamma)\pi(y|\theta, z, k)$$

where the dependence on the number of changepoints, $k$, is made explicit. A prior $\pi(k)$ may be introduced so that the posterior of interest is the joint posterior of $(k; z, \theta)$,

$$\pi(k; z, \theta|y, \xi, \gamma) \propto \pi(k)\pi(z, \theta|y, k, \xi, \gamma). \quad (2.1)$$

This is a hierarchical changepoint model similar to that used in Green (1995).
2.3 Collapsing changepoint models

It is possible to construct a MCMC scheme to sample the posterior of (2.1) using RJMCMC (Green 1995). The sampler will explore the product space support of this posterior:

\[ X = \prod_k \{k\} \times \{z, \Theta_k|k\} \]

where \( Z_k, \Theta_k \) are respectively the sample spaces of \( z \) and \( \theta \) conditional on \( k \) changepoints. A switch in the number of changepoints in the model can be made by a RJ move switching between support subspaces. For the purposes of illustration a straightforward move of this type is now discussed. When proposing a switch from \( k \) to \( k + 1 \) changepoints one possibility is to generate a random variable \( u \in \mathbb{R}^d \) and form a bijection \( f : \Theta_k \times \mathbb{R}^d \rightarrow \Theta_{k+1} \) where \( d \) is the dimension of a single \( \theta_j \). This bijection gives the parameters for the proposed \( k + 1 \) changepoint model as a function of those for the \( k \) changepoint model; \( \theta' = (\theta'_1, \ldots, \theta'_{k+2}) = f(\theta_1, \ldots, \theta_{k+1}, u) \). The proposed switch in model is then accepted with probability \( \min(1, R) \) where

\[
R = \frac{\pi(k + 1, z', \theta' | y, \xi, \gamma) P(k + 1, k)}{\pi(k, z, \theta | y, \xi, \gamma) P(k, k + 1)} \frac{1}{q(u|\theta)} \left| \frac{\partial(\theta')}{\partial(\theta, u)} \right|.
\]

In the expression for \( R \), \( P(\cdot, \cdot) \) denotes the proposal probability for transitions between different numbers of changepoints, and \( q(\cdot|\theta) \) is the proposal density of \( u \). The last term on the right is a Jacobian term for the bijection \( f \). The reverse move in switching from \( k + 1 \) to \( k \) changepoints is accepted with probability \( \min(1, R^{-1}) \). More elaborate moves between support subspaces are possible which propose changes of more than one dimension or involve stochastic moves in both directions.

The key questions in a changepoint analysis are usually; how many changepoints are there and where are the changepoints? The segment parameters \( \theta \) can be viewed as a nuisance parameter in this regard. Choosing conjugate priors for the \( \theta_j \) allows these
to be collapsed in the model

$$\pi(k, z|y, \xi, \gamma) \propto \pi(k) \pi(z|k, \xi) \prod_{j=1}^{k+1} \int \pi(\theta_j|\gamma) \prod_{i=\tau_{j-1}+1}^{\tau_j} \pi(y_i|\theta_j) \, d\theta_j$$

$$= \pi(k) \pi(z|k, \xi) \prod_{j=1}^{k+1} \pi(y_{\tau_{j-1}+1:\tau_j}|\gamma), \quad (2.2)$$

where $\pi(y_{\tau_{j-1}+1:\tau_j}|\gamma)$ is the marginal likelihood of the data segment $y_{\tau_{j-1}+1:\tau_j}$ and is assumed to be available in closed form due to the conjugacy. The support of this posterior is

$$\mathcal{Y} = \prod_k \{k\} \times \{Z_k|k\}$$

and a switch from $k$ to $k+1$ changepoints does not require the design of a bijective function between support subspaces. The proposed switch in model is now accepted with Metropolis-Hastings probability $\min(1, A)$ where

$$A = \frac{\pi(k+1, z'|y, \xi, \gamma) P(k+1, k)}{\pi(k, z|y, \xi, \gamma) P(k, k+1)}. \quad (2.3)$$

This idea of collapsing has been used previously in Punskaya et al. (2002) and Lavielle & Lebarbier (2001) for Gaussian data models.

It can be seen that the first term on the right hand side of the acceptance ratio (2.3) is the Bayes factor for a model with $k+1$ changepoints at positions $z'$ versus a model with $k$ changepoints at positions $z$, assuming all models are equally likely, $a \ priori$. Noting this, it becomes apparent that sampling $k$ and $z$ is equivalent to a model search over a large model space. If there can be at most $\bar{k}$ changepoints, then the dimension of this space is $\sum_{k=0}^{\bar{k}} \binom{n-1}{k}$. So searching for up to 5 changepoints in a dataset of length 200 corresponds to a dimension $\sim 2.5 \times 10^9$. In the next section an MCMC scheme to search over these large model spaces, that is, sample from the posterior (2.2), is proposed.
2.4 Sampling changepoints

The MCMC scheme to generate samples of changepoints from the posterior (2.2) consists of three possible moves: add a changepoint; delete a changepoint; move a changepoint. Each sweep consists of the following:

- Choose to add or delete a changepoint with probabilities \(a_k\) and \(d_k = 1 - a_k\) respectively. Clearly \(a_k = d_0 = 0\).

- Select a changepoint and propose to move it to a position in the range of its closest neighbouring changepoints.

Add or delete a changepoint

This move has been discussed in Section 2.3 but more details are given here. Suppose there are currently \(k\) changepoints at positions \(z\). Let \(z\) correspond to changepoints at \(\tau_1, \ldots, \tau_k\). Randomly select one of the \(n - k - 1\) points where there could be a changepoint i.e. a \(t < n\) with \(z_t = 0\). Say this is currently in segment \(j\) given by \(y_{\tau_j - 1}^{\tau_j}\). Relabel the proposed changepoints in \(z'\) as \(\tau'_1, \ldots, \tau'_{k+1}\) with \(\tau'_{j+1} = t\). Cancellation of marginal likelihood terms then implies that

\[
\frac{\pi(k+1, z'|y, \xi, \gamma)}{\pi(k, z|y, \xi, \gamma)} = \frac{\pi(k+1) \pi(z'|k+1, \xi)}{\pi(k) \pi(z|k, \xi)} \frac{\pi(y_{\tau^*_j-1+1: \tau^*_j} | \gamma)}{\pi(y_{\tau_j-1+1: \tau_j} | \gamma)}
\]

so calculation of \(A\) in (2.3) only requires at most three marginal likelihood values.

Conversely, for the delete move, one of the \(k+1\) changepoints in \(z'\) is chosen at random and the calculation of the acceptance probability involves

\[
\frac{\pi(k, z|y, \xi, \gamma)}{\pi(k+1, z'|y, \xi, \gamma)} = \frac{\pi(k)}{\pi(k+1)} \frac{\pi(z|k, \xi)}{\pi(z'|k+1, \xi)} \frac{\pi(y_{\tau^*_j-1+1: \tau^*_j} | \gamma)}{\pi(y_{\tau_j-1+1: \tau_j} | \gamma)} \frac{\pi(y_{\tau^*_j+1: \tau^*_j+1} | \gamma)}{\pi(y_{\tau_j+1: \tau_j+1} | \gamma)}
\]

Finally, the proposal one step transition probabilities for the number of changepoints will be \(P(k, k+1) = a_k/(n - k - 1)\) and \(P(k+1, k) = d_{k+1}/(k + 1)\), so that \(A\) (2.3) can be computed. The acceptance probability for the add move is then \(\min(1, A)\) and the delete move is accepted with probability \(\min(1, A^{-1})\).
Move a changepoint

**Gibbs update:** Given the model assumption that the marginal likelihood for any segment is available in closed form, it is possible to update the position of any changepoint from its full conditional. Suppose $\tau_j$ is being updated. Then the conditional probability that $\tau_j = t, \tau_{j-1} < t < \tau_{j+1}$ is proportional to

$$\pi(z'_t | k) \pi(y_{\tau_{j-1}+1:t} | \gamma) \pi(y_{t+1:\tau_{j+1}} | \gamma)$$

where $z'_t$ corresponds to changepoints $\tau_1, \ldots, \tau_{j-1}, t, \tau_{j+1}, \ldots, \tau_k$. The effort required for the Gibbs update is $O(\tau_{j+1} - \tau_{j-1})$ and so may be computationally expensive for large datasets with changepoints far apart, or datasets with many changepoints. In this situation a local random walk update may be preferred.

**Local random walk update:** $t$ is drawn uniformly from the integers $\max(\tau_{j-1} - l, \tau_{j-1} + 1), \ldots, \min(\tau_{j+1} - l, \tau_{j+1} + 1)$ where $l$ specifies the locality of the proposed move. The move is accepted with probability $\min(1, B)$ where

$$B = \frac{\pi(y_{\tau_{j-1}+1:t} | \gamma) \pi(y_{t+1:\tau_{j+1}} | \gamma)}{\pi(y_{\tau_{j-1}+1:\tau_{j}} | \gamma) \pi(y_{\tau_{j}+1:\tau_{j+1}} | \gamma)}.$$

In the event that $\tau_j - l \leq \tau_{j-1}$ and $t < \tau_j$, $B$ must be multiplied by $(\tau_j - \tau_{j-1} + l)/(t - \tau_{j-1} + l)$. Similar modifications are needed if $t > \tau_j$ or $\tau_j + l \geq \tau_{j+1}$.

**Mixture of updates:** A mixture of the two moves above should improve mixing and not be overly computationally expensive. For example, choose the Gibbs update with probability $g_k = 1/\sqrt{k}$ ($k \geq 1$) and random walk with probability $r_k = 1 - g_k$.

### 2.5 Prior specification

There are many possible choices for $\pi(z | k, \xi)$. Yao (1984) considers a geometric distribution for the duration, $d$, of segments; $d \sim \text{Geometric}(p)$. The prior used by Green (1995) has been adapted by Fearnhead (2006) for the discrete time context discussed here. The $k$ changepoint locations are distributed as the even numbered order statistics in a sample of size $2k + 1$ from the integers $1, \ldots, n - 1$, drawn without replacement.
The geometric prior relies on specification of $\xi = p$. Ideally, one could simulate a segment specific $p_j$ in a similar vein to Chib (1998). However this leads to more difficult jump dynamics when adding or deleting a changepoint. The choice of $p$ may impact the analysis. If too small, then it will assign very small probability to changepoints, meaning small changes cannot be detected with high power. If too large, then spurious changepoints are inferred. For these reasons, it desireable to introduce a hyperprior on $p$. For example, a Beta($\alpha_1, \alpha_2$) prior with $1 < \alpha_1 < \alpha_2$ (more weight less than 0.5), would be an ideal choice if there is enough prior information to choose $\alpha_1, \alpha_2$. Otherwise, a non-informative Beta(1, 1) prior would suffice.

Segment parameters share a common hyperparameter $\gamma$ in Section 2.2. It is therefore possible to explore uncertainty in $\gamma$ also by introducing a hyperprior $\pi(\gamma)$.

Sampling $p$ and $\gamma$ can be easily incorporated into the MCMC scheme in Section 2.3. One sweep of the algorithm consists of:

1. Sample the changepoints.
3. Conditional on the changepoints sample $\theta$.
4. Conditional on $\theta$ sample $\gamma$ and discard the $\theta$ values.

For the last step here, it will often be possible to sample $\gamma$ using a Gibbs step. However, if this is not possible, a simple random walk Metropolis-Hastings could be used.

### 2.6 Analysis by filtering recursions

It is useful to give a brief recap of the filtering recursions analysis of Fearnhead (2006) based on a point process prior for changepoint positions. Liu & Lawrence (1999), Barry & Hartigan (1992) have also used these types of methods for the analysis of changepoint problems. Define

$$L_\gamma(t) = \Pr\{y_{t:n}|\text{changepoint at } t-1, \gamma\}.$$
It is possible to compute this quantity in a backward recursion. Defining $L_\gamma(n) = \pi(y_n|\gamma)$, for $t = n-1, \ldots, 2$

$$L_\gamma(t) = \sum_{s=t}^{n-1} \pi(y_{t:s}|\gamma)L_\gamma(s+1)g(s-t+1) + \pi(y_{t:n}|\gamma)(1 - G(n-t+1))$$

and

$$L_\gamma(1) = \sum_{s=1}^{n-1} \pi(y_{1:s}|\gamma)L_\gamma(s+1)g_0(s) + \pi(y_{1:n}|\gamma)(1 - G_0(n-1))$$

where the dependence of $L_\gamma(t)$ on the hyperparameter $\gamma$ has been made explicit. Here $g(\cdot)$ gives the point process for the changepoint positions and $G(\cdot)$ the corresponding cumulative distribution function (the subscript 0 on $g$ and $G$ in $L_\gamma(1)$ denotes the distribution of the first changepoint after 0). Yao (1984) takes this as geometric as do Barry & Hartigan (1992). Fearnhead (2006) suggests a negative binomial family in general for this process.

After computing the recursions, a sample of size $N$ of the changepoints can be efficiently simulated as follows:

1. Initialize all samples to have a changepoint at $t = 0$.

2. For $t = 0, \ldots, n-2$:
   
   (a) Get $n_t$, the number of samples for which the last changepoint was at time $t$.

   (b) If $n_t > 0$ compute the distribution of the next changepoint:

   $$\Pr\{\tau|y_{1:n}, t\} = \frac{\pi(y_{t+1:}\tau|\gamma)L_\gamma(\tau + 1)g(\tau-t)/L_\gamma(t+1)}{\pi(y_{t+1:}\tau|\gamma)L_\gamma(\tau + 1)g(\tau-t)/L_\gamma(t+1)}$$

   (c) Sample $n_t$ times from $\Pr\{\tau|y_{1:n}, t\}$ and update the $n_t$ samples that have the last changepoint at $t$.

There are two strengths of this approach. The first is that the samples of changepoints will be independent draws from the posterior distribution. The second is the fast sampling algorithm which avoids computing the distribution of the next changepoint for each possible time. The main weakness of this approach is that the generated
samples are dependent on a fixed value of the hyperparameters $\gamma$. Updating $\gamma$ using a hyperprior to correctly explore uncertainty in the value would involve recomputing the recursions $L_\gamma(t)$ for each new value of $\gamma$, a computation which is quadratic in $n$. This would lead to an infeasible computational overhead for any reasonably large sample from the posterior.

### 2.7 Poisson data: coal mining disasters

The sampler of Section 2.4 was applied to the coal-mining data of Jarrett (1979). The data records the dates of serious coal-mining disasters between 1851 and 1962. Disasters are assumed to arise from a Poisson process whose intensity is the height of a step function with an unknown number of steps. Given a segment $y_{s:t}$, each $y_t \sim \text{Poisson}(\mu)$ where $\mu$ is the height of the step function between times $s$ and $t$. Assume the prior for $\mu$ is Gamma$(\rho, \lambda)$ where $\gamma = (\rho, \lambda)$. The marginal likelihood for the segment is then

$$
\pi(y_{s:t} | \gamma) = \int_{0}^{\infty} \frac{\lambda^\rho}{\Gamma(\rho)} \mu^{\rho-1} \exp\{-\lambda \mu\} \prod_{i=s}^{t} \frac{\mu^{y_i}}{y_i!} \exp\{-\mu\} \, d\mu
$$

$$
= \frac{\lambda^\rho}{\Gamma(\rho)} \int_{0}^{\infty} \frac{1}{F_{s:t}} \mu^{S_{s:t}+\rho-1} \exp\{- (t - s + \lambda + 1) \mu\} \, d\mu
$$

where $F_{s:t} = \prod_{i=s}^{t} y_i$ and $S_{s:t} = \sum_{i=s}^{t} y_i$. Completing the integral of the Gamma density gives

$$
\pi(y_{s:t} | \gamma) = \frac{\lambda^\rho}{\Gamma(\rho)} \frac{1}{F_{s:t}} \frac{\Gamma\{S_{s:t} + \rho\}}{(t - s + \lambda + 1)^{S_{s:t}+\rho}}
$$

For comparison with Fearnhead (2006), time is discretized in weeks and the hyperparameters are set to be $\rho = 1$ and $\lambda = 200/7$ a priori. Conditional on $k$ change-points, the prior on their positions was taken to be the same as the distribution of even numbered order statistics of a sample of size $2k + 1$ drawn without replacement from $\{1, \ldots, n - 1\}$ (Fearnhead 2006),

$$
\pi(\tau_1, \ldots, \tau_k | k) = \frac{n - 1}{2k + 1} \prod_{j=0}^{k} (\tau_{j+1} - \tau_j - 1).
$$
See Appendix A for a derivation of this prior.

The MCMC algorithm was run for 500,000 sweeps after 10,000 burn in. Every 50th sample was taken to reduce dependency in the MCMC iterates. This took 10 seconds on a 2.5GHz processor. Figure 2.1(a) shows that the posterior number of changepoints is almost identical to that obtained from long runs of a RJMCMC sampler and methods based on recursions (see Fearnhead (2006), Figure 2.1(a)).

2.8 Streakiness in sports

A sportsperson is considered “streaky” if instead of having a constant success rate over time, they have periods of high success rate. Such data will generally be a binary sequence with a “0” denoting a loss and a “1” denoting a win. The data concerning Tiger Woods’ championship wins from September 1996-June 2001 is given and analyzed by Yang (2004), and are reanalyzed using the sampler of Section 2.4. The cumulative
Figure 2.2: Streakiness data: Cumulative counts of Tiger Woods’ tournament wins counts are shown in Figure 2.2 (a). Following Yang (2004) the data is assumed to arise as a sequence of Bernoulli trials, with a possible changing probability of success. The data is ordered by subsequent tournament, and if a changepoint occurs, it is assumed to do so at some tournament.

Let $s_j = \sum_{i=\tau_{j-1}+1}^{\tau_j} y_i$, the number of successes in a segment. From the model assumptions, $y_i \sim_{\text{iid}} \text{Bernoulli}(\phi)$. Taking a Beta($\alpha, \beta$) prior on $\phi$, the marginal likelihood is obtained from

$$
\pi(y_{\tau_{j-1}+1:\tau_j} | \gamma) = \int_0^1 \frac{\Gamma\{\alpha + \beta\}}{\Gamma\{\alpha\} \Gamma\{\beta\}} \phi^{\alpha-1} (1-\phi)^{\beta-1} \prod_{i=\tau_{j-1}+1}^{\tau_j} \phi^{y_i} (1-\phi)^{1-y_i} \, d\phi
$$

where $\gamma = (\alpha, \beta)$. This reduces to

$$
\pi(y_{\tau_{j-1}+1:\tau_j} | \gamma) = \frac{\Gamma\{\alpha + \beta\}}{\Gamma\{\alpha\} \Gamma\{\beta\}} \int_0^1 \phi^{s_j+\alpha-1} (1-\phi)^{\tau_j-\tau_{j-1}-s_j+\beta} \, d\phi.
$$

Completing the Beta integral gives

$$
\pi(y_{\tau_{j-1}+1:\tau_j} | \gamma) = \frac{\Gamma\{\alpha + \beta\} \Gamma\{s_j + \alpha\} \Gamma\{\tau_j - \tau_{j-1} - s_j + \beta\}}{\Gamma\{\alpha\} \Gamma\{\beta\} \Gamma\{\tau_j - \tau_{j-1} + \alpha + \beta\}}.
$$
The parameters $\alpha$ and $\beta$ were both set equal to 1. The distribution between changepoints was taken to be Geometric($p$). The specification of $p$ may have an effect on the outcome of the analysis. It is thus desirable to investigate uncertainty in its value. This is done in two ways. Firstly, a simulation study using the sampler of Section 2.4 is carried out, where there is a hyperprior placed on $p$. Secondly, outputs of analyses using filtering recursions (Fearnhead 2006) for a range of values $p$ are compared.

