Towards Automatic Reversible Jump Markov Chain Monte Carlo

By

David Hastie

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Science

March 2005

Department of Mathematics

Abstract

Since its introduction by Green (1995), reversible jump MCMC has been recognised as a powerful tool for making posterior inference about a wide range of statistical problems. Despite enjoying considerable application across a variety of disciplines, the method's popularity has been tempered by the common perception that reversible jump samplers can be difficult to implement.

Beginning with a review of reversible jump methods and recent research in the area, this thesis introduces steps towards the design of an automatic reversible jump sampler with the aim of taking the method outside of the domain of the MCMC expert. The need for a sampler is discussed and motivated by an application of reversible jump to a recent problem in biology. We build upon recent developments in the area of adaptive sampling, analysing and extending existing methods for their inclusion in an automatic reversible jump sampler.

The automatic sampler that we introduce in the penultimate chapter of the thesis builds upon the first steps taken by Green (2003). Requiring minimal user input, it uses adaptive techniques to perform self-tuning and calibration for many trans-dimensional statistical problems. The broad applicability of the sampler is detailed, as are typical results, indicating performance comparable to problem-specific samplers, designed and tuned by hand. The thesis ends with some suggestions for further work.

Acknowledgements

Without the contribution of many people the research presented within this thesis would not have been possible. I am especially grateful to my supervisor Peter Green for his patient guidance and encouragement and for allowing me to have control over how I carried out my research. I would also like to thank Michael Newton, who welcomed me to the University of Wisconsin for six months and allowed me to help him with some very exciting biostatistical applications. Thank you to the World Universities Network for making the visit possible.

Thanks also to Christophe Andrieu, my second Bristol supervisor and adaptive sampling mentor, for providing me with some fantastic ideas and always being available to discuss my work. Graeme Ambler was also a source of advice on Markov chain Monte Carlo sampling issues and my thanks are extended to him. Both Peter Green and Christophe Andrieu deserve further thanks for reading early drafts of this thesis, as does Phil Turner who kindly read the manuscript for errors. Any errors that remain are my own.

On a more personal level I am grateful to my family and Wendy for supporting my decision to give up my job and return to academia and to my office mate Steve O'Keefe. Without your support and constant encouragement throughout these three years I would not have been able to do this. Thank you.

Declaration

I, the author, declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree.

The views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

David Hastie

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The importance of stochastic models has long been recognised in many disciplines of science. Not only have such methods contributed to a greater understanding of the scientific problems themselves, but different scientific perspectives have also spawned new and improved statistical methods. Indeed, some of our most popular statistical tools have their origins in disciplines such as physics or the medical sciences. Due to the ease with which scientists are able to share information, the use of statistical models will continue to be an important part of an increasing number of disciplines in the future.

As the use of statistical models increases, so must we develop techniques that are widely applicable to the diverse range of stochastic models that will proliferate. Moreover, to maximise the efficiency of the community as a whole, we must always remember that for many scientists it is the conclusion that is important and not the statistical tools that helped them reach this end. Concisely put, we must concentrate our efforts on making inferential techniques easily adoptable by scientists so that they can focus upon the field in which they specialise.

The research that we describe in this thesis aims to try to aid this process for

the methods arising from one small branch of statistics. We focus on a collection of powerful computational statistical tools, collectively known as reversible jump Markov chain Monte Carlo, that have the capacity to be applied to problems across a full spectrum of disciplines. We begin this chapter by motivating the need for our research.

Many problems arising from the application of statistical models can be reformulated in terms of an integral containing some target distribution π. Examples include the evaluation of the expectation of a function g, given by

$$\mathbb{E}[g(\boldsymbol{x})] = \int_{\mathcal{X}} g(\boldsymbol{x})\pi(\mathbf{d}\boldsymbol{x}) \ ,$$

or the calculation of the probability of a certain event B, given by

$$\pi(B) = \int_{B} \pi(\mathbf{d}\boldsymbol{x}) \ .$$

The Bayesian statistical paradigm (see for example O'Hagan, 1994) provides us with many further examples, such as the calculation of a constant of proportionality or the integration of a joint posterior distribution to obtain a marginal distribution. Indeed, with the growth in popularity of the Bayesian approach, such integrals are more prominent in this context than in any other area of statistics.

A contributing factor to the recent spread of the Bayesian approach is the increase in computing capabilities that have become widely available over the last few decades. Until recently, the computational requirements that were necessary to make inference using a Bayesian framework were prohibitively expensive. However, with the arrival of an era where computers were a realistic option for all researchers, came the development of computationally

intensive statistical techniques. One such class of methods, collectively known as *Markov chain Monte Carlo (MCMC)*, has proved a particularly useful tool in allowing us to approximate the common integrals mentioned above. As such, MCMC has improved our understanding of complex and highly structured statistical models and has been particularly useful for making Bayesian inference.

MCMC methods were first introduced by Metropolis *et al.* (1953) and Hastings (1970). Despite the importance of these works (the statistical algorithms proposed remain among the most commonly used today) the lack of available computing power meant that little research was carried out to extend these methods until relatively recently. However, over the last two decades research in this area has grown significantly, so that now a multitude of modified MCMC algorithms exist. We revise the basic ideas and principles of MCMC in detail in chapter 2.

One of the most significant developments in MCMC research was provided by Green (1995). Green's research demonstrated how MCMC methods could be applied to a much wider class of problems, including those where the number of unknowns is one of the unknowns. More formally, the generalisation allowed the consideration of statistical problems where the parameter space can be written as the union of subspaces, each with a possibly different dimension. The class of extended MCMC methods was termed *reversible jump Markov chain Monte Carlo (RJMCMC)* and in essence provided the possibility of making inference about the most general of stochastic models.

Since its conception, RJMCMC has been used for many different applications throughout the scientific world. In section 2.3, some examples (with references)

of these applications are presented. Previous applications for which RJMCMC has been used have ranged from the analysis of times of coal mining disasters (Green, 1995) to the study of AIDS clinical trials (Han and Carlin, 2001), from image analysis (Al-Awadhi *et al.*, 2004) to modelling capture-recapture data in animals (King and Brooks, 2002), and from disease mapping (Green and Richardson, 2002) to understanding prehistoric tomb building technologies (Fan and Brooks, 2000). The statistical models underlying such applications are equally diverse and examples include time series models (where the order of the process is unknown), mixture models (where the number of components is unknown), and graphical models (where the number of edges is unknown). Reversible jump methods have also proved very useful in the analysis of data from recently developed biotechnologies such as gene expression data.

The variety of examples to which RJMCMC has been applied give a clear indication of how powerful the method is. However, the examples to date represent only a fraction of the possible applications for which reversible jump could facilitate statistical inference. The fact that the full potential of the method has not been realised is partly because reversible jump is seen by many scientists to be a complicated statistical tool that it difficult to implement in practice. While in some cases this difficulty is only perceived, in many other cases it is true that designing an efficient RJMCMC sampler is a difficult task that can often require specialist computational statistical knowledge. In chapter 2 we discuss some of the practical difficulties in more detail.

Despite an increase in research in improving the efficiency for general reversible jump algorithms (see 2.5 for a review of some of this research), reversible jump remains in the domain of the MCMC expert. Indeed, the majority of

reported applications of the method have involved considerable collaboration with specialist statisticians. In order to exploit the potential of RJMCMC as an inferential technique, research must be completed to allow effective reversible jump tools to be accessible to all. As several authors recognise (Godsill, 2003; Brooks *et al.*, 2003b; and Green, 2003) the development of more automatic generic RJMCMC tools is a very important goal.

Considerable previous research has concentrated on modifying the reversible jump algorithm to improve its efficiency, with the aim of allowing more effective inference. Other work considers questions of optimal parameterisation of proposal distributions to achieve the same aim. This thesis is dedicated to achieving a similar goal but from a different perspective. Our research focuses upon work towards the design and implementation of an automatic generic reversible jump sampler that requires minimum user input to effectively apply RJMCMC to a vast range of problems. It is hoped that ultimately such a technique could be adopted by scientists with no specialist reversible jump expertise to consider the specific problem of interest to them. We name the sampler that we develop the *AutoMix* sampler.

A necessary stage in the design process is to properly understand reversible jump tools that work efficiently regardless of the problem under consideration. We design the AutoMix sampler by drawing upon important previous research which we discuss fully in later chapters. Particular attention is given to recent relevant research on adaptive sampling (see chapter 4), mixture fitting (see section 5.3.3) and previous work on generic sampler design (see section 5.2).

The thesis is divided into six chapters including this short introductory chapter.

Before proceeding we briefly summarise the aims of each of the chapters and how they fit into the context of automatic sampler design.

Our research begins with chapter 2, where we revise MCMC methods and formally introduce RJMCMC. The chapter continues by considering in further depth some of the many applications of RJMCMC, with the aim of giving a clear demonstration of the potential of the method. This is followed by a consideration of some of the difficulties that are often confronted when implementing reversible jump. The chapter ends with a brief review of some recent research into tackling these difficulties and improving the efficiency of the reversible jump algorithm.

The purpose of chapter 3 is to highlight the involved process of RJMCMC sampler design, motivating why an automatic reversible jump sampler is appealing. This is done by commentating on the development of a stochastic model and appropriate reversible jump technique for a recent problem within biology. The chapter concentrates on the statistical modelling and sampler design aspects. Attention is also given to the results of the biological problem as the results are often of more interest to the non-specialist statistician than the inferential method. It is hoped that the richness of such results may demonstrate to the unfamiliar reader the power of reversible jump as an inferential tool.

Having motivated the need for our research into an automatic sampler, chapter 4 considers the statistical field of adaptive sampling that forms a basis for many of the techniques employed by the AutoMix sampler. The chapter begins with a substantial review of some recent research in this relatively new area of statistics. Our review includes a consideration of some existing adaptive samplers and a discussion of some very important theoretical results. We then focus our

attention on a particular adaptive sampling algorithm (introduced by Atchadé and Rosenthal, 2003), looking at an important question affecting the convergence of the algorithm that may not have been previously considered. As we discuss in chapter 5 this algorithm plays an important role in our AutoMix sampler. The final contribution of chapter 4 is to introduce two new adaptive algorithms, specifically for reversible jump samplers, that we use in the AutoMix sampler. We discuss these algorithms and demonstrate some important properties that they satisfy.

Chapter 5 shares its title with that of the thesis. The research that we present therein draws on the adaptive sampling methods of the previous chapter and other previous research such as the automatic sampler introduced by Green (2003). This sampler is reviewed in section 5.2. The chapter progresses by outlining the AutoMix sampler, providing motivation for each of the components that make up the sampler. We pay special attention to the adaptive techniques employed by the sampler to achieve automaticity. Having reviewed the AutoMix algorithm in detail, we briefly discuss the software that we have written to implement the algorithm. Chapter 5 includes several varied examples of the AutoMix sampler in practice. We demonstrate how well the generic AutoMix sampler performs, even compared to samplers that have been designed for the specific problem by reversible jump experts. The chapter closes with some of our conclusions about the AutoMix sampler.

We end this introductory chapter by acknowledging that the development of a fully automatic reversible jump sampler is indeed a very ambitious goal and we certainly do not claim to have achieved this goal with this research. However, our work and the AutoMix sampler that we develop move us nearer to this aim.

In the final chapter we consider some of the areas of the AutoMix sampler which might benefit from future research. We hope that our research will stimulate other researchers to improve upon our methods and bring a successful automatic method closer.

Chapter 2

Reversible Jump Markov Chain Monte Carlo: A Review

In this chapter we revise the ideas behind reversible jump Markov chain Monte Carlo and review some recent literature on the subject. Throughout the chapter we assume a basic knowledge of Markov chains, including familiarity with notions such as time homogeneity, irreducibility, aperiodicity, recurrence, invariance distributions and reversibility. A review of these concepts can be found in Robert and Casella (2002) and many other introductory texts. The chapter forms a basis for the remainder of the thesis, introducing notation and formulations that we will use in later chapters.

2.1 Standard Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a collection of computer intensive methods which allow the approximate computation of integrals that are analytically intractable. Although much of the theory behind many popular MCMC algorithms was developed in the 1950s and 1970s (Metropolis *et al.*, 1953; Hastings, 1970; Peskun, 1973) it has not been until the last few years, with the increase in the availability and specification of computers, that the potential of the method has been realised.

The principle behind MCMC is the ergodic theorem applied to Markov chains. Suppose we wish to make inference about a target distribution π defined on a general state space $\mathcal{X}$. Suppose also that we can construct an irreducible, aperiodic, Harris recurrent, time homogeneous Markov chain with one step transition kernel $\mathcal{K}$, which has π as its invariant distribution. Denoting the observations of our Markov chain by $\boldsymbol{x}^0, \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$, the ergodic theorem says that for a function f satisfying $\int |f(\boldsymbol{x})|\pi(\mathrm{d}\boldsymbol{x}) < \infty$,

$$\lim_{m\to\infty} \frac{1}{m+1} \sum_{t=0}^{m} f(\boldsymbol{x}^{\boldsymbol{t}}) = \mathbb{E}_\pi[f(\boldsymbol{x})].$$

In effect, if such a Markov chain can be constructed, integrals such as those listed in chapter 1 can be approximated using Monte Carlo averages based upon realisations of the Markov chain, rather than samples from π. Clearly if we could sample directly from π we would use just normal Monte Carlo techniques, but MCMC allows us to gain integral approximations where direct sampling from π is not easy.

The MCMC problem is the inverse of the common Markov chain problem which requires us to find the unique stationary distribution of a Markov chain given the transition kernel. Our new problem is to find a transition kernel so that the resulting Markov chain has equilibrium distribution π. Fortunately, MCMC methods provide us with off-the-shelf recipes for constructing these transitions, allowing us to build the necessary Markov chains. Often (although not exclusively) the transition kernels prescribed are based upon the extra condition that the resulting Markov chain is (time) reversible. Formally, for a Markov chain $\boldsymbol{X}^0, \boldsymbol{X}^1, \boldsymbol{X}^2, \ldots$, reversibility means that the distribution of $\boldsymbol{X}_{n+1}$ given $\boldsymbol{X}_n = \boldsymbol{x}$ is the same as the distribution of $\boldsymbol{X}_{n+1}$ given $\boldsymbol{X}_{n+2} = \boldsymbol{x}$. This

means that the chain would have exactly the same statistical properties if it was run in reverse. In such cases, the detailed balance condition also holds for the unique stationary distribution π. Moreover, for reversible chains, central limit theorems for the Monte Carlo averages can be shown to apply.

The two most common MCMC algorithms are known as the (systematic sweep) Gibbs sampler and the Metropolis-Hastings algorithm. A lengthy discussion of both samplers including extensions and special cases of each can be found in Robert and Casella (2002). For the purpose of brevity, only a short summary of each is included below.

The Gibbs sampler can be used when it is possible to sample from the full conditionals of the target distribution π. Suppose $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ and that for each i we can sample from the full conditional $\pi(x_i|\boldsymbol{x}_{(i)})$, where $\boldsymbol{x}_{(i)}$ denotes $\boldsymbol{x}$ with x_i removed. The Gibbs sampler moves from $\boldsymbol{x}^t$ to $\boldsymbol{x}^{t+1}$ by updating each x_i in turn, by sampling x_i^{t+1} from $\pi(\cdot|x_1^{t+1}, \ldots, x_{i-1}^{t+1}, x_{i+1}^t, \ldots, x_n^t)$. Although each component update is reversible, the systematic nature of each sweep as a whole means that the Gibbs sampler is not. Nonetheless, with a little work the sampler can be shown (Smith and Roberts, 1993; Gamerman, 1997) to have stationary distribution π. Gibbs updates can also be combined in different ways to the systematic sampler, for example the random sweep Gibbs sampler or the reversible Gibbs sampler. A discussion of such alternatives and the relative merits of each can be found in Roberts and Sahu (1997).

The Metropolis-Hastings algorithm (MH algorithm) is an alternative to the Gibbs sampler that does not require the ability to sample directly from the full conditionals. It can be shown that a Gibbs update is a special case of an update

using the MH algorithm. The MH algorithm proceeds as follows. Suppose the chain is currently in state $\boldsymbol{x}$. We propose a move to some new state $\boldsymbol{x}'$ sampled from a proposal distribution $q(\boldsymbol{x}, \boldsymbol{x}')$ (this can be any distribution over $\mathcal{X}$ that we can directly sample from). This new state $\boldsymbol{x}'$ is accepted as the new state in our Markov chain with probability $\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{1, \frac{\pi(\boldsymbol{x}')q(\boldsymbol{x}', \boldsymbol{x})}{\pi(\boldsymbol{x})q(\boldsymbol{x}, \boldsymbol{x}')}\right\}$. Otherwise we reject $\boldsymbol{x}'$ and chain remains in the same state $\boldsymbol{x}$. The acceptance probability has this form because it is the optimal acceptance probability (in the sense of minimising the variance of the resulting Monte Carlo estimates, see Peskun, 1973 and Tierney, 1998) that ensures that the resulting transition kernel is reversible. It has the added bonus that the target distribution need only be known up to the constant of proportionality, as this cancels in the ratio.

Importantly, these two MCMC methods are by no means exhaustive and can be combined to build hybrid samplers that update some components using Gibbs kernels and others using Metropolis-Hastings steps. More sophisticated extensions include the Langevin algorithm, tempering algorithms and the algorithm designed by Tjelmeland and Hegstad (2001) which cleverly combines pairs of MCMC moves to improve performance for target distributions which typically cause problems for standard samplers. Another example is the innovative slice sampler, introduced by Neal (2003)[1] and subsequently studied by many authors. As shown by Roberts and Rosenthal (1999) the slice sampler demonstrates very appealing properties, but can often be hard to implement.

Much research has been completed on many aspects concerning the performance of standard MCMC algorithms. An interesting example of such work is the

[1] Although this paper appeared after many other papers on the subject, a preprint appeared in 1997. Additionally, similar ideas had previously been suggested by several authors, see for example Besag and Green (1993).

extensive research into the optimal scaling and acceptance probability of various Metropolis-Hastings algorithms, details of which can be found in Roberts *et al.* (1997), Roberts and Rosenthal (1998), Roberts and Rosenthal (2001) and more recently Christensen *et al.* (2004). Other examples include research into the rates of convergence of various Markov chains and various diagnostics for assessing and comparing algorithms. Much of this research is now well established and can be found in many basic reviews of the subject, such as Brooks (1998) or Robert and Casella (2002). However, active research continues in the area, for example the work presented by Neal (2004), advocating the use of non-reversible chains, proving that for finite state spaces such chains lead to lower asymptotic variance in MCMC estimates.

We mention finally a related class of methods based on MCMC. These methods, known as *coupling from the past (CFTP)* or *perfect sampling*, were introduced by Propp and Wilson (1996). The methods use a (possibly infinite) number of Markov chains to create exact samples from the posterior distribution π (rather than samples from a Markov chain which has π as a stationary distribution). Although the methods have enjoyed considerable popularity (see Brooks, 1998, Green, 1999 or Brooks *et al.*, 2002 for a review), it is fair to say that the difficulty in implementing exact samplers for many problems has so far prevented them from achieving the success that was originally envisaged. As this area is not the focus of this thesis we do not provide further details but direct the interested reader to the aforementioned papers for further details.

Having briefly revised standard MCMC algorithms and mentioned some recent research in this area, we now move on to consider reversible jump Markov chain Monte Carlo.

2.2 Reversible Jump MCMC

Since its introduction by Green (1995), it has been widely recognised that reversible jump MCMC (RJMCMC) encompasses many common MCMC algorithms including the standard MCMC algorithms discussed in the previous section. What is often not appreciated however, is that RJMCMC is really nothing more than a tightening up of the terms in the Metropolis-Hastings algorithm, to allow consideration of problems involving general state spaces, including those comprised of subspaces of different dimensions.

Despite the fact that we present RJMCMC as a simple generalisation of the MH algorithm, it is often viewed as difficult to understand. This perception in part originates from the measure theoretical presentation used when the method was first introduced. Although this formulation was necessary to demonstrate the method's validity, RJMCMC can be formulated in a way that avoids much of the measure theory. Applying the method to real problems necessitates no direct consideration of measure theory.

There have been numerous subsequent explanations of the algorithm that perhaps may be viewed as more user-friendly than the original formulation of Green. Such presentations are invaluable and will be further explored below in an attempt to demystify the reversible jump algorithm. Nonetheless, the measure theoretical approach plays an irreplaceable role in understanding the full generality of the method. Therefore we now consider the algorithm from this perspective. Readers are directed to Green (1995), Tierney (1998), Green (2003) or Waagepetersen and Sorensen (2001) for more details.

Let us consider again a general state space $\mathcal{X}$, and suppose that we are interested in some target distribution π that is defined on this state space. We are now interested in constructing a Markov chain, with transition kernel $\mathcal{K}$, that has π as its invariant distribution. With this general state space we now consider π as a probability measure on $\mathcal{X}$ and the equilibrium equation that we need to hold is

$$\int_{\mathcal{X}} \pi(\mathrm{d}\boldsymbol{x})\mathcal{K}(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}') = \pi(\mathrm{d}\boldsymbol{x}') \ .$$

Just as in the case of the MH algorithm, we make the further requirement that the resulting Markov chain is reversible. In particular we require that the chain satisfies the integrated detailed balance equation, so that for all Borel sets $\mathcal{C}$, $\mathcal{C}' \subset \mathcal{X}$,

$$\int_{\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x})\mathcal{K}(\boldsymbol{x}, \mathcal{C}') = \int_{\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}')\mathcal{K}(\boldsymbol{x}', \mathcal{C}) \ . \tag{2.1}$$

Rather than following the approach of Green (1995) and Green (2003) and expressing equation 2.1 in its full integral form, it is perhaps more instructive to look at $\mathcal{K}(\boldsymbol{x}, \mathcal{C}')$ itself. We proceed just as if we were using a normal MH update. We first draw a proposed new state $\boldsymbol{x}'$ from a proposal measure $q(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')$. We accept $\boldsymbol{x}'$ as our new state with probability $\alpha(\boldsymbol{x}, \boldsymbol{x}')$. Otherwise our old state $\boldsymbol{x}$ becomes our new state. Our transition kernel is now given by

$$\mathcal{K}(\boldsymbol{x}, \mathcal{C}') = \int_{\mathcal{C}'} q(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')\alpha(\boldsymbol{x}, \boldsymbol{x}') + \mathcal{R}(\boldsymbol{x})1_{\{\boldsymbol{x}\in\mathcal{C}'\}} \ , \tag{2.2}$$

where $1_{\{.\}}$ is the indicator function and $\mathcal{R}(\boldsymbol{x})$ is the probability of rejecting the proposed move, given by

$$\mathcal{R}(\boldsymbol{x}) = \int_{\mathcal{X}} q(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')(1 - \alpha(\boldsymbol{x}, \boldsymbol{x}')) \ .$$

Substituting equation 2.2 into 2.1 gives

$$\int_{\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}) \int_{\mathcal{C}'} q(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')\alpha(\boldsymbol{x}, \boldsymbol{x}') + \int_{\mathcal{C}\cap\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x})\mathcal{R}(\boldsymbol{x}) = \int_{\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}') \int_{\mathcal{C}} q(\boldsymbol{x}', \mathrm{d}\boldsymbol{x})\alpha(\boldsymbol{x}', \boldsymbol{x}) + \int_{\mathcal{C}'\cap\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}')\mathcal{R}(\boldsymbol{x}') \; . \tag{2.3}$$

The last terms on each side are equal. This means that equation 2.3 reduces to the following neat equation, as given in Green (2003),

$$\int_{(\boldsymbol{x},\boldsymbol{x}')\in\mathcal{C}\times\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x})q(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')\alpha(\boldsymbol{x}, \boldsymbol{x}') = \int_{(\boldsymbol{x},\boldsymbol{x}')\in\mathcal{C}\times\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}')q(\boldsymbol{x}', \mathrm{d}\boldsymbol{x})\alpha(\boldsymbol{x}', \boldsymbol{x}) \; . \tag{2.4}$$

Green (1995) discusses the case where the transition kernel is a mixture over a number of move types, so that a move of type m, taking the chain to $\boldsymbol{x}'$ is proposed with probability $q_m(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')$. This argument also allows the possibility of no move being attempted. Green shows that a sufficient condition for the reversibility of such an algorithm is exactly that given in equation 2.4 but this time for each $q_m(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')$. It is worth noting that in the notation above, if there is more than one move type (for example a birth-death step and a split-combine step, Richardson and Green, 1997), each q_m must include the probability of choosing the specific move type being attempted. Allowing different move types is an important feature of the reversible jump algorithm. Often the mixing of a reversible jump algorithm is improved by alternating reversible jump updates with standard MCMC updates. As the above formulation encompasses standard MCMC, this can be achieved by designating a reversible jump update a different type from a standard MCMC update.