For the MCMC simulation study using the sampler proposed earlier, the hyperparameter given to $p$ was uniform on $[0, 1]$. After each update of the changepoints the value of $p$ was updated by drawing from its full conditional distribution which is Beta($k + 1, n - k$). A discrete uniform prior on $\{0, \ldots, 10\}$ was taken for the number of changepoints. This gives no discriminating prior weight on a particular number of changepoints. The sampler was run 100 times each for 100,000 burn in iterations and a subsequent 1,000,000 iterations. To reduce dependency in the sample, only every 100th sample was stored. Each run took about 1.5 min on a 2.5GHz processor. Changepoints were updated using the mixture of moves discussed in Section 2.4. Figure 2.2 (b) shows the output from one of these runs, with the posterior probability of a changepoint at any tournament indicated by the dashed line and a scaled counts curve overlain. Figure 2.3 (a) shows posterior probability of the number of changepoints over the 100 runs of the sampler. It can be seen that the sampler performs consistently, giving similar results over the 100 runs. Figure 2.3 (b) shows a histogram for the sampled values of $p$ from the last run. Posterior support for $p$ is highest over the range $[0, 0.1]$.

For the filtering recursions analysis (Fearnhead 2006), the recursions of Section 2.6 were computed for $p \in (0.001, 0.1)$ following the analysis above. A sample of size 100,000 changepoints was generated and the posterior of the number of changepoints was computed for each value of $p$. The modal number of changepoints was recorded from this for each value of $p$ and is shown in Figure 2.4. It is clear that the number of changepoints inferred in the filtering recursions analysis is very sensitive to the value of $p$ for this data. It is questionable whether such an analysis would be useful for a practitioner since it is unclear how one could objectively choose $p$ in this situation.
Figure 2.3: (a) Boxplots of posterior probability for a given number of changepoints for 100 independent runs of the sampler. (b) Histogram of marginal draws of $p$ from one run in the MCMC sampler simulation study.

Certainly an exploratory analysis would be necessary before choosing the value of $p$ to compute the filtering recursions. One suggestion is to use the sampler proposed here for an exploratory analysis of the posterior allowing for uncertainty in the specification of $p$. The MCMC sampler simulation study suggests that two changepoints is most likely although there is relatively strong support for up to five changepoints. In this case, specification of one value of $p$ to generate samples of changepoints will not fully explore uncertainty in the posterior. As before, the output of the MCMC sampler shown in Figure 2.2 (b) shows that one change is clearly identified, but that there is considerable uncertainty in the other positions, hence the support for up to five changepoints.
Figure 2.4: Modal number of changepoints from a filtering recursions analysis over a range of values of $p$.

2.9 Gaussian changepoint models

Gaussian changepoint models are widely used and studied. Models can include those with changing mean and/or variance across segments. The model assumed for the purposes of the example here is piecewise constant, where data in any segment is Gaussian distributed. Segments share a common error variance. Data point $y_i$ in segment $j$ is assumed to arise independently from a $N(\mu_j, \sigma^2)$ distribution. The segment means $\mu_j$ are assumed to arise from a Gaussian distribution with mean $\mu_0$ and variance $\nu^2\sigma^2$, \textit{a priori}. Denote $\gamma = (\sigma^2, \mu_0, \nu^2)$. Segment length is assumed to have a geometric distribution with parameter $p$. Let $ss_j = \sum_{i=\tau_{j-1}+1}^{\tau_j} y_i^2$ and $s_j = \sum_{i=\tau_{j-1}+1}^{\tau_j} y_i$. The model for
all the data may be manipulated to isolate the segment means;

$$\pi(k, z, \theta | y, p, \gamma) \propto \pi(z | k) \pi(\theta | k, z, \mu_0) \pi(y | k, z)$$

$$\propto p^k(1 - p)^{n-k-1} \prod_{j=1}^{k+1} \frac{1}{\nu \sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\nu^2\sigma^2}(\mu_j - \mu_0)^2 \right\}$$

$$\times \prod_{i=\tau_{j-1}+1}^{\tau_j} \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2}(y_i - \mu_j)^2 \right\}$$

$$= \frac{(2\pi)^{-n/2}\nu^{k+1}\sigma^{n+k+1}}{\nu^{k+1}\sigma^{n+k+1}} p^k(1-p)^{n-k-1} \prod_{j=1}^{k+1} \exp \left\{ -\frac{1}{2\sigma^2} \left( \tau_j - \tau_{j-1} + \frac{1}{\nu^2} \right) \mu_j^2 - 2 \left( s_j + \frac{\mu_0}{\nu^2} \right) \mu_j + ss_j + \frac{\mu_0^2}{\nu^2} \right\}.$$  

Completing the square on $\mu_j$ and then performing integration of $\mu_j$ over $(-\infty, \infty)$ gives the posterior,

$$\pi(k, z | y, p) \propto \frac{(2\pi)^{-n/2}}{\nu^{k+1}\sigma^{n+k+1}} p^k(1-p)^{n-k-1} \prod_{j=1}^{k+1} \left( \tau_j - \tau_{j-1} + \frac{1}{\nu^2} \right)^{1/2}$$

$$\times \exp \left\{ -\frac{1}{2\sigma^2} \left( ss_j + \frac{\mu_0^2}{\nu^2} - \frac{(s_j + \frac{\mu_0}{\nu^2})^2}{\tau_j - \tau_{j-1} + \frac{1}{\nu^2}} \right) \right\}.$$  

### Application to Well-log data

The Well-log data (Ó Ruanaidh & Fitzgerald (1996)) records measurements of nuclear-magnetic response of underground rocks obtained by lowering a probe into a bore-hole. The probe records the response at regular points in time. As well as Fearnhead (2006) this data is also analyzed in Fearnhead & Clifford (2003). The data consists of 4050 measurements, some of which are outliers and were removed before analysis. The data are shown in Figure 2.5.

The purpose of this example is to demonstrate how results from an analysis with filtering recursions may be sensitive to the choice of hyperparameters $\gamma$ and how a short run of the sampler could possibly provide good starting values. It is possible to fit a
Figure 2.5: Top: Well-log data. Bottom: Posterior probability of a changepoint in any position from 100,000 samples using the sampler with hyperpriors.

more elaborate state space model to the Well-log data, however, this is not considered here.

Fearnhead (2006) chose the values $p = 0.013, \sigma = 2,330, \nu = 4.3, \mu_0 = 115,000$ when analyzing the Well-log data in the section on inclusion of hyperpriors. Two simple experiments were performed here to investigate sensitivity of the posterior distribution to prior specification. One of $p$ (Experiment 1) or $\sigma$ (Experiment 2) was varied over a grid on a small range keeping all other hyperparameter values fixed (details in Table 2.1). The recursions of Section 2.6 were computed for each value on the grid and a sample of size 100,000 was generated from the posterior of the changepoints. The empirical
posterior distribution of the number of changepoints was computed for each of these samples and the modal number of changepoints recorded. The results are summarized in Figure 2.6. It can be seen that the modal value of the posterior number of changepoints is sensitive to the values of both $p$ and $\sigma$. Thus choosing these values, \textit{a priori}, places the posterior mass $\pi(k, z|y, p, \gamma)$ in the area determined by $p$ and $\sigma$ and may not correctly represent the true posterior over all $p, \sigma$.

For the Well-log data it would seem most sensible to carry out an analysis with inclusion of hyperpriors on $p, \sigma$ and $\mu_0$ using the scheme outlined in Section 2.5. The hyperpriors used are $\pi(p) \propto 1$, $\pi(\mu_0) \propto 1$, $\pi(\nu) \propto 1/\nu$, $\pi(\sigma) \propto 1/\sigma$. The bottom of Figure 2.5 shows the posterior probability of a change output from an algorithm run for 10,000 burn-in and 100,000 subsequent iterations using a random walk update for changepoint positions. Ergodic mean estimators of the hyperparameters were $\hat{\sigma} = 2360$, $\hat{p} = 0.014$, $\hat{\nu} = 3.99$, $\hat{\mu_0} = 113771.0$. This took about 10 sec on a 2.5GHz processor with very diffuse starting values. This Gaussian model infers many changepoints as it picks up small changes in the mean and thus performs well for this data.

A long run of the sampler was implemented so as to obtain a near independent sample ($1.8 \times 10^7$ iterations taking every $1,800^{th}$ sample; estimated integrated autocorrelation time of the number of changepoints $\approx 1$, see Appendix B) of size 10,000 from the posterior distribution of changepoints and hyperparameters. In the independence proposal MCMC scheme suggested in Fearnhead (2006), a sample of changepoints is generated using filtering recursions conditional on $p = 0.013, \sigma = 2.330, \mu_0 = 115,000, \nu = 4.3$. This sample is then used for an independence proposal and hyperparameters are updated in the same way as done here. Figure 2.7 shows kernel density estimates constructed from samples of the hyperparameters for the sampler (dashed line) and independence proposal (solid line). It can be seen that there is a slight discrepancy in that the independence proposal leads to more peaked densities.

In our implementation an independence proposal based on a sample of size 10,000 was used. This updating scheme for hyperparameters and changepoints was then run for 50,000 iterations. Although the acceptance rate for moving between different change-
Figure 2.6: Modal number of changepoints for a filtering recursions analysis of the Well-log data for Experiment 1 and Experiment 2. Experiment 1 varies $p$ (left) and Experiment 2 varies $\sigma$ (right).

point configurations was high, the independence proposal distribution was highly degenerate. Only ten unique changepoint configurations were sampled in the 50,000 iterations of the MCMC scheme. For other datasets where less information is available to choose the hyperparameters to generate the independence proposal, it is possible that this could lead to highly biased sampling from the hyperpriors.

In the sense of hyperprior incorporation and full exploration of the posterior distribution the MCMC sampler proposed performs better than the independence proposal. However, generating independent samples may be more costly in large datasets with many changepoints. Nonetheless, it is clear that the inclusion of hyperpriors circumvents the sensitivity of posterior distribution of the changepoints to specification of the hyperparameters. This is a main advantage of the approach proposed here and makes the detection of changepoints more automatic.
Figure 2.7: Comparison of long run of sampler to MCMC scheme with independent proposals from filtering recursions. Dashed lines give the density from the MCMC sampler output and solid lines give the density output from analysis using the independence proposal in Fearnhead (2006).
<table>
<thead>
<tr>
<th>Recursion Sensitivity</th>
<th>Fixed</th>
<th>Varied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>$\sigma = 2,330, \nu = 4.3, \mu_0 = 115,000$</td>
<td>$p \in [0.005, 0.03]$</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>$p = 0.013, \nu = 4.3, \mu_0 = 115,000$</td>
<td>$\sigma \in [2250, 2750]$</td>
</tr>
</tbody>
</table>

Table 2.1: Experiments to investigate sensitivity of results of filtering recursions to prior specification

2.10 Discussion

This chapter has presented an MCMC method to perform retrospective inference for changepoint models which are collapsable. The multiple changepoint problem is rephrased as a stochastic model search over a large models space, with the Bayes factors for competing models appearing in the acceptance probabilities for the MCMC sampling scheme.

The performance of the sampler was verified for the benchmark coal mining disasters data. Application of the sampler to a streakiness dataset from sports revealed that posteriors for the number of changepoints can be diffuse. It was demonstrated that prior specification on the duration of segments plays a crucial role in the analysis of the models considered. Incorporation of hyperpriors to account for this revealed features of the posterior that would be missed by a popular filtering recursions analysis for changepoints. Application to the Well-log data further highlighted sensitivity of analysis by filtering recursions to prior specification. It was shown that output from a short run of our sampler can be used to give good values of the hyperparameters for this prior specification.

In conclusion, the sampling scheme presented is shown to work well and can provide further insight and account for prior uncertainty in some difficult situations. It can be used as a useful exploratory tool or for a full analysis of multiple changepoint problems.
Chapter 3

Collapsed sampling for Bayesian Latent Block models

3.1 Introduction

Many data sets arise as a result of a number of features or variables being observed for a collection of objects. As examples; shoppers and the items which they do or do not buy; whether a document contains specific words or not; the expression levels of a gene under a series of conditions in a DNA experiment. Such data will be recorded in a matrix, say, with rows indexing objects and columns indexing features or variables. Often interest will focus on clustering rows and further, clustering the features which distinguish these row clusters. This task is referred to as block clustering, although it is also known as block modelling, bichustering, co-clustering and two-mode clustering.

One of the first approaches to block clustering was suggested by Hartigan (1972) and since then, many have been proposed. Much recent work in block clustering and related areas has been either on the analysis of microarray data (Tibshirani, Hastie, Eisen, Ross, Botstein & Brown 1999, Cheng & Church 2000, Getz, Levine & Domany 2000, Lazzeroni & Owen 2002, Kluger, Basri, Chang & Gerstein 2003) or document classification (Hofmann 2001, Blei, Ng & Jordan 2003, Griffiths & Steyvers 2004). Approaches vary in whether they allow clusters to overlap or not. In the case considered in this chapter,
the problem can be thought of as permuting the rows and columns of the data matrix to make a “chessboard” of blocks of data having similar value.

While many approaches to document classification are model-based i.e. a parametric underlying model is assumed when clustering data, this is often not the case in microarray analysis. Some exceptions are Lazzeroni & Owen (2002) who assume a Gaussian error model for gene expression with additive effects for gene and condition clusters and Sheng, Moreau & Moor (2003) who assume a multinomial model for expression level in a discretized microarray. It is common to use two-way hierarchical clustering or other partitioning methods for this data (for example Getz et al. (2000)). One drawback of these methods is the lack of probabilistic justification, as noted by Wit & McClure (2004) (Chapter 7, page 171). A model-based approach allows explicit modelling of noise in the data. This can be an advantage in data, such as microarrays, which is particularly prone to noise, resulting in uncertainty in cluster membership.

This chapter considers an extension of the latent block model (LBM) approach of Govaert & Nadif (2008). The LBM was developed as an intuitive extension of the finite mixture model used in model-based clustering (Fraley & Raftery 2002) to allow clustering of objects and features. The proposed extension is a Bayesian LBM. This has been considered previously by van Dijk, van Rosmalen & Paap (2009). In their approach the number of clusters in objects and features is assumed known and Gibbs sampling is used to find clusterings. They choose the number of clusters using an information criterion based on maximum likelihood.

The main aim of this chapter is to demonstrate that it is possible to sample the number of clusters and the cluster membership jointly, using simple Markov chain Monte Carlo (MCMC) on a collapsed model, so that uncertainty in the number of clusters is naturally incorporated as part of the Bayesian LBM. The collapsed model is obtained by integrating out block parameters analytically. This is possible using standard prior assumptions. There is no need to resort to a trans-dimensional sampler, such as the reversible jump sampler of Green (1995). The idea proposed here extends the allocation sampler of Nobile & Fearnside (2007) to two directions, with slight modifications. Com-
parisons are made to block clustering methods which are most comparable qualitatively. The sampler is applied to both simulated and real datasets to gauge performance.

The remainder of this chapter is organized as follows. Section 3.2 reviews the LBM and introduces the collapsed Bayesian LBM. Section 3.3 gives the MCMC sampler which is used. The differences between this MCMC sampler and reversible jump samplers are also discussed. Procedures to deal with label switching in the context of the sampler are reviewed, and the section concludes with approaches to summarize the output of the sampler. Section 3.4 applies the approach to simulated data. In Section 3.5 voting records data from the U.S. congress are analyzed. The results are compared with those from a maximum likelihood analysis. In Section 3.6 the sampler is used to analyze some microarray data from a biological experiment on budding yeast. The chapter then concludes with a discussion.

3.2 Models

The data is $Y = (y_{ij})$, an $n \times m$ matrix. It is assumed rows and columns may be reordered so that the matrix can be represented as $K \times G$ blocks, with data in blocks modelled by the same density. Here, $K$ and $G$ are the number of row and column clusters respectively. This could be imagined as a “chessboard” effect, with $K - 1$ divisions in the direction of the rows and $G - 1$ in the direction of the columns. The parameters of the data density are conditional on the block and $\theta_{kg}$ denotes the parameters for block $(k, g)$, with $\Theta$ denoting the collection of these. We now give a review of the LBM of Govaert & Nadif (2008).

3.2.1 Latent block models

Conditional on $K$ and $G$, let $U$ be a latent space indexing the set of all possible clusterings of rows and columns. Then the distribution of the data $Y$ can be written

$$p(Y|K, G, \Theta, \phi) = \sum_{u \in U} p(u|K, G, \phi)p(Y|K, G, u, \Theta)$$
where $\phi$ are parameters for the distribution of $u$. Govaert & Nadif (2008) make the assumption that row and column clusterings are independent a priori, so that $p(u|K, G, \phi) = p(z|K, \omega)p(w|G, \rho)$ where $z_i = k$ if row $i$ is in cluster $k$ and $w_j = g$ if column $j$ is in cluster $g$. The probability of a row belonging to cluster $k$ is $\omega_k$ and $\rho_g$ denotes the probability that a column belongs to cluster $g$. The LBM is then

$$p(Y|K, G, \Theta, \omega, \rho) = \sum_{(z, w) \in Z \times W} p(z|K, \omega)p(w|G, \rho)p(Y|K, G, z, w, \Theta) \quad (3.1)$$

where $Z$ and $W$ denote the latent spaces of all row and column clusterings respectively. The assumption of local independence is made when constructing the data likelihood conditional on the latent allocations. That is, within a block, data are independent. This gives data likelihood conditional on $z, w$,

$$p(Y|K, G, z, w, \Theta) = \prod_{k=1}^{K} \sum_{i: z_i = k} \prod_{j: w_j = g} p(y_{ij}|\theta_{kg}).$$

As $|Z \times W| = K^n G^m$, it is not feasible to calculate (3.1). An approach to fitting this model using an iterative maximum likelihood procedure based on Expectation-Maximization (EM) (Dempster, Laird & Rubin 1977) is reviewed below.

**Estimation using BEM2**

The BEM2 algorithm of Govaert & Nadif (2008) will be compared with the Bayesian LBM approach later (Section 3.5). It is thus useful to give a review of this algorithm here. Let the random variables $r_{ik} = I\{z_i = k\}$ and $c_{jg} = I\{w_j = g\}$. Then

$$r_i \sim \text{Multinomial}(\omega_1, \ldots, \omega_K)$$

$$c_j \sim \text{Multinomial}(\rho_1, \ldots, \rho_G).$$

The complete (or classification) log-likelihood associated with the LBM (3.1) is

$$\mathcal{L}(r, c, \omega, \rho, \Theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \log \omega_k + \sum_{j=1}^{m} \sum_{g=1}^{G} c_{jg} \log \rho_g + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{K} \sum_{g=1}^{G} r_{ik} c_{jg} \log p(y_{ij}|\theta_{kg}). \quad (3.2)$$
The E step using this log-likelihood directly is intractable due to the dependence structure among the rows and columns. Govaert & Nadif (2008) suggest a variational approximation to the joint distribution of the latent $r, c$ which leaves $r$ and $c$ independent. Then using the interpretation of EM due to Neal & Hinton (1998) this leads to a new “fuzzy” criterion for block clustering

$$G(s, t, \omega, \rho, \Theta) = \mathcal{L}(s, t, \omega, \rho, \Theta) - \sum_{i=1}^{n} \sum_{k=1}^{K} s_{ik} \log s_{ik} - \sum_{j=1}^{m} \sum_{g=1}^{G} t_{jg} \log t_{jg}$$

which can be alternately maximized with respect to $s, t$ and $\omega, \rho, \Theta$ where $s_{ik} = \Pr(z_i = k)$ and $t_{jg} = \Pr(w_j = g)$.

The possible ways in which this criterion may be maximized determines different algorithms. The BEM2 algorithm maximizes it as follows.

1. Initialize the unknowns $s, t, \omega, \rho, \Theta$ at some sensible value.
2. • Maximize $G$ with respect to $s, \omega$ and $\Theta$ keeping $t$ and $\rho$ fixed.
   • Maximize $G$ with respect to $t, \rho$ and $\Theta$ keeping $s$ and $\omega$ fixed.
3. Iterate step 2 until convergence.

It is noted that each sweep of BEM2 has two maximizations of $\Theta$. This maximization procedure is reported to have outperformed the other schemes considered in Govaert & Nadif (2008), so it is used here to compare with the Bayesian LBM approach.

**Choosing $K$ and $G$ when using BEM2**

In (3.1), it is assumed that $K$ and $G$ are known. The number of clusters assumed can have a considerable effect on the output of clustering algorithms. Usually, many runs, each with a different number of clusters, are necessary. These are then compared to find the best clustering, either based on some information criterion or visual inspection of plots.

Since the LBM is defined in terms of the latent allocation vectors $z$ and $w$, it is not clear how one could use a standard information criterion (e.g. BIC (Schwarz 1978))
to choose the number of components best supported by the data. One approach is to use the maximized complete log-likelihood treating the row and column allocations as unknown parameters. van Dijk et al. (2009) have used this approach for LBM when using AIC-3 (Bozdogan 1994) to choose $K$ and $G$. In this case the number of parameters to be estimated is $n(K-1)+m(G-1)+dKG+(K-1)+(G-1)$ where $d$ is the dimension of any $\theta_{kg}$. A separate model estimation is required for each $K$ and $G$ combination over a grid of plausible models. With little prior knowledge, this grid could be large to ensure all the models supported by the data are entertained.

The Bayesian LBM proposed seeks to incorporate uncertainty in $K$ and $G$ into the model. This is so that the clustering task is also one of cluster model determination. The model determination task and the allocation task are dealt with simultaneously through a fully Bayesian approach. This has analogy with some other block clustering strategies, which undertake greedy searches to find new row and column clusters. See for example Hartigan (2000). An advantage here is that the search has a probabilistic justification based on a posterior distribution for $K$ and $G$. The next section introduces the Bayesian LBM which is at the core of the clustering procedures proposed.

### 3.2.2 Bayesian latent block models

The Bayesian LBM is formed by taking prior densities on $K$, $G$, $\Theta$, $\omega$ and $\rho$. Let $\pi(\cdot)$ denote prior and posterior densities. Then the posterior of the number of clusters and latent cluster allocations follows from Bayes’ theorem

$$
\pi(K, G, z, w, \omega, \rho, \Theta|Y) \propto \pi(K, G)p(z|K, \omega)p(w|G, \rho)p(Y|K, G, z, w, \Theta)
\times \pi(\Theta|K, G)p(\omega|K)p(\rho|G)
$$

(3.3)

Adopting a conjugate prior for $\omega$, $\rho$ and each $\theta_{kg}$ allows one to integrate these from the posterior analytically. This is referred to as collapsing. Doing this obtains the marginal posterior $\pi(K, G, z, w|Y)$. Samples can be generated from this posterior using the MCMC sampler of Section 3.3. This is similar to the general approach of Nobile &
Fearnside (2007). The idea of collapsing has been used by Sheng et al. (2003) in the analysis of a discretized microarray and by Griffiths & Steyvers (2004) in latent Dirichlet analysis for document classification. It would be possible to estimate this model without integrating out parameters by using reversible jump MCMC (RJMCMC) (Green 1995). This is discussed further in Section 3.3.2.

Standard conjugate priors are chosen for the parameters to be integrated out. For example, \( \omega \sim \text{Dirichlet}(\alpha, \ldots, \alpha) \) and \( \rho \sim \text{Dirichlet}(\beta, \ldots, \beta) \) a priori. For the examples considered in this paper, the non-informative values \( \alpha = 1, \beta = 1 \) are used. The prior on \( \theta_{kg} \) will depend on the distribution assumed for the data. For the most widely used models, a standard conjugate prior will be available. The \( \theta_{kg} \) are assumed independent a priori.

Collapsed joint posterior

Let \( n_k \) be the number of rows in cluster \( k \) and \( m_g \) is the number of columns in cluster \( g \). Then writing out all posterior terms gives

\[
\pi(K, G, z, w, \omega, \rho, \Theta | Y) \propto \pi(K) \pi(G) \times \prod_{k=1}^{K} \omega_k^{n_k} \times \prod_{g=1}^{G} \rho_g^{m_g} \times \prod_{k=1}^{K} \prod_{g=1}^{G} \prod_{i: z_i = k} \prod_{j: w_j = g} p(y_{ij} | \theta_{kg})
\]

\[
\times \prod_{k=1}^{K} \prod_{g=1}^{G} \pi(\theta_{kg}) \times \frac{\Gamma\{\alpha K\}}{\Gamma\{\alpha\}^K} \prod_{k=1}^{K} \omega_k^{\alpha - 1} \times \frac{\Gamma\{\beta G\}}{\Gamma\{\beta\}^G} \prod_{g=1}^{G} \rho_g^{\beta - 1}
\]

\[
= \pi(K) \pi(G) \frac{\Gamma\{\alpha K\}}{\Gamma\{\alpha\}^K} \frac{\Gamma\{\beta G\}}{\Gamma\{\beta\}^G} \prod_{k=1}^{K} \omega_k^{n_k + \alpha - 1} \prod_{g=1}^{G} \rho_g^{m_g + \beta - 1}
\]

\[
\times \prod_{k=1}^{K} \prod_{g=1}^{G} \pi(\theta_{kg}) \prod_{i: z_i = k} \prod_{j: w_j = g} p(y_{ij} | \theta_{kg}).
\]

Integrating the left hand side of the above with respect to \( \omega, \rho \) and \( \Theta \) gives \( \pi(K, G, z, w | Y) \).

Integration of the right hand side with respect to \( \omega, \rho, \Theta \) then gives the desired joint
posterior;

\[
\pi(K) \pi(G) \int \int \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\Gamma(\beta G)}{\Gamma(\beta)^G} \prod_{k=1}^{K} \omega_k^{n_k+\alpha-1} \prod_{g=1}^{G} \rho_g^{m_g+\beta-1} \\
\times \prod_{k=1}^{K} \prod_{g=1}^{G} \pi(\theta_{kg}) \prod_{i:z_i=k} \prod_{j:w_j=g} p(y_{ij}|\theta_{kg}) \, d\omega \, d\rho \, d\Theta
\]

\[
= \pi(K) \pi(G) \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\Gamma(\beta G)}{\Gamma(\beta)^G} \times \int \prod_{k=1}^{K} \omega_k^{n_k+\alpha-1} \, d\omega \times \int \prod_{g=1}^{G} \rho_g^{m_g+\beta-1} \, d\rho
\]

\[
= \pi(K) \pi(G) \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\Gamma(\beta G)}{\Gamma(\beta)^G} \times \prod_{k=1}^{K} \Gamma\left\{ n_k + \alpha \right\} \frac{\Gamma\left\{ n + \alpha K \right\}}{\Gamma\left\{ n + \alpha K \right\}} \times \prod_{g=1}^{G} \Gamma\left\{ m_g + \beta \right\} \frac{\Gamma\left\{ m + \beta G \right\}}{\Gamma\left\{ m + \beta G \right\}}
\]

\[
\times \prod_{k=1}^{K} \prod_{g=1}^{G} \int \pi(\theta_{kg}) \prod_{i:z_i=k} \prod_{j:w_j=g} p(y_{ij}|\theta_{kg}) \, d\theta_{kg}.
\]

The two integrals in \( \omega \) and \( \rho \) are obtained as the inverse of the normalizing constant of a Dirichlet density. Making the definition

\[
M_{kg} = \int \pi(\theta_{kg}) \prod_{i:z_i=k} \prod_{j:w_j=g} p(y_{ij}|\theta_{kg}) \, d\theta_{kg},
\]

which will be referred to as the integrated block likelihood, the posterior may be written

\[
\pi(K, G, z, w|Y) \propto \pi(K) \pi(G) \frac{\Gamma(\alpha K) \prod_{k=1}^{K} \Gamma\left\{ n_k + \alpha \right\} \Gamma(\beta G) \prod_{g=1}^{G} \Gamma\left\{ m_g + \beta \right\}}{\Gamma(\alpha)^K \Gamma\left\{ n + \alpha K \right\} \Gamma(\beta)^G \Gamma\left\{ m + \beta G \right\}} \prod_{k=1}^{K} \prod_{g=1}^{G} M_{kg}.
\]

**Priors for the number of clusters**

The priors for the number of clusters, \( \pi(K) \) and \( \pi(G) \) are taken to be truncated Poisson(1) over the ranges \( 1, \ldots, K_{\text{max}} \) and \( 1, \ldots, G_{\text{max}} \). Examples of Poisson priors being adopted for the number of components include Phillips & Smith (1996) and Stephens (2000). The use of a truncated Poisson(1) prior has been justified in Nobile (2005). Experiments with a uniform prior on \( K \) and \( G \) seemed to give unnecessary empty clusters in some situations, possibly due to the fact that there was no penalization for adding clusters.
Calculation of the integrated block likelihood

As noted, the calculation of $M_{kg}$ requires a fully conjugate prior for the $\theta_{kg}$ parameter. Such a conjugate prior exists for most commonly used data models. The $M_{kg}$ is now derived for two useful models which appear in the examples and applications later.

Bernoulli model for binary data

Assume that $Pr(y_{ij} = 1|z_i = k, w_j = g) = \theta_{kg}$. The natural conjugate prior for $\theta_{kg}$ is Beta($\gamma, \delta$). Let $s_{kg} = \sum_{i:z_i = k} \sum_{j:w_j = g} y_{ij}$, the sufficient statistic for block $(k, g)$. Then

$$
\pi(\theta_{kg}) \prod_{i:z_i = k} \prod_{j:w_j = g} p(y_{ij}|\theta_{kg}) = \frac{\Gamma\{\gamma + \delta\}}{\Gamma\{\gamma\} \Gamma\{\delta\}} \theta_{kg}^{\gamma - 1} (1 - \theta_{kg})^{\delta - 1} \prod_{i:z_i = k} \prod_{j:w_j = g} \theta_{kg}^{y_{ij}} (1 - \theta_{kg})^{1 - y_{ij}}
$$

$$
= \frac{\Gamma\{\gamma + \delta\}}{\Gamma\{\gamma\} \Gamma\{\delta\}} \theta_{kg}^{s_{kg} + \gamma - 1} (1 - \theta_{kg})^{n_k m_g - s_{kg} + \delta - 1}.
$$

Integrating the left and right hand side of this with respect to $\theta_{kg}$ gives

$$
M_{kg} = \frac{\Gamma\{\gamma + \delta\} \Gamma\{s_{kg} + \gamma\} \Gamma\{n_k m_g - s_{kg} + \delta\}}{\Gamma\{\gamma\} \Gamma\{\delta\} \Gamma\{n_k m_g + \gamma + \delta\}}.
$$

Gaussian model for continuous data

Assume the distribution of $y_{ij}$ conditional on $z_i = k, w_j = g$ is $N(\mu_{kg}, \sigma_{kg}^2)$. Take the priors $\mu_{kg}|\sigma_{kg}^2 \sim N(\xi, \tau^2 \sigma_{kg}^2)$ and $\sigma_{kg}^2 \sim \text{InvGamma}(\delta/2, \gamma/2)$ where InvGamma($a, b$) is the Inverse-Gamma density $\pi(x) = \frac{b^a}{\Gamma(a)} x^{-(a+1)} \exp\{-b/x\}$. Then

$$
\pi(\mu_{kg}|\sigma_{kg}^2) \pi(\sigma_{kg}^2) \prod_{i:z_i = k} \prod_{j:w_j = g} p(y_{ij}|\mu_{kg}, \sigma_{kg}) = \frac{(\gamma/2)^{\delta/2}}{\Gamma(\delta/2)} \sigma_{kg}^{-(\delta/2+1)} \exp\{-\gamma/2\sigma_{kg}^2\}
$$

$$
\times (2\pi\tau^2 \sigma_{kg}^2)^{-1/2} \exp\{-(\mu_{kg} - \xi)^2/2\tau^2 \sigma_{kg}^2\}
$$

$$
\times (2\pi \sigma_{kg}^2)^{-n_k m_g/2} \exp\{-s_{skg} - 2\mu_{kg} s_{kg} + n_k m_g \mu_{kg}^2/2\sigma_{kg}^2\}
$$

where $s_{skg} = \sum_{i:z_i = k} \sum_{j:w_j = g} y_{ij}^2$. Completing the square on $\mu_{kg}$ and integrating with respect
to it gives
\[ (2\pi)^{-nkmg/2} \frac{(\gamma/2)^{\delta/2}}{\Gamma(\delta/2)} \sigma_{kg}^{-(n_k m_g + \delta)/2+1} \]
\[ \times (n_k m_g \tau^2 + 1)^{-1/2} \exp \left\{ -\frac{1}{2\sigma_{kg}^2} \left( ss_{kg} - \frac{\tau^2 (s_{kg} + \xi/\tau)^2}{n_k m_g \tau^2 + 1} + \frac{\xi^2}{\tau^2} + \gamma \right) \right\} . \]

Finally, integrating with respect to \( \sigma_{kg}^2 \) and tidying up gives

\[ M_{kg} = \frac{\gamma^{\delta/2} \Gamma\{(n_k m_g + \delta)/2\}}{\pi^{nkmg/2} \Gamma(\delta/2)} \left( \frac{n_k m_g \tau^2 + 1}{ss_{kg}} \right)^{1/2} \left( \frac{n_k m_g \tau^2 + 1}{ss_{kg} - \frac{\tau^2 (s_{kg} + \xi/\tau)^2}{n_k m_g \tau^2 + 1} + \frac{\xi^2}{\tau^2} + \gamma} \right)^{-(n_k m_g + \delta)/2} . \]

### 3.3 MCMC sampling of clusterings

The sampler which is used consists of four different moves. The first is just a standard Gibbs update for the row/column label. The second proposes to reallocate collections of rows and columns. The final two moves propose to add or remove clusters. The moves are described for rows, but apply to columns analogously. When running the algorithm, the moves are each applied to the rows and columns in a single sweep. Since the LBM will be invariant to cluster labelings, it is possible for label switching to occur. A procedure to deal with this is outlined as well as ways to summarize the output from the sampler.

#### 3.3.1 MCMC moves

**Gibbs sampling to update the allocation of one row**

Suppose row \( i \) is currently in cluster \( k \). Then its new allocation, \( \tilde{z}_i \) is sampled from the distribution

\[ p(\tilde{z}_i = k'|Y, K, G, z_{-i}, w) \propto \frac{n_{k'} + \alpha}{n_k - 1 + \alpha} \prod_{g=1}^G \frac{M_{k'g}^{(+i)} M_{kg}^{(-i)}}{M_{kg}^{(+i)} M_{k'g}^{(-i)}} , k' \neq k \]  

(3.5)

and \( p(\tilde{z}_i = k|Y, K, G, z_{-i}, w) \propto 1 \) where \( M_{kg}^{(-i)} \) and \( M_{k'g}^{(+i)} \) are obtained respectively by removing row \( i \) from cluster \( k \) and adding it to cluster \( k' \) within column cluster \( g \). This
distribution is arrived at as follows. Consider moving row \( i \) from cluster \( k \) to cluster \( k' \). The only terms effected in the posterior are \( n_k, n_{k'} \) and \( M_{kg}, M_{k'g}, g = 1, \ldots, G \). The posterior probability of this move is thus proportional to

\[
\Gamma\{n_k - 1 + \alpha\}\Gamma\{n_{k'} + 1 + \alpha\}\prod_{g=1}^{G} M_{kg}^{(-i)} M_{k'g}^{(+i)}.
\]

There is a slight computational saving to be made by dividing the posterior probability of each of the candidate clusters \( k' \) by the posterior probability that row \( i \) remains in its current cluster;

\[
\frac{\Gamma\{n_k - 1 + \alpha\}\Gamma\{n_{k'} + 1 + \alpha\}}{\Gamma\{n_k + \alpha\}\Gamma\{n_{k'} + \alpha\}}\prod_{g=1}^{G} \frac{M_{kg}^{(+i)} M_{k'g}^{(-i)}}{M_{kg} M_{k'g}}.
\]

The computational saving arises since the quotient involving the gamma functions simplifies. This simplification follows from the gamma function recurrence relation, \( \Gamma\{z + 1\} = z\Gamma\{z\}, z \in \mathbb{R} \), implying

\[
\frac{\Gamma\{x + \alpha\}}{\Gamma\{x + \alpha + 1\}} = \frac{\Gamma\{x + \alpha\}}{(x + \alpha)\Gamma\{x + \alpha\}} = \frac{1}{x + \alpha}
\]

for any \( x \in \mathbb{Z}, \alpha \in \mathbb{R} \). Thus evaluation of the gamma functions when computing probability of membership for each cluster is not required. This gives the un-normalized posterior probability of moving to cluster \( k' \) as that stated above.

The total computational effort required for the Gibbs sweep on rows and columns is \( O((n + m)KG) \) which may be prohibitive for large \( K, G, n \) or \( m \). It is possible to move one row and column between clusters using a Metropolis-Hastings move. This could be alternated with a Gibbs update to reduce computational overhead or some mixture of the two moves could be used.

**Move to update the allocation of more than one row**

This move is similar to move M3 in Nobile & Fearnside (2007). Its role is to move more than one row at a time. The way in which new row allocations are proposed should isolate clusters more quickly than just performing one row Gibbs updates. The
procedure is as follows. Choose two row clusters $k$ and $k'$ at random. Let $S$ be the index set of rows currently belonging to clusters $k$ and $k'$. The members of $S$ are randomly reordered. Imagining clusters $k$ and $k'$ to be empty initially and $S$ to be full, sequentially take each row from $S$ and allocate it to $k$ or $k'$. This allocation is done using the probability that the current clusters $k$ or $k'$ generated that row conditioning on rows that have already been reallocated to $k$ or $k'$. For row $i$ in $S$ these probabilities are denoted by $p_k^{(i)}$ and $p_{k'}^{(i)}$ with $p_k^{(i)} + p_{k'}^{(i)} = 1$. To write down the proposal probability of this move we use $\bar{M}_{kg}, g = 1, \ldots, G$ to represent the integrated likelihood of the members placed in cluster $k$ before member $i$ has been processed. Similarly $\tilde{n}_k$ represents the number of rows in cluster $k$ before $i$ has been processed. Then using similar notation to the Gibbs move it can be shown (see Appendix A.2, Nobile & Fearnside (2007)) that

\[
\frac{p_k^{(i)}}{p_{k'}^{(i)}} = \frac{\tilde{n}_{k'} + \alpha}{\tilde{n}_k + \alpha} \prod_{g=1}^{G} \frac{\bar{M}_{k'g} \bar{M}_{kg}}{\bar{M}_{k'g} \bar{M}_{kg}}.
\]

Using $p_k^{(i)} + p_{k'}^{(i)} = 1$, the above can be solved for $p_{k'}^{(i)}$. The proposed allocation of row $i$, $\tilde{z}_i$ may then be sampled. Once the quantities $\tilde{n}_{k'}$, $\tilde{n}_k$, $\bar{M}_{k'g}$ and $\bar{M}_{kg}$ have been updated based on $\tilde{z}_i$, the next row in $S$ can be dealt with.