Green assumes the existence of a symmetric measure μ on $\mathcal{X}\times\mathcal{X}$ which dominates

$\pi(\mathrm{d}\boldsymbol{x})q(\boldsymbol{x},\mathrm{d}\boldsymbol{x}')$. Under this assumption, $\pi(\mathrm{d}\boldsymbol{x})q(\boldsymbol{x},\mathrm{d}\boldsymbol{x}')$ has density $f(\boldsymbol{x},\boldsymbol{x}')$ (the Radon-Nikodym derivative) with respect to μ. This means that equation 2.4 can be rewritten

$$\int_{(\boldsymbol{x},\boldsymbol{x}')\in\mathcal{C}\times\mathcal{C}'} \alpha(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x},\boldsymbol{x}')\mu(\mathrm{d}\boldsymbol{x},\mathrm{d}\boldsymbol{x}') = \int_{(\boldsymbol{x},\boldsymbol{x}')\in\mathcal{C}\times\mathcal{C}'} \alpha(\boldsymbol{x}',\boldsymbol{x})f(\boldsymbol{x}',\boldsymbol{x})\mu(\mathrm{d}\boldsymbol{x}',\mathrm{d}\boldsymbol{x}) \ . \tag{2.5}$$

Clearly this holds for all Borel $\mathcal{C},\ \mathcal{C}'$ if

$$\alpha(\boldsymbol{x},\boldsymbol{x}') = \min\left\{1, \frac{f(\boldsymbol{x}',\boldsymbol{x})}{f(\boldsymbol{x},\boldsymbol{x}')}\right\} \ .$$

As Green notes, if we express this ratio less exactly and replace the density f by the appropriate measures, the resulting acceptance probability,

$$\alpha(\boldsymbol{x},\boldsymbol{x}') = \min\left\{1, \frac{\pi(\mathrm{d}\boldsymbol{x}')q(\boldsymbol{x}',\mathrm{d}\boldsymbol{x})}{\pi(\mathrm{d}\boldsymbol{x})q(\boldsymbol{x},\mathrm{d}\boldsymbol{x}')}\right\} \ , \tag{2.6}$$

clearly resembles the acceptance probability of the standard MH algorithm.

The abstract nature of the above approach is noted by almost all authors that introduce the method in this way. To counter this, as mentioned above, many authors present an alternative, more constructive, formulation. By doing so, the measure μ and Radon-Nikodym derivative become indirect bi-products of considering the actual technicalities behind the moves.

In general, the less formal discussion of RJMCMC adopts one of two styles. Several authors (Brooks, 1998; Green, 1999; Green and Mira, 2001; and Green, 2003) explore the actual mechanics of the transitions in terms of random numbers, while still retaining a relatively general framework. Other authors choose to demonstrate the method in the context of a general problem requiring jumps between models. Such authors include Godsill (2001), Brooks *et al.* (2003b) and Cappé *et al.* (2003). For convenience we refer to this class of

problems simply as model jumping problems throughout the remainder of the thesis. We return to the general model jumping problem in the next section.

To continue our discussion, we follow the approach of the first set of authors (in particular Green, 2003) and try to make RJMCMC more transparent whilst still maintaining a relatively general setting. However, in order to make the method more definite, it is necessary to move away from the total generality of the measure theoretical approach. In particular, we suppose now that $\mathcal{X} \subset \mathbb{R}^d$. (It is important to note that Green goes on to show that much more general state spaces, for example $\mathcal{X} = \bigcup_k (\{k\} \times \mathbb{R}^{n_k})$, can be considered with little change to the resulting expressions.) In accordance with Green, we make the further restriction that π has a density (which, in an abuse of notation, is denoted by $\pi(\cdot)$) with respect to the d-dimensional Lebesgue measure.

We prescribe the proposed move from $\boldsymbol{x} \in \mathbb{R}^d$ to $\boldsymbol{x}' \in \mathbb{R}^d$ as follows. First we generate r random numbers, $\boldsymbol{u}$, which have known density g. The proposed new state is then given by $\boldsymbol{x}'$, where $(\boldsymbol{x}', \boldsymbol{u}') = \boldsymbol{t}(\boldsymbol{x}, \boldsymbol{u})$ for some known deterministic function $\boldsymbol{t}$. The variables $\boldsymbol{u}'$ are those that would be needed for the reverse transition (see below). Comparing the standard MCMC proposal $q(\boldsymbol{x}, \boldsymbol{x}')$ with this formulation, it can be seen that all standard MCMC proposals can be formed in this way. As Green comments, the redundancy of splitting the generation of the random numbers $\boldsymbol{u}$, from the transformation onto $\boldsymbol{x}'$ using $\boldsymbol{t}$, is deliberate, allowing the same proposal to be expressed in several different ways. This permits the user to choose the most convenient formulation for the particular problem.

Having considered the transition from $\boldsymbol{x}$ to $\boldsymbol{x}'$ we next consider the reverse

transition. This transition is made in an analogous fashion to the original proposal, so that $(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{t}'(\boldsymbol{x}', \boldsymbol{u}')$, where $\boldsymbol{u}'$ are random numbers with known density g', and $\boldsymbol{t}'$ is again a deterministic function.

With the transition and its reverse so defined, equation 2.1 becomes

$$\int_{(\boldsymbol{x},\boldsymbol{x}')\in \mathcal{C}\times\mathcal{C}'} \pi(\boldsymbol{x})g(\boldsymbol{u})\alpha(\boldsymbol{x},\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{u} = \int_{(\boldsymbol{x},\boldsymbol{x}')\in \mathcal{C}\times\mathcal{C}'} \pi(\boldsymbol{x}')g'(\boldsymbol{u}')\alpha(\boldsymbol{x}',\boldsymbol{x})\mathrm{d}\boldsymbol{x}'\mathrm{d}\boldsymbol{u}' \ . \tag{2.7}$$

Green (2003) notes that provided that the transformation from $(\boldsymbol{x}, \boldsymbol{u})$ to $(\boldsymbol{x}', \boldsymbol{u}')$ is a diffeomorphism (i.e. the transformation and its inverse are differentiable) then by basic calculus

$$\mathrm{d}\boldsymbol{x}'\mathrm{d}\boldsymbol{u}' = |J|\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{u} \ , \tag{2.8}$$

where $J = \frac{\partial(\boldsymbol{x}',\boldsymbol{u}')}{\partial(\boldsymbol{x},\boldsymbol{u})}$ is the Jacobian of the transformation from $(\boldsymbol{x}, \boldsymbol{u})$ to $(\boldsymbol{x}', \boldsymbol{u}')$. Substituting equation 2.8 into equation 2.7, it is evident that a suitable (and optimal) choice of $\alpha(\boldsymbol{x}, \boldsymbol{x}')$ is given by

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{1, \frac{\pi(\boldsymbol{x}')g'(\boldsymbol{u}')}{\pi(\boldsymbol{x})g(\boldsymbol{u})}\,|J|\right\} \ . \tag{2.9}$$

Equation 2.9 provides a much more appealing expression for constructing the acceptance probabilities for a given algorithm. However, an important point should be recognised about the above formulation. Green remarks upon the frequently held inaccurate understanding that the Jacobian comes from the dimension jump often associated with reversible jump algorithms. As Green correctly asserts, the Jacobian factor actually arises from the specification of $\boldsymbol{x}'$ and $\boldsymbol{u}'$ indirectly, in terms of $\boldsymbol{x}$ and $\boldsymbol{u}$. The misunderstanding no doubt arises

from the fact that the standard MCMC algorithm does not include this Jacobian factor. However, this is because by the design of the proposal in the standard case, this Jacobian factor is always equal to 1.

The realisation that the Jacobian is not a relic of dimension change is further emphasised by the fact that until this point we have only considered a problem of fixed dimension d. However, following Green (1999) and Green (2003), consideration of the variable dimension problem is an easy extension of our above framework.

Suppose now that our state space $\mathcal{X}$ is no longer a subset of $\mathbb{R}^d$, but rather a countable union of spaces $\mathcal{X}_k$, with possibly different dimensions n_k, so that $\mathcal{X} = \cup_k \mathcal{X}_k$. Again we are interested in π, the distribution over $\mathcal{X}$. Clearly π no longer has a density with respect to the d-dimensional Lebesgue measure. However, if we suppose $\mathcal{X}_k \subseteq \mathbb{R}^{n_k}$, we make the restriction that for each k, π has a density over $\mathcal{X}_k$ (denoted by $\pi(\cdot|k)$ in an abuse of notation) with respect to the n_k-dimensional Lebesgue measure.

Suppose we now wish to propose a move from the current state $\boldsymbol{x}$ that has dimension d to a new state $\boldsymbol{x}'$ that has dimension d'. Just as before, the move from $\boldsymbol{x}$ to $\boldsymbol{x}'$ is made by generating r random numbers $\boldsymbol{u}$. Again we set $(\boldsymbol{x}', \boldsymbol{u}') = \boldsymbol{t}(\boldsymbol{x}, \boldsymbol{u})$, but the deterministic function $\boldsymbol{t}$ is such that $\boldsymbol{t} : \mathbb{R}^d \times \mathbb{R}^r \rightarrow \mathbb{R}^{d'} \times \mathbb{R}^{r'}$. The reverse move is similarly constructed, so that with the aid of r' random numbers $\boldsymbol{u}'$, $(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{t}'(\boldsymbol{x}', \boldsymbol{u}')$ where $\boldsymbol{t}' : \mathbb{R}^{d'} \times \mathbb{R}^{r'} \rightarrow \mathbb{R}^d \times \mathbb{R}^r$ is a deterministic function. Following the same argument as the above we are able to show that provided the transformation from $(\boldsymbol{x}, \boldsymbol{u})$ to $(\boldsymbol{x}', \boldsymbol{u}')$ remains a diffeomorphism, equation 2.9 holds without change. As Green notes, so that this mapping and its inverse

remain differentiable (i.e. for this transformation to remain a diffeomorphism) we need the following relationship to hold

$$d + r = d' + r' \ .$$

This is frequently referred to as the dimension matching assumption.

This formulation provides a template for the most general reversible jump algorithm. As illustrated above, the method can be applied to both variable and fixed dimension problems. For more general problems it may not be sufficient to consider only the Lebesgue measure. However, the measure plays an important role in the large majority of RJMCMC problems, even if only as a factor in the overall measure and the restrictions of the above illustration are easily generalised. This is best seen in the context of real problems, such as the model jumping problem (see section 2.3).

In the following section we provide a brief review of some of the many problems to which RJMCMC has been applied. The literature from which these applications are taken provides further details of how the RJMCMC algorithm can be used as an invaluable statistical tool to make inference about a wide variety of problems.

2.3 Applications of Reversible Jump MCMC

The statistical problems to which the reversible jump method has been applied can all be formulated as a generic model jumping problem. We consider the general model jumping problem before looking at the specific examples that originate from this formulation. The general model jumping problem has been discussed by several authors, including Godsill (2001), Brooks *et al.* (2003b) and

Green (2003). The reader is referred to these papers for further details.

Suppose that we have a countable set of possible models $\mathcal{M}$, indexed by variable k. We suppose that model k has associated parameter vector $\boldsymbol{\theta}_k$ of length n_k. Our task is to make inference about the joint distribution π of $(k, \boldsymbol{\theta}_k)$. Referring to the terminology of the last section, $\mathcal{X} = \cup_k(\{k\} \times \mathcal{X}_k)$, where $\mathcal{X}_k$ is some n_k-dimensional subspace. To be consistent with the existing literature, we take $\mathcal{X}_k \subseteq \mathbb{R}^{n_k}$. To indicate that this restriction holds, we replace the notation $\mathcal{X}_k$ by $\boldsymbol{\Theta}_k$. Then $\pi(\cdot)$ denotes a density across $\mathcal{X} = \bigcup_k(\{k\} \times \boldsymbol{\Theta}_k)$ and for each k, $\pi(k, \cdot)$ is absolutely continuous in $\mathbb{R}^{n_k}$ with respect to the n_k-dimensional Lebesgue measure.

In most applied cases the model jumping problem is Bayesian in nature and π is a posterior distribution. In this context, we might typically assign some prior distribution over $\mathcal{X}$, with density $p(k, \boldsymbol{\theta}_k)$. Denoting the likelihood of the data $\boldsymbol{Y}$ by $\mathcal{L}(\boldsymbol{Y}|k, \boldsymbol{\theta}_k)$, our posterior distribution, which in an abuse of notation we denote $\pi(k, \boldsymbol{\theta}_k)$, is given by

$$\pi(k, \boldsymbol{\theta}_k) = \pi(k, \boldsymbol{\theta}_k|\boldsymbol{Y}) \propto p(k, \boldsymbol{\theta}_k)\mathcal{L}(\boldsymbol{Y}|k, \boldsymbol{\theta}_k) \ .$$

The constant of proportionality is not required as it cancels out of the numerator and denominator of the acceptance ratio (see equation 2.9).

As Green (2003) notes, the Bayesian formulation, when allied with RJMCMC provides a sensible way of making combined inference about $(k, \boldsymbol{\theta}_k)$. Making joint inference from a frequentist perspective is unnatural, because the approaches used for k and $\boldsymbol{\theta}_k$ would, in general, be very different. Nonetheless, it is worth commenting that it is the statistical modelling of the problem rather than the

RJMCMC algorithm itself that relies on the Bayesian paradigm.

Using the same formulation with a slightly altered notation, Brooks *et al.* (2003b) show that the acceptance probability (given in equation 2.9) for a proposed jump from state $\boldsymbol{\theta}_k$ in model k, to state $\boldsymbol{\theta}'_{k'}$ in model k', becomes

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{1, A(\boldsymbol{x}, \boldsymbol{x}')\right\} \ , \tag{2.10}$$

where $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$, $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$ and

$$A(\boldsymbol{x}, \boldsymbol{x}') = \frac{\pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}g'(\boldsymbol{u}')}{\pi(k, \boldsymbol{\theta}_k)\rho_{k,k'}g(\boldsymbol{u})} \left|\frac{\partial(\boldsymbol{\theta}'_{k'}, \boldsymbol{u}')}{\partial(\boldsymbol{\theta}_k, \boldsymbol{u})}\right| \ . \tag{2.11}$$

Here, $\rho_{k_0,k_1} = \rho_{k_0,\boldsymbol{\theta}_{k_0}}(k_1)$ is the probability of proposing a move to model k_1 given that the chain is currently in model k_0 at state $\boldsymbol{\theta}_{k_0}$. If there is more than one move type, ρ_{k_0,k_1} must include the probability of choosing the particular move type under consideration (given the current state). Throughout this thesis we do not require the dependence of ρ_{k_0,k_1} on $\boldsymbol{\theta}_{k_0}$, so we use this notation for simplicity. In addition, g now denotes the density of the r random numbers $\boldsymbol{u}$ such that $(\boldsymbol{\theta}'_{k'}, \boldsymbol{u}') = \boldsymbol{t}(\boldsymbol{\theta}_k, \boldsymbol{u})$, and similarly for the reverse transition. The dimension matching constraint is now $n_k + r = n_{k'} + r'$.

The generic nature of the model jumping problem has been adapted to many individual applications. One example that has been applied to many datasets is a change point model. This was used by Green (1995) in the original RJMCMC paper as an example that highlighted the capabilities of the reversible jump method. This change point formulation models a series of times (the data) as a Poisson process with rate $x(t)$, where $x(t)$ is constrained to be a step function with an unknown number of steps. In terms of the generic model jumping problem, model k corresponds to the rate having k steps, and $\boldsymbol{\theta}_k$ is a vector of

dimension $2k + 1$, corresponding to the step heights, and times at which the rate changes. This model has been applied to the point process of times of coal mining disasters (Green, 1995; Green, 2003) and cyclone occurrences (Green, 1999; Green and Mira, 2001).

There are several other problems that are direct derivatives of the model jumping problem. One such example is the modelling of an autoregressive time series, where the order of the process is taken to be unknown. This model has been used by Godsill (2001) for simulated data and Brooks *et al.* (2003b) to study sea pressures. Here, model k corresponds to the k^{th} order process, where $\boldsymbol{\theta}_k$ is the $k + 1$-dimensional vector of the k autoregression coefficients and the standard deviation of the noise. As noted by many authors, this type of problem is particularly nice because of its nested structure. In other words, the likelihood given $k = k_0$ and $\boldsymbol{\theta}_{k_0} = (a_1, a_2, \ldots, a_{k_0}, \sigma)$ is identical to that given $k = k_0 + 1$ and $\boldsymbol{\theta}_{k_0+1} = (a_1, a_2, \ldots, a_{k_0}, 0, \sigma)$. This facilitates the construction of across dimension jumps.

Another area to which RJMCMC methods have been applied is image analysis. Al-Awadhi *et al.* (2004) use a marked point process approach to fitting an unknown number of ellipsoids to recover an image. Again the relationship to the model jumping problem is immediately apparent, with each model corresponding to the number of ellipsoids that are fitted. A slightly different problem was studied by Green (1995), where the goal was to segment a digital image into an unknown number of homogeneous regions as a prelude to further analysis.

The general model jumping problem also forms a basis for the final example considered in Green (1995). For these types of problem, termed partition models,

each model k corresponds to a particular partition of some overall indexing set. By partitioning the indexing set, a different likelihood is applicable for the data set under study. Following the example of Consonni and Veronese (1995), Green sets the indexing set to be the integers from 1 to n, corresponding to n data responses $y_1, \ldots, y_n$. In this case, each response is binomially modelled, with the binomial probabilities implicitly dependent upon the particular partition. Green notes that such a model arises naturally in many problems including ANOVA, factorial experiments, and variable selection in regression.

A particularly popular application of RJMCMC is mixture modelling, where the number of components k of the mixture is unknown. Richardson and Green (1997) formulate the basic mixture model by assuming that the data $\boldsymbol{y}$ have density function given by

$$f(\boldsymbol{y}|\boldsymbol{\lambda}_k, \boldsymbol{\phi}_k) = \sum_{l=1}^{k} \lambda_k^l f_l(\boldsymbol{y}|\boldsymbol{\phi}_k^l) \ , \tag{2.12}$$

where f_l is the density of the l^{th} component of the mixture and $\lambda_k^l \geq 0$ satisfy $\sum_{l=1}^{k} \lambda_k^l = 1$. In such a setting, model k corresponds to the mixture of k components. Inference is then made about the joint distribution of the number of components k, the component weights $\boldsymbol{\lambda}_k$ and the model component parameters $\boldsymbol{\phi}_k^l$, $l = 1, \ldots, k$. Here, $\boldsymbol{\lambda}_k$ and $\boldsymbol{\phi}_k$ correspond to $\boldsymbol{\theta}_k$. These models often have hidden complexities such as the need to make inference about hidden allocation variables and the need for ordering constraints for identifiability. Such difficulties are well discussed in the considerable literature on the subject. For problems involving normal mixtures, readers are referred to: Richardson and Green (1997) (applied to enzymatic activity in the blood, lake acidity in northern Wisconsin and velocities of galaxies diverging from our own); Brooks (1998); Brooks *et al.* (2003b) (enzyme activity in the blood); and Cappé *et al.*

(2003) (volatility of IBM stock over a period of 5 years and wind intensity in Athens). Along the same lines, Green and Richardson (2002) look at mixtures of spatial Poisson processes to model disease mapping for simulated datasets.

The large number of disciplines in which reversible jump techniques are now being applied means that there is little chance of reviewing even a representative sample of the applications. Other interesting examples worth mentioning include the application of RJMCMC to surface approximation by marked Poisson processes (Heikkinen, 2003) and the application of the technique to fitting graphical Gaussian models (Brooks *et al.*, 2003b). Despite the examples of RJMCMC for a wide variety of problems, there are many more cases where the technique has not been adopted but which may have benefited from employing the method. In the next section we review some of the difficulties associated with the method which may have contributed to the reluctance of parts of the scientific community to embrace reversible jump methods.

2.4 Difficulties with Reversible Jump MCMC

As with the majority of algorithms, when reversible jump is applied to practical problems, difficulties can occur. Some of the difficulties may be problem-specific or may arise due to misunderstanding, naivety or even errors on the part of the user. Nonetheless there are several recurring themes amongst the literature about reversible jump applications and these are worthy of review.

One difficulty frequently encountered when designing reversible jump algorithms is the construction of efficient proposals. Problems of this nature have been reported by many different authors. As noted by Al-Awadhi *et al.* (2004),

inefficient proposals often result in low acceptance rates. Typically, dimension jumping moves in reversible jump samplers display much lower acceptance rates than other types of move, where the dimension is held fixed and other parameters are updated. Al-Awadhi *et al.* note that models with multimodal target distributions give particularly low acceptance rates. As Liu *et al.* (2001) explain, this is often because samplers become trapped in local modes.

Low acceptance rates essentially lead to poor mixing, which in turn leads to slow convergence. As noted by Green and Mira (2001), low acceptance rates intuitively increase autocorrelation in the Markov chain. This intuition is formalised by results detailed in Peskun (1973) and Tierney (1998).

Many authors (Godsill, 2001; Al-Awadhi *et al.*, 2004; Brooks *et al.*, 2003b) observe that for the trans-dimensional jumps often associated with RJMCMC it is not possible to simply extend standard MCMC analysis of optimal scaling of proposals (see references in section 2.1). The hurdle occurs because there is no longer the concept of Euclidean closeness. Indeed, for complex models the absence of any obvious concept of closeness can make designing even sensible proposals a very difficult task. Such models occur frequently throughout the literature. Heikkinen (2003) demonstrates that in some extreme cases, it is not only designing efficient proposals that becomes troublesome but also designing valid proposals.

In order to design more efficient problems it has become standard practice to *tune* proposals. Tuning is the practice of doing several short runs of a RJMCMC algorithm, each time changing certain aspects of a proposal. The specification associated with the run that maximises acceptance rates may then be chosen

as the one to be used for the main RJMCMC analysis. Green (1999) gives examples of aspects of the proposal, such as scaling, blocking of updates and re-parameterisation, that might be tuned. However, as Green (2003) notes, algorithm construction followed by tuning is often seen as an awkward and difficult process.

A further problematic area when using reversible jump is the determination of whether or not a particular realisation of a Markov chain has converged to the stationary distribution. This has been the subject of much research for standard MCMC, with some, but not extensive, progress being made. For Markov chains over general spaces the topic becomes yet more difficult (Green, 1995). One reason why this proves hard is that assessing convergence within models that are rarely visited is very difficult. Some progress has been made by Brooks and Giudici (2000) who develop a diagnostic applicable to reversible jump that can be decomposed into separate terms explained by fixed dimension and across dimension variation. Brooks *et al.* (2003a) also address the problem, offering alternative methods to assess the closeness of multiple replications of the chains.

A related but different problem is the effective comparison of samplers. Brooks *et al.* (2003b) discuss a range of techniques, both graphical and numerical that can be used to assess reversible jump methods. The graphical techniques include trace plots and autocorrelation plots, whereas the numerical statistics include acceptance probabilities, total number of models visited and estimates of the effective sample size (the number of independent samples that would be equivalent to the dependent MCMC sample). Although none of these comparisons is sufficient on its own, by using a collection of such methods sensible conclusions can be drawn.

Another aspect of concern with regard to RJMCMC is that there is conflicting literature as to whether across model jumps should be avoided or encouraged. Although reversible jump samplers may demonstrate poor mixing, the situation can be envisaged where moving from one model to another escapes a local mode.

Green suggests that whether we use RJMCMC or just separate within-model MCMC runs for a number of models should depend upon the particular situation. For example, RJMCMC may be better if we want to make joint inference about the model and the model parameters. On the other hand, if we are solely interested in the model parameters for each (and in particular the most probable) model, then a within-model approach may be more suitable. However, some authors (for example Dellaportas *et al.*, 2002 and Han and Carlin, 2001) suggest reversible jump need not be used even if posterior model probabilities are desired. This alternative approach involves the estimation of the marginal likelihood of each model using within-model samplers. These marginal likelihoods can then be used to compare models by means of Bayes factors. For more details of these methods see, for example, Meng and Wong (1996), Chib and Jeliazkov (2001), Meng and Schilling (2002) and Mira and Nicholls (2004). In comparing the merits of reversible jump and this alternative approach, Green (2003) notes that if there are not too many models under consideration, each with a good within-model MCMC sampler, then it may be considerably less efficient to design a reversible jump sampler from scratch. However, the various methods for computing marginal likelihoods from within-model runs are often themselves difficult to implement, meaning that this may not necessarily be the case.

It is hopefully clear from this section that there are many areas of reversible jump MCMC that require further research. In this thesis we concentrate only on alleviating the problems of inefficient proposals for reversible jump samplers. Our research attempts to address this problem from the perspective of automatic reversible jump methods. We detail our research towards providing a generic automatic sampler that will perform well for a wide variety of problems, hopefully eliminating the need to expend considerable effort on designing efficient proposals for each particular problem. Before we continue with this thesis, the final section in this chapter details some recent contributions on the subject of efficient proposal design that adopt alternative perspectives to those that we offer in later chapters.

2.5 Methodological Developments

We begin our discussion of research into efficient proposal design by considering an extension of the reversible jump algorithm known as the *delayed rejection sampling* algorithm.

Delayed rejection sampling was first introduced by Tierney and Mira (1999) as an extension to the standard Metropolis-Hastings method. Just as in the standard MCMC algorithm, given that the chain is currently in $\boldsymbol{x}$, a new state $\boldsymbol{x}'$ is proposed from a proposal distribution q and accepted with the usual probability. The difference between delayed rejection sampling and standard Metropolis-Hastings algorithms occurs if the move is rejected. Whereas standard MCMC takes the new state of the chain to be $\boldsymbol{x}$, the delayed rejection sampler proposes a second candidate state $\boldsymbol{x}''$, which is allowed to depend upon the rejected state $\boldsymbol{x}'$. This proposed state is accepted as the new state of the chain

with an appropriate probability. Mira (1998) shows this probability ensures detailed balance holds for the second stage of the transition and Tierney and Mira show this results in the full transition kernel being reversible.

Green and Mira (2001) extend the delayed rejection algorithm to reversible jump samplers. This is achieved by relaxing the requirement that the reverse move must also propose and reject the state $\boldsymbol{x}'$ in order to move from $\boldsymbol{x}''$ to $\boldsymbol{x}$. An interesting feature of the work is the fact that the delayed rejection algorithm requires three dimension matching constraints.

The modest increase in trans-dimensional acceptance rates reported by Green and Mira comes at the price of increased computational cost. Although the method could be extended beyond two stages, the gain in computational expense would outweigh any benefits. However, the authors suggest that by being more inventive in their second stage proposals (for example using bold first proposals combined with more timid second stage proposals) the increase in acceptance rate may have been enhanced. Nonetheless, because the question of sensible proposal design for general reversible jump problems remains difficult the benefits of the method are limited.

Several other authors have also tried to adapt the reversible jump algorithm to improve the acceptance rates. Al-Awadhi *et al.* (2004) propose another example of an algorithm designed to address this goal. By a clever use of an intermediate within-model chain which maintains detailed balance with respect to an alternative distribution $\pi^\star$, the method is designed to allow the proposed state $\boldsymbol{x}'$ to escape the mode that $\boldsymbol{x}$ may be in. This relies on a suitable choice of $\pi^\star$, which Al-Awadhi *et al.* suggest could be a flatter version of π, perhaps

achieved by tempering. Essentially the method is not a true generalisation of reversible jump MCMC but rather just a particular choice of proposal mechanism.

One drawback of Al-Awadhi *et al.*'s algorithm is the convoluted reverse move which requires the intermediate within-model moves to occur before the trans-dimensional move. This could be relaxed if a method could be found where the chain did not have to pass through the same path for the reverse transition. Unfortunately this is not a trivial task. In addition, with the exception of a toy example, the method has so far only been applied to problems where the existing sampler had acceptance rates of almost zero. While the algorithm increases these rates, they remain prohibitively small. It remains an interesting task to study the behaviour of the algorithm for problems where the original acceptance probability was not so small.

Perhaps the biggest recent step towards improving the understanding of efficient proposal design is the considerable work done by Brooks *et al.* (2003b). This work aims at developing methods for good choices of the parameters of the proposal densities g that are used in the construction of trans-dimensional proposals. Underlying this work is the assumption that for each proposal the deterministic transition function $\boldsymbol{t}$ is fixed. The different methods introduced by the authors broadly fall into two general classes. As Brooks *et al.* demonstrate, the two classes of methods can be used in tandem but we choose to summarise each of these approaches in turn. We begin with the class of methods that the authors call *order methods*.

Order methods proceed by analysing the acceptance ratio $A(\boldsymbol{x}, \boldsymbol{x}')$ in an attempt to achieve automatic parameterisation of the proposal distribution g. The

methods are based on the idea that for any current state $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ it is possible to identify a special *centring point* $c(\boldsymbol{\theta}_k)$ in a proposed new model k'. The authors discuss ways of choosing this centring point, including a method for nested models and a more general approach which they call *conditional maximisation*. This approach sets $c(\boldsymbol{\theta}_k) = \boldsymbol{\theta}^{\star}_{k'}$, where $\boldsymbol{\theta}^{\star}_{k'}$ is the value of $\boldsymbol{\theta}'_{k'}$ that maximises $\pi(k', \boldsymbol{\theta}'_{k'})$, conditional on the current parameter value $\boldsymbol{\theta}_k$.

At the chosen centring point the order methods impose various constraints on $A(\boldsymbol{x}, \boldsymbol{x}')$ and its derivatives, resulting in equations that can be solved to yield the parameters of g. For example, the J^{th} order method imposes the conditions $A(\boldsymbol{x}, \boldsymbol{x}')|_{\boldsymbol{x}'=(k', c(\boldsymbol{\theta}_k))} = 1$ and $\frac{\partial^j}{\partial \boldsymbol{u}^j} A(\boldsymbol{x}, \boldsymbol{x}')\Big|_{\boldsymbol{x}'=(k', c(\boldsymbol{\theta}_k))} = \mathbf{0}$, for $j = 1, \ldots, J$. These equations can be solved to give parameters of g.

While it is not clear that requiring the acceptance probability to be 1 at the chosen centring point is a good thing, the high order criteria are appealing (scaling the proposal to maintain a high acceptance rate for a wide range of values) and yield good numerical results. This leads to the possibility (recognised by the authors) that the zeroth order criteria could be dropped from higher order methods.

The second class of methods introduced by Brooks *et al.* is entitled the *saturated state space approach*. The idea behind the saturated state space approach is to use auxiliary variables $\boldsymbol{u}$ to augment the state space, so that all models have the same dimension. Similar to the product space approach introduced by Godsill (2001) (see below), the purpose of the auxiliary variables is to aid future proposal design. One feature of the method is the use of deterministic proposals for the new state $\boldsymbol{\theta}'_{k'}$, conditional on a new model k' having been chosen

using a random kernel. These deterministic transitions are combined with suitable within-model updates of the state variable $\boldsymbol{\theta}_k$ and the auxiliary variables $\boldsymbol{u}_k$.

The authors suggest the use of within-model MCMC for the state variables and either autoregressive or Gibbs updates for the auxiliary variables (depending on whether the auxiliary variables are assumed to be independent or correlated). This approach allows control over how far a proposed state $\boldsymbol{\theta}'_{k'}$ will be from the last time the chain was in model k'. As Godsill (2003) comments, the examples cited in the paper provide evidence that the temporal memory of the auxiliary variables allows the resulting Markov chain to better explore the tails of the stationary distribution. In particular, when used with higher order methods the numerical results offer a significant improvement over those of the standard reversible jump scheme.

As Brooks *et al.* express, there remain many directions in which the work could be extended. Indeed, Ehlers and Brooks (2002) develop the methods a little further, considering more flexible trans-dimensional moves than those considered in the original paper. The authors also provide examples of the method for a wider class of problems. Godsill (2003) notes that the saturated space approach is open for development, including the possibility of random between model proposals involving some of the components of $\boldsymbol{u}_k$. More generally, the choice of the deterministic transition function $\boldsymbol{t}$ remains an important and challenging question (see for example Hastie, 2003). These subjects and many others motivated by this paper will undoubtedly be the subject of future research.

Although this section summarises some significant progress in the direction of efficient proposal design for RJMCMC, reversible jump is not the only approach

for making inference about trans-dimensional problems. One class of alternative models to RJMCMC is known as the product space approach and was first used to consider trans-dimensional problems by Carlin and Chib (1995). Since then, interest in the methods has increased, leading to an improved formulation proposed by Godsill (2001) and Godsill (2003), that encompasses both Carlin and Chib's original sampler and the reversible jump method.

The product space approach is similar to the saturated state space approach introduced by Brooks *et al.* and discussed above. Essentially, the idea is that by using auxiliary variables to augment the state space, a fixed dimensional sampler can be used, avoiding the need for tricky trans-dimensional moves. For each model k, the method relies on the specification of a distribution, which is termed the *pseudo prior*, for the variables that do not contribute directly to that particular model.

The choice of pseudo priors has a considerable effect upon the efficiency of the MCMC sampler and the reversible jump problem of finding suitable proposal distributions is directly replaced by finding suitable pseudo priors. Brooks *et al.*'s saturated state space approach resolves some of the issues with the product space approach, for example replacing the specification of pseudo priors by the mechanism for updating $\boldsymbol{u}_k$. Furthermore, Brooks *et al.* (2003b) also note that their algorithm does not have the same extensive storage requirements that are typically a feature of the product space approach. Research continues into the product space approach but we omit further details as it is not the focus of this thesis. We direct the reader to Green and O'Hagan (1998), Godsill (2001) and Godsill (2003) for further details.

For the specific case of mixture models with an unknown number of components, a second alternative to RJMCMC is the point process approach introduced by Stephens (2000). Cappé *et al.* (2003) suggest how this method can be extended to more general problems. The method is similar to reversible jump, but works in continuous time and accepts all jumps between models. In order to maintain detailed balance, the acceptance probabilities are replaced by the length of time the process spends in each model between moves. Despite the elegance of the method, Cappé *et al.* (2003) compare the approach to reversible jump and conclude that low acceptance rates of RJMCMC samplers are simply replaced by phases where the point process approach does not move between models. In essence, the point process approach does not offer a significant gain in performance over reversible jump. To emphasise the similarity between the approaches the authors construct a sequence of reversible jump samplers, so that by taking a particular continuous time version of each of these samplers, Stephens's point process sampler can be viewed as a limit of this sequence.

Our discussion of some other work on the subject of efficient proposal design and automatic samplers has been deliberately delayed until later chapters, so that the direct contribution of such work to our new research can be fully appreciated. Since we do not draw directly upon any of the contributions included in this section, our discussion included here has been limited. Indeed, only a flavour of the important concepts has been provided and we recommend that researchers keen to make progress in this area should seek full appreciation of the existing literature.

Chapter 3

An Application of Reversible Jump MCMC in Biology: The Rb9 Problem

In this chapter we present an extended new application of RJMCMC to a question arising from the field of biology. It is intended that applying the abstract concepts discussed in chapter 2 to a specific example will highlight some practical issues of the design of a reversible jump algorithm. Furthermore, we return to this example in chapter 5 to demonstrate the applicability of the automatic sampler introduced therein. An alternative commentary to the work in this chapter, along with additional classical statistical analysis can be found in Newton and Hastie (2004).

3.1 Introduction

Contrary to prior expectation, biological scientists have frequently observed that the number of cancerous tumours occurring in a tissue during a fixed time exhibit extra-Poisson variation (Drinkwater and Klotz, 1981; Moser *et al.*, 1992; and Ramachandran *et al.*, 2002). Before such observations, it was believed that tumours occurred independently at some unknown rate, suggesting that the Poisson distribution for tumour multiplicities might be a natural candidate

(Moolgavkar and Knudson, 1981 and Kokoska, 1987). In many experimental settings however, it is easy to imagine various factors contributing to the introduction of extra-Poisson variation in the stochastic process underlying tumour development. While the identification of the sources of extra-Poisson variation is of interest to cancer researchers, equally important to them is an understanding of the biological conditions that seem to result in tumours forming independently of one another.

In this chapter we concentrate on a small laboratory study involving the adenometous polyposis coli gene that appears to demonstrate independence of intestinal tumours in certain mice. In section 3.2 we propose an appropriate statistical model for the stochastic process underlying the study. Section 3.3 then discusses how we use RJMCMC to make inference under this model. The results and conclusions are presented in sections 3.4 and 3.5. Before proceeding we introduce the experiment and data in more detail.

The adenometous polyposis coli gene (Apc in mice, APC in humans) plays the role of a tumour suppressor gene for intestinal cancer (see for example Hardy *et al.*, 2000). Specifically, individuals with a defective allele corresponding to the Apc gene have a considerably increased susceptibility to developing the disease. One particular strain of laboratory mouse, known as the multiple intestinal neoplasia (Min) mouse, has a deficient copy of the Apc gene and if the healthy Apc allele is inactivated tumours develop (Moser *et al.*, 1992). Use of such mice allows experimental biologists the opportunity to study and better understand the genetics behind the formation of intestinal cancer tumours.

We consider a small study of four groups of Min mice as presented by Haigis and

Dove (2003). The mice studied by the authors were created to be genetically identical, apart from harbouring a group dependent form of a chromosome translocation known as the Robertsonian translocation (Rb9). The result of this translocation, whereby chromosomes 7 and 18 (the second of which carries the Apc gene) are fused together, is that the genome is altered in its organisation, while the genomic content is retained.

Throughout this paper we retain the labelling of the original paper. The +/+ group refers to the control group with no Robertsonian translocation. The Rb9 trans and Rb9 cis each have one fused Rb9 chromosome and one each of chromosomes 7 and 18, the difference being that the Min allele is on the fused chromosome in the Rb9 cis case and on the chromosome 18 in the Rb9 trans case. Finally the Rb9/Rb9 is homozygous in the Rb9 chromosome (i.e. both pairs of chromosomes 7 and 18 have fused).

The tumour count data from Haigis and Dove's study is tabulated in table 3.1. The number of mice in each group is relatively small (16, 17, 15 and 18 respectively) but this is a feature of many mouse studies in biology. As can be seen from the data, the reduced tumour counts of the Rb9 groups is immediately apparent. The original study looked at ways in which Rb9 affects the inactivation of the wild-type Apc allele and concluded that mice with Rb9 demonstrated a significant reduction in tumour multiplicity. The fact that the presence of the Rb9 fused chromosome appears to suppress tumour formation demonstrates that chromosomal factors play a significant role in an individual's susceptibility to intestinal cancer. For further details the interested reader is referred to the original paper.

Group	Tumour Multiplicities	Mean	Var.
+/+	80,103,112,121,121,121,131,140,140,150,166,169, 194,199,199,262	150.5	2102.1
Rb9 trans	5,7,8,8,9,9,11,12,12,13,13,13,14,15,15,16,18	11.6	12.5
Rb9 cis	7,7,7,8,8,8,10,10,10,10,11,11,12,12,20	10.1	10.6
Rb9/Rb9	3,4,4,5,6,6,6,6,7,7,7,9,10,10,11,11,12,15	7.7	10.3

Table 3.1: Data from the Haigis and Dove study. Right-most columns are sample means and variances.

Another interesting feature is the close agreement of the sample means and variances for the Rb9 groups, yet not for the control group. This provides support for the hypothesis that presence of the Rb9 chromosome in Min mice may results in tumour counts consistent with a Poisson distribution. In such settings, the Poisson hypothesis has a natural interpretation. In particular, the intestinal tract (in both humans and mice) is formed of many organised groups of cells known as crypts. Since tumours first develop from aberrant crypts (Li *et al.*, 1994), a Poisson distribution for tumour multiplicities corresponds to each crypt being defective with a small probability, independent of all other crypts.

While this claim is briefly considered in the original paper, the interest into the biological significance of Poisson tumour counts indicates that further analysis may provide valuable insight into the area. The main aim of the work presented in this chapter is to understand the distribution of tumour multiplicities in each of the four groups of Haigis and Dove mice. In particular, we are interested in whether the tumour counts (see table 3.1) follow a Poisson distribution or if they exhibit extra-Poisson variation. In addition to this aim, we would like to make inference about the parameters of the underlying distributions in order to make formal conclusions about the relative frequencies of intestinal tumours in each of the four groups. In the following section we introduce a suitable statistical

model upon which to base this inference and discuss an appropriate Bayesian approach for proceeding.

3.2 A Bayesian Model Choice Approach

An alternative to the Poisson distribution for tumour multiplicities is to model the counts as arising from the two-parameter negative binomial distribution. Such a distribution captures extra-Poisson variation and has been suggested as being suitable for tumour count data (Drinkwater and Klotz, 1981). Additionally, the negative binomial distribution includes the Poisson distribution as a limiting case. We formalise this model as follows.

Let $i = 1, \ldots, 4$ denote the four groups +/+, Rb9 trans, Rb9 cis and Rb9/Rb9 of Haigis and Dove mice respectively. Denote the number of mice in group i as n_i. We model the number of tumours observed in mouse j of the i^{th} group as having a negative binomial distribution with parameters $\lambda_i > 0$ and $\kappa_i > 0$. Specifically, if Y_j^i is the number of tumours in mouse j in group i, then

$$\mathbb{P}(Y_j^i = y) = \begin{cases} C_i \frac{\lambda_i^y \Gamma(y+1/\kappa_i)}{\Gamma(y+1)(\lambda_i+1/\kappa_i)^{(y+1/\kappa_i)}} & y \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

for $i = 1, \ldots, 4$ and $j = 1, \ldots, n_i$. Here $C_i = \kappa_i^{-1/\kappa_i}/\Gamma(1/\kappa_i)$ is a normalising constant. As in the case of the Poisson(λ_i) distribution, λ_i is the expected tumour count for group i. The variance of this negative binomial distribution is given by $\lambda_i(1 + \lambda_i\kappa_i)$ and the parameter κ_i characterises the over-dispersion relative to the Poisson distribution for group i.

The Poisson distribution with parameter λ is the limit of the negative binomial

distribution with parameters λ and κ, as κ tends to 0. This is most easily motivated by noting that if $r \sim \text{Gamma}(1/\kappa, 1/\kappa)$ and $Y \sim \text{Poisson}(\lambda r)$, then Y has a negative binomial distribution with parameters λ and κ. It is clear then that as κ tends to 0, the Gamma distribution tends to the distribution with a point mass on 1.

A separate negative binomial distribution for the tumour counts in each of the four groups clearly provides a way of modelling the tumour count data collected by Haigis and Dove. However, this statistical model, with 8 free parameters, does not help address the question of interest, namely whether each of the groups exhibits Poisson tumour counts or not. In order to make inference in this direction we consider the possibility of allowing sub-models of this 8 parameter full model.

Using the notation above it is possible to place constraints on $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ and $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3, \kappa_4)$. Examples of such restrictions might be $\lambda_2 = \lambda_3$ corresponding to groups 2 and 3 having a common mean parameter, or $\kappa_3 = \kappa_4 = 0$ corresponding to groups 3 and 4 having tumour counts from a Poisson distribution. In total there are 15 such equality constraints for $\boldsymbol{\lambda}$, ranging from the simplest $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$ (the four groups have a shared mean parameter) to the most complicated where each group is allowed a different mean parameter. The possible equality structures of $\boldsymbol{\lambda}$ are tabulated in table 3.2 and each is denoted by the indexing parameter l.

Similar restrictions exist for $\boldsymbol{\kappa}$ with the addition that the restriction $\kappa_i = 0$ has the special meaning of tumour counts in group i following a Poisson distribution. This means that there are 52 allowable equality constraints of $\boldsymbol{\kappa}$ ranging from

	$\{\lambda_i\}$ structure				
Index l	$i=1$	$i=2$	$i=3$	$i=4$	Dimension $d_L(l)$
1	1	1	1	1	1
2	1	1	1	2	2
3	1	1	2	1	2
4	1	2	1	1	2
5	1	2	2	2	2
6	1	1	2	2	2
7	1	2	1	2	2
8	1	2	2	1	2
9	1	1	2	3	3
10	1	2	1	3	3
11	1	2	3	1	3
12	1	2	2	3	3
13	1	2	3	2	3
14	1	2	3	3	3
15	1	2	3	4	4

Table 3.2: Possible mean structures across the four groups. Equality of the entries within a row means equality of the mean parameters. The first column indexes the structure and the last column gives the number of free parameters.

$\kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = 0$ (the tumour multiplicities in each group are all from Poisson distributions) to the most complicated, where each group follows a negative binomial distribution with different dispersion parameters. We index the equality structures of $\boldsymbol{\kappa}$ by k but do not tabulate the 52 possibilities (choosing only to give explanation of the interesting equality structures in subsequent sections).

A sub-model is thus created by considering both a structure for $\boldsymbol{\lambda}$ and a structure for $\boldsymbol{\kappa}$. In particular, a model M is a pair of indices (l, k) denoting the structure of the mean and dispersion parameters for the 4 groups of mice. In total there are $15 \times 52 = 780$ such models.

Having constructed sub-models in this manner, the study of whether or not the counts in each group are Poisson becomes a simple model choice problem. Explicitly, we wish to make inference about the sub-models that might best explain the Haigis and Dove data. Analysis of this type can be performed using a variety of statistical methods. A range of classical methods, including hypothesis tests and classical model choice based on the Akaike information or Bayesian information criteria (AIC and BIC respectively) are presented in Newton and Hastie (2004). However, in this thesis we adopt a Bayesian approach and use reversible jump methods to make inference. The Bayesian framework provides a natural setting for making simultaneous conclusions about both the model M and the values of the underlying model parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$.

In order to apply the Bayesian paradigm, we denote the joint posterior distribution of the model M and parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$ by π. Then, by Bayes theorem we have,

$$\pi(M, \boldsymbol{\lambda}, \boldsymbol{\kappa}) \propto \mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa}) p(M, \boldsymbol{\lambda}, \boldsymbol{\kappa}).$$

Here $\mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$ is the likelihood of the data $\boldsymbol{y} = (y_1^1, \ldots, y_{n_1}^1, y_1^2, \ldots, y_{n_4}^4)$ given the model M and the model parameters. Under the framework presented above, this is given by $\mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa}) = \prod_{i=1}^4 L_i$ where

$$L_i = \begin{cases} \prod_{j=1}^{n_i} C_i \frac{\lambda_i^{y_j^i} \Gamma(y_j^i + 1/\kappa_i)}{\Gamma(y_j^i+1)(\lambda_i + 1/\kappa_i)^{(y_j^i + 1/\kappa_i)}} & \kappa_i > 0 \\ \prod_{j=1}^{n_i} \frac{\lambda_i^{y_j^i} e^{\lambda_i}}{y_j^i!} & \kappa_i = 0. \end{cases} \tag{3.2}$$

To analyse the posterior distribution it remains only to specify a suitable prior distribution $p(M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$ for the model and underlying parameters. We concentrate on specification of two alternative prior distributions which we call A and B.

Before observing the data we have very little prior information about either

the true sub-model or $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$. It is therefore appropriate that we adopt non-informative priors (i.e. priors that do not import strong beliefs about the model or underlying parameters). The two priors that we consider are non-informative in different ways that we discuss below. For convenience we impose independence between $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$ and assume that both priors satisfy the conditional decomposition $p(M, \boldsymbol{\lambda}, \boldsymbol{\kappa}) = p(\boldsymbol{\lambda}|M)p(\boldsymbol{\kappa}|M)p(M)$. The factors $p(\boldsymbol{\lambda}|M)$ and $p(\boldsymbol{\kappa}|M)$ are the same for both priors (see below), meaning that priors A and B differ only in the factor $p(M)$, the prior probability that the data comes from sub-model M.

A natural candidate for a vague distribution over the 780 sub-models M is to set $p(M) = 1/780\ \forall M$. We take this uniform distribution to be our prior A. An additional feature of this prior is that it is related to the BIC. (For an introduction to the BIC, see Schwarz, 1978). In particular, consider the posterior marginal probability of model M, given by $\pi(M|\boldsymbol{y}) \propto \mathcal{L}(\boldsymbol{y}|M)p(M)$. The factor $\mathcal{L}(\boldsymbol{y}|M)$ is the marginal likelihood of the data given model M, evaluated as $\mathcal{L}(\boldsymbol{y}|M) = \int\int \mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa})p(\boldsymbol{\lambda}|M)p(\boldsymbol{\kappa}|M)\mathrm{d}\boldsymbol{\lambda}\mathrm{d}\boldsymbol{\kappa}$. Model choice on the basis of the posterior distribution is equivalent to comparison of any two models by their posterior odds. For two models, M_0 and M, the posterior odds of M over M_0 are given by

$$\frac{\pi(M|\boldsymbol{y})}{\pi(M_0|\boldsymbol{y})} = \frac{\mathcal{L}(\boldsymbol{y}|M)}{\mathcal{L}(\boldsymbol{y}|M_0)} \times \frac{p(M)}{p(M_0)}. \tag{3.3}$$

The ratio of marginal likelihoods in equation 3.3 is the Bayes factor, $B(M, M_0)$, for model M over model M_0. By expanding the log-likelihood of the data for models M and M_0 in a Taylor series about the maximum likelihood estimates $\hat{\boldsymbol{\lambda}}_M$, $\hat{\boldsymbol{\kappa}}_M$ and $\hat{\boldsymbol{\lambda}}_{M_0}$, $\hat{\boldsymbol{\kappa}}_{M_0}$ respectively, it can be shown (see, for example, O'Hagan, 1994) that asymptotically as the number of data points increases, $-2\log B(M, M_0) = \mathrm{BIC}(M, M_0) + a$, where a is $O(1)$. If we ignore a, choosing

the model that minimizes the BIC with respect to some arbitrary reference model M_0 is equivalent to choosing the model that maximizes the (log) Bayes factor over model M_0. If, as is the case for prior A, $p(M) = p(M_0)$ for all M, equation 3.3 shows that choosing the model with the maximum Bayes factor over model M_0 is equivalent to choosing the model that maximizes the posterior odds with respect to model M_0.

Being uniform, the term $p(M)$ in prior A is non-informative in the sense of prior beliefs about which of the 780 models is the true model. However, we are aiming to make inference about whether or not the tumour counts in each group are Poisson and prior A is not non-informative in this sense. Specifically, under prior A, simple combinatorics demonstrate that for each of the 4 Haigis and Dove groups, $\mathbb{P}(\text{Group } i \text{ is Poisson}) = \mathbb{P}(\kappa_i = 0) = 15/52 \simeq 0.29$. This shows that prior A favours the negative binomial hypothesis for each group. The reason for this is simply because, for each group, the number of models corresponding to the negative binomial case is considerably greater than the number corresponding to the Poisson assumption.