When all members of $S$ have been processed the proposal probability of moving from the current allocation $z$ to the new allocation $\tilde{z}$ is

\[
\frac{1}{K(K-1)} \prod_{i \in S} p_{\tilde{z}_i}^{(i)}.
\]

For the reverse move the proposal probability is

\[
\frac{1}{K(K-1)} \prod_{i \in S} p_{z_i}^{(i)}.
\]

The new allocation $\tilde{z}$ is then accepted with probability $\min(1, A)$ where

\[
A = \frac{\Gamma\{\tilde{n}_k + \alpha\} \Gamma\{\tilde{n}_{k'} + \alpha\}}{\Gamma\{n_k + \alpha\} \Gamma\{n_{k'} + \alpha\}} \prod_{g=1}^{G} \frac{\bar{M}_{k'g} \bar{M}_{kg}}{\bar{M}_{k'g} \bar{M}_{kg}} \times \prod_{i \in S} \frac{p_{\tilde{z}_i}^{(i)}}{p_{z_i}^{(i)}},
\]

and $\tilde{n}_k, \tilde{n}_{k'}, \bar{M}_{k'g}, \bar{M}_{kg}$ are the proposed cluster sizes and integrated block likelihoods when all the members of $S$ have been processed.
Moves to split or combine clusters

To add a cluster, first randomly propose a cluster, \( k \), to “split”. The new cluster will be labelled \( K + 1 \) if the current number of clusters is \( K \). In the same way as the move to reallocate more than one row (Section 3.3.1), the probability of a row proposed as being in cluster \( k \) or \( K + 1 \) is given by the conditional probability it was generated by that cluster, the rows being processed sequentially. Clearly the order in which rows are processed is important. Thus for the split and combine moves we place an ordering on the members of cluster \( k \), that is, the order in which the members are arranged in cluster \( k \) is important. As well as taking members out from cluster \( k \) and placing them in cluster \( K + 1 \), all members are placed back into cluster \( k \) in the combine move. It is possible to propose a label swap of \( K + 1 \) with any other label selected at random (itself included), say \( k' \). This then would split cluster \( k \) into clusters \( k \) and some \( \{1, \ldots, K + 1\} \setminus k \).

Let \( S \) denote the index set of rows currently belonging to cluster \( k \). A split move is chosen with probability \( p^K_S \). For the split move, the denominator in the proposal ratio will be

\[
p(z \rightarrow \tilde{z}) = p^K_S \frac{1}{K(K + 1)} \frac{1}{n_k!} \prod_{i \in S} p^{(i)}_{z_i}.
\]

where the second term accounts for selecting the cluster to split, and then the cluster to swap labels with, the third term accounts for the number of ways in which members may be arranged (processed), and the fourth term is the product of conditional probabilities (see Section 3.3.1).

For the combine move, two clusters are selected at random, say \( k \) and \( k' \) from the \( K + 1 \) available. Then all members of cluster \( k' \) are proposed to be placed back in \( k \). Thus the numerator in the proposal probability for the split move is

\[
p(\tilde{z} \rightarrow z) = (1 - p^{K+1}_S) \frac{1}{K(K + 1)} \frac{1}{n_k!}.
\]

where the first term is the probability of proposing a combine move, the second accounts for the clusters selected, and the third accounts for the number of ways in which the members of cluster \( k \) may be arranged.
The acceptance probability for the split move is then \( \min(1, A) \) where

\[
A = \frac{\pi(K+1)}{\pi(K)} \frac{\Gamma\{n + \alpha K\}}{\Gamma\{n + \alpha(K+1)\}} \frac{\Gamma\{\alpha(K+1)\}}{\Gamma\{\alpha\Gamma\{\alpha K\}\}} \frac{\Gamma\{n_k + \alpha\}}{\Gamma\{n_{k'} + \alpha\}} \times \prod_{g=1}^{G} \frac{\tilde{M}_{kg}\tilde{M}_{kg'}}{\bar{M}_{kg}} \times \frac{1 - \bar{p}_s^{K+1}}{\bar{p}_s^K} \left( \prod_{i \in S^p(i)} \frac{s}{\tilde{p}_z(i)} \right)^{-1}
\]

and \( \tilde{n}_k, \tilde{n}_{k'}, \tilde{M}_{kg}, \tilde{M}_{kg'} \) give the proposed sizes and integrated block likelihoods of the proposed clusters.

The acceptance probability for the combine move is \( \min(1, A^{-1}) \). These moves are similar to the “split and combine” moves discussed by Richardson & Green (1997). Experiments showed that these split and combine moves gave more satisfactory mixing and higher acceptance rates for the Bayesian LBM than proposing empty clusters.

### 3.3.2 Form of reversible jump sampler

As noted, the moves discussed in the previous section resemble moves used in the RJMCMC sampler of Richardson & Green (1997) and other RJ samplers for related classification problems (for example Robert et al. (2000)). The difference with the sampler here is that the posterior sampled from is of fixed dimension. This is due to collapsing. Performing an equivalent RJ analysis to that presented here would be challenging for LBMs. This would mean extending the Gibbs sampler of van Dijk et al. (2009) to include variable dimension moves for splitting or combining clusters. The construction of proposal densities for variable dimensional moves in RJ samplers can be crucial to their performance. Work has been done in this area (Brooks et al. 2003, Green 2003), but for many applications construction of proposals is case specific. The reason for considering a RJ analysis here is that concern is not in only finding a cluster allocation for a specified LBM. Exploring different cluster models is of interest too, and so the task also becomes one of model determination as discussed in Section 3.2.1.

Consider splitting a row cluster \( k \) into \( k \) and \( k' \) in a typical RJ approach. This is more difficult than the component splitting case in Richardson & Green (1997), since splitting each row cluster gives rise to \( d(G+1) \) new parameters where \( d \) is the dimension.
of any $\theta_{kg}$. Finding a proposal that will mix well may require lots of trial and error, especially if $d > 1$ or $G$ is even moderately large. Moreover, computational time would increase dramatically with respect to the collapsed LBM in these situations.

Using a collapsed model, is, in a sense, a form of variance reduction for this model. Variability in sampling of allocations is reduced by integrating out $\omega, \rho$ and $\Theta$. Akin, in some sense, to the idea of Rao-Blackwellization (Robert & Casella 2004). This should give better sampling of the high probability clusterings of the data, since uncertainty due to parameter values has vanished.

### 3.3.3 Label switching

The joint posterior of cluster models and allocations or labels (3.4) is only identifiable up to permutations of the labels. If there is one labelling $1, \ldots, K$ of the rows, then any permutation of this, say, $\sigma(1), \ldots, \sigma(K)$, gives exactly the same information about clustering relationships. The posterior on row labels has $K!$ indistinguishable modes. Generally as the Markov chain progresses, switches between these equivalent modes will occur. For the Bayesian LBM model label switching can occur for both row and column labels independently. There are many approaches for dealing with the label switching problem (Stephens 2000, Celeux, Hurn & Robert 2000). The approach adopted here is due to Nobile & Fearnside (2007). It is ideal for the Bayesian LBM since it does not involve loss functions based on sampled model parameters (which are no longer in the collapsed model). It only uses the sampled $z$ and $w$.

Row and column allocation vectors are post-processed separately. To post-process the label vectors $z_1, z_2, \ldots$ output from MCMC the first step is to arrange these in order of increasing number of non-empty components. This gives the ordering $z^{(1)}, z^{(2)}, \ldots$, where for $s < t$, $z^{(s)}$ uses either the same number of components as $z^{(t)}$ in total, or less. For example, with $K = 4$, $z^{(s)} = (3, 3, 2, 2, 1)$ would come before $z^{(t)} = (4, 4, 3, 3, 1, 2)$. Suppose the vectors $z^{(t)}$ have been processed and re-labelled up to time $T − 1$, and there
are $K_{T-1}$ non-empty components in $z^{(T-1)}$. Compute a cost matrix with general element

$$C(k_1, k_2) = \sum_{t=1}^{T-1} \sum_{i=1}^{n} I\{z_i^{(t)} \neq k_1, z_i^{(T)} = k_2\}.$$ 

Then the more $z^{(T)}$ disagrees with the vectors already processed, the higher this cost will be.

The role of the loss function is made clearer by rewriting the general entry of the cost matrix:

$$C(k_1, k_2) = n(T - 1) - \sum_{t=1}^{T-1} \sum_{i=1}^{n} I\{z_i^{(t)} = k_1, z_i^{(T)} = k_2\}$$

$$= n(T - 1) - \sum_{i=1}^{n} N_i(T - 1, k_1) I\{z_i^{(T)} = k_2\}$$

(3.6)

where $N_i(T - 1, k_1)$ gives the number of processed samples up to $z^{(T-1)}$ which have given label $k_1$ to row $i$. For the sake of discussion, consider processing a sample where no label switching has occurred, $K$ is fixed and there are no changes in labels from one MCMC sample to the next. In this case the costs will be

$$C(k, k) = n(T - 1) - \sum_{i=1}^{n} N_i(T - 1, k) I\{z_i^{(T)} = k\}$$

$$= n(T - 1) - n_k(T - 1)$$

$$= (n - n_k)(T - 1)$$

and for $k' \neq k$

$$C(k, k') = n(T - 1) - \sum_{i=1}^{n} N_i(T - 1, k) I\{z_i^{(T)} = k'\}$$

$$= n(T - 1) - 0$$

$$= n(T - 1).$$

Zero cost can only be obtained when all rows have the same label, that is, when there is no clustering. Of course this discussion simplifies the problem somewhat. The key is
in finding a permutation of the labels to minimize all costs. This permutation is found by solving the square assignment problem using the algorithm of Carpaneto & Toth (1980) in our case. This algorithm returns the permutation $\sigma(\cdot)$ of the labels in $z^{(T)}$ which minimizes the total cost $\sum_{k=1}^{K_T} C(k, \sigma(k))$. Then $z^{(T)}$ is relabelled by permuting the labels according to $\sigma(\cdot)$.

The equation (3.6) can be exploited to give an efficient on-line post processing procedure. Define the $K \times n$ matrix $S^{(T-1)}$ with general entry

$$S_{ki}^{(T-1)} = \sum_{t=1}^{T-1} I\{z_i^{(t)} = k\}.$$ 

Then

$$C(k, k') = n(T - 1) - \sum_{i=1}^{n} S_{ki}^{(T-1)} I\{z_i^{(T)} = k'\}.$$ 

After calling the square assignment algorithm and permuting the labels $z^{(T)}$ according to its solution, $S$ can be updated using

$$S_{ki}^{(T)} = S_{ki}^{(T-1)} + I\{z_i^{(T)} = k\}.$$ 

This approach will become computationally more expensive as the number of clusters $K$ increases (each call to the square assignment algorithm is at worst $O(K^4)$ for each MCMC iterate).

### 3.3.4 Summarizing MCMC output

Having sampled both the number of clusters and cluster memberships, it will be of interest to give a summary of the sampling. As different $(K, G)$ cluster models are structurally different, it is not possible to give an “average” of cluster membership. Two summaries are suggested.

**Using the modal cluster model**

The first summary focuses on using the modal, or most visited model from the MCMC output. It takes the series of $(K, G)$ visited models and chooses the pair which appear
most often. Call this pair \((\hat{K}, \hat{G})\). Suppose this pair has occurred \(N\) times in the post burn-in sample. The \(N\) pairs of label vectors \(z\) and \(w\) corresponding to these occurrences are extracted from the MCMC sample. These are then post-processed to undo possible label switching. This is necessary to compute posterior distributions of row and column cluster membership. After computing the posterior distributions of row membership, row \(i\) has distribution \((q_{i1}, \ldots, q_{iK})\) where \(q_{ik}\) is the estimated posterior probability row \(i\) belongs to cluster \(k\) in the \((\hat{K}, \hat{G})\) cluster model. For the summary, assign \(i\) to cluster \(\text{arg max}_k q_{ik}\). The columns are given the same treatment.

**Using the MAP**

Since sampling is from the fixed dimensional posterior \(\pi(K, G, z, w|Y)\), the maximum a posteriori (MAP) cluster model and cluster membership \((K, G, z, w)_{MAP}\) is also a useful summary of the MCMC output. The MAP gives the visited \((K, G, z, w)\) having highest probability a posteriori from the samples obtained.

### 3.4 Simulated data

To see how the sampler discriminates between different cluster models, it was run on some simulated data. This data consisted of three \(200 \times 200\) binary matrices with 4 and 4, 2 and 5, and 1 and 4 row and column clusters respectively. In each case, the block parameter \(\theta_{kg}\) was drawn uniformly from \([0, 1]\). The blocks were then generated using \(\text{Bernoulli}(\theta_{kg})\) random variates. This is shown in the left of Figure 3.1. Clusters were made less distinguishable by transforming the generated \(\theta_{kg}\) to the intervals \([0.2, 0.8]\) and \([0.3, 0.7]\) using \(\theta_{kg}^{[a,b]} = a + \theta_{kg}(b - a)\) and generating two further matrices. The rows and columns of the resulting matrices were then randomly reordered, disguising the data structure. The chain was run for 1000 burn-in iterations and a further 16,000 iterations on each data set. A Beta\((1, 1)\) prior was assumed for \(\theta_{kg}\) in all cases. The priors for \(\omega\) and \(\rho\) are as in Section 3.2.2.

Two performance diagnostics of the sampler were considered. The first was the
Figure 3.1: Simulated data with decreasing distinguishability. Columns left to right correspond to A, B, C.
posterior model probability (PMP) of the model used to generate the data and the second was the integrated autocorrelation time (IAT) of sampled cluster models. Computing the PMP just amounts to counting the number of times the model in question was visited and dividing by the total number of samples. For the IAT, cluster models \((K, G)\) are identified by a model index \(R = 1, \ldots, K_{\text{max}}G_{\text{max}}\). Then the quantity \(\hat{\tau} = 1 + 2 \sum_{t=1}^{\infty} \rho_R(t)\) is estimated, where \(\rho_R(t)\) is the autocorrelation of the series of post burn-in samples \(R_1, R_2, \ldots\) at lag \(t\). The series here refers to the cluster models sampled from the posterior (3.4). If the estimate of the IAT is \(\hat{\tau}\), then roughly \(\hat{\tau}\) MCMC draws are required to get the equivalent of one independent draw from the full posterior distribution. This can be used as a measure of efficiency for MCMC algorithms. See for example Roberts (1996). A more detailed description is given in Appendix B.

The results are shown in Table 3.1. The \(\theta_{kg}\) column is coded A for \(\theta_{kg} \sim \text{Uniform}[0, 1]\), B for transformation to \([0.2, 0.8]\) and C for \([0.3, 0.7]\). As the noise in the data increases, the ability to identify the model which generated the data decreases. This is to be expected. The estimated IAT indicates that there is less efficient sampling as the noise

<table>
<thead>
<tr>
<th>((K, G))</th>
<th>(\theta_{kg})</th>
<th>PMP</th>
<th>(\hat{\tau})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,4)</td>
<td>A 0.9550</td>
<td>8.79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B 0.9463</td>
<td>10.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C 0.9014</td>
<td>17.43</td>
<td></td>
</tr>
<tr>
<td>(2,5)</td>
<td>A 0.9343</td>
<td>4.55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B 0.8886</td>
<td>9.79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C 0.8369</td>
<td>13.66</td>
<td></td>
</tr>
<tr>
<td>(1,4)</td>
<td>A 0.8035</td>
<td>7.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B 0.3000</td>
<td>8.97</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C 0.1494</td>
<td>4.61</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Results of simulation experiment. The PMP gives the posterior model probability of the generating model. \(\hat{\tau}\) is the estimated IAT.
increases, with the exception of the \((1, 4)\) cluster model. This is a particularly challenging situation, since two of the clusters are very similar (see Figure 3.1). The transformed \(\theta_{11}, \theta_{13}\) were for \([0.2, 0.8] : 0.288, 0.320\) and for \([0.3, 0.7] : 0.36, 0.38\). The fact that these two clusters are practically indistinguishable would make the sampler choose a \((1, 3)\) model as the best model after scrambling of the data. In fact the most visited model in both these cases had 1 row cluster and 3 column clusters (62.37% and 82.66% of the posterior probability). In this situation, the best cluster model was not the same as the generating model. This is an artifact of the simulation process but shows that a sensible clustering can be achieved.

### 3.5 Congressional voting in US senate

The sampler was applied to the UCI Congressional Voting data assuming the Bernoulli model of Section 3.2.2. The data records whether 435 members of the 98th congress (267 democrats, 168 republicans) voted “yay”, “nay”, abstained or were absent in votes on 16 different key issues. Here the members of congress are represented by rows, and the issues are represented by columns. The data is available from


and is shown in the left panel of Figure 3.2. The aim was to see whether the sampler could discover any clustering by party and issue. For example, one may expect that democrats voted differently to republicans on certain issues. It was thought best to ignore absent and abstain votes. Here, this is equivalent to treating these votes as a “nay”, since the focus is on clustering rows and columns. The only sample sizes entering into the posterior calculations are the number of rows and columns in each cluster. For the Bernoulli model, the block sufficient statistic is the sum of the data. This is not affected by a missing data point. It is possible to include missing data by using an extra Gibbs step to sample the value of any point from its full conditional. This isn’t considered here however.

The sampler was run for 110,000 iterations with 10,000 as a burn-in initialized at
the no cluster model. To reduce correlation in samples, every 10th sample was stored after burn-in. The move of Section 3.3.1 had a 16% acceptance rate for rows and 51% for columns. The cluster split and combine moves had about a 1.5% acceptance rate for rows and 8% for columns. The run took just over an hour on a 2.5GHz processor.

Table 3.2 shows the distribution of the number of row and column clusters. It can be seen that about 60% of the posterior probability is placed on 6/7 row clusters and 12/13 column clusters. The samples with 7 row clusters and 12 column clusters were extracted to construct an estimated clustering following Section 3.3.4. The estimated clustering is shown in the middle panel of Figure 3.2. The red horizontal lines here divide the clusters of congressmen, and the blue vertical lines divide the issue clusters. When referencing Figure 3.2, congressman clusters (rows) are numbered 1 to 7 top-bottom, and the issue clusters (columns) are numbered 1 to 12 left-right.

Issues have only three non-singleton clusters. The first contains “anti-satellite-test-ban”, “aid-to-nicaraguan-contras” and “mx-missile” (column cluster 1). The second contains “physician-fee-freeze” and “education-spending” (column cluster 3) and the third has “handicapped-infants” and “duty-free-exports” (column cluster 6). Row cluster composition by party is shown in Table 3.3. The majority democrat party roughly splits into four clusters, while the republican party splits into two. The main discrepancy between the two large democrat clusters, 2 and 3, appear to be the issues “religious-groups-in-schools” and “crime” in issue clusters 9 and 12. Row cluster 6 which is also mainly democrat appears to vote similarly to the republican cluster 1. Row clusters 4 (democrat) and 5 (republican) appear to deviate from their core party vote.

The results obtained from the Bayesian LBM were compared with those obtained from the BEM2 algorithm of Govaert & Nadif (2008), reviewed in Section 3.2.1. The algorithm was run using 7 row clusters and 12 column clusters. It should be noted that BEM2 requires the number of row and column clusters to be assumed known in advance. To obtain an estimated clustering, we took the cluster with the maximum probability of membership. The composition by party of the row clusters from BEM2 is shown in
Figure 3.2: Voting data. Colour key: white = “nay”, black = “yay”. Left panel: Raw data. Right panels: summary cluster membership from the modal 7 row and 12 column cluster model and the cluster membership obtained from BEM2. Row clusters are numbered 1-7, top-bottom. Column clusters are numbered left-right. The red lines divide clusters of congressmen, and the blue lines divide the issue clusters.
Table 3.2: Distribution of cluster models for voting data

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Democrat</th>
<th>Republican</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (131)</td>
<td>8</td>
<td>123</td>
</tr>
<tr>
<td>2 (125)</td>
<td>125</td>
<td>0</td>
</tr>
<tr>
<td>3 (77)</td>
<td>71</td>
<td>6</td>
</tr>
<tr>
<td>4 (38)</td>
<td>37</td>
<td>1</td>
</tr>
<tr>
<td>5 (36)</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>6 (23)</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>7 (5)</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.3: Party distribution over row clusters from collapsed sampling.

Table 3.4. Row cluster 1 is similar in both, but there are some differences in the other clusters. The BEM2 clustering only used 10 column clusters of the 12 available. The right panel of Figure 3.2 shows the clustering from BEM2. For comparison purposes with the collapsed LBM clustering, the columns have been arranged in the same order. The collapsed LBM appears to identify more small clusters, leading to a marginally more homogeneous blocking of the data.