Motivated by the previous paragraph, a natural choice for prior B is to choose $p(M)$ so that for each group i, independent of all other groups, there is an equal prior probability of tumor counts being distributed according to a Poisson or negative binomial distribution (i.e. $\mathbb{P}(\kappa_i = 0) = 1/2$). Conditional upon whether each of the 4 groups are Poisson or otherwise we assign an equal probability to each value of k that satisfies the selected Poisson / negative binomial categorization. For a configuration with 0 negative binomial groups there is 1 possible structure index k satisfying this configuration. Similarly, for each configuration with 1, 2, 3 or 4 negative binomial groups, there are

1, 2, 5, or 15 possible structure indices k respectively. Independent of k, the distribution of the index l is modelled as uniform among any of the 15 candidates.

Given the sub-model M, the terms $p(\boldsymbol{\lambda}|l)$ and $p(\boldsymbol{\kappa}|k)$, corresponding to the underlying parameter priors, are identical for priors A and B. In particular, for a model $M = (l, k)$ we can determine the number of free parameters $d_L(l)$ in $\boldsymbol{\lambda}$ and the number of free parameters $d_K(k)$ in $\boldsymbol{\kappa}$. Denoting the $d_L(l)$ unique values of the λ_i's by $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{d_L(l)}$ (such that $\min\{i : \lambda_i = \tilde{\lambda}_r\}$ is increasing in r) , we model $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{d_L(l)}$ as i.i.d. Gamma$(\alpha_\lambda, \beta_\lambda)$ random variables. Similarly, letting $\tilde{\kappa}_1, \ldots, \tilde{\kappa}_{d_K(k)}$ be the unique non-zero κ_i values (such that $\min\{i : \kappa_i = \tilde{\kappa}_r\}$ is increasing in r), we model these as independent draws from a Gamma$(\alpha_\kappa, \beta_\kappa)$ distribution. To ensure that these priors are vague we set the hyper-parameters to be $\alpha_\lambda = 2.0$, $\beta_\lambda = 0.1$, $\alpha_\kappa = 1.0$, and $\beta_\kappa = 2.0$.

Regardless of the choice of prior A or prior B, the resulting joint posterior distribution for M, $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$ is a non-standard distribution. Furthermore, sampling directly from such a distribution would prove a difficult task. However, being a simple model choice problem an obvious way to proceed is to use MCMC techniques. To make inference about the full joint posterior we require the MCMC sampler we design to jump between models. Since different models have different numbers of free parameters, it is necessary to apply a trans-dimensional approach. In keeping with the main theme of this thesis we choose to design and apply a reversible jump algorithm (Green, 1995). The design of this algorithm is discussed in detail in the following section.

3.3 A Reversible Jump MCMC sampler

In many applications of reversible jump, models are indexed by a single variable and the reversible jump step is used to move from one model to another. For the statistical model presented in the previous section, each model is indexed by M which is a pair (l, k). Implementing reversible jump for this problem, we choose to move between models by including reversible jump moves that update l (and $\boldsymbol{\lambda}$) while maintaining the current structure k (and $\boldsymbol{\kappa}$) and reversible jump moves that update k (and $\boldsymbol{\kappa}$) while keeping l (and $\boldsymbol{\lambda}$) fixed. In designing these between model moves we exploit a partial ordering of the models. Models are partially ordered in the index l given a fixed value of k and also in the index k for a fixed value l. We illustrate this partial nesting in l, assuming for the moment that the dispersion structure index k is fixed.

It is clear that different sub-models have different dimensions. For example, sub-model $M_1 = (2, k)$ has two free parameters in $\boldsymbol{\lambda}$ but sub-model $M_2 = (9, k)$ has three such free parameters regardless of the choice of k (see table 3.2). Moreover, we can consider model M_1 as being nested within model M_2, in the sense that model M_2 is the result of relaxing the requirement in M_1 that the tumour counts in mice in the Rb9 cis share a mean parameter with counts from mice in the +/+ and Rb9 trans group. The nesting is only partial because it is possible to choose 2 sub-models, M and M' say, such that M is not nested in M' and M' is not nested in M. As a specific example, if $M = (2, k)$ and $M' = (14, k)$, then since M' has fewer equality constraints (i.e. more free parameters) than M, M' cannot be nested in M. However, it is not possible to arrive at the mean equality restrictions of M' only by relaxing some of the constraints required by model M (since model M' requires the constraint $\lambda_3 = \lambda_4$

which is not imposed in model M). Thus model M is not nested in model M'. For a fixed value of k all models are nested in model $M = (15, k)$ and similarly model $M = (1, k)$ is nested within all other models.

By fixing the value of l, there is a partial ordering of sub-models in the index k in an analogous fashion, remembering that for a group i, the requirement $\kappa_i = 0$ is just another equality constraint. In this sense, model $M = (l, 1)$, where $\kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = 0$, is nested within all other models, whereas all sub-models are nested within model $M = (l, 52)$, where the dispersion parameter for each Haigis and Dove group is allowed to take a different non-zero value.

Before describing how we incorporate this partial nesting into our sampler design we define two functions which are needed to formally express between model moves. Let $v_L : \{1, \ldots, 15\} \times \{1, \ldots, 4\} \rightarrow \{1, \ldots, d_L(l)\}$ and $v_K : \{1, \ldots, 52\} \times \{1, \ldots, 4\} \rightarrow \{0, 1, \ldots, d_K(k)\}$. For a mean structure index l, the function $v_L(l, i)$, $i = 1, \ldots, 4$, is implicitly defined by the relation $\lambda_i = \tilde{\lambda}_{v_L(l,i)}$, $i = 1, \ldots, 4$. Given a dispersion equality structure k, for $i = 1, \ldots, 4$, if $\kappa_i > 0$, $\kappa_i = \tilde{\kappa}_{v_K(k,i)}$ otherwise $v_K(k, i) = 0$. With this notation in place we are able to explicitly detail the reversible jump move types.

Consider first the move type that updates the model M by altering the mean structure l (updating $\boldsymbol{\lambda}$ in the process). We emphasise that for this move type the dispersion structure index k and dispersion vector $\boldsymbol{\kappa}$ remain unchanged. Suppose the current state of the Markov chain is $\boldsymbol{x} = (M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$, where $M = (l, k)$ is the current model. The first stage in this move type is to propose a new mean structure index l'. With probability $B_L(l)$ we propose a move to a structure l' such that $d_L(l') = d_L(l) + 1$ (a birth move). Alternatively, with probability $D_L(l)$

we propose a move to a structure l' such that $d_L(l') = d_L(l) - 1$ (a death move). The dimension of the model is never altered by more than 1 in a single move. We set $B_L(0) = 1,\ D_L(0) = 0$ and $B_L(15) = 0,\ D_L(15) = 1$ and for all other values of l we take $B_L(l) = D_L(l) = \frac{1}{2}$.

Suppose now that a birth is chosen. At this stage we take advantage of the partial nesting of l given k. We choose the particular value of l' uniformly from the $r_L(l, d_L(l'))$ mean structures that have $d_L(l')$ free parameters and satisfy the property that sub-model $M = (l, k)$ is nested within $M' = (l', k)$. For each value of l, table 3.3 tabulates the allowable candidates l' for a birth move.

With the proposed model $M' = (l', k)$ in place, we propose new parameter values $\boldsymbol{\lambda}'$. Importantly, we note we are actually proposing a move from the $d_L(l)$ unique parameter values $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{d_L(l)}$ to the $d_L(l')$ unique parameters $\tilde{\lambda}'_1, \ldots, \tilde{\lambda}'_{d_L(l')}$. This is achieved by generating a variable u from a LogNormal$(0, \sigma_\lambda)$ distribution and then setting

$$\tilde{\lambda}'_{v_L(l',i_1)} = \tilde{\lambda}_{v_L(l,i_1)} u, \quad \tilde{\lambda}'_{v_L(l',i_2)} = \frac{\tilde{\lambda}_{v_L(l,i_1)}}{u} \quad \text{and } \tilde{\lambda}'_{v_L(l',j)} = \tilde{\lambda}_{v_L(l,j)} \text{ for } j \notin \{i_1, i_2\}.$$

The values of indices i_1 and i_2 are formally defined to be

$$i_1 = \min\{i : \exists j > i \text{ s.t. } \lambda_i = \lambda_j \text{ and } \lambda'_i \neq \lambda'_j\}$$

and

$$i_2 = \min\{j > i_1 : \lambda_{i_1} = \lambda_j \text{ and } \lambda'_{i_1} \neq \lambda'_j\}$$

but in practice are obvious from the current and proposed mean structures l and l' as shown in the example below. The proposed parameter vector $\boldsymbol{\lambda}'$ is immediate from $\tilde{\lambda}'_1 \ldots, \tilde{\lambda}'_{d_L(l')}$ and our proposed new state is then $\boldsymbol{x}' = (M', \boldsymbol{\lambda}', \boldsymbol{\kappa})$, where $M' = (l', k)$.

Current mean structure l	Permitted birth proposals l'	Permitted death proposals l'	Permitted switch proposals l'
1	2, 3, 4, 5, 6, 7, 8	-	-
2	9, 10, 12	1	3, 4, 5, 6 ,7 ,8
3	9, 11, 13	1	2, 4, 5, 6 ,7 ,8
4	10, 11, 14	1	2, 3, 5, 6 ,7 ,8
5	12, 13, 14	1	2, 3, 4, 6 ,7 ,8
6	9, 14	1	2, 3, 4, 5 ,7 ,8
7	10, 13	1	2, 3, 4, 5 ,6 ,8
8	11, 12	1	2, 3, 4, 5 ,6 ,7
9	15	2, 3, 6	10, 11, 12, 13, 14
10	15	2, 4, 7	9, 11, 12, 13, 14
11	15	3, 4, 8	9, 10, 12, 13, 14
12	15	2, 5, 8	9, 10, 11, 13, 14
13	15	3, 5, 7	9, 10, 11, 12, 14
14	15	4, 5, 6	9, 10, 11, 12, 13
15	-	9, 10, 11, 12, 13, 14	-

Table 3.3: The values of the mean structure index l' that can be proposed when the current mean structure index is l, for the birth, death and switch moves. (Mean structure indices as in table 3.2).

To clarify this birth move, consider the example of moving from model $M = (l, k)$ with $l = 8$ to model $M = (l', k)$ with $l' = 12$. This is a birth move since $d_L(l) = 2$ and $d_L(l') = 3$ and is allowed because $l' = 12$ is tabulated as an allowable candidate for $l = 8$ (see table 3.3) since $M = (8, k)$ is nested within $M = (12, k)$. Suppose $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_4)$. From table 3.2 we can see that $v_L(8, 1) = v_L(8, 4) = 1$ and $v_L(8, 2) = v_L(8, 3) = 2$ so that the 2 unique parameters are $\tilde{\lambda}_1 = \lambda_1 = \lambda_4$ and $\tilde{\lambda}_2 = \lambda_2 = \lambda_3$. For the proposed mean structure $l' = 12$, table 3.2 shows that $v_L(12, 1) = 1$, $v_L(12, 2) = v_L(12, 3) = 2$ and $v_L(12, 4) = 3$. Here $i_1 = 1$ and $i_2 = 4$, which means we propose a move to $\tilde{\lambda}'_1, \ldots, \tilde{\lambda}'_3$, by generating a LogNormal$(0, \sigma_\lambda)$ and setting $\tilde{\lambda}'_1 = \tilde{\lambda}_1 u$, $\tilde{\lambda}'_2 = \tilde{\lambda}_2$ and $\tilde{\lambda}'_3 = \frac{\tilde{\lambda}_1}{u}$. Finally this results in $\boldsymbol{\lambda}' = (\lambda'_1, \ldots, \lambda'_4) = (\tilde{\lambda}'_1, \tilde{\lambda}'_2, \tilde{\lambda}'_2, \tilde{\lambda}'_3)$.

Before considering the acceptance probability for this birth move it is important to consider the reverse death move. For convenience, suppose now that the current state of the Markov chain is $\boldsymbol{x}' = (M', \boldsymbol{\lambda}', \boldsymbol{\kappa})$, where the number of free parameters in l' is $d_L(l')$. As noted above, a death move is selected with probability $D_L(l')$. A new mean structure l is chosen at random from the $r_L(l', d_L(l))$ possible candidates (see table 3.3). The new mean parameter $\boldsymbol{\lambda}$ is given by

$$\tilde{\lambda}_{v_L(l,i_1)} = \sqrt{\tilde{\lambda}'_{v_L(l',i_1)}\tilde{\lambda}'_{v_L(l',i_2)}}, \quad \tilde{\lambda}_{v_L(l,j)} = \tilde{\lambda}'_{v_L(l',j)} \text{ for } j \notin \{i_1, i_2\},$$

and the dummy random variable u is given by

$$u = \sqrt{\frac{\tilde{\lambda}'_{v_L(l',i_1)}}{\tilde{\lambda}'_{v_L(l',i_2)}}},$$

where i_1 and i_2 are as above. The proposed new model is then $\boldsymbol{x} = (M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$, where $M = (l, k)$.

Following the methods reviewed in chapter 2 it is easily shown that the acceptance probabilities for the birth move and reverse death moves are given by $\min\{1, A_{BD,L}(\boldsymbol{x}, \boldsymbol{x}')\}$ and $\min\{1, A_{BD,L}(\boldsymbol{x}', \boldsymbol{x})\}$ respectively, where

$$A_{BD,L}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\mathcal{L}(\boldsymbol{y}|M', \boldsymbol{\lambda}', \boldsymbol{\kappa})p(M', \boldsymbol{\lambda}', \boldsymbol{\kappa})}{\mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa})p(M, \boldsymbol{\lambda}, \boldsymbol{\kappa})} \times \frac{D_L(l')}{B_L(l)} \frac{r_L(l, d_L(l'))}{r_L(l', d_L(l))} \frac{1}{q_{\sigma_\lambda}(u)} \times |J|, \tag{3.4}$$

and $A_{BD,L}(\boldsymbol{x}', \boldsymbol{x}) = [A_{BD,L}(\boldsymbol{x}, \boldsymbol{x}')]^{-1}$. The first factor contains the ratio of posteriors, comprising the ratio of likelihoods $\mathcal{L}$ given by equation 3.2 and the ratio of priors which depends upon whether prior A or prior B is used. The second factor is the ratio of the proposal distributions, where q_{σ_λ} is the probability density function of the Lognormal$(0, \sigma_\lambda)$ distribution. The final factor is the absolute

value of the Jacobian J, where J simplifies to

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial \tilde{\lambda}'_{v_L(l',i_1)}}{\partial \tilde{\lambda}_{v_L(l,i_1)}} & \frac{\partial \tilde{\lambda}'_{v_L(l',i_2)}}{\partial \tilde{\lambda}_{v_L(l,i_1)}} \\ \frac{\partial \tilde{\lambda}'_{v_L(l',i_1)}}{\partial u} & \frac{\partial \tilde{\lambda}'_{v_L(l',i_2)}}{\partial u} \end{vmatrix} \\ &= \begin{vmatrix} u & \frac{1}{u} \\ \tilde{\lambda}_{v_L(l,i_1)} & -\frac{\tilde{\lambda}_{v_L(l,i_1)}}{u^2} \end{vmatrix} \\ &= -\frac{2\lambda_{i_1}}{u}. \end{aligned}$$

Having discussed in detail the birth/death move type to update the model through the mean structure index l while keeping the dispersion index k fixed, we briefly mention the equivalent move type for updating k for a given value of l. The move proceeds in exactly the same fashion as above, first choosing a birth move with probability $B_K(k)$ or death move with probability $D_K(k)$. As with the update of l, $B_K(k) = D_K(k) = \frac{1}{2}$ for all dispersion structures k, except the minimal and maximal values $k = 1$ and $k = 52$, where $B_K(1) = 1$, $D_K(1) = 0$, $B_K(52) = 0$, and $D_K(52) = 1$. The next stage is to choose a candidate model k' at random from the permissable candidates (which as for the update of l depend on the choice of a birth or death move and which models nest within each other).

The one slight difference between the update of k and that of l, is how the underlying model parameters are updated. Suppose firstly that the current state is $\boldsymbol{x} = (M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$ where $M = (l, k)$. Consider first a birth move to a new model M' with the same mean structure but dispersion structure k', where $d_K(k') = d_K(k) + 1$. We proceed by first trying to find indices i_1 and i_2 defined as above (with κ replacing λ). If such indices can be found and $\kappa_{i_1} > 0$ (i.e. tumour counts in group i_1 follow a negative binomial distribution under the current model), then we proceed exactly as above, with the same transformation, resulting in

an entirely analogous acceptance probability. However, if we are trying to move from a model where one of the groups is Poisson, to another model where the same group is negative binomial, it is possible that no such i_1 exists, or if it does, that $\kappa_{i_1} = 0$. In such cases, we proceed in a different way. Firstly, we define an integer $i_3 = \min\{i : \kappa_i = 0 \text{ and } \kappa'_i \neq 0\}$. Having found such an integer, we generate a Lognormal$(0, \sigma_\kappa)$ random variable u and define the new unique parameters $\tilde{\kappa}'_1, \ldots, \tilde{\kappa}'_{d_K(k')}$ as

$$\tilde{\kappa}'_{v_K(k',i_3)} = u \text{ and } \tilde{\kappa}'_{v_K(k',j)} = \tilde{\kappa}_{v_K(k,j)} \text{ for } j \neq i_3 \text{ and } v_K(k,j) \neq 0.$$

We give an example of this case with a birth move from model $M = (l, 14)$, (where $\kappa_1 = \kappa_2 = \kappa_4 > 0$ and $\kappa_3 = 0$) to model $M' = (l, 36)$ (where $\kappa_1 = \kappa_2 = \kappa_4 > 0$). This is a birth move since $d_K(14) = 1$ and $d_K(36) = 2$ and is permitted since model M is nested within model M'. Also for model $M = (l, 14)$, $v_K(14, 1) = v_K(14, 2) = v_K(14, 4) = 1$ and $v_K(14, 3) = 0$. For model $M = (l, 36)$, $v_K(36, 1) = v_K(36, 2) = v_K(36, 4) = 1$ and $v_K(36, 3) = 2$. Since it is not possible to find i_1 and i_2 as defined above, we are in this alternative case. It is easy seen that $i_3 = 3$. We thus generate a LogNormal$(0, \sigma_\kappa)$ random variable u, and set $\tilde{\kappa}'_1 = \tilde{\kappa}_1$ and $\tilde{\kappa}'_2 = u$. The proposed new dispersion vector $\boldsymbol{\kappa}$ is then given by $\boldsymbol{\kappa} = (\tilde{\kappa}_1, \tilde{\kappa}_1, \tilde{\kappa}_2, \tilde{\kappa}_1)$.

The reverse death move for the dispersion structure is obvious from the above discussion. In particular, if the current state is $\boldsymbol{x}' = (M', \boldsymbol{\lambda}, \boldsymbol{\kappa}')$, having chosen a death move with appropriate probability and chosen a proposed new dispersion structure k uniformly from the allowable candidates, we try to identify indices i_1 and i_2. If such indices can be found, the move progresses in the same manner as the death move for the mean structure l (described above). Otherwise, we find an index i_3, and then set our $d_K(k)$ proposed new unique model parameters to

be

$$\tilde{\kappa}_{v_K(k,j)} = \tilde{\kappa}_{v_K(k',j)} \text{ for } j \neq i_3 \text{ and } v_K(k,j) \neq 0,$$

and the dummy variable u to be

$$u = \tilde{\kappa}_{v_K(k',i_3)}.$$

The acceptance probability for the birth move is given by $\min\{1, A_{BD,K}(\boldsymbol{x}, \boldsymbol{x}')\}$ and for the death move is given by $\min\{1, A_{BD,K}(\boldsymbol{x}', \boldsymbol{x})\}$, where $A_{BD,K}(\boldsymbol{x}', \boldsymbol{x}) = [A_{BD,K}(\boldsymbol{x}, \boldsymbol{x}')]^{-1}$. The value of $A_{BD,K}(\boldsymbol{x}, \boldsymbol{x}')$ is as in equation 3.4, with the constants and functions for l replaced with the equivalent constants and functions for k. Importantly, if the birth/death step for k was of the type where indices i_1 and i_2 were found, the Jacobian is analogous to that for the birth/death set for l. However, if no such i_1 and i_2 existed and we proceeded using index i_3, it is easy to show that the Jacobian factor $|J|$ is equal to one.

In addition to the birth/death moves described in detail above, the sampler we design includes two other move types at each sweep. In the remainder of this section we mention these briefly, omitting full details. We begin by looking at a second type of move that alters the current model M, again by changing either the mean structure l (while keeping k fixed) or the dispersion structure k (while keeping l fixed). We call this move a switch move. At each sweep, an update of both l and k is attempted using this move. Although the purpose of the move is to try to move to a different model with the same dimension, we retain the reversible jump framework. We describe the move for updating l (for a fixed k) but the equivalent move for updating k is identical.

Once again we suppose that the current state of the chain is $\boldsymbol{x} = (M, \boldsymbol{\lambda}, \boldsymbol{\kappa})$, where $M = (l, k)$. The first stage in this move is to choose an alternative mean

structure l', such that $d_L(l) = d_L(l')$, uniformly from the $r_L(l, d_L(l))$ possibilities (see table 3.3). If no such model exists, the move is not attempted. Having proposed a sub-model $M' = (l', k)$, a new parameter vector $\boldsymbol{\lambda}'$ is proposed as a replacement for $\boldsymbol{\lambda}$ by setting $\tilde{\lambda}'_i = u_i$ where the independent random variables u_i are such that $u_i e^{-\mu_\lambda} \sim \text{LogNormal}(0, \sigma_\lambda)$, $i = 1, \ldots, d_L(l)$, for some constant μ_λ.

It is easily shown that the Jacobian factor equals 1 and thus the resulting acceptance probability is given by $\min\{1, A_{SW,L}(\boldsymbol{x}, \boldsymbol{x}')\}$ where

$$A_{SW,L}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\mathcal{L}(\boldsymbol{y}|M', \boldsymbol{\lambda}', \boldsymbol{\kappa})p(M', \boldsymbol{\lambda}', \boldsymbol{\kappa})}{\mathcal{L}(\boldsymbol{y}|M, \boldsymbol{\lambda}, \boldsymbol{\kappa})p(M, \boldsymbol{\lambda}, \boldsymbol{\kappa})} \times \frac{\prod_{i=1}^{d_L(l)} q_{\mu_\lambda, \sigma_\lambda}(\tilde{\lambda}_i)}{\prod_{i=1}^{d_L(l)} q_{\mu_\lambda, \sigma_\lambda}(u_i)}.$$

The final move type attempted at each sweep attempts to update the model parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\kappa}$ without changing the model M. This is done by componentwise updates for each unique parameter using simple Metropolis-Hastings steps (see for example chapter 2 or Robert and Casella, 2002). For the update of the i^{th} unique parameter value, the proposed new value $\tilde{\lambda}'_i$ (or $\tilde{\kappa}'_i$) is the product of $\tilde{\lambda}_i$ (or $\tilde{\kappa}_i$) and u_i where $u_i \sim \text{LogNormal}(0, \sigma_\lambda)$ (or $u_i \sim \text{LogNormal}(0, \sigma_\kappa)$). The acceptance probability is then easily derived by applying the standard Metropolis-Hastings expression.

Including several different types of move at each sweep of a sampler is standard practice. One benefit of doing so is that if (as is normally the case) the chain is only sampled (at most) at the end of each sweep, including many several move types can help to ensure the necessary irreducibility of the Markov chain. For this particular problem, irreducibility is ensured by the inclusion of the birth/death move by using the partial nesting of the sub-models M. The other

move types, which, if used alone, would not result in an irreducible sampler, are included with the aim of improving mixing of the sampler.

3.4 Numerical Results

For both priors we run our MCMC sampler for 1,100,000 sweeps and discard the first 100,000 as burn-in. A run of our sampler with these settings takes approximately 12 minutes when prior A is used and 11 minutes when prior B is used.[1] To reduce storage requirements, we sub-sample every 10 observations leaving 100,000 observations. Our sampler typically gives trans-dimensional acceptance rates of between 6% and 12% for prior A and 5% and 8% for prior B, depending on the move type. In both cases fixed dimensional acceptance rates vary between 10% and 47%. Although some tuning has been carried out, the sampler has not been fully optimised and higher acceptance rates could no doubt be achieved with increased effort. However, we believe that with the current parameter settings the mixing is adequate and focusing extra effort in this direction is not warranted. The adequate performance of the sampler for the Haigis and Dove data is supported by the trace plots in Figure 3.1. In order to confirm that the chains had converged to the stationary distribution we ran several MCMC chains started from random initial states. The resulting numerical results were consistent across these runs, suggesting that the Markov chains had converged. Reported posterior probabilities are averages across 2 independent MCMC runs. In all instances we estimate the Monte Carlo standard error (using blocking, see Green, 1999) to be less than 0.005.

[1] Our sampler was written in C, and compiled using the gcc GNU compiler with optimisation level 3. The sampler was run on a machine with a 1600MHz Intel Pentium M processor

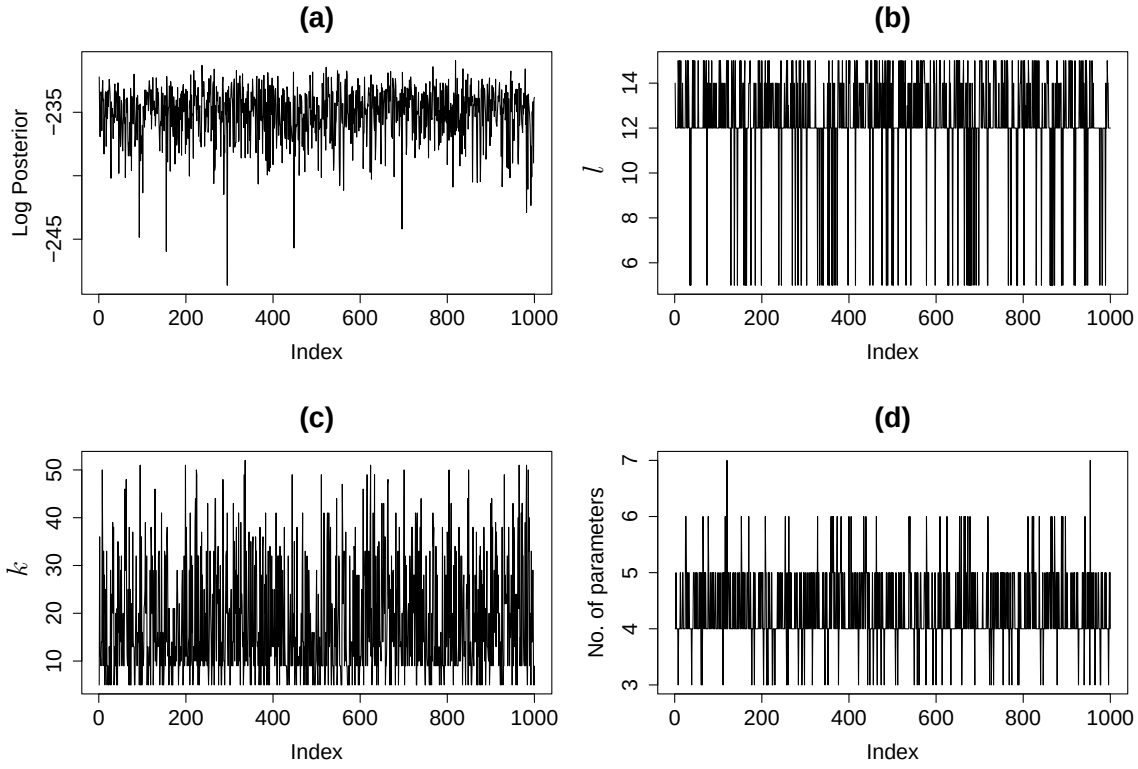

Figure 3.1: Trace plots from MCMC runs (chains thinned to 1000 observations for clarity): (a) log-posterior value (prior A); (b) mean structure index l (prior B); (c) dispersion structure index k (prior A); and (d) number of free parameters, $d_L(l) + d_K(k)$ (prior B).

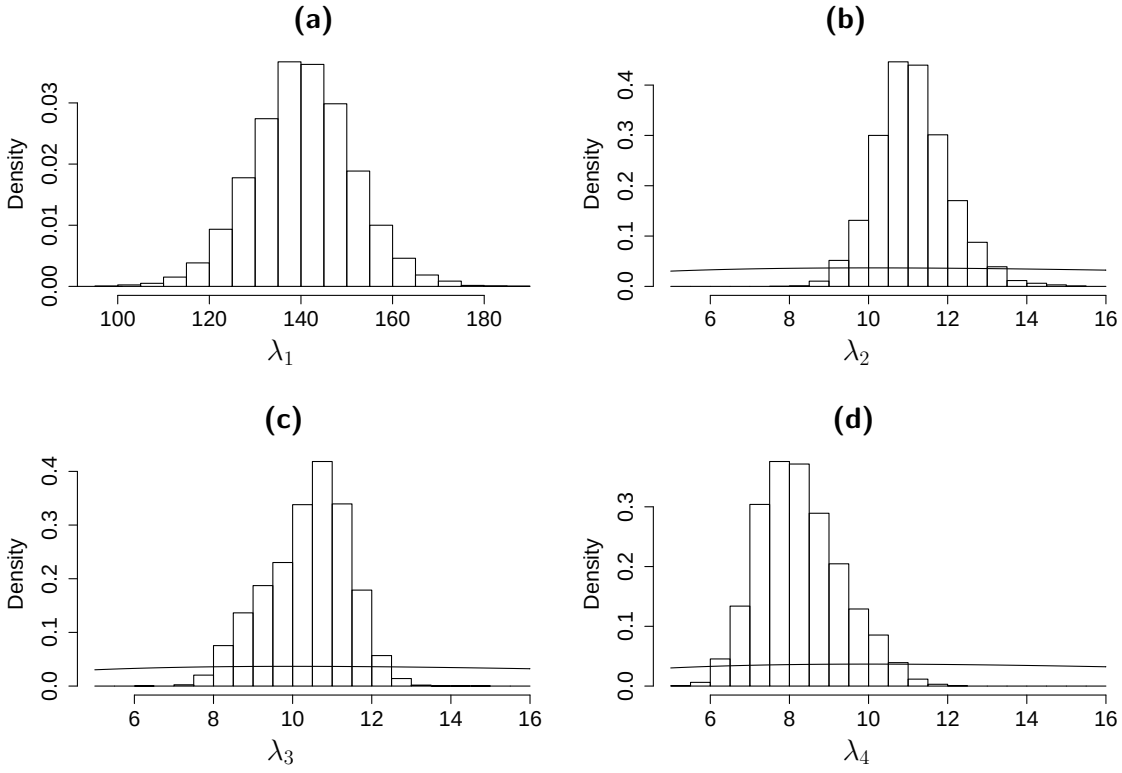

Figure 3.2: Histograms of marginal posterior samples of the components of $\boldsymbol{\lambda}$ (prior A, solid lines denote the prior): (a) +/+; (b) Rb9 trans; (c) Rb9 cis; and (d) Rb9/Rb9.

The output from the MCMC allows us easy access to marginal properties of the posterior distribution. Figures 3.2 and 3.3 show histograms of posterior samples for the group-specific parameters λ_i (using prior A) and κ_i (using prior B) averaged across all models M. The main features of the histograms are not sensitive to choice of prior. The solid lines are the prior densities which can be seen to be non-informative for the means, but carry slightly more information for the dispersion parameters. The mean for the marginal distribution of $\boldsymbol{\lambda}$ is (140.3,11.1,10.3,8.3) using prior A and (139.9,11.1,10.4,8.2) when prior B is used. As we expect from the data (table 3.1), in both cases the marginal posterior mean for the +/+ group is significantly higher than the other groups. Moreover, figure 3.2 shows that the marginal distribution for λ_1 does not even overlap the distributions for the other components.

The marginal posterior distribution for $\boldsymbol{\kappa}$ has mean (0.10,0.02,0.03,0.06) with prior A and (0.10,0.01,0.01,0.04) using prior B. Figure 3.3 shows that the marginal distribution for the dispersion parameter for the control group (κ_1) has nearly all weight on $\kappa_1 > 0.0025$. The distributions for the Rb9 groups do not share this feature. If the true tumour occurrence rate for the +/+ group was the posterior mean of 140, a dispersion parameter of 0.1 (the marginal posterior mean), would increase the variance by a factor of 15 when compared to Poisson variance.

While making inference about the rate parameters provides us with some insight into the problem, of greater interest for this study are the underlying models. Tabulated in table 3.4 are the (non-zero) marginal posterior probabilities of the mean structure index l. The modal pattern $l = 12$, which accounts for over half the posterior mass under either prior, corresponds to there being 3 different rates

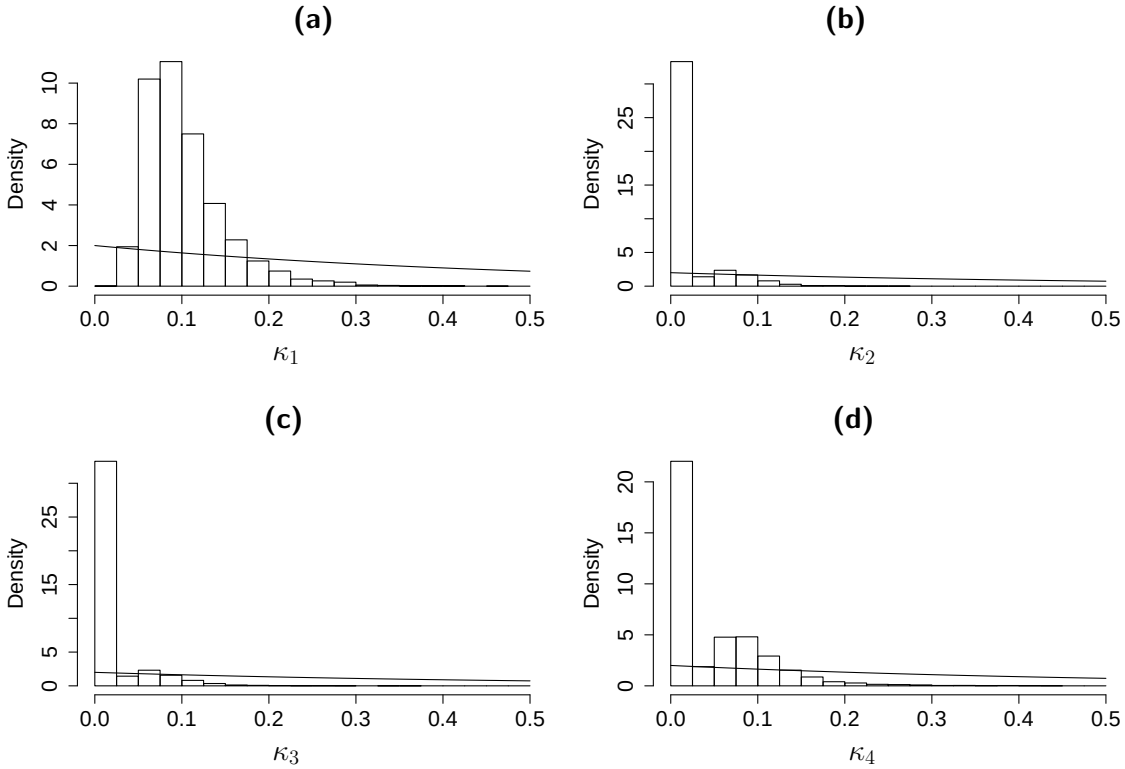

Figure 3.3: Histograms of marginal posterior samples of the components of $\boldsymbol{\kappa}$ (prior B, solid lines denote the prior): (a) +/+; (b) Rb9 trans; (c) Rb9 cis; and (d) Rb9/Rb9.

Index		$\{\lambda_i\}$ structure				Posterior probability	
l	$i=$	1	2	3	4	prior A	prior B
5		1	2	2	2	0.09	0.07
12		1	2	2	3	0.57	0.59
13		1	2	3	2	0.01	0.01
14		1	2	3	3	0.19	0.18
15		1	2	3	4	0.15	0.15

Table 3.4: Mean structures with non-zero marginal posterior probabilities.

of tumour occurrence, one each for the +/+ and Rb9/Rb9 groups, and a shared rate for the Rb9 trans and Rb9 cis groups. There is no posterior support for any of the values of l in which the control group +/+ shares a mean parameter with any of the other groups. This is consistent with the marginal posterior distributions in figure 3.2, indicating the strong effect of the Rb9 translocation on expected tumour count.

Analysing the marginal distributions of the dispersion structure k in the same way, two structures contain most of the posterior mass under both priors. They are $k = 5$, in which only the +/+ group has extra-Poisson variation (i.e. $\kappa_1 > 0$, $\kappa_2 = \kappa_3 = \kappa_4 = 0$) and $k = 9$, where the two homozygous groups +/+ and Rb9/Rb9 share a non-zero dispersion parameter and the other groups are Poisson (i.e. $\kappa_1 = \kappa_4 > 0$, $\kappa_2 = \kappa_3 = 0$). Other dispersion structures are considerably less probable, although there is more mass outside the modal two values using prior A than using prior B.

Although it is instructive to look at marginal posterior probabilities for l and k independently, we are more interested in making inference about likely sub-models M, (i.e. l, k pairs). Table 3.5 shows 11 sub-models $M = (l, k)$ which include the top 6 sub-models for each prior, ranked by posterior probability. For both priors, the top two sub-models, $M = (12, 5)$ and $M = (12, 9)$ are the same, although for other models the ranking is dependent upon the prior used. These top two sub-models account for 18% of the total posterior mass under the uniform prior A and 34% under prior B.

As motivated in section 3.1 the main interest for the Haigis and Dove data set is the posterior probability of group i being Poisson, i.e. $\pi(\kappa_i = 0|\boldsymbol{y})$ for each

Sub-model index		$\{\lambda_i\}$ structure					$\{\kappa_i\}$ structure				Posterior probability (ranking)	
$M=(l,k)$	$i=$	1	2	3	4	$i=$	1	2	3	4	prior A	prior B
(12 , 5)		1	2	2	3		1	0	0	0	0.09 (1)	0.23 (1)
(12 , 9)		1	2	2	3		1	0	0	1	0.09 (2)	0.11 (2)
(14 , 9)		1	2	3	3		1	0	0	1	0.04 (3)	0.05 (5)
(15 , 5)		1	2	3	4		1	0	0	0	0.03 (8)	0.06 (3)
(12 , 20)		1	2	2	3		1	0	0	2	0.04 (5)	0.04 (6)
(14 , 5)		1	2	3	3		1	0	0	0	0.02 (11)	0.05 (4)
(12 , 13)		1	2	2	3		1	0	1	1	0.04 (4)	0.02 (12)
(15 , 9)		1	2	3	4		1	0	0	1	0.03 (9)	0.03 (9)
(12 , 14)		1	2	2	3		1	1	0	1	0.03 (6)	0.02 (15)
(15 , 16)		1	2	3	4		1	1	1	1	0.00 (65)	0.00 (90)
(15 , 52)		1	2	3	4		1	2	3	4	0.00 (162)	0.00 (-)

Table 3.5: Posterior probabilities and rankings for 11 sub-models containing the 6 most highly ranked sub-models for both priors.

Posterior	$\mathrm{BF}(\kappa_i = 0, \kappa_i > 0)$			
type	+/+	Rb9 trans	Rb9 cis	Rb9/Rb9
Prior A	0.00	3.86	3.70	1.16
Prior B	0.00	4.00	4.00	1.13

Table 3.6: Bayes factors for $\kappa_i = 0$ vs $\kappa_i > 0$, for each group. Bayes factors are calculated under prior A that has $p(\kappa_i = 0) = 15/52$ and prior B that has $p(\kappa_i = 0) = 1/2$.

$i = 1, \ldots, 4$. To remove the effect of the prior, we calculate the Bayes factors for $\kappa_i = 0$ versus $\kappa_i > 0$, for $i = 1, \ldots, 4$. The estimates of the Bayes factors under both priors are recorded in table 3.6. From this table we can see that the data provides positive evidence that tumour counts follow a Poisson distribution for the two heterozygous Rb9 groups, Rb9 trans and Rb9 cis. It is also obvious that there is no support for Poisson counts for the control group. The question of whether tumour counts are Poisson for the Rb9/Rb9 group is less clear although the data seems to be slightly in favour of the Poisson hypothesis.

3.5 Conclusions

Although much is known about cancer biology, relatively little is understood about the early stages in tumour formation. For intestinal cancer, it has been established that loss of functionality of the APC (Apc) gene is a necessary preliminary event. This study, based on data collected from the Haigis and Dove experiment, involved mice for which the occurrence of an Rb9 translocation prevents one mechanism for Apc inactivation, meaning that Apc failure must occur by some other method. The reduced tumour counts evident in the Rb9 mice provide strong evidence of this phenomenon.

While the original study (Haigis and Dove, 2003) highlighted the above effect of the Rb9 translocation on tumour counts, another biologically significant scenario was observed: the Rb9 translocation appears to result in tumour counts that can be well modelled by a Poisson distribution. Historical data has indicated that tumour counts can often be better modelled by a negative binomial distribution which accounts for the extra variation often recorded in laboratory experiments. Nonetheless, the Poisson distribution represents the ideal biological reference point corresponding to tumours forming independently of one another. Finding the biological conditions that appear to lead to Poisson tumour counts has therefore been a subject of interest to scientists and will help to motivate research into the biological factors that contribute to non-Poisson counts. Due to the interest into the Rb9 Poisson claim and the fact that previous biological studies have not achieved Poisson tumour counts, we have used a Bayesian framework and MCMC to statistically assess the assertion.

As demonstrated from the results in the previous section, we have found and

quantified positive evidence for Poisson tumour counts in mice with a single Rb9 translocation. Our conclusions will therefore hopefully be of use to scientists to investigate what specifically it is about the Rb9 translocation that reduces the variation in observed tumour counts.

Although here we adopt the Bayesian paradigm to analyse the problem, a variety of other statistical tools and methods have also been employed, the details and results of which can be found in Newton and Hastie (2004). In addition to other more classical techniques, this paper uses posterior predictives (Gelman *et al.*, 2003) to demonstrate the validity of the negative binomial model upon which the Bayesian analysis rests. As demonstrated above, and concluded by Newton and Hastie, the Bayesian formulation provides the most natural framework to make rigorous inference about the problem of interest. In particular, inference about the sub-model and underlying model parameters can be carried out simultaneously, avoiding potential problems inherent in other methods.

In the context of this thesis, the biological study provides a real example to demonstrate the process of reversible jump sampler design and implementation. While the design of a reversible jump sampler for this particular problem was not unduly difficult, it is hoped that the section highlights the involved nature of sampler design, perhaps motivating the reluctance of non-specialists to adopt RJMCMC. In this case, little sampler tuning was required to obtain adequate performance. Often, however, this is not the case and this tuning process is a further hinderance to the popularity of the reversible jump algorithm. As such it is hoped that this chapter provides motivation to look at steps in which we can make the process of sampler design more automatic and less of a specialist statistical topic.

The remainder of this thesis looks further at these ideas, attempting to work towards an automatic RJMCMC sampler.

Chapter 4

Adaptive Sampling Methods

In this chapter we explore the area of adaptive MCMC samplers. We report on previous research, looking at the recent advances in the literature, concerning both the theoretical properties of such samplers and implementation issues. The chapter then details some of our own research in this area, including our efforts at designing adaptive algorithms specifically for RJMCMC problems. The algorithm that we propose at the end of this chapter is adopted in chapter 5 where we introduce our automatic reversible jump sampler.

4.1 Introduction

The quest for automatic MCMC samplers has received a large boost from the recent surge in research and literature on the subject of adaptive samplers. As a considerably simplified summary, an adaptive sampler does exactly what we might expect from its name, it adapts how it samples from the target distribution during each MCMC run. This adaptation is based on the samples that it has already drawn up to the current point.

Before the advent of adaptive samplers, an integral feature in MCMC sampler

design was an initial tuning process to ensure the efficiency of the resulting sampler. This process necessitated running the MCMC algorithm multiple times (such runs being known as pilot runs), each time using different values for the underlying parameters of the various proposal distributions. The final MCMC algorithm would then use the parameters identified as best achieving some measure of performance.

As tuning is often a sensitive, involved and expensive process, this means that specialist experience is necessitated every time an MCMC sampler is implemented. For an automatic sampler, where the goal is to minimise the need for specialist MCMC knowledge, this is clearly not desirable. As such, the idea of an adaptive sampler that learns from its own experience is certainly an attractive one. For such a sampler, the need for pilot runs is reduced and in some cases eliminated. This offers the prospect of a wider, non-specialist, user base.

Despite the appeal of adaptive samplers, adapting the proposal distribution by using past realisations of the chain means that the chain is typically no longer Markovian. Although it is possible to augment the chain to restore the Markov property, the augmented chain is often no longer time homogenous. Questions regarding the ergodic properties of the resulting chain and the validity of using such a chain to compute Monte Carlo estimates of target integrals are no longer covered by the standard arguments. Until recently this problem had prevented the adaptive sampler from being anything more than an attractive concept. However, over the last few years research into the area has grown considerably and adaptive samplers have become a reality.

One approach to adaptive samplers that side steps the difficulty of establishing ergodicity has been to allow a sampler to have an initial adaptive phase. Once the adaptation had been performed to a suitable degree, adaptation could stop and the adapted transition kernel would be used. The initial sample would be discarded and only the ergodic chain resulting from this unchanging adapted kernel would be used in the Monte Carlo estimates. A recent example of such a sample is provided by Pasarica and Gelman (2004).

Whilst such partially adaptive samplers seem to automatically tune themselves, the important question arises of how to decide when to stop the adaptive phase. A sampler that has not sufficiently adapted to the target distribution would perform as inefficiently as a standard MCMC sampler that had not been tuned. On the other hand, requiring some kind of convergence of the adaptation process might result in a sampler that takes a prohibitively long time to run. In many cases, any solution to this problem seems arbitrary and contradicts the underlying philosophy of an adaptive sampler.

Fortunately, recent research has been concerned with fully adaptive algorithms that are allowed to adapt for the duration of the MCMC run. For such algorithms, two distinct areas have received considerable attention. Firstly, research has been concentrated on ensuring the convergence of the parameters being adapted (we refer to this simply as the convergence of the algorithm throughout this chapter). Secondly, work has been done to establish the ergodicity properties of the resulting chain (we refer to this as the ergodicity of the algorithm). Considerable success has been made in each of these areas and various results have demonstrated sufficient conditions to establish convergence and sufficient conditions for ergodicity. Before reviewing some of this research we devote

the remainder of this section to briefly formalising the idea of an adaptive sampler.

Suppose we are interested in sampling from a target distribution π over some space $\mathcal{X}$. Consider a general MCMC algorithm and within that algorithm consider a single move type m. Typically, the proposal distribution $Q_{\boldsymbol{\psi}}$ for the move m will depend upon a $n_{\boldsymbol{\psi}}$-vector of (tuning) parameters $\boldsymbol{\psi} \in \Psi$. For simplicity, we assume for the remainder of the chapter that $\Psi \subseteq \mathbb{R}^{n_\psi}$. The traditional approach when running MCMC is to do several pilot runs of the sampler with various values of $\boldsymbol{\psi}$ to find the value $\boldsymbol{\psi}^\star$ that best achieves some desired measure of performance.

As a specific example, consider the case of the simple univariate Normal random walk Metropolis (RWM) algorithm sampling from some target distribution defined over the real line. This algorithm proposes a new value x' for the chain by sampling from a $N(x, \sigma^2)$ distribution, where x is the current state of the chain. In this example, $\boldsymbol{\psi}$ is the single parameter σ^2.

Adaptive samplers replace the need for pilot runs by tuning themselves while running. Over the length of an MCMC run, the proposal distribution for the move type of interest is no longer static. Rather, if we index the sweep by n, the proposal distribution, $Q_{\boldsymbol{\psi}_n}$, depends on the value $\boldsymbol{\psi}_n$ at sweep n. Importantly, $\boldsymbol{\psi}_n$ (and therefore the proposal $Q_{\boldsymbol{\psi}_n}$) may depend upon the previous $n-1$ samples of the chain $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{n-1}$.

The dependence of $\boldsymbol{\psi}_n$ on the history of the chain $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{n-1}$ demonstrates why the resulting chain is no longer Markovian. Although the joint chain $(\boldsymbol{X}_n, \boldsymbol{\psi}_n)$ is Markovian, it is no longer time homogeneous. Two important

questions that are of interest are how to ensure convergence so that $\boldsymbol{\psi}_n \to \boldsymbol{\psi}^\star$ and how to ensure that $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ retain the important ergodic properties that allow us to produce Monte Carlo estimates.

In the next section we provide a brief summary of some of the main results from a small sample of recent literature in the area. The review is not supposed to be comprehensive but considers a few papers which provide useful tools in the search for automatic samplers. In particular, we concentrate upon the papers written by Haario *et al.* (2001), Andrieu and Robert (2001), Andrieu and Moulines (2004), Atchadé and Rosenthal (2003), Andrieu *et al.* (2004) and Gilks *et al.* (1998). We begin our review looking at the first five of these papers, all of which are instances of a type of adaptation referred to as *diminishing adaptation*. Having reviewed these papers we briefly look at the last paper in the above list which provides us with an insight into an alternative type of adaptation, known as *adaptation by regeneration*. For an alternative review of the research into adaptive sampling the reader is referred to Erland (2003).

Following the review in section 4.2, section 4.3 then looks more closely at the particular algorithm proposed by Atchadé and Rosenthal (2003) and raises some important questions about the details underlying the convergence of the algorithm. These questions are considered more carefully in section 4.4 where we look at an example where the use of a similar but simpler algorithm to that proposed by Atchadé and Rosenthal can result in problems. In section 4.5 we discuss a small simulation study to attempt to understand whether these problems are shared by the original algorithm. Finally, moving away from the Atchadé and Rosenthal algorithm, section 4.6 introduces two adaptive algorithms applicable for reversible jump problems.