There is an advantage here over an EM approach to fitting a cluster model in that the number of clusters need not be assumed known in advance. Here, model uncertainty
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Democrat</th>
<th>Republican</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (131)</td>
<td>8</td>
<td>123</td>
</tr>
<tr>
<td>2 (104)</td>
<td>104</td>
<td>0</td>
</tr>
<tr>
<td>3 (62)</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>4 (60)</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>5 (35)</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>6 (30)</td>
<td>26</td>
<td>4</td>
</tr>
<tr>
<td>7 (13)</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.4: Party distribution over row clusters from BEM2 algorithm.

is naturally inbuilt into the approach and it is dealt with automatically by the sampler. There is no user intervention to choose the cluster model. User intervention is only in the choice of prior hyperparameters and priors on the number of clusters. Generally the priors for the data model parameters here, and the truncated Poisson(1) prior on the number of clusters (as argued in Nobile (2005)) have been found to perform well on various binary datasets. The computations for the collapsed LBM are also numerically stable if clusters empty out. In EM algorithms this causes instability, as was experienced in applying BEM2 above. Empty clusters can easily occur when the chosen cluster model is not well supported by the data.

### 3.6 Microarray experiments

A DNA microarray experiment records expression levels of a large number of genes over a number of conditions or samples. The number of conditions or samples is usually less than 100, while the number of genes could be in the thousands. Discovering which genes behave similarly and under which subgroups of conditions is the aim of analysis. One way to do this is to group together genes with similar expression levels. Methods
differ in whether they allow clusters to overlap or not. Here clusters are not allowed to overlap due to the form of the LBM.

Analyzing DNA experiments can be challenging, due to the large row dimension and the general uncertainty in how many clusters may be present in the data. The sampler is applied to data from DNA experiments on the budding yeast *Saccharomyces Cerevisiae*. The microarray contains 419 genes and records the expression level of these under 70 conditions. It was obtained from the R package *biclust* (Kaiser, Santamaria, Sill, Theron, Quintales & Leisch. 2009). Expression levels lay between $-6$ and $+7$. The aim is to see how much structure the sampler can uncover, so the rows and columns of the microarray were randomly reordered (Figure 3.3 (a)). In this application the rows represent the genes and the columns represent the conditions.

The Gaussian model of Section 3.2.2 is used for expression level. This model requires specification of four hyperparameters. Two of these ($\gamma$ and $\delta$) are for the prior on the block error variances and two are for the prior on the block means ($\xi$ and $\tau^2$). The were chosen to be $\gamma = \delta = 0.02$ and $\xi = 0$, $\tau^2 = 100$. This choice of $\gamma$ and $\delta$ gives a proper density on the error variance which is non-informative (see for example Spiegelhalter, Best, Gilks & Inskip (1996)). Similarly, choosing $\xi = 0$ is a reasonable non-informative choice given the range of the data. Setting $\tau^2 = 100$ says that the prior information on a block mean is equal to 1% of the information in the observed expression level of one gene under one condition within that block. This is also non-informative.

The sampler was run for 220,000 iterations with 20,000 taken as burn-in. Every 20th iteration was stored after burn-in. The run was time consuming, taking approximately 3 hours. This said, the large gene dimension of such an array does pose a challenge when searching for two way clusters. The initial cluster model assumed had 1 row and column cluster i.e. no cluster structure. Acceptance rates for the move of Section 3.3.1 were 25% for rows and 18% for columns. Split and combine acceptances were about 0.5% for rows and about 25% each for columns. The low acceptance rates of split and combine moves for rows would be expected, since finding clusters will be more difficult in a larger dimension.
Table 3.5 gives the PMP of the visited models from the MCMC output. The model space visited by the sampler is large. The modal model (25 row clusters and 4 column clusters) gave a posterior probability of 11.98%. There is posterior support for anything from 3 to 5 column clusters, for 23 to 26 row clusters (PMP > 0.02 in all cases). This is an example with considerable model uncertainty and it may be difficult to know the models to include using an information criterion over a grid of possible models as discussed in Section 3.2.1 and adopted by van Dijk et al. (2009). An obvious advantage of exploring the uncertainty in the posterior model space and attaching a probability to each model is that it by-passes these issues.

Instead of constructing a summary based on the modal model, the MAP clustering was used here (shown in Figure 3.3(b)). The MAP had 26 row clusters and 4 column clusters. To have a closer look at the row clusters, a selection of these are plotted in Figure 3.4. The plots show the gene expression profiles for genes in the same row cluster over conditions arranged by condition cluster. It can be seen that in certain cases, there is a clear clustering of genes with similar profiles. The gene clusters shown are arranged by size (left-right, top-bottom). Some of the larger clusters appear quite noisy, while some follow a common trend closely. Auxiliary runs of the sampler on the subsets of row clusters could be performed to try and isolate further cluster structures.

### 3.7 Discussion

This chapter considered a collapsed Bayesian extension of the Latent Block Model of Govaert & Nadif (2008). It was shown that an MCMC sampler could be used to sample both the cluster model and the cluster memberships when clustering a data matrix into blocks. The approach was demonstrated on simulated data and two real data examples. The application to simulated data suggested that the sampler’s performance deteriorates as clusters become less distinguishable. The sampler was applied to Congressional voting records from the U.S. senate. It was shown to perform well in isolating clusters of congressmen and “yay”, “nay” votes in the data. In the second
Figure 3.3: Yeast data. (a) Original microarray (b) MAP clustering from sampler
<table>
<thead>
<tr>
<th>Rows</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>19</td>
<td>0.0005</td>
<td>0.0002</td>
<td>0.0005</td>
<td>0.0001</td>
<td>0.0019</td>
<td>0.0044</td>
<td>0.0013</td>
<td>0.0011</td>
<td>0.0006</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>20</td>
<td>0.0005</td>
<td>0.0000</td>
<td>0.0002</td>
<td>0.0032</td>
<td>0.0110</td>
<td>0.0192</td>
<td>0.0121</td>
<td>0.0057</td>
<td>0.0012</td>
<td>0.0004</td>
<td>0.0000</td>
</tr>
<tr>
<td>21</td>
<td>0.0029</td>
<td>0.0027</td>
<td>0.0019</td>
<td>0.0023</td>
<td>0.0128</td>
<td>0.0147</td>
<td>0.0090</td>
<td>0.0028</td>
<td>0.0008</td>
<td>0.0003</td>
<td>0.0001</td>
</tr>
<tr>
<td>22</td>
<td>0.0076</td>
<td>0.0119</td>
<td>0.0074</td>
<td>0.0053</td>
<td>0.0109</td>
<td>0.0129</td>
<td>0.0085</td>
<td>0.0033</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0001</td>
</tr>
<tr>
<td>23</td>
<td>0.0368</td>
<td>0.0409</td>
<td>0.0242</td>
<td>0.0094</td>
<td>0.0089</td>
<td>0.0041</td>
<td>0.0030</td>
<td>0.0015</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>24</td>
<td>0.0643</td>
<td>0.0706</td>
<td>0.0420</td>
<td>0.0144</td>
<td>0.0073</td>
<td>0.0049</td>
<td>0.0023</td>
<td>0.0005</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>25</td>
<td>0.1016</td>
<td>0.1198</td>
<td>0.0702</td>
<td>0.0272</td>
<td>0.0082</td>
<td>0.0018</td>
<td>0.0011</td>
<td>0.0003</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>26</td>
<td>0.0373</td>
<td>0.0457</td>
<td>0.0228</td>
<td>0.0095</td>
<td>0.0020</td>
<td>0.0008</td>
<td>0.0003</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>27</td>
<td>0.0900</td>
<td>0.0109</td>
<td>0.0058</td>
<td>0.0021</td>
<td>0.0008</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>28</td>
<td>0.0008</td>
<td>0.0015</td>
<td>0.0010</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 3.5: Posterior distribution of cluster models for the microarray data.
Figure 3.4: Selection of row clusters from the MAP clustering. Each plot corresponds to a different row cluster. Each profile (black line) gives the gene expression level for over all conditions. Conditions are arranged by condition cluster membership. The condition clusters are separated by the red dashed line.
real data example, the sampler was used for analysis of a DNA microarray experiment. This demonstrated that there can be considerable uncertainty in the number of clusters in certain situations. Knowing even the range of possible models may be difficult. Results from the microarray experiment demonstrated that clusters could be found by a search strategy with a probabilistic basis using the collapsed LBM sampler. Overall, the approach seems to be a robust way to block cluster a data matrix. The user need only specify prior hyperparameters and priors on the number of clusters.
Chapter 4

Changepoint modelling using
Gaussian Markov random fields

4.1 Introduction

Gaussian Markov random fields (GMRFs) are widely used tools in spatial statistics and image analysis. Although there are many possibilities for GMRFs, at the most basic level, a GMRF is a multivariate Gaussian random variable whose entries follow a conditional independence structure specified by the precision (inverse covariance) matrix. A GMRF is characterized by its conditional independence structure. GMRFs are often used in hierarchical models to model dependency at the low level while conditional on the field, at the top level, observed data is assumed to arise independently from some distribution. Dependency in the data is then accounted for by the field. The applicability of such models is by no means restricted to spatial statistics and image analysis. Rue & Held (2005) and Rue et al. (2009) have demonstrated that such models can be applied in the analysis of structural time-series, longitudinal data, graphical models, and others.

A recent major advance in the analysis of hierarchical GMRF models is the methodology proposed by Rue et al. (2009). This methodology provides computationally efficient approximations for GMRF posteriors, which have been demonstrated to out-
perform MCMC in certain situations (Rue et al. 2009), giving high accuracy approximations. These approximations are termed integrated nested Laplace approximations or INLAs. An obvious advantage to such approximations is that they avoid lengthy MCMC runs to fully explore the posterior support and they also avoid the need to demonstrate that these runs have converged. Another advantage is that the approximations may be used to estimate quantities such as the marginal likelihood of the data under a given GMRF model. This is key to the central ideas of this chapter.

Chapter 2 discussed how changepoint models could be analyzed using MCMC or the filtering recursions approach of Fearnhead (2006). A slight disadvantage of the models discussed in that chapter was their simplicity. Such simple models can not account for time dependency in the data (where “time” refers to some natural ordering of the data). However, in many situations it would be unreasonable not to include a longitudinal aspect in the data model if it is possible. Take for example the coal-mining disasters data discussed in Section 2.7 and later in Section 4.6. This data records the occurrences of disasters on a weekly basis over a period of 112 years. One may expect there to be week-to-week dependency, possibly due to changes in practice, industrial disputes and other day-to-day business.

However, fitting these models with dependency in an MCMC setting, along with a reversible jump MCMC search for the most likely number of changepoints would be a formidable computing task. This chapter explores the possibility of analysing such changepoint models outside the MCMC setting by combining the INLA methodology of Rue et al. (2009) and the filtering recursions approach for changepoints from Fearnhead (2006). This is done by replacing the marginal likelihoods necessary to compute the filtering recursions with the approximated marginal likelihood computed by INLAs. This leads to the potential for many new and useful changepoint models.

The remainder of this chapter is organised as follows. Hierarchical GMRF models and the INLAs of Rue et al. (2009) are explained more in Section 4.2. Section 4.3 gives a brief review of recursions for performing inference conditional on a particular number of changepoints as given in Fearnhead (2006). In Section 4.4 possible computational
difficulties are discussed and solutions for these are proposed. Sections 4.5, 4.6 and 4.7 analyze real data examples; analysis of data arising from comparative genomic hybridization studies; the coal-mining data is re-analyzed using a model with dependency and this is compared with the original analysis; the Well-log data is reanalyzed with a more realistic model. Section 4.8 explores the possibility of detecting changepoints under the assumption of a stochastic volatility model. The chapter concludes with a discussion.

4.2 Hierarchical GMRF models

4.2.1 General model definition

The hierarchical GMRF model considered here has two levels. At the low level, a vector of hyperparameters \( \theta_1 \) specifies a distribution for the GMRF \( x \). At the top level the data \( y \) is assumed to be conditionally independent given the latent field \( x \). In many typical examples, \( y \) and \( x \) will not be of the same dimension. For example, \( x \) could include an additional common intercept term in a generalized linear regression model. This intercept would be given a Gaussian prior and would be a member of the GMRF. Since this is the case, the elements of \( x \) that correspond directly to an element of \( y \) are indexed by the set \( C \) and also \( y = \{ y_i : i \in C \} \). Further, the dimension of \( x \) is \( n_x \) and that of \( y \) is \( n \), the number of observed data points. The distribution of \( y \) may depend on another hyperparameter \( \theta_2 \). The hyperparameters are given independent priors \( \pi(\theta_1), \pi(\theta_2) \). The GMRF may be written as follows:

\[
\begin{align*}
\theta_1 &\sim \pi(\theta_1) \\
\theta_2 &\sim \pi(\theta_2) \\
x &\sim \pi(x|\theta_1) \\
y_i &\sim_{\text{ind}} \pi(y_i|x_i, \theta_2), \quad i \in C.
\end{align*}
\]
The joint posterior of the latent field and hyperparameters is

$$
\pi(x, \theta_1, \theta_2 | y) \propto \pi(\theta_1) \pi(\theta_2) \pi(x | \theta_1) \prod_{i \in C} \pi(y_i | x_i, \theta_2).
$$

To simplify the notation $\theta_1$ and $\theta_2$ are amalgamated into $\theta$ to give

$$
\pi(x, \theta | y) \propto \pi(\theta) \pi(x | \theta) \prod_{i \in C} \pi(y_i | x_i, \theta).
$$  \hspace{1cm} (4.1)

Here, $\pi(x | \theta) = N_n(0, Q^{-1}(\theta))$ due to the GMRF structure, where $Q^{-1}(\theta)$ is the inverse covariance matrix, also known as the precision matrix. The posterior (4.1) can be written in the full form:

$$
\pi(x, \theta | y) \propto \pi(\theta) |Q(\theta)|^{1/2} \exp \left\{ -\frac{1}{2} x^T Q(\theta) x + \sum_{i \in C} \log \pi(y_i | x_i, \theta) \right\}.
$$  \hspace{1cm} (4.2)

Usually the matrix $Q(\theta)$ will be quite sparse due to the conditional independence structure of the GMRF. This is exploited in computations for the INLA approximation of Rue et al. (2009) who use sparse matrix computations. This gives a huge computational advantage, since $Q(\theta)$ is often of large dimension and it is inefficient to work with it directly.

A well known example of a sparse precision matrix is that for a first order autoregressive, AR(1) model. For this model, the latent field $x$ is defined by

$$
x_1 \sim N(0, \sigma^2/(1 - \phi^2))
$$

and

$$
x_i = \phi x_{i-1} + \varepsilon_i, \hspace{0.5cm} i > 1
$$

where $\varepsilon_i \sim N(0, \sigma^2)$, so that $x \sim N_n(0, \Sigma)$. In this situation, $x_i$ will be conditionally independent of $x_j$ when $|i - j| > 1$. This is since, conditioning on the rest of the field being known, $x_i$ is only determined by $x_{i-1}$ and $x_{i+1}$. This does not mean however that $x_i$ is not correlated with the rest of the field. It can be shown that

$$
\text{Corr}(x_i, x_j) = \frac{\phi^{|i-j|}}{1 - \phi^2}
$$
which means that \( \Sigma \) will be completely dense. It is also possible to show that \( Q(\theta) = \Sigma^{-1} \) has entries

\[
Q_{ij} = \begin{cases} 
-\phi/\sigma^2 & |i - j| = 1 \\
1/\sigma^2 & |i - j| = 0 \\
0 & |i - j| > 1 
\end{cases}
\]

thus verifying the conditional independence properties of the field. Moreover, this matrix is now very sparse. Only \( 3n_x - 2 \) of the \( n_x^2 \) entries are non-zero. Sparse matrix computations can be used on this matrix instead of working with the dense matrix \( \Sigma \) in which all \( n^2 \) entries are non-zero.

### 4.2.2 Integrated nested Laplace approximations

The quantity which is of main interest for the purposes of this chapter is the marginal likelihood of the data \( \pi(y) \). From the hierarchical form of the GMRF model

\[
\pi(x, \theta | y) \propto \pi(\theta) \pi(x | \theta) \pi(y | x, \theta).
\]

In addition, one has the relation

\[
\pi(x, \theta | y) \propto \pi(x | \theta, y) \pi(\theta | y)
\]

where \( x \) has been integrated out in the second term on the right hand side. Thus,

\[
\pi(\theta | y) \propto \frac{\pi(x, \theta, y)}{\pi(x | \theta, y)}
\]

where \( \pi(x, \theta, y) = \pi(\theta) \pi(x | \theta) \pi(y | x, \theta) \). Since it will rarely be possible in practice to compute \( \pi(\theta | y) \) analytically, the idea of INLAs is to build up an approximation \( \tilde{\pi}(\theta | y) \) of the marginal posterior of \( \theta \) as

\[
\tilde{\pi}(\theta | y) \propto \frac{\pi(x, \theta, y)}{\pi_c(x | \theta, y)} \bigg|_{x = x^*(\theta)},
\]

where \( \pi_c(x | \theta, y) \) is the Gaussian approximation to the full conditional of \( x \), and \( x^*(\theta) \) is the mode of the full conditional for \( x \), for a given value of \( \theta \).
An approximation to the marginal likelihood, \( \hat{\pi}(y) \), can be obtained from the approximation (4.3) as the normalizing constant of \( \hat{\pi}(\theta | y) \),

\[
\hat{\pi}(y) = \int \frac{\pi(x, \theta, y)}{\hat{\pi}_G(x | \theta, y)} \bigg|_{x=x^*(\theta)} \, d\theta.
\]

The approximation of \( \pi(y) \) is completed by numerically integrating over a grid of ‘suitable’ \( \theta \) values to give

\[
\hat{\pi}(y) \approx \sum_k \pi(x_k, \theta, y) \bigg|_{x=x^*(\theta_k)} \times \Delta_k
\]

where \( \Delta_k \) represents the volume element of \( \theta \) space attached to grid point \( \theta_k \). It should be noted here that the Gaussian approximation \( \hat{\pi}_G(x | \theta, y) \) and modal value \( x^*(\theta) \) must be computed for each grid point \( \theta_k \). This leads to the main computational overhead in the approximations, as will be discussed in the following sections.

4.2.3 Gaussian approximations

The approximation (4.3) has the Gaussian approximation to \( \pi(x | \theta, y) \) as a main ingredient. This density will be of the form

\[
f(x) \propto \exp \left\{ -\frac{1}{2} x^T Q x + \sum_{i \in C} h_i(x_i) \right\}
\]

where \( h_i(x_i) = \log \pi(y_i | x_i, \theta) \) with the dependence on \( \theta \) suppressed (for convenience). To obtain the Gaussian approximation, expand \( h_i(x_i) \) about some point \( x_i^{(0)} \) to the second order,

\[
h_i(x_i) \approx h_i(x_i^{(0)}) + a_i x_i - \frac{1}{2} b_i x_i^2.
\]

The constants \( a_i, b_i \) depend on the value of \( x_i^{(0)} \). Using these expansions,

\[
-\frac{1}{2} x^T Q x + \sum_{i \in C} h_i(x_i) \approx -\frac{1}{2} x^T Q x + \sum_{i \in C} \left[ h_i(x_i^{(0)}) + a_i x_i - \frac{1}{2} b_i x_i^2 \right]
\]

\[
= -\frac{1}{2} x^T Q x + a^T x - \frac{1}{2} x^T \text{diag}(b)x + \sum_{i \in C} h_i(x_i^{(0)})
\]

\[
= -\frac{1}{2} x^T [Q + \text{diag}(b)]x + a^T x + \sum_{i \in C} h_i(x_i^{(0)}).
\]

83
This shows that the precision matrix of the Gaussian approximation will be \( Q + \text{diag}(b) \).
To find the mode \( \mu \), the quadratic forms may be matched:

\[
(x - \mu)^T (Q + \text{diag}(b))(x - \mu) = x^T (Q + \text{diag}(b))x - 2\mu^T (Q + \text{diag}(b))x + \text{constant}
\]

to give

\[
[Q + \text{diag}(b)]\mu = a. \tag{4.4}
\]

The above optimizations can be done by iteratively

- solving for \( \mu \)
- updating \( a \) and \( b \)

until convergence.