4.2 Adaptive Sampling: A Brief Review

We begin our discussion by briefly considering the adaptive algorithm proposed by Haario *et al.* (2001). In this paper, the authors address the specific case of online adaptation of the d-dimensional Normal random walk Metropolis (RWM) algorithm, naming their new adaptive method the *Adaptive Metropolis algorithm* or *AM algorithm*.

Although adaptive samplers existed before the AM algorithm, previous methods fell into two restrictive classes. Firstly, many proposed algorithms resulted in chains that were no longer ergodic (see e.g. the Adaptive Proposal algorithm, Haario *et al.*, 1999). Alternatively, the required conditions and assumptions that allowed the resulting chains to be used in Monte Carlo estimates were difficult to verify (see e.g. Holden, 1998). The results proved by Haario *et al.* (2001) represented an important step towards providing sufficient conditions that could be more easily modified to verify the convergence and ergodicity of other adaptive samplers. We now outline the AM algorithm in more detail noting that we find it convenient to use alternative notation to that used within the original paper.

Let $\boldsymbol{x} \in \mathbb{R}^d$. Recall, for the case of the d-dimensional RWM sampler parameterised by covariance matrix Σ, the density function $q_\Sigma(\boldsymbol{x}, \cdot)$ of the proposal distribution Q_Σ is given by

$$q_\Sigma(\boldsymbol{x}, \boldsymbol{x}') = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{x}' - \boldsymbol{x})^T \Sigma^{-1} (\boldsymbol{x}' - \boldsymbol{x}) \right),$$

where $\boldsymbol{x}' \in \mathbb{R}^d$ is the proposed new state. The underlying idea behind the

AM algorithm is that Σ is updated as the sampler progresses, so that the Normal proposal best captures the shape and correlation features of the target distribution π. If the current state is $\boldsymbol{X}_n$ at time n, the proposed value for the next state of the chain is drawn from the proposal distribution Q_{Σ_n} with density $q_{\Sigma_n}(\boldsymbol{X}_n, \cdot)$. Σ is adapted at each iteration of the MCMC algorithm by setting $\Sigma_n = \lambda_d \text{Cov}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) + \lambda_d \varepsilon I_d$. Here, λ_d is a scale parameter depending on the dimension d, $\text{Cov}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ is the sample covariance matrix of $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, I_d is the d-dimensional identity matrix and $\varepsilon \geq 0$ is a constant. Taking $\varepsilon > 0$ ensures that the adaptive estimates of Σ_n $(n \geq 1)$ remain positive definite matrices, although the authors claim that this precaution is not necessary in practice. The adaptation begins after running the chain for an initial period t_0.

By appealing to a Mixingale process (as defined by McLeish, 1975), Haario *et al.* demonstrate sufficient conditions for which a strong law of large numbers (SLLN) of the form,

$$\frac{1}{N} \sum_{n=1}^{N} g(\boldsymbol{X}_n) \xrightarrow{\text{a.s.}} \mathbb{E}_\pi(g),$$

holds for the chain $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ resulting from the AM algorithm. This result is a special case of a more general result applicable to other adaptive processes also proved within the paper. For concise statements and proofs of these results we direct the interested reader to the original paper.

The general results introduced by Haario *et al.* provide motivation to develop new adaptive algorithms that can easily be checked to see if they satisfy this SLLN. Furthermore, the introduction of the Mixingale approach as a tool for proving ergodicity results has permitted other researchers to further study the properties of general adaptive algorithms. A particular example of the

application of the Mixingale approach is the paper by Atchadé and Rosenthal (2003). In this work the authors extend the general ergodicity results, relaxing some of the requirements used in the original proofs. While we do not give specific details of the results or assumptions, we note briefly two features of their results. Firstly, as remarked by the authors, one of the assumptions required for their ergodicity results equates to the need for each kernel in the family of adaptive kernels to be geometrically ergodic (with respect to the target distribution π) with a common rate ρ. This observation allows classical results about geometric ergodicity (such as the use of drift conditions, see for example Meyn and Tweedie, 1993) to be used to verify the assumptions. Secondly, the results presented by Atchadé and Rosenthal remove the restrictive requirement of the proofs by Haario *et al.* that the state space must be bounded. For further details on these aspects readers are referred to Atchadé and Rosenthal (2003).

In addition to the theoretical results presented by Atchadé and Rosenthal, the authors also introduce a specific algorithm which we detail now. As was the case for the AM algorithm proposed by Haario *et al.*, the adaptive algorithm developed by Atchadé and Rosenthal is based on adapting a Normal RWM proposal distribution. Whereas Haario *et al.* consider a $d-$dimensional Normal proposal distribution parameterised by Σ, the algorithm introduced by Atchadé and Rosenthal concentrates on a univariate RWM algorithm, with proposal density q_σ parameterised by $\sigma > 0$. Atchadé and Rosenthal suggest that the method can be extended to the multivariate case by imposing the condition that $\Sigma = \sigma I_d$. However, due to the restrictiveness of this condition, we detail the univariate case.

The aim of the adaptive scheme proposed by Atchadé and Rosenthal is to adapt the variance parameter σ to achieve some optimal value $\tau^\star$ for the expected

acceptance probability, $\tau(\sigma)$, given by

$$\tau(\sigma) := \int_{\mathbb{R}\times\mathbb{R}} \alpha(x, x')q_\sigma(x, x')\pi(x)\mathrm{d}x\mathrm{d}x'.$$

We refer to the algorithm as the *adaptive acceptance probability* (AAP) algorithm for the remainder of the thesis.

In order to achieve the adaptive aim, Atchadé and Rosenthal modify the standard RWM algorithm in the following way. Suppose that at iteration n, the current state of the Markov chain is x. Suppose further that the current value of the parameter to be adapted is σ_n. We propose a new value for the chain x' by sampling from the proposal distribution Q_{σ_n} with density $q_{\sigma_n}(x, \cdot)$. We then calculate the standard acceptance probability given by $\alpha(x, x') = \min\left(1, \frac{\pi(x')}{\pi(x)}\right)$. With probability $\alpha(x, x')$ we set the new state of the chain to be x' and set the new value of the adaptive parameter to be

$$\sigma_{n+1} = \max\left\{a, \min\left(\sigma_n + \frac{c}{n}(1 - \tau^\star), A\right)\right\}. \tag{4.1}$$

Otherwise the chain remains in the state x and

$$\sigma_{n+1} = \max\left\{a, \min\left(\sigma_n + \frac{c}{n}(0 - \tau^\star), A\right)\right\}. \tag{4.2}$$

Here c is a positive constant, for which the authors suggest $c = \sigma_0$ is an appropriate value. The constants a and A satisfy $0 < a < A < \infty$ and the min and max parts of the update ensure that the adaptive parameter remains within the compact set $[a, A]$. We note that the original AAP algorithm allows for the adaptive parameter to be updated only after every w moves of the Markov chain, replacing the 1 (or 0 in the above formulae) by the empirical average acceptance rate over these w moves. However, the authors claim that the algorithm behaves equally well for $w = 1$ (which is the algorithm described above) as for larger

values of w, so for simplicity in the subsequent discussion we keep $w = 1$. We return to the issue of the algorithm's sensitivity to σ_0 in section 4.5 and again in the examples presented in chapter 5.

Atchadé and Rosenthal demonstrate that the AAP algorithm satisfies the necessary assumptions for their general ergodicity results, meaning that the chains resulting from the algorithm are ergodic. Furthermore, the authors prove that for the AAP algorithm, the adaptive parameter σ_n also converges to the optimal value $\sigma^\star$ (where $\tau(\sigma^\star) = \tau^\star$). Although the convergence results concerning this algorithm appear well founded, they rely on a more basic assumption which the authors fleetingly mention but do not attempt to justify. In the next three sections we look more closely at this algorithm, paying particular attention to this specific assumption. However, before progressing in this direction we continue with our review of the literature on adaptive techniques.

An important step in the study of adaptive algorithms came with the realisation that the general adaptive algorithm could be formulated as a stochastic approximation (SA) problem. By applying Martingale arguments to this general formulation, several authors have achieved even more powerful results than those derived from the Mixingale arguments used in the papers reviewed above. We now consider the SA formulation, discussing three important papers that have been integral in the development of the powerful new convergence and ergodicity criteria for adaptive MCMC.

The first published insight into the usefulness of stochastic approximation algorithms for adaptive MCMC occurred in Andrieu and Robert (2001). In this paper, the authors observe that the problem of adapting MCMC proposal

parameters is in most cases equivalent to the problem of finding solutions to the equation $\boldsymbol{h}(\boldsymbol{\psi}) = \mathbf{0}$ for $\boldsymbol{\psi} \in \Psi$, and $\boldsymbol{h} : \Psi \to \mathbb{R}^m$ for some integer m. In this thesis we concentrate on $\boldsymbol{h}(\boldsymbol{\psi}) = \int_{\mathcal{X}} \boldsymbol{H}(\boldsymbol{\psi}, \boldsymbol{X})\pi(\mathrm{d}\boldsymbol{X})$, for some function $\boldsymbol{H} : \Psi \times \mathcal{X} \to \mathbb{R}^m$. We observe that all arguments hold for more general functions $\boldsymbol{h}$, for example replacing π by some more general distribution $\mu_{\boldsymbol{\psi}}$ over a more general state space, where π might be a marginal distribution of $\mu_{\boldsymbol{\psi}}$.

Stochastic approximation is a class of algorithms particularly suited to solving $\boldsymbol{h}(\boldsymbol{\psi}) = \mathbf{0}$ when only noisy observations of the mean field $\boldsymbol{h}(\boldsymbol{\psi})$ are available for any particular $\boldsymbol{\psi}$. In particular, the authors concentrate on the Robbins-Monro algorithm (as introduced by Robbins and Monro, 1951), whereby the following iterative process

$$\boldsymbol{\psi}_0 \in \Psi, \quad \boldsymbol{\psi}_{n+1} = \boldsymbol{\psi}_n + \gamma_{n+1}\boldsymbol{\zeta}_{n+1} \tag{4.3}$$

is used to solve such equations. In this algorithm, $\boldsymbol{\zeta}_n = \boldsymbol{h}(\boldsymbol{\psi}_n) + \boldsymbol{\xi}_n$ is a noisy measurement of $\boldsymbol{h}(\boldsymbol{\psi}_n)$. The convergence of this algorithm (in terms of $\boldsymbol{\psi}_n \to \boldsymbol{\psi}^\star$) has been studied extensively in the literature (see Andrieu *et al.*, 2004 for references) and requires various conditions on the noise sequence $\{\boldsymbol{\xi}_n, n \geq 0\}$ and the sequence of step sizes $\{\gamma_n, n \geq 0\}$. One common such condition requires $\gamma_n \downarrow 0$ at a suitable rate.

For the case of adaptive MCMC, the noise sequence $\{\boldsymbol{\xi}_n, n \geq 0\}$ is given by

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{H}(\boldsymbol{\psi}_n, \boldsymbol{X}_{n+1}) - \boldsymbol{h}(\boldsymbol{\psi}_n), \tag{4.4}$$

where $\boldsymbol{X}_{n+1}$ is the state of the Markov chain at time $n+1$. Despite demonstrating the applicability of the Robbins-Monro framework, Andrieu and Robert do not provide any theoretical results about the convergence of adaptive MCMC algorithms. However, the authors develop an adaptive multivariate RWM

algorithm for adapting Σ in order to achieve an optimal acceptance probability property. Unfortunately the specific choice of algorithm is a little convoluted and the iterative procedure relies on running two parallel Markov chains. Nonetheless, the algorithm demonstrates clear empirical evidence of convergence.

To demonstrate the general applicability of the SA framework, Andrieu and Robert also note that Haario *et al.*'s AM algorithm could also be formulated in this manner. In particular, defining $\boldsymbol{\mu}_n = \frac{1}{n}\sum_{r=1}^{n} \boldsymbol{X}_r$, and noting

$$\text{Cov}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) = \frac{1}{n-1}(\sum_{r=1}^{n} \boldsymbol{X}_r \boldsymbol{X}_r^T - n\boldsymbol{\mu}_n \boldsymbol{\mu}_n^T),$$

the AM algorithm prescribed above (with $\varepsilon = 0$) can be written using the following recurrence relation. For $n \geq 0$ and some initial estimates $\boldsymbol{\mu}_0$ and Σ_0,

$$\begin{aligned} \boldsymbol{\mu}_{n+1} &= \boldsymbol{\mu}_n + \frac{1}{n+1}(\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n) \\ \Sigma_{n+1} &= \Sigma_n + \frac{1}{n+1}(\lambda_d(\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n)^T - \nu_n \Sigma_n), \end{aligned}$$

where, $\nu_n = (n+1)/n$.

Setting $\boldsymbol{\psi} = (\boldsymbol{\mu}, \Sigma)$, $\gamma_n = 1/n$ and defining

$$\boldsymbol{H}(\boldsymbol{\psi}_n, \boldsymbol{X}_{n+1}) = (\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n, \lambda_d(\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{X}_{n+1} - \boldsymbol{\mu}_n)^T - \nu_n \Sigma_n)^T,$$

it is clear that the AM algorithm can be rewritten as

$$\boldsymbol{\psi}_{n+1} = \boldsymbol{\psi}_n + \gamma_{n+1} \boldsymbol{H}(\boldsymbol{\psi}_n, \boldsymbol{X}_{n+1})$$

as required.

The first theoretical results about the ergodicity and convergence of general adaptive samplers using the general SA formulation are presented in the two

interdependent papers by Andrieu and Moulines (2004) and Andrieu *et al.* (2004). In these papers, the authors introduce an abstract general adaptive algorithm and then prove theoretical results about this algorithm. Illustrative examples demonstrate that adaptive versions of both the Normal random walk Metropolis and the Independent Metropolis-Hastings algorithms meet the conditions necessary to ensure that the resulting chains are ergodic.

The first contribution of Andrieu *et al.* was to demonstrate sufficient conditions for convergence of the adaptive parameters, where the sequence $\{\boldsymbol{\xi}_n, n \geq 0\}$ (see equation 4.3) is deterministic. The authors show that if it is possible to establish the existence of a global Lyapunov function $\boldsymbol{w}$, for the mean field $\boldsymbol{h}$, then mild conditions on the sequence $\{\boldsymbol{\xi}_n, n \geq 0\}$ and the step sizes $\{\gamma_n, n \geq 0\}$ will ensure convergence of the Robbins-Monro algorithm.

The existence of a Lyapunov function ensures that the updates of the parameter $\boldsymbol{\psi}$ occur correctly. In particular, when such a function exists, the updates prescribed by equation 4.3 are such that adding $\boldsymbol{h}(\boldsymbol{\psi}_n)$ will add a contribution which will move the current parameter value $\boldsymbol{\psi}_n$ in the direction of the target value $\boldsymbol{\psi}^\star$. The specific conditions that can be used to verify whether a function is a Lyapunov function are detailed later in this section.

Andrieu *et al.* use the existence of a Lyapunov function as a basis for considering the more complicated and interesting case of Markov state-dependent noise. It is this case that is of interest for adaptive MCMC. The results rely on the conditions for the noise and stepsize sequences to be satisfied almost surely. The boundedness conditions required by Haario *et al.* are removed and convergence results are proved for the general case.

Recall, we are interested in the case where the adaptive MCMC algorithm can be written

$$\boldsymbol{\psi}_{n+1} = \boldsymbol{\psi}_n + \gamma_{n+1}\boldsymbol{H}(\boldsymbol{\psi}_n, \boldsymbol{X}_{n+1}), \tag{4.5}$$

for $\boldsymbol{X}_0 \in \mathcal{X}$, $\boldsymbol{\psi}_0 \in \Psi$ and $\boldsymbol{X}_{n+1} \sim \mathcal{K}_{\boldsymbol{\psi}_n}(\boldsymbol{X}_n, \cdot)$. We make the restriction that $\boldsymbol{\gamma} = \{\gamma_n : n \in \mathbb{N}\}$ is some non-increasing sequence, with $\gamma_0 \leq 1$. In order to study this algorithm the authors make some preliminary assumptions about the function $\boldsymbol{H}$ and the family of proposal kernels $\{\mathcal{K}_{\boldsymbol{\psi}}, \ \boldsymbol{\psi} \in \Psi\}$. As with the Lyapunov conditions, we state them explicitly with the main convergence and ergodicity results below. We note here that the conditions amount to the target distribution π being the unique stationary distribution for $\mathcal{K}_{\boldsymbol{\psi}}$ for every value of $\boldsymbol{\psi} \in \Psi$, and the expectation of the absolute value of $\boldsymbol{H}$ with respect to the target distribution being finite for every value of $\boldsymbol{\psi} \in \Psi$.

The approach taken by Andrieu *et al.* towards determining convergence for Markov state-dependent noise is to construct a general algorithm by amending the classical Robbins-Monro algorithm. This is achieved by truncating the proposal parameters $\boldsymbol{\psi}$ to compact sets which are themselves permitted to adapt. Before considering this general algorithm we introduce some additional notation.

For a vector $\boldsymbol{v} \in \mathbb{R}^d$, we denote the norm of the vector by $|\boldsymbol{v}| = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle}$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Also, for a set $B \subset \mathbb{R}^d$, we define the delta function $\delta_{\boldsymbol{v}}(B)$, which takes the value 1 if $\boldsymbol{v} \in B$ and 0 otherwise. Suppose now we define $\{\boldsymbol{Y}_n\} = \{(\boldsymbol{X}_n, \boldsymbol{\psi}_n)\}$ to be the inhomogeneous Markov chain on the space $\mathcal{X} \times \Psi$ arising from the adaptive scheme described in equation 4.5. Following Andrieu *et al.*, for $(\boldsymbol{x}, \boldsymbol{\psi}) \in \mathcal{X} \times \Psi$ and sets $\mathcal{A}_{\mathcal{X}} \in \mathcal{B}(\mathcal{X})$ and $\mathcal{A}_{\Psi} \in \mathcal{B}(\Psi)$, we

define the sequence of transition probabilities $\{Q_{\gamma_n}\}$ that generates the $\boldsymbol{Y}-$chain as

$$Q_{\gamma_n}((\boldsymbol{x},\boldsymbol{\psi}),(\mathcal{A}_{\mathcal{X}}\times\mathcal{A}_{\Psi}))=\int_{\mathcal{A}_{\mathcal{X}}}\mathcal{K}_{\boldsymbol{\psi}}(\boldsymbol{x},\mathrm{d}\boldsymbol{x}')\delta_{\boldsymbol{\psi}+\gamma_n\boldsymbol{H}(\boldsymbol{\psi},\boldsymbol{x}')}(\mathcal{A}_{\Psi}).$$

We also define a sequence $\{\mathcal{C}_l:\ l\in\mathbb{N}\}$ of compact subsets of Ψ, satisfying

$$\bigcup_{l\geq 0}\mathcal{C}_l=\Psi \text{ and } \mathcal{C}_l\subset\mathrm{int}(\mathcal{C}_{l+1}).$$

Finally, we introduce another non-increasing sequence $\boldsymbol{\epsilon}=\{\epsilon_n\}$ of positive numbers and three sequences of indexing variables $\{\kappa_n\}$, $\{\varsigma_n\}$ and $\{\nu_n\}$.

With this notation in place the general algorithm proceeds by considering the homogeneous Markov chain $\{\boldsymbol{Z}_n\}$, where $\boldsymbol{Z}_n=(\boldsymbol{X}_n,\boldsymbol{\psi}_n,\kappa_n,\varsigma_n,\nu_n)\in\mathcal{X}\times\Psi\times\mathbb{N}^3$ is generated according to the following updating regime. Let $A\subset\mathcal{X}$ and let $\Phi:\mathcal{X}\times\Psi\rightarrow A\times\mathcal{C}_0$ be a measurable function. Let $(\kappa_0,\varsigma_0,\nu_0)=(0,0,0)$. At iteration $n+1$,

$$(\boldsymbol{X}_{n+1},\boldsymbol{\psi}_{n+1}) \sim \begin{cases} Q_{\gamma_{\varsigma_n}}(\Phi(\boldsymbol{X}_n,\boldsymbol{\psi}_n),\cdot) & \text{if } \nu_n=0 \\ Q_{\gamma_{\varsigma_n}}((\boldsymbol{X}_n,\boldsymbol{\psi}_n),\cdot) & \text{otherwise,} \end{cases} \tag{4.6}$$

$$(\kappa_{n+1},\varsigma_{n+1},\nu_{n+1}) = \begin{cases} (\kappa_n,\varsigma_n+1,\nu_n+1) & \text{if } |\boldsymbol{\psi}_{n+1}-\boldsymbol{\psi}_n|<\epsilon_{\varsigma_n} \\ & \text{and } \boldsymbol{\psi}_{n+1}\in\mathcal{C}_{\kappa_n} \\ (\kappa_n+1,\varsigma_n+1,0) & \text{otherwise.} \end{cases} \tag{4.7}$$

Although this general algorithm appears quite abstract, it is clearly explained and motivated in Andrieu *et al.*. In brief, the original Markov chain variable $\boldsymbol{X}$ and adaptive parameters $\boldsymbol{\psi}$ are updated according to the basic Robbins-Monro updating scheme, with the exception that the algorithm is reinitialised if $\boldsymbol{\psi}_{n+1}$ differs too much from $\boldsymbol{\psi}_n$ or if $\boldsymbol{\psi}_{n+1}$ strays outside the current compact set. The counter variables κ, ς and ν are simply tools to achieve this reprojection.

We observe that in the original general algorithm, the updates of the counter variables are allowed to be more general than those prescribed by equation 4.7, but we detail this version for simplicity. Importantly, the reprojections ensure that the recursion remains within a random compact set and control the noise sequence $\{\boldsymbol{\xi}_n\}$ (see equation 4.4). In addition, the adaptive nature of the truncation sets guarantees that after sufficiently many iterations and reprojections the solution set of $\boldsymbol{h}(\boldsymbol{\psi}) = \mathbf{0}$ will intersect the compact set to which $\{\boldsymbol{\psi}_n\}$ is confined. These two features, coupled with a proof that reprojection will occur only finitely often, combine to allow rigorous results to be obtained about the convergence of this general algorithm.

Andrieu and Moulines also adopt the general algorithm to produce results about the ergodicity of the resulting chains. Both the convergence and ergodicity results detailed in these papers are based upon a shared set of assumptions, although the assumptions are numbered differently in the two papers. In order to fully state the important results of these papers we reproduce these assumptions here. We choose to adopt the numbering of Andrieu *et al.*. Having stated these assumptions and briefly remarked about their purposes we then restate the main results of these papers. We do not reproduce the proofs of these papers as they are lengthy and involved. The interested reader will find full details in the original papers. Before stating the assumptions we borrow the following extra notation from Andrieu *et al.*. For a transition kernel $\mathcal{K}$, and a measurable function $\mu : \mathcal{X} \to [0, \infty)$, define $\mathcal{K}\mu(\boldsymbol{x}) = \int_{\mathcal{X}} \mathcal{K}(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')\mu(\boldsymbol{x}')$. For a function $w : \Psi \to [0, \infty)$ define the level set $\mathcal{W}_M = \{\boldsymbol{\psi} \in \Psi : \ w(\boldsymbol{\psi}) \leq M\} \subseteq \Psi$. Also, for functions $V : \mathcal{X} \to [1, \infty)$ and $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^{n_\psi}$ we define the following norm

$$\|\boldsymbol{g}\|_V = \sup_{\boldsymbol{x} \in \mathcal{X}} \frac{|\boldsymbol{g}(\boldsymbol{x})|}{V(\boldsymbol{x})}.$$

Given this norm, we define the set $\mathcal{L}_V := \{\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^{n_\psi} : \|\boldsymbol{g}\|_V < \infty\}$. Finally, for the function $\boldsymbol{H}(\boldsymbol{\psi}, \boldsymbol{x})$ (introduced above) we also use the alternative notation $\boldsymbol{H}_{\boldsymbol{\psi}}(\boldsymbol{x})$.

The first set of assumptions are as follows.

Assumptions (A1)

Suppose Ψ is an open subset of $\mathbb{R}^{n_\psi}$ and the mean field $\boldsymbol{h} : \Psi \to \mathbb{R}^{n_\psi}$ is continuous. Suppose that there exists a continuously differentiable function $w : \Psi \to [0, \infty)$ such that

(i) there exists $M_0 > 0$ such that

$$\mathcal{L} := \{\boldsymbol{\psi} \in \Psi : \langle \nabla w(\boldsymbol{\psi}), \boldsymbol{h}(\boldsymbol{\psi}) \rangle = 0\} \subset \{\boldsymbol{\psi} \in \Psi : w(\boldsymbol{\psi}) < M_0\};$$

(ii) there exists $M_1 \in (M_0, \infty)$ such that $\mathcal{W}_{M_1}$ is a compact set;

(iii) for any $\boldsymbol{\psi} \in \Psi \setminus \mathcal{L}$, $\langle \nabla w(\boldsymbol{\psi}), \boldsymbol{h}(\boldsymbol{\psi}) \rangle < 0$; and

(iv) the closure of $w(\mathcal{L})$ has an empty interior.

These assumptions imply that the function w is a global Lyapunov function for the mean field $\boldsymbol{h}$. As noted by Andrieu *et al.* (2004), if we can find some function J such that $\boldsymbol{h} = -\nabla J$, then the choice $w = J$ satisfies these conditions. They also note that Sard's theorem of differential geometry can be used to establish assumption (**A1**-iv). We now look at the second collection of assumptions that are required to hold.

Assumptions (A2)

For any $\boldsymbol{\psi} \in \Psi$, the Markov kernel $\mathcal{K}_{\boldsymbol{\psi}}$ has a single stationary distribution π, i.e. $\int_{\mathcal{X}} \pi(\mathrm{d}\boldsymbol{x}) \mathcal{K}_{\boldsymbol{\psi}}(\boldsymbol{x}, \mathcal{A}) = \pi(\mathcal{A})$, for all Borel sets $\mathcal{A} \subseteq \mathcal{X}$. In addition,

$\boldsymbol{H} : \Psi \times \mathcal{X} \to \Psi$ is measurable and for all $\boldsymbol{\psi} \in \Psi$, $\int_{\mathcal{X}} |\boldsymbol{H}(\boldsymbol{\psi}, \boldsymbol{x})| \pi(\mathrm{d}\boldsymbol{x}) < \infty$.

The third set of conditions (**A3**) assumed by Andrieu *et al.* guarantee the existence and regularity of a solution to Poisson's equation for a family of transition kernels $\{\mathcal{K}_{\boldsymbol{\psi}} : \boldsymbol{\psi} \in \Psi\}$. The authors discuss Poisson's equation and note that the existence of such solutions is known to occur if the kernels are geometrically ergodic.

Despite the necessity of assumptions (**A3**), the authors note that in practice the requirements are difficult to verify directly. However, to overcome this problem the authors provide an alternative set of drift conditions (**DRI**). They demonstrate that if these drift conditions are satisfied, then assumptions (**A3**) will automatically hold. In nearly all practical cases it is these drift conditions that will be applied and therefore we choose only to state these conditions and omit further details of assumptions (**A3**).

Assumptions (DRI) For any $\boldsymbol{\psi} \in \Psi$, $\mathcal{K}_{\boldsymbol{\psi}}$ is irreducible and aperiodic. In addition, there exist a function $V : \mathcal{X} \to [1, \infty)$, and constants $p \geq 2$ and $\beta \in (0, 1]$ such that for any compact subset $\mathcal{C} \subset \Psi$,

(i) there exists a small set $\mathcal{S} \subset \mathcal{X}$, an integer m, constants $0 < \lambda < 1$ and $b, \kappa, \delta > 0$ and a probability measure ν such that

$$\begin{aligned}
\sup_{\boldsymbol{\psi} \in \mathcal{C}} \mathcal{K}_{\boldsymbol{\psi}}^m V^p(\boldsymbol{x}) &\leq \lambda V^p(\boldsymbol{x}) + b\mathbf{1}_{\{\boldsymbol{x} \in \mathcal{S}\}} && \forall \boldsymbol{x} \in \mathcal{X}, \\
\sup_{\boldsymbol{\psi} \in \mathcal{C}} \mathcal{K}_{\boldsymbol{\psi}} V^p(\boldsymbol{x}) &\leq \kappa V^p(\boldsymbol{x}) && \forall \boldsymbol{x} \in \mathcal{X}, \\
\inf_{\boldsymbol{\psi} \in \mathcal{C}} \mathcal{K}_{\boldsymbol{\psi}}^m(\boldsymbol{x}, \mathcal{A}) &\geq \delta \nu(\mathcal{A}) && \forall \boldsymbol{x} \in \mathcal{S},\ \forall \mathcal{A} \in \mathcal{B}(\mathcal{X});
\end{aligned}$$

(ii) there exists a constant C_1 such that, for all $\boldsymbol{x} \in \mathcal{X}$,

$$\sup_{\boldsymbol{\psi}\in\mathcal{C}} |\boldsymbol{H}_{\boldsymbol{\psi}}(\boldsymbol{x})| \leq C_1 V(\boldsymbol{x}),$$
$$\sup_{(\boldsymbol{\psi},\boldsymbol{\psi}')\in\mathcal{C}\times\mathcal{C}\setminus\{\boldsymbol{\psi}\}} |\boldsymbol{\psi}-\boldsymbol{\psi}'|^{-\beta}|\boldsymbol{H}_{\boldsymbol{\psi}}(\boldsymbol{x}) - \boldsymbol{H}_{\boldsymbol{\psi}'}(\boldsymbol{x})| \leq C_1 V(\boldsymbol{x}); \quad \text{and}$$

(iii) there exists a constant C_2 such that, for all $(\boldsymbol{\psi}, \boldsymbol{\psi}') \in \mathcal{C} \times \mathcal{C}$,

$$\begin{aligned} \|\mathcal{K}_{\boldsymbol{\psi}}\boldsymbol{g} - \mathcal{K}_{\boldsymbol{\psi}'}\boldsymbol{g}\|_V &\leq C_2\|\boldsymbol{g}\|_V|\boldsymbol{\psi}-\boldsymbol{\psi}'|^{\beta} \quad \forall \boldsymbol{g} \in \mathcal{L}_V, \\ \|\mathcal{K}_{\boldsymbol{\psi}}\boldsymbol{g} - \mathcal{K}_{\boldsymbol{\psi}'}\boldsymbol{g}\|_{V^p} &\leq C_2\|\boldsymbol{g}\|_{V^p}|\boldsymbol{\psi}-\boldsymbol{\psi}'|^{\beta} \quad \forall \boldsymbol{g} \in \mathcal{L}_{V^p}. \end{aligned}$$

As noted by Andrieu *et al.*, these drift conditions ensure that the kernel $\mathcal{K}_{\boldsymbol{\psi}}$ has a stationary distribution π, and that the resulting chain is V^p-uniformly ergodic.

The final set of assumptions involves the rate at which the updates diminish.

Assumptions (A4)

Let $\alpha \in (0, \beta)$, where β is defined in assumptions (**DRI**). The sequences $\boldsymbol{\gamma} = \{\gamma_n\}$ and $\boldsymbol{\epsilon} = \{\epsilon_n\}$ (from the general algorithm in equations 4.6 and 4.7) are non-increasing, positive and satisfy $\sum_{n=0}^{\infty} \gamma_n = \infty$, $\lim_{n\to\infty} \epsilon_n = 0$ and

$$\sum_{n=0}^{\infty} \{\gamma_n^2 + \gamma_n\epsilon_n^{\alpha} + (\epsilon_n^{-1}\gamma_n)^p\} < \infty,$$

where p is defined in (**DRI**).

If the conditions (**A3**) are proved directly then α is as defined therein. For most adaptive samplers, if assumptions (**A1**) to (**A3**) can be satisfied then this last set of assumptions can be achieved by design.

Having reproduced the assumptions of Andrieu *et al.* (2004) and Andrieu and Moulines (2004) we can quote two powerful results proved within these papers.

We first define the distance d between a point $\boldsymbol{x} \in \mathbb{R}^l$ and a set $A \subset \mathbb{R}^l$ (where l is an integer), as $d(\boldsymbol{x}, A) = \inf_{\boldsymbol{y} \in A} |\boldsymbol{x} - \boldsymbol{y}|$. We also require the probability measure $\bar{\mathbb{P}}_{\boldsymbol{x}_0, \boldsymbol{\psi}_0}$ associated with the Markov chain $(\boldsymbol{Z}_n)$ resulting from the general algorithm (described in equations 4.6 and 4.7) started at $\boldsymbol{z}_0 = (\boldsymbol{x}_0, \boldsymbol{\psi}_0, 0, 0, 0)$.

Theorem 4.1. *[Andrieu* et al. *(2004), Theorem 5.5] Assume (**A1**) to (**A4**). Let $A \subset \mathcal{X}$ (defined in the general algorithm) be such that $\sup_{\boldsymbol{x} \in A} V^p(\boldsymbol{x}) < \infty$. Suppose we choose $\mathcal{C}_0 \subset \mathcal{W}_{M_0}$, for M_0 as defined in assumptions (**A1**). Let $\{\boldsymbol{Z}_n\}$ be the Markov chain resulting from the general algorithm described in equations 4.6 and 4.7, with $(\boldsymbol{x}_0, \boldsymbol{\psi}_0) = (\boldsymbol{x}, \boldsymbol{\psi})$. Then for all $(\boldsymbol{x}, \boldsymbol{\psi}) \in \mathcal{X} \times \Psi$, we have $\lim_{n \to \infty} d(\boldsymbol{\psi}_n, \mathcal{L}) = 0$ almost surely, where $\mathcal{L}$ is as defined in (**A1**).*

Theorem 4.2. *[Andrieu and Moulines (2004), Theorem 6] Assume (**DRI**) with $p > 2$, $\beta \in (0, 1]$ and $V : \mathcal{X} \to [1, \infty)$. Let $\{\boldsymbol{Z}_n\}$ be the Markov chain resulting from the general algorithm described in equations 4.6 and 4.7, with $(\boldsymbol{x}_0, \boldsymbol{\psi}_0) = (\boldsymbol{x}, \boldsymbol{\psi})$. Let $\sum_{n=1}^{\infty} n^{-1/2} \epsilon_n^{\alpha} < \infty$, for some $\alpha \in (0, \beta)$, and for all $(\boldsymbol{x}, \boldsymbol{\psi}) \in \mathcal{X} \times \Psi$ let $\bar{\mathbb{P}}_{\boldsymbol{x}, \boldsymbol{\psi}}(\lim_{n \to \infty} \kappa_n < \infty) = 1$, where ϵ_n and κ_n are defined in the general algorithm. Then, for $f \in \mathcal{L}_V$,*

$$\frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{X}_n) \xrightarrow{\text{a.s.}} \mathbb{E}_{\pi}[f(\boldsymbol{X})].$$

The first of these theorems states that the algorithm will converge. The second asserts that a Law of Large Numbers can be applied to the chain resulting from the algorithm. This result demonstrates that if an adaptive sampler can be formulated in terms of the general algorithm and it can be demonstrated that the required assumptions hold, then the chain resulting from the sampler can be used in the same way as the sample from a normal MCMC sampler. Even more powerfully, Andrieu and Moulines (2004) also state conditions for which a central limit theorem applies to the resulting chain. For more details of this result and proofs of the above theorems the reader is invited to consult the original papers.

While theoretical results are undoubtedly important, equally important is the application of such results to practical algorithms. Both, Andrieu *et al.*

and Andrieu and Moulines also make an important step in this direction, demonstrating that adaptive algorithms based on either independence MCMC samplers or random walk Metropolis samplers are special cases of the general algorithm for which the above results hold. As a particular example Andrieu *et al.* use the reformulation of the AM algorithm proposed by Haario *et al.* to extend its convergence and ergodicity properties, removing the need for the state space to be bounded.

The assumptions (**A1**) to (**A4**) are perhaps more easily verifiable than those used for the Mixingale results. Nonetheless, it is worth emphasising that for all adaptive algorithms the required conditions are not minimal and some technical statistical knowledge is required to design new algorithms and check that they demonstrate the appropriate properties.

Having concentrated the majority of this section on diminishing adaptation algorithms, we now briefly consider an alternative form of adaptation known as adaptation by regeneration. We begin by introducing the ideas and underlying concepts for such methods and briefly discuss the research by Gilks *et al.* (1998). Once again, Erland (2003) provides a useful and more detailed insight into this type of adaptation, including comprehensive further references.

Underlying the process of adaptation by regeneration is the requirement that there exists a subset B of the state space $\mathcal{X}$, such that if $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ is the output from the Markov chain, then given $\boldsymbol{X}_n \in B$, $(\boldsymbol{X}_{n+1}, \boldsymbol{X}_{n+2}, \ldots)$ is independent of $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n-1})$ with some probability p. The set B is known as a *proper atom* for the Markov chain, and the chain is said to *regenerate* each time it visits the set B and this independence occurs. We denote the time of the i^{th} regeneration by T_i.

The idea behind adaptation by regeneration is that each time the chain regenerates we can adapt the transition kernel that is used for subsequent transitions for the Markov chain. In particular, suppose the chain starts at state $\boldsymbol{X}_0 \in B$, so that $T_0 = 0$. The chain proceeds using transition kernel $\mathcal{K}_0$ until some regeneration time, T_1, when the chain enters the set B and regenerates. Since subsequent realisations of the Markov chain are independent of the history of the chain $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{T_1-1})$, when the chain regenerates it is valid to define a new transition kernel $\mathcal{K}_1$ which may depend on these past realisations $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{T_1-1})$. The new kernel is then used until the next regeneration time T_2, when the kernel may again be updated. This process continues so that at regeneration time T_i, a new kernel K_i is defined which can use the whole history of the chain $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{T_i-1})$ up to this point.

Most important in establishing the validity of adaptation by regeneration are the results of Gilks *et al.* (1998). Suppose f is a function of interest, and define $F_i = \sum_{j=T_i}^{T_{i+1}-1} f(\boldsymbol{X}_j)$. Also define $N_i = T_{i+1} - T_i$ and $Z_i = F_i - N_i \mathbb{E}_\pi(f)$. Given $\mathbb{E}_\pi(Z_i^2) < b < \infty$ for all i, if $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ is the Markov chain resulting from adapting the transition kernel at each regeneration, then

$$\bar{f}_n = \frac{1}{n} \sum_{j=1}^{n} f(\boldsymbol{X}_j) \text{ is MSE consistent for } \mathbb{E}_\pi(f).$$

In addition to this result, Gilks *et al.* state sufficient further conditions for a central limit theorem to hold.

While the importance of these results is unquestionable, the authors acknowledge that the necessary assumptions are hard to demonstrate in practice. Indeed, although Gilks *et al.* introduce adaptive regeneration schemes for both an

independence sampler and a random walk Metropolis sampler, they do not present results guaranteeing their ergodicity. Nonetheless, the numerical results provide convincing support for the authors' claim that as long as adaptation is done in a sensible manner, with the aim of improving mixing, then "the regularity conditions generally will be satisfied" (Gilks *et al.*).

We return to the discussion of adaptation by regeneration in section 4.6 in the context of a new reversible jump adaptive algorithm. The chapter continues by resuming our review of diminishing adaptation methods, in particular by looking more closely at the adaptive acceptance probability proposed by Atchadé and Rosenthal.

4.3 A Closer Look at the AAP Algorithm

Recall from section 4.2, that the idea of the algorithm proposed by Atchadé and Rosenthal is to adapt the variance parameter $\psi > 0$ of a Normal random walk Metropolis kernel in order to achieve an optimal average acceptance rate $\tau^\star = \tau(\psi^\star)$, where

$$\tau(\psi) = \int_{\mathbb{R}\times\mathbb{R}} \alpha(x, x')q_\psi(x, x')\pi(x)\mathrm{d}x\mathrm{d}x'. \tag{4.8}$$

Here, q_ψ is the density function of a Normal distribution with variance ψ centred on the current state x, and our state space is $\mathcal{X} = \mathbb{R}$. The term $\alpha(x, x') = \min\{1, \pi(x')/\pi(x)\}$ is the standard acceptance probability of a proposed move from x to x'. (Our notation differs from that of the original paper for convenience in the following discussion. Specifically, we parameterise by the variance ψ rather than the standard deviation σ.)

The authors use Mixingale arguments to demonstrate convergence of the adaptive parameter and ergodicity of the resulting chains. However, underlying the convergence of the proposed algorithm is the assumption that the function τ is a non-increasing function of ψ. Despite the importance of this assumption it receives little attention from the original authors. As such, we use this section and the following two to explore this supposition more carefully.

The natural place to begin is to check whether the assumption can be verified directly. It is obvious that the assumption is equivalent to the statement $\frac{\partial \tau}{\partial \psi} \leq 0$ for all ψ. Letting $z = x' - x$ and noting that for the symmetric random walk $\alpha(x, x')$ does not depend on ψ we have

$$\begin{aligned} \frac{\partial \tau}{\partial \psi} &= \frac{\partial}{\partial \psi} \int_{\mathbb{R}\times\mathbb{R}} \alpha(x, x+z) q_\psi(z) \pi(x) \mathrm{d}x \mathrm{d}z \\ &= \int_{\mathbb{R}\times\mathbb{R}} \alpha(x, x+z) \frac{\partial}{\partial \psi} [q_\psi(z)] \pi(x) \mathrm{d}x \mathrm{d}z, \end{aligned}$$

where $q_\psi(z) = \frac{1}{\sqrt{2\pi\psi}} \exp\left\{-\frac{z^2}{2\psi}\right\}$. (See for example Andrieu and Robert, 2001 for confirmation that we can take the derivative inside the integral). Observing that $\frac{\partial}{\partial \psi} q_\psi(z) = q_\psi(z) \frac{\partial}{\partial \psi} \log[q_\psi(z)]$, simple differentiation yields

$$\frac{\partial \tau}{\partial \psi} = \int_{\mathbb{R}\times\mathbb{R}} \frac{(z^2 - \psi)}{2\psi^2} \alpha(x, x+z) q_\psi(z) \pi(x) \mathrm{d}x \mathrm{d}z. \tag{4.9}$$

In order to be certain of the validity of the algorithm it remains only to show that the integral on the right-hand side of equation 4.9 (we shall call this integral $I(\psi)$) is less than or equal to 0. By splitting the real line into the sets $Z := \{z \in$

$\mathbb{R} : \ z^2 > \psi\}$ and $\mathbb{R} \setminus Z$, it is possible to show,

$$\begin{aligned} I(\psi) &= \int_{\mathbb{R}\times Z} \frac{(z^2-\psi)}{2\psi^2}\alpha(x,x+z)q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z \\ &\quad + \int_{\mathbb{R}\times\mathbb{R}\setminus Z} \frac{(z^2-\psi)}{2\psi^2}\alpha(x,x+z)q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z \\ &\leq \int_{\mathbb{R}\times Z} \frac{(z^2-\psi)}{2\psi^2}q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z \\ &\leq \int_{\mathbb{R}\times Z} \frac{z^2}{2\psi^2}q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z \\ &\leq \int_{\mathbb{R}\times\mathbb{R}} \frac{z^2}{2\psi^2}q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z \\ &= \frac{1}{2\psi}. \end{aligned} \tag{4.10}$$

By analogous arguments it can also be shown that $I(\psi) \geq -\frac{1}{2\psi}$.

Although we can provide an upper bound for $I(\psi)$, since it is positive it does not help in our quest to check that the condition $\frac{\partial\tau}{\partial\psi} \leq 0$ holds. Similarly, since the lower bound is negative, this does not contradict the requirement. While it is apparent that the bounds provided above are not very tight, we have been unable to find tighter bounds that would resolve the problem. Indeed, for a general target distribution π, we are yet to ascertain whether or not the necessary inequality holds or not.

An interesting feature of the above problem is that we can re-write the integral $I(\psi)$ as,

$$I(\psi) = \frac{1}{2\psi^2}\left[\int_{\mathbb{R}\times\mathbb{R}} z^2\alpha(x,x+z)q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z - \int_{\mathbb{R}\times\mathbb{R}} \psi\alpha(x,x+z)q_\psi(z)\pi(x)\mathrm{d}x\mathrm{d}z.\right]$$

Thus the condition that we are trying to verify is equivalent to

$$\mathbb{E}_{\pi,q_\psi}\left[Z^2\alpha(X,X+Z)\right] - \mathbb{E}_{\pi,q_\psi}\left[Z^2\right]\mathbb{E}_{\pi,q_\psi}\left[\alpha(X,X+z)\right] \leq 0. \tag{4.11}$$

In words, the algorithm works if, under the stationary distribution π, the correlation between the squared distance of proposed jump, Z^2, and acceptance probability, $\alpha(X, Z)$, is less than or equal to zero for every value of $\psi > 0$. At first glance, this might seem like an intuitively obvious relation. However, for the algorithm to work for any target distribution π, the condition must hold for all distributions π making it more general than on first appearances. Despite careful consideration we are unable to demonstrate that this condition must always hold. Nonetheless, we hope that other researchers may be more successful in considering this problem. However, in the next section we demonstrate that for a similar algorithm to the AAP algorithm, using uniform rather than Normal proposals, the same important condition does not hold.

4.4 A Counter Example for a Similar Algorithm

Consider again the adaptive acceptance probability (AAP) algorithm (Atchadé and Rosenthal) discussed in the previous section. Suppose now however, that instead of adapting a Normal random walk kernel, we wish to adapt a random walk algorithm that uses uniform proposal distributions. In particular, for a parameter $\psi > 0$, suppose that we now redefine

$$q_\psi(x, x') := \begin{cases} \frac{1}{2\psi} & x' \in (x - \psi, x + \psi) \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Clearly, if the current state of the Markov chain is x, $\mathbb{E}(X') = x$ and $\mathrm{Var}(X') = \frac{\psi^2}{3}$, so that as in the case of the Normal random walk, the parameter ψ controls the variance of the proposal distribution. Just as in the case of the Normal random walk, our aim is to adapt the value of ψ, in order to achieve some target value $\tau^\star$ for the expected acceptance probability $\tau(\psi)$, where τ is as

in equation 4.8, but with the new uniform proposal $q_\psi(x, x')$.

Regardless of whether we choose uniform or Normal random walk proposals, increasing ψ increases the variance of the proposal distribution q_ψ. It therefore seems natural that to achieve the optimal expected acceptance rate, at each iteration we should update the proposal parameter ψ according to the AAP algorithm with our new proposals. Again, the convergence of this algorithm relies upon the condition $\frac{\partial \tau}{\partial \psi} \leq 0$, i.e. the expected acceptance probability must be a non-increasing function of ψ. Whilst in the previous section we were unable to resolve whether or not this requirement always held when the proposal distributions were Normal, in the remainder of this section we demonstrate that it is possible to choose a target distribution $\pi(x)$ such that the condition does not hold when the proposals are uniform. Thus we are able to conclude that the AAP algorithm is not appropriate for random walks with uniform proposals.

In order to highlight our result, we consider the case where the target distribution is a mixture of two separated uniform distributions. To be precise, we let the density of a point $x \in \mathbb{R}$ be given by

$$\pi(x) = \begin{cases} \frac{1}{2} & x \in \mathcal{X}_\beta \\ 0 & \text{otherwise,} \end{cases} \tag{4.13}$$

where $\beta > 0$ and $\mathcal{X}_\beta = \{x \in \mathbb{R} : \ -\beta - 1 \leq x \leq -\beta\} \cup \{x \in \mathbb{R} : \ \beta \leq x \leq \beta + 1\}$. An example of this density function with $\beta = 1$ is illustrated in figure 4.4 in the following section. For this target distribution, we must have $\psi > 2\beta$ otherwise the Markov chain sampler is no longer irreducible. We now show by direct computation that with this choice of π, $\tau(\psi)$ is not a non-increasing function for the permissable range of ψ values.