The large computational overhead in solving for \( \mu \) at each iteration is reduced drastically by the use of sparse matrix computations in INLAs. These can be employed to factorize the precision matrix into its Cholesky decomposition, \( Q + \text{diag}(b) = LL^T \), at each iteration as the sparseness of \( Q + \text{diag}(b) \) is inherited by \( L \). Thus, terms which are known to be zero in \( L \) need not be computed. Note that the sparseness in \( Q \) remains unchanged by adding \( \text{diag}(b) \) at each step, as this only contributes to the non-zero diagonal entries. See Rue et al. (2009) and Rue & Held (2005) for more details of these calculations in the context of GMRFs.

### 4.2.4 Inference for the field and the hyperparameters

It is also possible to use INLAs to approximate the marginal posteriors of the field and hyperparameters. In the case of the marginal posteriors of the field, this requires an extra level of approximation. The marginals are

\[
\pi(x_i | y) = \int \pi(x_i | \theta, y) \pi(\theta | y) \, d\theta \\
\pi(\theta_j | y) = \int \pi(\theta | y) \, d\theta_{-j}
\]
and their approximations may be written
\[
\tilde{\pi}(x_i|y) = \int \tilde{\pi}(x_i|\theta, y) \tilde{\pi}(\theta|y) d\theta
\]
\[
\tilde{\pi}(\theta_j|y) = \int \tilde{\pi}(\theta|y) d\theta_{-j}.
\]

The approximation \(\tilde{\pi}(x_i|\theta, y)\) is obtained from a simplified version of a Laplace approximation (see Rue et al. (2009) for more details). The Laplace approximation is given by
\[
\tilde{\pi}(x_i|\theta, y) \propto \frac{\pi(x, \theta, y)}{\tilde{\pi}_{GG}(x_{-i}|x_i, \theta, y)}|_{x_i = x_{i*}(x_i, \theta)}
\]
where \(\tilde{\pi}_{GG}\) is a Gaussian approximation to the distribution \(\pi(x_{-i}|x_i, \theta, y)\) and \(x_{i*}(x_i, \theta)\) is the modal configuration of \(x_{-i}\) for a given value of \(x_i\) and \(\theta\). The simplified version of this approximation is necessary since the Gaussian approximation to \(\tilde{\pi}(x_{-i}|x_i, \theta, y)\) must be computed for each \(x_i\) and \(\theta\), thus implying a large computational overhead if computing density estimates for \(x_i\).

The marginal posteriors for the \(\theta_j\) can be obtained by numeric integration on a suitable grid of points \(\theta_k\):
\[
\tilde{\pi}(\theta_j|y) \approx \sum_k \tilde{\pi}((\theta_j, \theta_{-j,k})^{T}|y) \times \Delta_k.
\]
where \(\theta_{-j,k}\) is \(\theta_k\) with the \(j^{th}\) element removed. The marginal posteriors for the \(x_i\) require an extra level of approximation to obtain \(\tilde{\pi}(x_i|\theta, y)\) and then numeric integration on a suitable grid. The reader is referred to Rue et al. (2009) for details of suitable approximations to \(\tilde{\pi}(x_i|\theta, y)\).

4.2.5 The R-INLA package

The R-INLA package Rue et al. (2009) for R-2.11.1 may be used to do all of the calculations above for a range of GMRF hierarchical models. It aims to give an off-the-shelf tool for INLAs. Currently the package implements many exponential family models; Gaussian with identity-link; Poisson with log-link; Binomial with logit-link; for many different temporal GMRFs; random effects models; first order auto-regressive; first and
second order random walk (neither of these lists are exhaustive!). The package also implements spatial GMRFs in two and three dimensions and is currently still evolving with new additions on a regular basis. Use of this package avoids programming for specific models as it allows the selection of any observational data model and selection of the desired GMRF through a one line call to the R-INLA package. The R-INLA package is used for all the computations on hierarchical GMRF models in this chapter.

4.3 Changepoint models

Chapter 2 outlined the exact inference for changepoint models using filtering recursions following Fearnhead (2006). The analysis there focused on using a Markov point process prior for the number and position of the changepoints. It was demonstrated that the posterior distribution may sometimes be sensitive to the choice of the parameters for the point process. In this chapter the focus will be more on performing inference for a given number of changepoints. Denote \( k \) ordered changepoints by \( \tau_1, \ldots, \tau_k \). The prior taken on changepoints is assumed to have the product form

\[
\pi^{cp}_k(\tau_1, \ldots, \tau_k) = \prod_{j=0}^{k} \pi^{cp}_k(\tau_j|\tau_{j+1}).
\]

where \( \tau_0 = 0, \tau_{k+1} = n \). Note that this prior is conditional on a given number of changepoints, \( k \). The idea is to introduce a prior on \( k \) and use the hierarchical form

\[
\pi(k|\mathbf{y}) \propto \pi(\mathbf{y}|k \text{ changepoints})\pi(k)
\]

to find the most likely number of changepoints. Using this, the most likely positions for the changepoints can then be found.

4.3.1 Recursively computing the posterior

Let \( L_j^{(k)}(t) = \Pr(\mathbf{y}_{t:n}|\tau_j = t - 1 \text{ and } k \text{ changepoints}) \). It is possible to compute \( L_j^{(k)}(t) \) in a backward recursion;

\[
L_j^{(k)}(t) = \sum_{s=t}^{n-k+j} P(t, s)L_{j+1}^{(k)}(s + 1)\pi^{cp}_k(\tau_j = t - 1|\tau_{j+1} = s)
\]
with \( j \) going from \( k \) to 1 and \( t \) going from \( n - k + j - 1 \) to \( j + 1 \), where \( P(t, s) = \pi(y_{t:s}) \) is the marginal likelihood of the segment \( y_{t:s} \). The marginal likelihood of \( y_{1:n} (= y) \) under a \( k \) changepoint model may be computed as

\[
\Pr(y_{1:n}|k \text{ changepoints}) = \sum_{s=1}^{n} P(1, s)L_{1}^{(k)}(s + 1)\pi_{k}^{p}(\tau_1 = s). \tag{4.6}
\]

### 4.3.2 Choice of changepoint prior and computational cost

It will be necessary to compute the marginal likelihood of the data under a \( k \) change-point model for a range of possibilities, say \( k = 0, \ldots, K \) in order to do inference for \( k \) using (4.5). This requires computational effort in \( O(n^2K^2) \) and storage requirements in \( O(nK^2) \) which could be costly. Both of these may be reduced by choosing an appropriate changepoint prior. One such prior, as used and noted by Fearnhead (2006), is to take changepoint positions distributed as the even numbered order statistics of \( 2k + 1 \) uniform draws from the set \( \{1, \ldots, n - 1\} \) without replacement. Doing this gives

\[
\pi_{k}^{cp}(\tau_1, \ldots, \tau_k) = \frac{1}{Z_k} \prod_{j=0}^{k} \delta(\tau_j | \tau_{j+1})
\]

where \( \delta(s|t) = t - s - 1 \) (see the Appendix A for details) and the normalizing constant \( Z_k = \binom{n-1}{2k+1} \). Using this prior restricts the dependence of the prior on the number of changepoints to the normalizing constant only, meaning that

\[
L_{j+r}^{(k+r)}(t) = \sum_{s=t}^{n-[k+r-(j+r)]} P(t, s)L_{j+r+1}^{(k+r)}(s + 1)\delta(\tau_{j+r} = t - 1|\tau_{j+r+1} = s)
\]

\[
= \sum_{s=t}^{n-k+j} P(t, s)L_{j+r+1}^{(k+r)}(s + 1) \times (s - t)
\]

\[
= \sum_{s=t}^{n-k+j} P(t, s)L_{j+1}^{(k)}(s + 1) \times (s - t) = L_{j}^{(k)}(t).
\]

Reusing these values gives a reduction by a factor of \( K \) in computational effort and storage requirements. The recursions are now

\[
L_{j}^{(k)}(t) = \sum_{s=t}^{n-k+j} P(t, s)L_{j+1}^{(k)}(s + 1)\delta(\tau_j = t - 1|\tau_{j+1} = s) \tag{4.7}
\]
Then (4.8) is divided by $Z_k$ to correctly normalize the prior and (4.5) is obtained by multiplying this by the prior weight for $k$ changepoints $\pi(k)$. This prior will be used in the examples later.

4.3.3 Posterior of any changepoint

Since the prior on changepoints makes the changepoint model factorizable, it is possible to write down the posterior distribution of $\tau_j$ conditional on $\tau_{j-1}$ and $k$;

$$
\text{Pr}(\tau_j | \tau_{j-1}, y_1:n, k \text{ changepoints}) \propto P(\tau_j + 1, \tau_j)L_j^{(k)}(\tau_j + 1)\delta(\tau_{j-1} = s)/L_{j-1}^{(k)}(\tau_{j-1} + 1).
$$

This is used for the forward simulation of changepoints once the backward recursions have been computed. It can also be used to give the modal changepoint configuration as in the examples later.

4.4 Approximate changepoint inference using IN-LAs

As mentioned earlier, the main idea of this chapter is to allow for segment models with dependency for multiple changepoint problems. The essential ingredient is to replace the segment marginal likelihood $P(t, s)$ in the recursions

$$
L_j^{(k)}(t) = \sum_{s=t}^{n-k+j} P(t, s)L_{j+1}^{(k)}(s + 1)\delta(\tau_j = t - 1|\tau_{j+1} = s)
$$

with a segment marginal likelihood approximated using INLA. It is the case that $P(t, s)$ needs to be available in closed form to use a filtering recursions approach. This will never be the case for hierarchical GMRF models, which can account for within segment dependency. However, INLAs can be used to get a good approximation to $P(t, s)$ for hierarchical GMRF segment models. This opens up the opportunity for more realistic
data models in many cases. There are also two other advantages; the posterior of the number of changepoints may be well approximated for model selection; the posterior of any given changepoint can be computed to a high degree of accuracy.

There are two potential drawbacks of the proposed approach however. The first is that it usually would not make sense to fit a GMRF model to a very small amount of data. For example, at least five data points would be required to make fitting a first order auto-regressive random field feasible. This means that for the approach to be reasonable it may be necessary to expect changepoints to be quite well separated. The second potential drawback contrasts with the first. For large amounts of data, using INLAs to compute the $\frac{n(n+1)}{2}$ segment marginal likelihoods necessary to compute the recursions (4.7) could be costly. The next section proposes a way to overcome both of these problems simultaneously, while still retaining almost all of the advantages of using a filtering recursions approach. This proposed solution is termed reduced filtering recursions for changepoints (RFRs).

### 4.4.1 Reduced filtering recursions for changepoints

The main idea of RFRs is to use all the data, but to do recursions on a smaller portion of it, in order to approximate the full recursions (4.7). What is meant by this is that the recursion is not computed at every data point which takes $O(n^2)$ computation. Generally if segments have a reasonable duration, changepoints can be detected in the region where they have occurred. The change in regime will be detectable for a period after the actual changepoint position, possibly until a time is reached where the support for a one segment model may be greater than that for a two segment model. An analysis using RFRs only permits a changepoint to occur at some point in the reduced time index set $\{t_1, \ldots, t_N\}$ with $t_i < t_j$ for all $i < j$. For convenience, define $t_0 = 0$ and $t_{N+1} = n$. So to clarify, the assumption is that if there is a changepoint between $t_i$ and $t_{i+2}$ it can be detected at $t_{i+1}$. The spacing of the $t_i$ is clearly an important issue. If the spacing is too wide, then changepoints will not be detected. If the spacing is too narrow, many points are required for the reduced time index set to cover the
entire data, consequently increasing the computation time. The most natural choice is to take equally spaced points if there is little prior knowledge of where changepoints occur. This corresponds to $t_i = ig$ for some choice of $g$. The following example briefly explores the choice of $g$ and makes the preceding discussion clearer.

Consider the data simulated from a Gaussian changepoint model shown at the top of Figure 4.1(a) with a clear change at 97. Searching for one changepoint, the bottom three plots in Figure 4.1(a) show the posterior probability of a changepoint for reduced time index sets given by $g = 1, 5, 10$. Note that $g = 1$ corresponds to the original recursions (1.7). For $g = 5$ the changepoint is detected at 95 and $g = 10$ detects it at 100. In both cases the changepoint is identified as the closest possible point in the reduced time index set to its actual position. Figure 4.1(b) shows a similar example, where this time one of the segments is very short (only 13 points). Again, the changepoint is identified at the closest possible position in the cases of $g = 1, 5$. In the case of $g = 10$ it is the second closest, possibly due to the noise in the data contaminating the separation of the two regimes.

**Recursions on the reduced time index set**

The changepoints are $\tau_1, \ldots, \tau_k$. The reduced time index set is $\{t_1, \ldots, t_N\}$. The changepoint prior is now defined on the set of numbers $\{1, \ldots, N\}$ and we let $c_j = r$ if $\tau_j = t_r$. That is, $c_j$ corresponds to the changepoint position if time is indexed by $\{1, \ldots, N\}$ whereas $\tau_j$ gives the changepoint position in the reduced time index set $\{t_1, \ldots, t_N\}$. Define

$$R_j^{(k)}(r) = \Pr(y_{t_r+1:n}|\tau_{j-1} = t_r, k \text{ changepoints}).$$

For $r = N, \ldots, k + 1$

$$R_k^{(k)}(r) = P(t_r + 1, n) \delta(c_k = r|c_{k+1} = N + 1).$$

Then recursively, for $j = k - 1, \ldots, 1$ and $r = N - k + j - 1, \ldots, j + 1$

$$R_j^{(k)}(r) = \sum_{s=r+1}^{N-k+j} P(t_r + 1, t_s) R_{j+1}^{(k)}(s) \delta(c_j = r|c_{j+1} = s).$$
Figure 4.1: Results when searching for one changepoint in simulated Gaussian data for $g = 1, 5, 10$. It can be seen that the changepoint is detected at one of its closest neighbouring points in the reduced time index set.
After computing these, the approximate marginal likelihood of the data conditional on $k$ changepoints follows as,

$$Pr(y_{1:n}|k \text{ changepoints}) \approx \frac{1}{N-k} \sum_{s=1}^{N-k} P(1, t_s) R^{(k)}_1(s) \delta(c_0 = 0|c_1 = s)/Z_k.$$ 

Once the grid spacing $g$ is not too large, the approximation to the marginal probability of $k$ changepoints should be reasonable for the competing models. There are many computational savings with this approach. Using the RFRs decreases the number of marginal likelihood evaluations required to $n_r(n_r + 1)/2$ where

$$n_r = \lceil n/g + 1 - I(g = 1) \rceil.$$ 

See Appendix C for details of this calculation.

**Distribution of any changepoint**

When the maximum a posteriori number of changepoints has been found, it is determined where the changepoints are most likely to occur on the reduced time index set. The distribution of $c_j$ is

$$Pr(c_j|c_{j-1}, y_{1:n}, k) \propto P(t_{c_{j-1}} + 1, t_{c_j}) R^{(k)}_j(c_j) \delta(c_j|c_{j+1})/R^{(k)}_{j-1}(c_{j-1}). \quad (4.9)$$

Instead of generating samples of changepoints, our focus is to deterministically search for the most probable changepoint positions a posteriori. The first changepoint detected on the reduced time index set will be

$$\hat{c}_1 = \arg \max_{c_1} Pr(c_1|c_0 = 0, y_{1:n}, k).$$

Conditioning on $\hat{c}_1$ the search proceeds for $c_2, \ldots, c_k$ in the same way. In general,

$$\hat{c}_j = \arg \max_{c_j} Pr(c_j|\hat{c}_{j-1}, y_{1:n}, k).$$

This procedure is repeated until the $k$ changepoints $t_{\hat{c}_1}, t_{\hat{c}_2}, \ldots, t_{\hat{c}_k}$ are found.
Refining changepoint detection

Following detection of changepoints on the reduced time index set, it is possible to refine the search and hone in on the most likely position of the changepoint. To begin, the changepoints obtained from the search above, \( \tau^{(0)}_1, \ldots, \tau^{(0)}_k \) where \( \tau^{(0)}_j = t_{c_j} \), will all be multiples of \( g \). Condition on the value of \( \tau^{(0)}_2 \) to update \( \tau^{(0)}_1 \). Compute

\[
P(1, \tau)P(\tau + 1, \tau^{(0)}_2)
\]

using INLAs for \( \tau \in \{\tau^{(0)}_1 - g + 1, \ldots, \tau^{(0)}_1 + g - 1\} \). Then take \( \tau^{(1)}_1 \) to be the \( \tau \) which maximizes this. Similarly \( \tau = \tau^{(1)}_j \) maximizes

\[
P(\tau^{(1)}_j, \tau)P(\tau + 1, \tau^{(0)}_{j+1}).
\]

This procedure can be carried out just once, or repeated until there is no difference between updates.

This step does of course mean additional computation. It may not be necessary in all cases to carry out a refined search. For example, the case of large \( n \) and small \( g \) would mean that refining the search would probably give little additional information.

Exploring approximation error and computational savings in a DNA segmentation example

To get a rough idea of the approximation error and the possible computational savings to be made by using RFRs, the methods were applied in a DNA segmentation task with a conditional independence model. This deviates from the general theme of the chapter (to fit models relaxing conditional independence), however, it is included to offer some insight into RFRs in general.

DNA sequence data is a string of the letters A,C,G and T representing the four nucleic acids, adenine, cytosine, guanine and thymine. Interest focuses on segmenting the sequence into contiguous segments characterized by their C+G content. It is assumed that within a segment the frequency of constituent acids follows a multinomial
distribution, so that
\[ \pi(y_{t:s} | \theta) = \prod_{i=t}^{s} \theta_{A}^{I(y_i=A)} \theta_{C}^{I(y_i=C)} \theta_{G}^{I(y_i=G)} \theta_{T}^{I(y_i=T)}. \]

With a Dirichlet(\(\alpha, \alpha, \alpha, \alpha\)) prior on \(\theta_{(t:s)}\) the marginal likelihood for a segment is
\[ P(t, s) = \frac{\Gamma\{4\alpha\}}{\Gamma\{\alpha\}^4 \Gamma\{s - t + 1 + 4\alpha\}} \prod_{j \in \{A,C,G,T\}} \Gamma\{n_{j}^{(t:s)} + \alpha\} \]

where \(n_{j}^{(t:s)}\) is the number of occurrences of acid \(j \in \{A,C,G,T\}\) in the segment from \(t\) to \(s\) inclusive.

The data analyzed is the genome of a parasite of the intestinal bacterium \textit{Escherichia coli}. The sequence consists of 48,502 base pairs, and so will provide a good measure of the computational savings to be made for larger datasets when using RFRs. This data has previously been analyzed by Boys & Henderson (2004), who implemented a hidden Markov model using RJMCMC to select the Markov order within a segment. Here however, a changepoint model assuming data in segments are independent is applied. Cumulative counts of the nucleic acids over location along the genome are shown in Figure 4.2.

The RFRs were applied to this data using an equally spaced reduced time index set with \(g = 1, 5, 10, 15, 20, 25\). The prior taken on the number of changes was uniform on \(\{0, 1, \ldots, 20\}\). All runs were on a 2.5GHz processor implemented in the C programming language, and the segment marginal likelihoods calculated in a step before the recursions were computed. Table 4.1 gives the identified changepoints and the computing time for each analysis. The value \(g = 1\) corresponds to filtering recursions on the entire data. It can be seen that using RFRs does not appear to have a considerable effect on the detected changepoints. However, there are drastic differences in computing time; the RFRs for \(g = 25\) give a 450 fold decrease in computing time with respect to recursions on the full data set.
Figure 4.2: Cumulative counts of A,C,G,T for the DNA data. Identified changepoints are overlain (vertical lines).
Table 4.1: Location of changepoints and computing time for DNA segmentation example. As $g$ increases there is little deviation in changepoint estimates. Reported changepoints are found after a refined search.
4.5 CGH studies of Coriel cell lines

Comparative genomic hybridization or CGH studies are used to detect chromosomal aberrations in the genome in tumor tissue. Two tissue samples, one tumor and the other healthy, are dyed with different fluorochromes (red and green). The two samples are then mixed together. Aberrations present in the tumor DNA are detected by examining the colour of the fluorescence emitted by the mixture of the two samples. A yellow fluorescence indicates that there have been no amplifications or deletions in the tumor sample. If the healthy tissue has been dyed red or green however, and the mixture of the samples emits a red or green fluorescence, then there has been chromosomal aberrations in the tumor tissue. After dying and mixing the tissue samples the emitted fluorescence is translated into DNA copy number.

The data studied here is chromosome 11 of Coriel.05296 which has been previously studied by Erdman & Emerson (2007) and Fridlyand, Snijders, Pinkel, Albertson & Jain (2004). The data is available from the R package bcp (Erdman & Emerson 2007). In this type of application, analysis is usually carried out on the log-to-base-two ratio of the red-green intensities obtained from the DNA copy numbers from the fluorescence experiments. The data is shown in Figure 4.3 for chromosome 11 \((n = 185)\). The task is to detect changepoints in this series. There has been some pre-processing of this particular data to remove points with a negligible level of intensity and to correct for background noise. The specifics of this pre-processing are described in Section 3.1 of Erdman & Emerson (2007). Even after pre-processing, this data can still be prone to outliers or “short-lived changes” which can be attributable to false signals or a true signal on a single strand of DNA.

Erdman & Emerson (2007) compare different changepoint analyses for this data. They use models which do not explicitly account for dependence in the chromosome. Fridlyand et al. (2004) use a Hidden Markov Model to account for the spatial dependency in the copy number values. Following Fridlyand et al. (2004), a GMRF model is fitted to account for dependency in the series of copy numbers. The observational data is assumed
Gaussian with some variance $\sigma^2_y$ and mean $\mu$ given by identity link to an AR(1) field. To be more specific,

$$y_i \sim N(\mu_i, \sigma^2_y)$$

where $\mu_i = \alpha + x_i$. The parameter $\alpha$ is an intercept and

$$x_i = \phi x_{i-1} + \varepsilon_i, \quad i = 2, \ldots, n$$

where $\varepsilon_i \sim_{\text{iid}} N(0, \sigma^2_x)$. The definition is completed by assuming the marginal distribution of $x_1$ is $N(0, \sigma^2_x/(1 - \phi^2))$.

There are a few advantages with this model; the mean copy number is modelled at each location by the field; the model exploits the spatial dependence along the chromosome; this model will be more robust to outliers, as the field can model extra intra segment variability. In addition to allowing dependence between neighbouring locations along the chromosome, the approach adopted allows for changes in this dependence pattern across different segments through the persistence parameter ($\phi$) of the AR(1) field. A drawback of the approach in general is that both changepoints corresponding to small segments (with length less than about three) may not be detected or correctly located. In many applications however, this is usually not an issue, as segments of duration less than three would generally not be expected. For a discussion on minimum segment duration and priors in this context, see Girón et al. (2007).

Using the R-INLA package requires choosing parameters for the priors of $\sigma^2_y, \sigma^2_x$ and $\phi$. A Gamma prior is used for $\sigma^2_y$ and $\sigma^2_x$. The prior on $\phi$ is specified through $\kappa = \logit \left( \frac{1 + \phi}{2} \right)$ where $\kappa \sim N(\mu_\kappa, \sigma^2_\kappa)$. Priors which are too diffuse could cause problems in approximating marginal likelihoods for small segments, or may demand lots of gridding in $\theta$ to get accurate approximations. This problem is not exclusive to INLAs. As noted by Kass & Raftery (1995) Bayes factors (and marginal likelihoods), tend to be sensitive to the choice of priors on the parameters. These issues can be overcome by doing an initial INLA on the entire data using relatively non-informative priors. Constructing the marginal posterior distributions of $\sigma^2_y, \sigma^2_x$ and $\phi$ from this should give a good guide for choosing the prior parameters. This procedure could be iterated a few times.
until there is no change in the marginal posterior distributions. In addition, one may use the mode of $\theta$ from the previous iteration as an initial value for the optimization in the next iteration. This should speed up the convergence of the Newton-Raphson scheme for finding the modal configuration of the GMRF. For the Coriel data, this procedure led to the priors

$$
\begin{align*}
\sigma_y^{-2} & \sim \text{Gamma}(75, 0.5) \\
\sigma_x^{-2} & \sim \text{Gamma}(15, 0.1) \\
\kappa & \sim \text{N}(2, 1).
\end{align*}
$$

The prior on changepoints was taken to be uniform on the integers $\{0, \ldots, 5\}$.

The results were obtained by running R-INLA on a 2.5GHz processor. The value of $g$ was taken to be 5. The approximate marginal likelihoods took just over 10 minutes to compute. There was overwhelming support for a two changepoint model. The posterior probability for this was 0.996. Changepoints were found at 50 and 65. A refined search then moved these to 51 and 66. Conditional on these changepoints, inference was performed for the segment fields. This is shown in the bottom of Figure 4.3. It can be seen that the AR(1) field gives a very good fit to the data. It is possible to assess qualitative differences between the three segments by comparing the approximated marginal posteriors of $\sigma_y^{-2}, \sigma_x^{-2}$ and $\phi$ (from INLAs) shown in Figure 4.4. It appears that the segment between 51 and 66 has more posterior support for a larger persistence parameter (mode is about 0.75 compared with 0.375). This segment also appears to have larger variance for the observations and the field, due to the noisier observations. Overall the segment from 66 to 185 is less noisy. This could be due to the larger number of data points used to fit the field.

### 4.6 Coal mining disasters

The coal mining disasters data which was discussed in Chapter 2 is now examined again. Recall that this data has been analyzed in Fearnhead (2006), Yang & Kuo (2001), Chib
Figure 4.3: Top: Log to base two of the red-green intensity ratio for Chromosome 11 of Coriell.05296 along the cell line. Note that the horizontal axis here is not scaled identically to that shown in Figure 1 of Erdman and Emerson (2008). (this does not affect the results). Bottom: Inferred changepoints (blue dashed vertical lines) and latent AR(1) field (solid red) using an RFR analysis and INLAs to estimate the field conditional on the detected changepoints.
Figure 4.4: Approximate marginal posterior densities of $\phi$, $\sigma_x^{-2}$ and $\sigma_y^{-2}$ for each of the three segments.
(1998), Green (1995), Carlin et al. (1992) and Raftery & Akman (1986), amongst others. In all of these analyses it is assumed that observations arise from a Poisson process. This Poisson process is assumed to have intensity which follows a step function with a known or unknown number of steps. These steps or “jumps” in intensity occur at the changepoints. Other models have also been fit to this data. For example, a smoothly changing log-linear function for the intensity of the Poisson process:

\[ \lambda(t) = \nu \exp\{-\gamma t\} \]

(see for example Cox & Lewis (1966) and the original source of this data Jarrett (1979)). The log-linear intensity model would favour more gradual change, rather than the abrupt changes implied by changepoint models. There is an argument for some of the elements of such a model that allows for gradual change. Although, as noted in Raftery & Akman (1986), abrupt changes in this data are most likely due to changes in the coal mining industry at the time, such as trade unionization, the possibility of more subtle changes in rate could and should be entertained. A GMRF model applied to this data should be able to model gradual as well as abrupt change.

As in Fearnhead (2006) a week is the basic time unit. The data spans 5,853 weeks over 112 years. The latent field is taken as AR(1). This allows for an inhomogeneous Poisson process within segments, opening up the possibility for gradual change. The rate of the Poisson process is related to the field through a log-link function. More specifically,

\[ y_i \sim \text{Poisson}(\lambda_i) \]

where

\[ \lambda_i = \exp\{\alpha + x_i\}, \quad i = 1, \ldots, n. \]

The parameter \( \alpha \) is an intercept and \( x_i \) follows an AR(1) process with persistence parameter \( \phi \).

Priors were chosen in the same way as the Coriel example by using INLAs on the entire data and using the results of this analysis to inform prior choices. The priors
chosen were

\[ \sigma_x^{-2} \sim \text{Gamma}(4, 0.01) \]
\[ \kappa \sim \text{N}(3, 1.89^2) \]
\[ \alpha \sim \text{N}(0, 10^2). \]

Following Fearnhead (2006) and Green (1995), the prior on the number of changepoints was taken to be Poisson with mean 3.

A spacing of \( g = 50 \) was used. Figure 4.5 (a) shows the posterior distribution of the number of changepoints for the AR(1) latent field model. A two changepoint model is most likely, \textit{a posteriori}. Figure 4.5 (b) shows the most likely position of these changepoints computed using the methods of Section 4.4.1. A plot of the log intensity of the Poisson process over the entire 5,853 weeks is shown in Figure 4.6, obtained by conditioning on the MAP changepoint positions from the two changepoint model. From this it can be argued that a model accounting for gradual changes in the rate of disasters is not entirely unjustified. There appears to be small fluctuations of rate around a mean rate. These fluctuations are treated differently to the two abrupt changes that are detected by the GMRF model.

There is a discrepancy between the posterior of the number of changepoints from RFRs given here and that given in Fearnhead (2006) and in Section 2.7 (Figure 2.1) which both allowed changepoints at all possible points in the data. This is a good opportunity to further investigate the approximation error introduced by using RFRs. Figure 4.7 shows the posterior number of changepoints obtained from using grids of size \( g = 1, 5, 10, 15, 25, 50 \) for the model and prior assumptions in Fearnhead (2006) and Section 2.7. It is clear that as the value of \( g \) increases, the RFRs become less sensitive to small or short lived changes for this model, as might be expected. However, at large values of \( g \) the ability to pick out two abrupt changes does not seem to diminish.

It is possible to compute approximate Bayes factors for the GMRF and independent data models conditional on there being a given number of changepoints. The marginal
Figure 4.5: Coal mining data: results from an analysis using INLAs and $g = 50$. The figure on the left is the posterior distribution of the number changes while that on the right shows the cumulative counts of disasters and the changepoints indicated (blue dashed line).

Figure 4.6: Coal mining data: Inferred log intensity by week.
Figure 4.7: Investigating approximation error in RFRs; results from analyses of coal mining disasters with different values of $g$ using the model from Fearnhead (2006).
likelihood of the data conditional on $k$ changepoints is approximately

$$\pi(y_{1:n}|k \text{ changepoints}) \approx \sum_{s=1}^{N-k} P(1, t_s) R^{(k)}_1(s) \delta(c_0 = 0|c_1 = s)/Z_k.$$  

The different models are characterized by model assumptions and consequently the way in which the segment marginal likelihoods are computed;

$$P_{\text{INLA}}(t, s) \quad \text{and} \quad P_{\text{ANALYTIC}}(t, s).$$  

The approximate Bayes factor for the GMRF model versus the analytic model conditioning on $k$ changepoints is given by

$$B_k = \frac{\pi_{\text{INLA}}(y|k \text{ changepoints})}{\pi_{\text{ANALYTIC}}(y|k \text{ changepoints})}.$$  

For a one changepoint model, this was $B_1 = 4.63$ and for two changepoints it was $B_2 = 5.25$. This implies that there is more support for the GMRF model in these cases, suggesting that modelling small scale variation in the rate of disasters is worthwhile. This supports the interpretation of Figure 4.6.

### 4.7 Well-log data

The Well-log data was introduced in Section 2.9 where a probe is lowered into a bore-hole in the Earth’s surface and is shown again in Figure 4.8. A more elaborate model is fitted here to account for dependency in the nuclear magnetic response as the probe is lowered into the bore-hole. This is an improvement on the independence model fitted in Section 2.9 as the probe lowers, it moves through different rock strata and some will have greater depth than others. Therefore, it would be expected to see some correlation between observations arising from rock strata of the same type. Fitting this model can also reduce the detection of false signals as changepoints. See Fearnhead & Clifford (2003) for a discussion of the issue of outliers in Well-log data.

Since this is a large data set ($n = 4050$) a larger value of $g$ should be used to isolate regions where changepoints occur. This vastly reduces the computational time required
Figure 4.8: Well-log data. Observations are the nuclear magnetic response recorded by a probe being lowered into a bore-hole in the Earth’s surface.
for the necessary approximations for data of this size. Analyses using $g = 10, 25, 50$
were carried out, choosing the prior parameters using the information obtained from
the MCMC analysis in Section 2.9. In each instance numerical instability prevented the
recursions on the reduced time index set from being computed. This happened because
the scale of the data is so large ($\sim 10^5$). In general, measures need to be introduced to
prevent numerical instabilities in these types of recursions. In the computations of the
RFRs a measure similar to those in Fearnhead (2005) (changepoint models) and Scott
(2002) (hidden Markov models) was employed. This consisted of two steps to ensure
stability. Firstly, compute

$$\frac{R_j^{(k)}(r)}{R_{j-1}^{(k-1)}(r + 1)} = \sum_{s=r+1}^{N-k+j} \delta(c_j = r|c_{j+1} = s) \exp \left\{ \log P(t_r + 1, t_s) + \log R_j^{(k)}(s) - \log R_{j-1}^{(k-1)}(r + 1) \right\}$$

and then

$$\log R_j^{(k)}(r) = \log R_j^{(k-1)}(r + 1) + \log \left( \frac{R_j^{(k)}(r)}{R_{j-1}^{(k-1)}(r + 1)} \right).$$

The reason these do not work here is that the large scale of the data means that
$\log P(t_r + 1, t_s)$ is much larger than usual, since it is the marginal likelihood of $g =
10, 25, 50$ points. It thus makes the argument to the exponential function in the first
stabilizing equation cause instabilities at some points. This then carries through the
removal of the recursions.

A simple way to overcome the issues is to do an equivalent analysis of the data on
a smaller scale, so that large $\log P(t_r + 1, t_s)$ is avoided. Simply dividing the data by
its sample standard deviation $s$ reduces the scale appropriately. The parameters for
the prior specification were also adjusted to allow for the difference in scale to give the
priors

$$\sigma_y^{-2} \sim \text{Gamma}(1, 0.01)$$
$$\sigma_x^{-2} \sim \text{Gamma}(1, 0.01)$$
$$\kappa \sim \text{N}(5, (\sqrt{10})^2).$$
Figure 4.9: Posterior of the number of changepoints for the Well-log data fitting an AR(1) GMRF model. This suggests the most likely number of changepoints \emph{a posteriori} is 19.

The prior on $\kappa$ here gives most prior weight to values of $\phi$ in $[0.9, 1)$ (about 93%). This will allow the possibility for the AR(1) GMRF model to closely approximate the behaviour of a random walk of order one. However, it still allows the freedom for the dependence pattern to vary across segments. Fearnhead (2006) fits a random walk model of order one to this data, showing that a latent field can be robust to short lived changes and outliers for Well-log data. A uniform prior on $\{0, \ldots, 30\}$ was taken for the number of changepoints.

For the final analysis $g$ was taken to be 25. This reduced the necessary number of approximate marginal likelihood approximations from roughly $8.2 \times 10^6$ (for $g = 1$) to $1.3 \times 10^4$; over 600 times less. The computations for these approximations took about a day of computing time. This appears lengthly, however this should be judged along with the fact that the model is more flexible and that the mean signal can be
Figure 4.10: Well-log data: results from RFRs and INLA (top) and independent data model (bottom).
estimated at every point in the data. Figure 4.9 shows the posterior probability of
the number of changepoints. The mode is at 19, but there appears to be support for
up to 22. Conditioning on 19 changepoints, their locations were determined using the
search strategy outlined in Section 4.4.1. These locations were then refined to hone
in on the actual changepoint positions. Conditioning on these positions inference was
carried out for the latent field. This is shown in the top figure of Figure 4.10. The
field appears to follow the trend of the data closely, while the changepoint model caters
for abrupt change. Fearnhead (2006) compared the results of a first order random walk
field to those from an independent Gaussian model for the data. Similarly, the results
from the GMRF model here are compared with those obtained using the sampler of
Section 2.9 on the Well-log data. For comparison, the 54 most likely changepoints
(mode of posterior) were taken from the independent Gaussian model, and segment
means were computed conditional on these (bottom of Figure 4.10). It can be seen that
the independent model is sensitive to changes in the mean and is conservative when
inferring changepoints (more rather than less). The GMRF model however appears to
be more robust to noisy data points and only infers changepoints when abrupt changes
occur in the field.

4.8 Stochastic volatility data

Stochastic volatility models were introduced in Chapter 1.3.3 along with a discussion
of some estimation techniques. In this example it is demonstrated how INLAs can be
used with RFRs to estimate changepoint models where the segment observations are
assumed to arise from a stochastic volatility model.

The segment model assumed is

\[ y_i \sim N\left(0, \beta^2 e^{x_i}\right), \quad i = 1, \ldots, n. \]

with \( x \) following an AR(1) process with persistence parameter \( \phi \) and innovation variance
\( \sigma^2_x \) where \( \exp\{\log \beta\} \) may be interpreted as an intercept for the volatilities. Data in
different segments are assumed independent, so that concern here is only in the complex intra segment correlation structure.

Two data sets were simulated to test out the methods. The first of these had two changepoints and is described in Table 4.2. The data is shown in Figure 4.11 along with a plot of the generated latent volatilities.

Prior parameters for computing the segment marginal likelihoods were roughly guessed based by applying INLAs to the entire data. The priors were

\[
\sigma^{-2}_x \sim \text{Gamma}(30, 0.02) \\
\kappa \sim \text{N}(3, 1).
\]

The required computations took about 5 minutes using a reduced time index set with equal spacing and \( g = 5 \). A two changepoint model had almost all of the posterior weight, although there was minor support for three changepoints. See Figure 4.12.

The approximation works well in this situation because the change is quite noticeable. In some other simulated data which the methods were applied to, it was found that the approach was poor at detecting any changepoints when there was only a small (or no) change in the intercept of the latent volatilities \( \log \beta \). It was desired that smaller changes, for example, a change in just the persistence parameters across segments, could be detected. This did not seem to be the case however. To demonstrate one such example, the data outlined in Table 4.3 was simulated and the INLAs were used to approximate the segment marginal likelihoods. In this data, the only change

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>100</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>( \log \beta )</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>( \phi )</td>
<td>0.8</td>
<td>0.9</td>
<td>0.7</td>
</tr>
<tr>
<td>( \sigma^2_x )</td>
<td>0.01²</td>
<td>0.05²</td>
<td>0.09²</td>
</tr>
</tbody>
</table>

Table 4.2: Stochastic volatility: parameters used to simulate data.
Figure 4.11: Stochastic volatility data: Simulated observed data with changepoints indicated with blue dashed line (top) and log of simulated latent volatilities (bottom).
occurs in the persistence parameter, and the noise is chosen to be small. The data, log latent volatilities and posterior distribution of the number of changepoints are shown in Figure 4.13 for $g = 5$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\log \beta$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.2</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.01$^2$</td>
<td>0.01$^2$</td>
<td>0.01$^2$</td>
</tr>
</tbody>
</table>

Table 4.3: Stochastic volatility data: parameter specifications for dataset with poor performance.
Figure 4.13: Stochastic volatility data: Second simulated dataset. Neither of two changepoints were detected.
4.9 Discussion

This chapter introduced and demonstrated two new useful approximate methods for changepoint problems when the assumption of independent data is relaxed. The first of these came in the form of INLAs, a new approximate inference method for GMRFs due to Rue et al. (2009). This allows the marginal likelihood for complex segment models to be evaluated approximately, so that it may be used for an approximate filtering recursions approach.

Some computational considerations led to the second proposed method. Instead of performing filtering recursions analysis on the entire data, RFRs were introduced so that recursions may be computed only on a reduced time index set, thus using all of the data, but only searching for changepoints in the general region where they occur. It was demonstrated that this method can be useful in cutting computation time for larger datasets by applying it to a DNA segmentation example with about 49,000 data points.

The hybrid INLAs-RFRs methodology was applied to four different data examples. The first of these involved detecting changepoints in DNA copy number in CGH studies of Coriel cell lines. The second example was a re-analysis of the coal mining disasters data, but where this time, the model allowed for small scale variation in the intensity of the process and allowed for week to week dependency. This new model was more supported by the data than the usual step function intensity models which are often fitted. This was demonstrated by approximate calculation of Bayes factors for the GMRF model and the independent data model over a range of numbers of changepoints. The GMRF model out-performed the independent data model every time. The third example was another re-analysis, this time of the Well-log data. It was shown that allowing for segment dependency can be more robust to noisy observations, and that unnecessary changepoints (short lived changes, outliers etc.) are not inferred in this case. For the final example, the methods were applied to some simulated stochastic volatility data. Performance was satisfactory when changes were large, but more subtle
changes in the underlying segments were not detected. This is an area for improvement of the proposed methodology.

It is worth noting again that RJMCMC would be practically infeasible for the data models considered here. This gives the approximate approach even more of an advantage. This is true especially in the case of models which require good corresponding proposal densities to perform well when it comes to MCMC, such as stochastic volatility models.

Hyperprior incorporation was mentioned in Chapter 2. In this chapter, reliance of the recursions on the hyperparameter values was not an issue for two reasons. The first was that inference was performed conditional on a given number of changepoints, so that a parameter for a Markov point process prior on changepoint positions ($p$) did not have to be chosen in advance. The second reason is that the hyperparameter space is explored by numerically integrating over a suitable grid on the space when applying INLAs. Thus, sensitivity of recursions to one chosen set of hyperparameters is not an issue for the work presented in this chapter.

Overall, this chapter has explored a promising new direction for estimation of changepoint models by creating a hybrid of two popular methods in their respective fields, namely INLAs in the GMRF field of study, and filtering recursions for sequential changepoint model estimation. Other data models are possible which have not been applied to any of the examples in this chapter. For example, it is possible to have higher order Markov dependencies for random walk fields in the R-INLA package. Zero inflated Poisson and Binomial data models are also possible.
Chapter 5

Conclusions and further work

5.1 Conclusions

Overall, the work in this thesis has two main outcomes. Firstly, it demonstrates that it is possible to use simpler MCMC methods than RJMCMC to perform model selection for some latent variable models. The second outcome is that approximate deterministic inference for changepoint models where conditional independence assumptions are relaxed is feasible.

Chapter 2 demonstrated how collapsed sampling approaches can be useful for the analysis of multiple changepoint models, where the number of changepoints is not known prior to analysis. The main achievement of this chapter was to give an MCMC sampling scheme where it can sometimes be possible to include hyperpriors, so that sampling is robust to prior sensitivity. It was shown that the filtering recursions approach of Fearnhead (2006) may be sensitive to prior specification in some scenarios. In these situations, the MCMC sampler proposed in this chapter could be used for an exploratory analysis or to give sensible starting values for a filtering recursions analysis.

In Chapter 3 a Bayesian extension of latent block models was introduced. This was a collapsed latent block model, which allowed for joint MCMC sampling of row and column allocations and the number of row and column clusters. The strength of this model is that it also allows for a probabilistically justified approach to model
selection for latent block models. An alternative to the approach presented in this chapter would be to use a RJMCMC algorithm instead for the model determination task. However, a discussion concluded that such an approach would be difficult to implement in this situation, due to the large change in parameter space dimension when adding or removing row or column clusters.

Chapter 4 used a hybrid of filtering recursions approaches for changepoint problems (Fearnhead 2006) and INLAs (Rue et al. 2009) to perform inference for complex data segment models in the presence of an unknown number of multiple changepoints. MCMC would not be feasible for many of the models considered, which are traditionally considered challenging for MCMC analyses, and so the approximate approach offers an interesting way forward, and could be a promising area for future research. An interesting part of the approximations considered are the reduced filtering recursions, which could be used to give savings in computation for larger datasets. This chapter contributes to a renewed desire of the Bayesian community for simulation-free and approximate methods for the analysis of models, traditionally analyzed using MCMC.

5.2 Further work

5.2.1 Collapsed sampling approaches for hidden Markov models

Recall the hidden Markov model of Section 1.3.4. Assume now a Gaussian distribution for observed data, so that
\[ y_i | z_i \sim N(y | \theta_{z_i}) \]
where \( z_i \) represents one of \( K \) Markov chain states, and is related to \( z_{i-1} \) by the \( K \times K \) transition matrix \( R = (r_{st}) \), and \( y_i \) is conditioned on \( z_i \). It turns out that it may be possible in this setting to perform collapsed sampling so that \( K \) may be unknown, and updated jointly with the state allocations in a scheme similar to those in chapters 2 and 4 and the allocation sampler of Nobile & Fearnside (2007).
Conditional on a set of allocations \( z \), the likelihood of data \( y \) may be written as

\[
\pi(y|K, z, \Theta) = \prod_{i=1}^{n} \phi(y_i|\mu_{z_i}, \sigma_{z_i}^2) = \prod_{k=1}^{K} \prod_{i: z_i = k} \phi(y_i|\mu_k, \sigma_k^2)
\]

where \( \phi(y|\mu, \sigma^2) \) is the density of a \( N(\mu, \sigma^2) \) distribution. Take independent priors \( \pi(\mu_k|\sigma_k^2)\pi(\sigma_k^2) \) where

\[
\mu_k|\sigma_k^2 \sim N(\xi, \sigma_k^2/\eta) \\
\sigma_k^{-2} \sim \text{Gamma}(\gamma/2, \lambda/2)
\]

for \( k = 1, \ldots, K \). Now turn attention to the transition matrix \( R \). Consider \( R_s \) the \( s \)th row of this. This is just the transition probabilities from state \( s \) to any other state. Thus

\[
\sum_{t=1}^{K} r_{st} = 1.
\]

Take a Dirichlet(\( \alpha, \ldots, \alpha \)) prior on each row of \( R \). The joint posterior can then be written as

\[
\pi(K, z, \Theta, R|y) \propto \pi(K)\pi(R|K)\pi(z|K, R)\pi(\Theta|K)\pi(y|K, z, \Theta, R)
\]

\[
= \pi(K) \times \left( \frac{\Gamma\{\alpha K\}}{\Gamma\{\alpha\}^K} \right)^K \prod_{s=1}^{K} \prod_{t=1}^{K} r_{st}^{\alpha - 1} \times \prod_{s=1}^{K} \prod_{t=1}^{K} n_{st}^{rs_{st}}
\]

\[
\times \prod_{k=1}^{K} \phi(\mu_k|\xi, \sigma_k^2/\eta) \text{Ig}(\sigma_k^2|\gamma/2, \lambda/2)
\]

\[
\times \prod_{i=1}^{n} \phi(y_i|\mu_{z_i}, \sigma_{z_i}^2)
\]

where \( n_{st} \) is the number of transitions from state \( s \) to state \( t \) in the chain \( z_{1:n} \) and \( \text{Ig}(x|a, b) \) represents the density function of an inverse gamma distribution with parameters \( a \) and \( b \) evaluated at \( x \).
Integrating out $\Theta$ and $R$ will be equivalent to the following expression

$$
\pi(K, z | y) \propto \pi(K) \left( \frac{\Gamma\{\alpha K\}}{\Gamma\{\alpha\} K} \right)^K \prod_{s=1}^K \left( \int \prod_{t=1}^K r_{st}^{n_{st}+\alpha-1} \, dR_s \right)
\times \prod_{k=1}^K \int \phi(\mu_k | \xi, \sigma_k^2 / \eta) \operatorname{lg}(\sigma_k^2 | \gamma / 2, \lambda / 2) \prod_{i : z_i = k} \phi(y_i | \mu_k, \sigma_k^2) \, d\mu_k \, d\sigma_k^2
$$

$$
= \pi(K) \left( \frac{\Gamma\{\alpha K\} K}{\Gamma\{\alpha\} K^2} \right) \prod_{s=1}^K \left( \prod_{t=1}^K \frac{\Gamma\{n_{st} + \alpha\}}{\Gamma\{n_s + \alpha K\}} \right)
\times \pi^{-n/2} \frac{\lambda \gamma K / 2}{\Gamma\{\gamma / 2\} K} \prod_{k=1}^K \sqrt{\frac{\eta}{n_k + \eta}} \Gamma\{(n_k + \gamma) / 2\}
\times \left\{ s_{sk} + \eta \xi^2 - (n_k + \eta)^{-1} (s_k + \eta \xi^2 + \lambda)^{-1} \right\}^{-(n_k + \gamma) / 2}
$$

where $ss_k = \sum_{i : z_i = k} y_i^2$ and $s_k = \sum_{z_i = k} y_i$.

Inference can be carried out in a similar way to that for the collapsed latent block model, where changes in the number of states $K$ are possible. For example, a Gibbs update for the label $z_i$ is possible by sampling it from the full conditional,

$$
\Pr(z_i = k' | \cdots) \propto \frac{\Gamma\{n_{k1} - 1 + \alpha\} \Gamma\{n_{k1} k' + 1 + \alpha\}}{\Gamma\{n_{k1} + \alpha\} \Gamma\{n_{k1} k' + \alpha\}}
\times \frac{\Gamma\{n_{kk2} - 1 + \alpha\} \Gamma\{n_{kk2} k' + 1 + \alpha\}}{\Gamma\{n_{kk2} + \alpha\} \Gamma\{n_{kk2} k' + \alpha\}}
\times \frac{\Gamma\{n_k + \alpha K\} \Gamma\{n_{k'} + \alpha K\}}{\Gamma\{n_k - 1 + \alpha K\} \Gamma\{n_{k'} + 1 + \alpha K\}}
\times \frac{M_k^{(-i)} M_{k'}^{(+i)}}{M_k M_{k'}}, \quad k' \neq k
$$

where $z_{i-1} = k_1, z_{i+1} = k_2$ and currently $z_i = k$. The probability that the label does not change is proportional to 1. The integrated likelihood is

$$
M_k = \pi^{-n/2} \frac{\lambda \gamma K / 2}{\Gamma\{\gamma / 2\} K} \prod_{k=1}^K \sqrt{\frac{\eta}{n_k + \eta}} \Gamma\{(n_k + \gamma) / 2\}
\times \left\{ s_{sk} + \eta \xi^2 - (n_k + \eta)^{-1} (s_k + \eta \xi^2 + \lambda)^{-1} \right\}^{-(n_k + \gamma) / 2}.
$$
It is intended to extend this early analysis in much the same way as the work in Chapters 2 and 3, to give an efficient method for MCMC analysis of hidden Markov models with an unknown number of hidden states. This involves adding moves for the birth and death of hidden states. Label switching will also be an issue here, but the algorithm outlined in Section 3.3.3 may be used in this situation also. Altogether, this would provide a simple alternative to the complex birth and death moves in the reversible jump approach of Robert et al. (2000) and it is envisaged that there would be gains in efficiency by using collapsing in the same way as the collapsed sampling schemes in Chapters 2 and 3.

5.2.2 A full assessment of the approximation error in RFRs

Chapter 4 gave a brief assessment of the error introduced in the posterior distribution of the changepoints by using the RFR approximations. A more thorough simulation study is to be undertaken to determine how this error varies with the size of the data and the spacing, $g$, in the reduced time index set. It is to be investigated whether rough analytic guides to the error rate can be derived.

5.2.3 Software for R

It is intended to write an R package to perform the collapsed latent block model calculations by a simple call to an R function. This would make the approaches available for wider use. Currently a front-end program written in C is available at www.ucd.ie/statdept/jwyse.
Appendix

A: Calculations for the prior on changepoints

The prior on changepoints assumes changepoint positions distributed as the even numbered order statistics of \(2k + 1\) uniform draws from the set \(\{1, \ldots, n - 1\}\) without replacement. This prior is now derived.

Consider drawing a sample of size \(m\) uniformly without replacement from the integers \(\{1, \ldots, N\}\), denoted \(u_1, \ldots, u_m\). Let the order statistics of the sample be \(u^{(1)}, \ldots, u^{(m)}\). Then there are \(\binom{N}{m}\) possible configurations of order statistics. Reduce attention to just the even numbered order statistics

\[
\begin{align*}
& u^{(2)}, u^{(4)}, \ldots, u^{(m)} & m \text{ even} \\
& u^{(2)}, u^{(4)}, \ldots, u^{(2\lfloor m/2 \rfloor)} & m \text{ odd}
\end{align*}
\]

The even numbered order statistics are determined by the entire set of order statistics, hence there are still \(\binom{N}{m}\) possibilities for these.

Consider a particular configuration of order statistics in the case of \(m\) being odd:

\[
u^{(2)} = i_1, u^{(4)} = i_2, \ldots, u^{(2\lfloor m/2 \rfloor)} = i_{\lfloor m/2 \rfloor}.
\]

The probability of this configuration is what is required.

\[
\Pr\{u^{(2)} = i_1, u^{(4)} = i_2, \ldots, u^{(2\lfloor m/2 \rfloor)} = i_{\lfloor m/2 \rfloor}\} \\
= \Pr\{1 \leq u^{(1)} \leq i_1 - 1, i_1 + 1 \leq u^{(3)} \leq i_2 - 1, \ldots, i_{\lfloor m/2 \rfloor} - 1 + 1 \leq u^{(2\lfloor m/2 \rfloor - 1)} \leq i_{\lfloor m/2 \rfloor} - 1, i_{\lfloor m/2 \rfloor} + 1 \leq u^{(m)} \leq N\} \\
= \binom{N}{m}^{-1} \times (i_1 - 1) \times (i_2 - i_1 - 1) \times \cdots \times (i_{\lfloor m/2 \rfloor} - i_{\lfloor m/2 \rfloor} - 1) \times (N - i_{\lfloor m/2 \rfloor})
\]
following from the assumption that sampling is independent.

In a similar way, for the case of \(m\) being even,

\[
\Pr\{u^{(2)} = i_1, u^{(4)} = i_2, \ldots, u^{(m)} = i_{m/2}\} = \left(\frac{N}{m}\right)^{-1} \times (i_1 - 1) \times (i_2 - i_1 - 1) \times \cdots \times (i_{m/2} - i_{m/2-1} - 1) \times (N - i_{m/2}).
\]
**B: Derivation of the Integrated Autocorrelation Time**

Consider using an MCMC sample $\theta^1, \ldots, \theta^N$ to estimate the integral

$$E(\eta(\theta)) = \int \eta(\theta) \pi(\theta) \, d\theta. \quad (5.1)$$

From the Ergodic theorem (Roberts 1996), as $N \to \infty$, the approximation

$$\bar{\eta} = \frac{1}{N} \sum_{t=1}^{N} \eta(\theta^t)/N \to E(\eta(\theta)),$$

asymptotically.

Consider the variance of the approximation $\bar{\eta}$ with respect to the target distribution of the MCMC algorithm $\pi(\cdot)$:

$$\text{Var}_\pi(\bar{\eta}) = \frac{1}{N^2} \text{Var} \left\{ \sum_{t=1}^{N} \eta(\theta^t) \right\}$$

$$= \frac{1}{N^2} \left\{ \sum_{t=1}^{N} \text{Var}(\eta(\theta^t)) + 2 \sum_{t=1}^{N-1} \sum_{s=t+1}^{N} \text{Cov}(\eta(\theta^t), \eta(\theta^s)) \right\}$$

$$= \frac{1}{N^2} \left\{ N \text{Var}(\eta(\theta^0)) + 2 \text{Var}(\eta(\theta^0)) \sum_{t=1}^{N-1} \sum_{s=t+1}^{N} \frac{\text{Cov}(\eta(\theta^t), \eta(\theta^s))}{\text{Var}(\eta(\theta^0))} \right\}$$

$$= \frac{\text{Var}(\eta(\theta^0))}{N} \left\{ 1 + \frac{2}{N} \sum_{t=1}^{N-1} \sum_{s=t+1}^{N} \rho_\eta(s-t) \right\}$$

$$= \frac{\text{Var}(\eta(\theta^0))}{N} \left\{ 1 + 2 \sum_{t=1}^{N-1} \left( 1 - \frac{t}{N} \right) \rho_\eta(t) \right\}$$

where $\rho_\eta(t)$ is the $t$-lag autocorrelation of $\eta(\theta^t)$ with respect to $\pi(\cdot)$. If the samples $\{\theta^1, \ldots, \theta^N\}$ were independent the autocorrelation terms would vanish. In this case the error incurred in estimating $E(\eta(\theta))$ by a sample of size $N$ decreases as $1/\sqrt{N}$. Thus to half the error one must use four times as many samples for a MCMC method which perfectly generates independent samples. Of course assuming the samples obtained are independent is incorrect, since ignoring the autocorrelation terms above may lead to underestimation of $\text{Var}(\bar{\eta})$.

The integrated autocorrelation time (IAT) may be used to assess the quality of the sample obtained from a MCMC run. The IAT of the series $\eta(\theta^t), t = 0, 1, 2, \ldots$, is
defined as
\[ \tau_\eta = \frac{1}{2} \sum_{t=0}^{\infty} \rho_\eta(t) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_\eta(t). \]
The last equality follows since \( \rho_\eta(0) = 1 \). As \( t \to \infty \), \( \rho_\eta(t) \to 0 \). When \( N \) is large,
\[
\left(1 - \frac{t}{N}\right) \rho_\eta(t) \approx \begin{cases} \rho_\eta(t) & t << N \\ 0 & t \approx N, \end{cases}
\]
so that
\[
\text{Var}(\bar{\eta}) \approx \frac{\text{Var}(\eta(\theta^0))}{N} \left\{ 1 + 2 \sum_{t=1}^{\infty} \rho_\eta(t) \right\} = 2\tau_\eta \times \frac{\text{Var}(\eta(X_0))}{N}.
\]
This says that the sample from the MCMC algorithm gives an estimator of \( \text{E}(\eta(\theta)) \) which is \( 2\tau_\eta \) times more variable than that obtained from an estimator using an iid sample of the same size. So effectively, there is the equivalent of \( N/2\tau_\eta \) worth of independent samples. This quantity is called the effective sample size (ESS) and is used as a measure of efficiency of MCMC algorithms. The closer this is to \( N \), that is, the closer \( 2\tau_\eta \) is to \( 1 \), the better the performance.

One can estimate the IAT from MCMC output. This can be done by firstly computing the sample autocorrelations \( \hat{\rho}_\eta(t) \) using a standard time series analysis. Defining
\[
\hat{\tau}_\eta(T) = \frac{1}{2} + \sum_{t=1}^{T} \hat{\rho}_\eta(t)
\]
and choose \( M \) to be the smallest integer such that \( M > c\hat{\tau}_\eta(M) \) where \( c = 6 \). Then \( \hat{\tau}_\eta \approx \hat{\tau}_\eta(M) \). This estimate is due to Madras & Sokal (1988) (“automatic windowing”) and should work well provided there is a large enough sample (roughly \( N > 1000\tau_\eta \)).
C: Number of marginal likelihood evaluations for RFRs

For equally spaced grid points, the number of points for a given value of \( g > 1 \) is the number of times \( g \) divides into \( n \) plus \( n \) itself. This gives \( \lfloor n/g \rfloor + 1 \). When \( g = 1 \) there is one point less as \( n \) is not included as a possible changepoint. Overall this gives \( \lfloor n/g \rfloor + 1 - I(g = 1) = \lfloor n/g + 1 - I(g = 1) \rfloor \). As there are usually \( n(n+1)/2 \) marginal likelihoods required, for the RFRs this becomes

\[
\lfloor n/g + 1 - I(g = 1) \rfloor (\lfloor n/g + 1 - I(g = 1) \rfloor + 1)/2.
\]
Bibliography