We begin by substituting our definitions of π and q_ψ (from equations 4.12 and 4.13) into equation 4.8. This gives,

$$\tau(\psi) = \int_{-\beta-1}^{-\beta}\int_{x-\psi}^{x+\psi} \alpha(x,x')\frac{1}{2\psi}\frac{1}{2}\mathrm{d}x'\mathrm{d}x + \int_{\beta}^{\beta+1}\int_{x-\psi}^{x+\psi} \alpha(x,x')\frac{1}{2\psi}\frac{1}{2}\mathrm{d}x'\mathrm{d}x.$$

Making two changes of variables in the first double integral and noting that π is symmetric, so that $\alpha(x,x') = \alpha(-x,x') = \alpha(x,-x')$, it is easily shown that

$$\tau(\psi) = 2\int_{\beta}^{\beta+1}\int_{x-\psi}^{x+\psi} \alpha(x,x')\frac{1}{2\psi}\frac{1}{2}\mathrm{d}x'\mathrm{d}x.$$

Since $\pi(x') = 0$ for $x' \notin \mathcal{X}_\beta$ and $\alpha(x,x') = 1$ for $x, x' \in \mathcal{X}_\beta$, this can be rewritten

$$\tau(\psi) = \frac{1}{2\psi}\int_{\beta}^{\beta+1}\left[\int_{\min(\max(-\beta-1,x-\psi),-\beta)}^{-\beta}\mathrm{d}x' + \int_{\max(\beta,x-\psi)}^{\min(\beta+1,x+\psi)}\mathrm{d}x'\right]\mathrm{d}x.$$

At this stage it is possible to integrate out x', giving

$$\tau(\psi) = \frac{1}{2\psi}\left[\int_{\beta}^{\beta+1}(-\beta - \min(\max(-\beta-1, x-\psi), -\beta))\,\mathrm{d}x\right.$$

$$\left. + \int_{\beta}^{\beta+1}(\min(\beta+1, x+\psi) - \max(\beta, x-\psi))\,\mathrm{d}x\right].$$

To make progress with the computation the next step is to split the range of the two integrals into separate parts, so that in each sub-interval the minimum and maximums are resolved. For clarity at this stage we just assume $\psi > 0$, and only implement the full restriction $\psi > 2\beta$ once we have an explicit expression for $\tau(\psi)$. Splitting each integral into 3 sub-integrals we get,

$$\tau(\psi) = \frac{1}{2\psi}\left[\int_{\beta}^{\min(\max(\beta,\psi-\beta-1),\beta+1)}\mathrm{d}x + \int_{\min(\max(\beta,\psi-\beta-1),\beta+1)}^{\min(\max(\beta,\psi-\beta),\beta+1)}(\psi-\beta-x)\,\mathrm{d}x\right.$$

$$+ \int_{\min(\max(\beta,\psi-\beta),\beta+1)}^{\beta+1} 0\,\mathrm{d}x + \int_{\beta}^{\max(\beta,\min(\beta+\psi,\beta+1-\psi))}(x+\psi-\beta)\,\mathrm{d}x$$

$$+ \int_{\max(\beta,\min(\beta+\psi,\beta+1-\psi))}^{\min(\max(\beta+\psi,\beta+1-\psi),\beta+1)}(\mathbf{1}_{\{\psi>\frac{1}{2}\}} + 2\psi\mathbf{1}_{\{\psi<\frac{1}{2}\}})\,\mathrm{d}x$$

$$\left. + \int_{\min(\max(\beta+\psi,\beta+1-\psi),\beta+1)}^{\beta+1}(\beta+1+\psi-x)\,\mathrm{d}x\right].$$

From this expression, simple integration yields the inelegant solution

$$\begin{aligned}\tau(\psi) = \frac{1}{2\psi}\Big[& \{\min(\max(\beta, \psi-\beta-1), \beta+1) - \beta\} \\ & + (\psi-\beta)\{\min(\max(\beta, \psi-\beta), \beta+1) - \min(\max(\beta, \psi-\beta-1), \beta+1)\} \\ & + \frac{1}{2}\left\{[\min(\max(\beta, \psi-\beta-1), \beta+1)]^2 - [\min(\max(\beta, \psi-\beta), \beta+1)]^2\right\} \\ & + (\psi-\beta)\{\max(\beta, \min(\beta+\psi, \beta+1-\psi)) - \beta\} \\ & + \frac{1}{2}\left\{[\max(\beta, \min(\beta+\psi, \beta+1-\psi))]^2 - \beta^2\right\} \\ & + \mathbf{1}_{\{\psi>\frac{1}{2}\}}\{\min(\max(\beta+\psi, \beta+1-\psi), \beta+1) - \max(\beta, \min(\beta+\psi, \beta+1-\psi))\} \\ & + 2\psi\mathbf{1}_{\{\psi<\frac{1}{2}\}}\{\min(\max(\beta+\psi, \beta+1-\psi), \beta+1) - \max(\beta, \min(\beta+\psi, \beta+1-\psi))\} \\ & + (\beta+1+\psi)\{\beta+1-\min(\max(\beta+\psi, \beta+1-\psi), \beta+1)\} \\ & + \frac{1}{2}\left\{[\min(\max(\beta+\psi, \beta+1-\psi), \beta+1)]^2 - (\beta+1)^2\right\}\Big].\end{aligned} \tag{4.14}$$

We may now focus on the case of interest, namely when $\psi > 2\beta$. To further simplify the calculation we suppose also that we choose $\frac{1}{2} < \beta < 1$, which implies $\psi > 1$. Considering only the range of values of ψ that result in an irreducible MCMC sampler, it can be checked that equation 4.14 can be simplified to

$$\tau(\psi) = \begin{cases} \frac{1}{2\psi} + \left(\frac{\psi}{4} - \beta + \frac{\beta^2}{\psi}\right) & 2\beta \leq \psi < 2\beta+1 \\ 1 + \beta - \frac{2\beta}{\psi} - \frac{\beta^2}{\psi} - \frac{\psi}{4} & 2\beta+1 \leq \psi < 2\beta+2 \\ \frac{1}{\psi} & \psi \geq 2\beta+2. \end{cases} \tag{4.15}$$

Figure 4.1 shows the function $\tau(\psi)$ against ψ for π with $\beta = 3/4$. It is clear from this plot, that for this choice of target distribution π, the requirement that underlies the convergence of the proposed adaptive algorithm (i.e. $\frac{\partial\tau}{\partial\psi} \leq 0$ for all ψ) does not hold. In particular, by differentiating equation 4.15, it is possible to

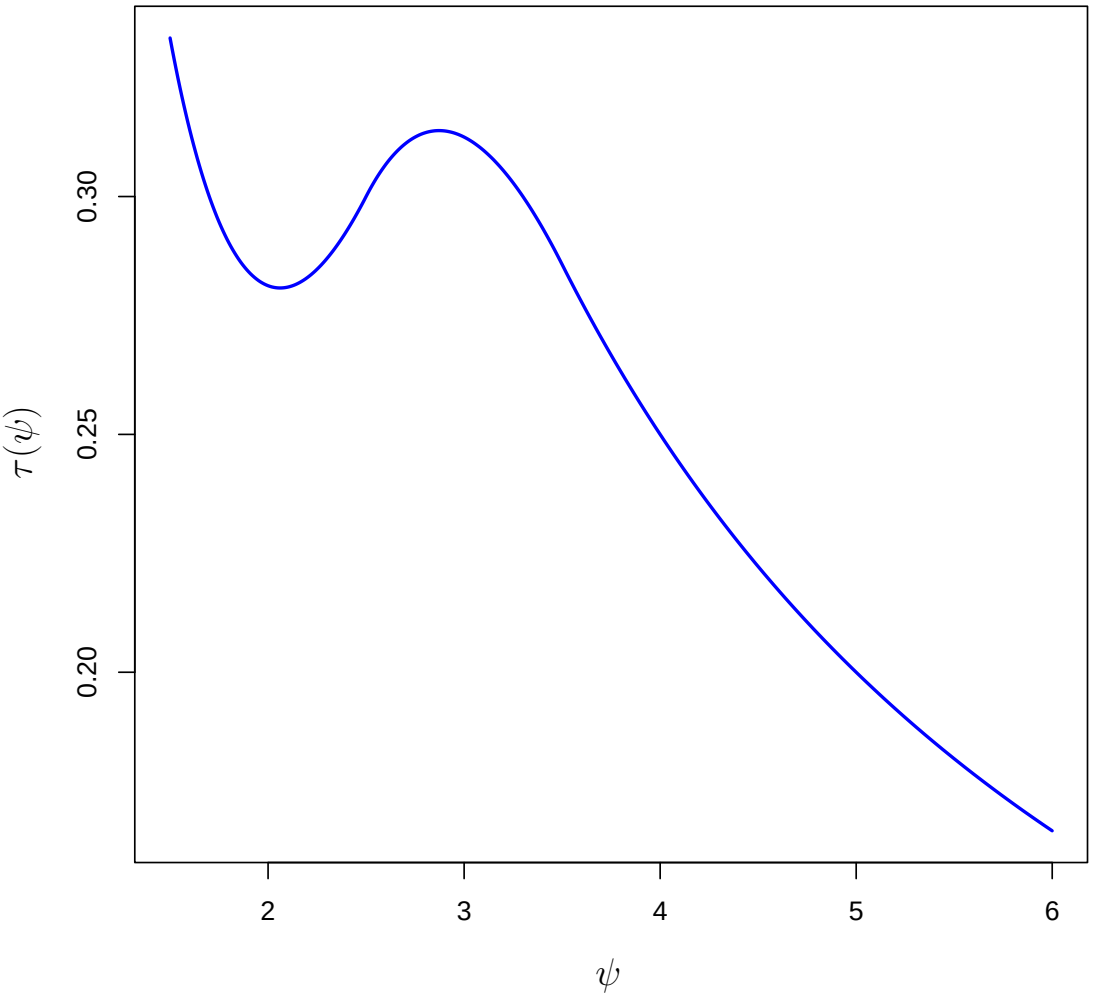

Figure 4.1: The function $\tau(\psi)$ plotted against ψ for $\beta = 3/4$.

show that for $2\beta < \sqrt{4\beta^2 + 2} < \psi < 2\sqrt{\beta^2 + 2\beta} < 2\beta + 2$, $\frac{\partial \tau}{\partial \psi} > 0$.

Although the example in this section deals with a modification of the AAP algorithm proposed by Atchadé and Rosenthal, we have shown that for the modified algorithm the assumption underlying convergence does not always hold. While this does not give us any direct insight into the case of the original AAP algorithm with Normal proposals, we can still take two important points from the results. Firstly, since we have presented a counter example for a very similar algorithm, we should be wary of the convergence results for the original algorithm until we can verify that the assumption holds for all target distributions. Secondly, we highlight that even for the simple case presented above, the calculations required to show the assumption does not hold are messy and involved. This suggests that for the original algorithm and for more general target distributions the case would be even more difficult.

In order to address these two final points, the following section presents an alternative approach to the theoretical study of the AAP algorithm. We return to the original AAP algorithm, but this time employ numerical simulations. In particular, we study the behaviour of the algorithm for two target different distributions for which we might expect the necessary assumption not to hold and thus the convergence to fail.

4.5 A Simulation Study for the AAP Algorithm

We study the behaviour of the algorithm when used with two simple one dimensional target distributions π_1 and π_2, specifically chosen as potential candidates for which the algorithm might fail. The first distribution π_1 is the distribution

π of the previous section whose density function is given by equation 4.13. The distribution is pictured in figure 4.4. The second of our chosen distributions π_2, is a mixture of two triangular distributions and has density given by

$$\pi_2(x) = \begin{cases} 1+\beta+x & x \in [-1-\beta, -\beta] \\ 1+\beta-x & x \in [\beta, 1+\beta] \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

An example of this distribution with $\beta = 1$ can be seen in figure 4.4 below.

In the previous section it was demonstrated that using a modified AAP algorithm with uniform proposals, the assumption underlying the convergence of the algorithm does not hold. Unsure of whether the assumption will hold for the AAP algorithm with Normal proposals we carry out the following simulation. Let G be a uniform 11×101 grid of (β, σ) pairs from $[0.05, 1.55] \times [0.01, 5.01]$. For each value in the grid and for each of the two target distributions π, we complete a simple Normal random walk Metropolis run of 1 million iterations. For each run we look at the average acceptance rate of the sampler as an empirical estimate of the expected acceptance probability.

At this stage no adaptation is being done and we are merely testing to see whether the assumption holds for 2 target distributions. If the assumption holds, for both target distributions and for each value of β we would expect the average acceptance probability to be a decreasing function of the variance $\psi = \sigma^2$ (or equivalently of the standard deviation σ).

Figure 4.2 shows the surface of the average acceptance probability as a function of β and σ, for target distribution π_1. A similar picture (not shown) arises when carrying out the simulation study for π_2. From this picture it is clear that

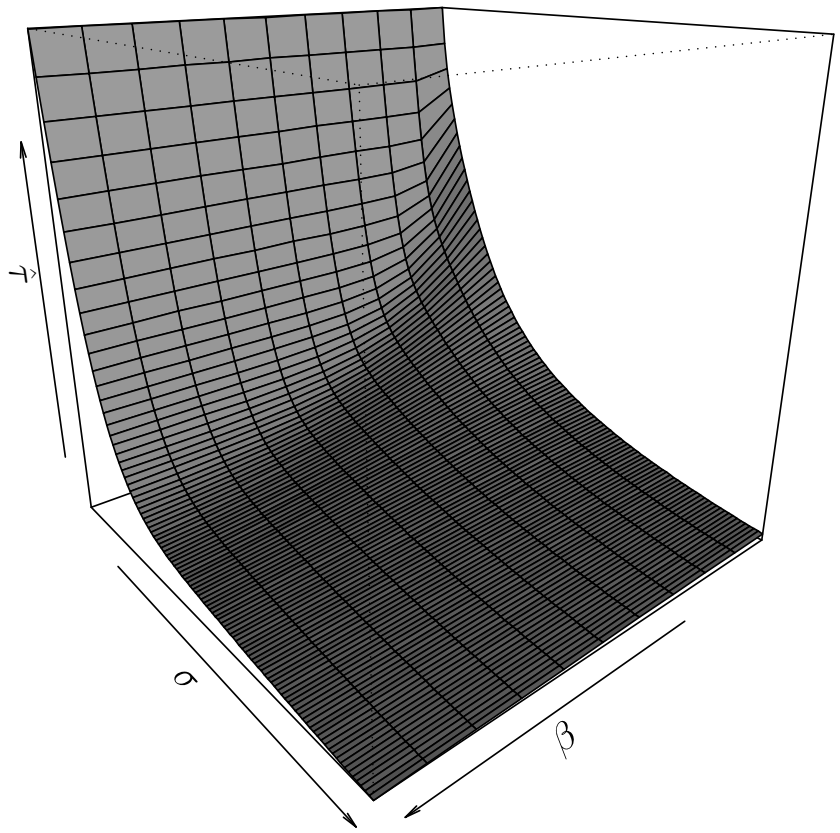

Figure 4.2: Graphical representation of the average acceptance probability $\hat{\tau}$ as a function of β and σ, for target distribution π_1.

for each value of β, the average acceptance probability is indeed a decreasing function of the standard deviation σ. This fact is seen even more clearly from the profile curves shown in figure 4.3 (this time the results shown are for π_2 and those for π_1 are omitted).

The above simulation provides us with evidence that when using Normal proposals the necessary assumption holds for the two target distributions under study. Given that the underlying assumption appears to hold, it is valid to next propose a simulation study of the adaptive algorithm in practice. For this study, we choose the value $\beta = 1$ and run the adaptive MCMC algorithm for 1 million iterations for each target distribution π_1 and π_2. We start the chain from a sample from the respective target distributions and begin adapting immediately. At each iteration we update the adaptive parameter $\psi = \sigma^2$ according to the AAP updating regime specified in equations 4.1 and 4.2.

The results of this simulation study can be seen in the various plots displayed in figure 4.4. Plots (a) and (b) appear to provide evidence that the chains resulting from the AAP algorithm are indeed ergodic for these two target distributions. Furthermore plots (c) to (e) demonstrate that the AAP algorithm appears to have converged, resulting in 'optimal' estimates of the spread parameters for the RWM. The final plot (f), shows the autocorrelation function for the x-chain over the last 10000 iterations of the run. This does not appear to demonstrate any striking differences from a usual non-adaptive MCMC run.

Despite the fact that π_1 and π_2 were specifically chosen as potentially troublesome target distributions, the AAP algorithm appears to work very well for these cases. The numerical results in this section cannot provide evidence that the

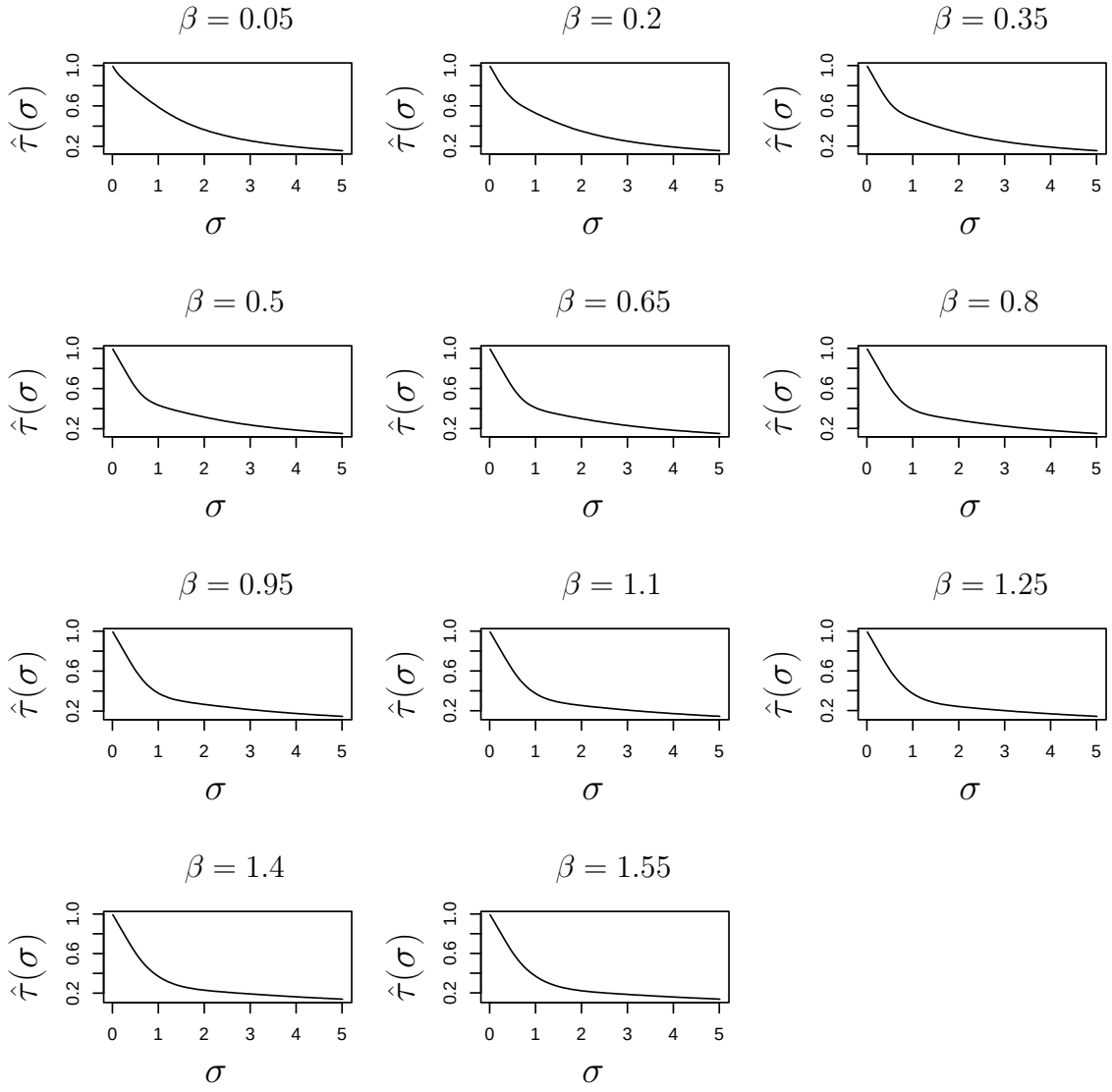

Figure 4.3: Profile curves of the average acceptance probability $\hat{\tau}(\sigma)$ against σ for a variety of β values, for target distribution π_2.

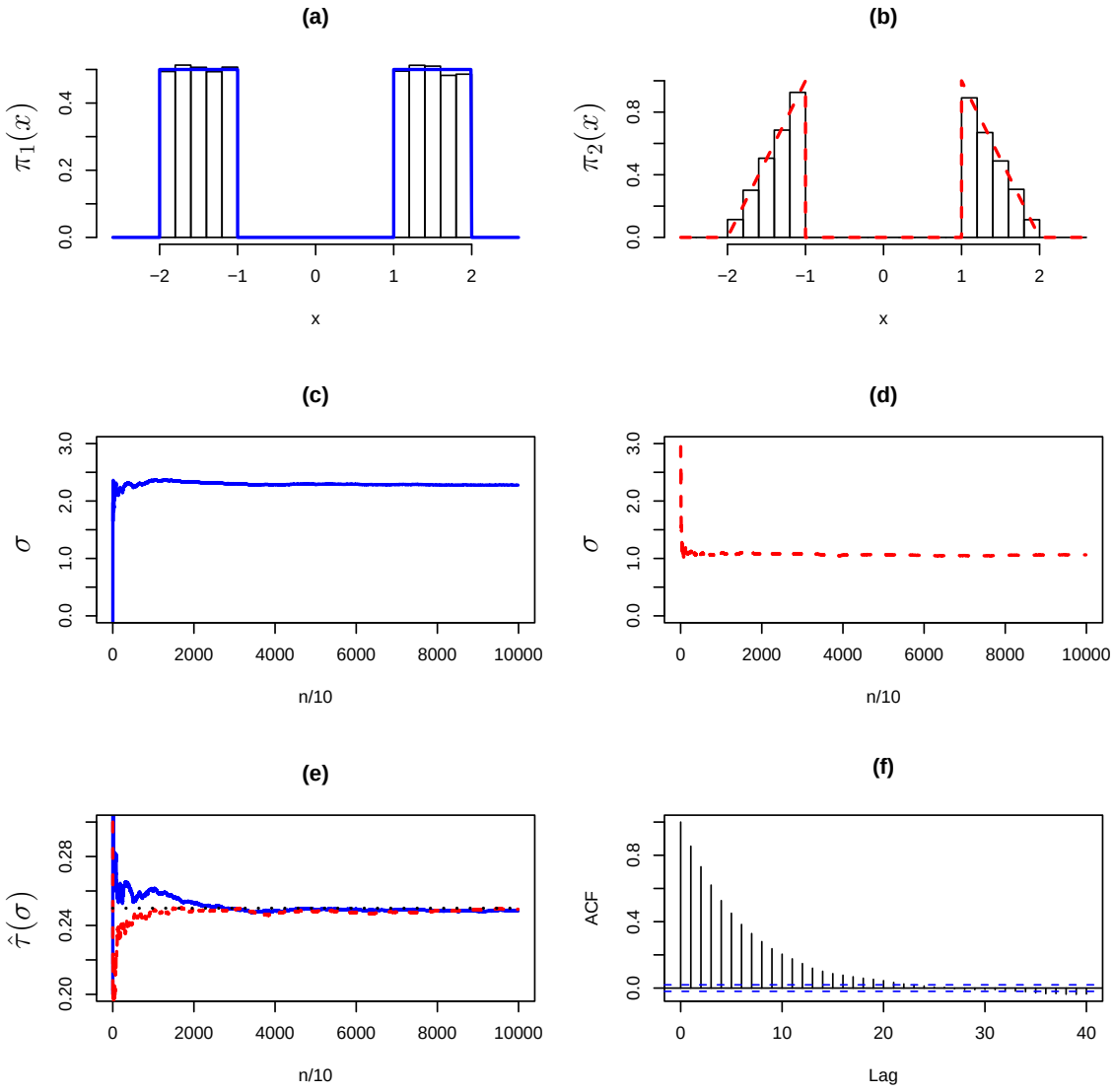

Figure 4.4: (a) Histogram of samples using the AAP algorithm with target distribution π_1 (solid blue line denotes target density); (b) Histogram of samples using the AAP algorithm with target distribution π_2 (dashed red line denotes target density); (c) Plot of adaptive parameter σ against time (sub-sampled every 1000 iterations), target distribution π_1; (d) Plot of adaptive parameter σ against time (sub-sampled every 1000 iterations), target distribution π_2; (e) Plot of average acceptance rate $\hat{\tau}(\sigma)$ against time (sub-sampled every 1000 iterations), blue solid line using π_1, red dashed line using π_2; and (f) autocorrelation function for ergodic averages of function $f(x) = x$, for last 10000 iterations (using target distribution π_1).

algorithm will converge regardless of target distribution. However, the fact that the desired performance was achieved for these two target distributions, must provide us with some confidence about the wide (if not universal) applicability of the AAP algorithm. Nonetheless, to move beyond such tentative conclusions we advocate further research in this area.

Before moving away from the AAP algorithm we mention briefly that contrary to the claims of Atchadé and Rosenthal (2003), our experiments showed that if we set $c = \sigma_0$ in equations 4.1 and 4.2 (as suggested by the authors) the algorithm is sensitive to the choice of σ_0. While the value of $\sigma_0 = 10$ that the authors use (and tentatively recommend) appears to produce good results for this problem, we found that changing the value to $\sigma_0 = 1$ seriously affects the rate of convergence of the algorithm. Repeated experiments with $c = \sigma_0 = 1$ showed that even after 10 million iterations the average acceptance probability had not converged to the target value.

Despite the slow convergence when σ_0 is too small, our experiments showed that this had no impact on the ergodicity of the resulting MCMC sample. However, one of the main aims of the AAP algorithm is the convergence of the RWM parameters towards optimal values and in certain scenarios this convergence may be of particular importance. In chapter 5 we introduce a context where the convergence is a particular reason for employing an adaptive algorithm.

4.6 Two Reversible Jump Adaptive Algorithms

We now move away from our discussion of previously proposed adaptive algorithms and introduce two new adaptive algorithms specifically designed

for RJMCMC samplers. For both adaptive strategies we delay examples of the algorithm in practice until section 5.5 in the next chapter. We begin by presenting details of an algorithm that adapts through regeneration, detailing the proposed method, the motivation behind it and a brief insight into why we are confident that the algorithm converges and results in an ergodic Markov chain. Having introduced this first adaptive method, we try to overcome the associated limitations by introducing an alternative diminishing adaptation algorithm. For the diminishing adaptation algorithm we prove a number of results that imply assumptions (**A1**), (**A2**), and (**A4**) of Andrieu *et al.* (2004). While we cannot guarantee the convergence of the adaptive parameters or the ergodicity of the resulting chains (since we do not imply that the algorithms satisfy assumptions (**A3**)), the theoretical results are an important step in this direction. Furthermore, the encouraging numerical results of our algorithm appear to provide evidence that indeed the chains are ergodic and that the algorithm does converge.

Let us first adopt the model jumping formulation for reversible jump problems (reviewed in Chapter 2) and write $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$, so that the state $\boldsymbol{x}$ is made up of a model index k and associated model parameters $\boldsymbol{\theta}_k$. The state space $\mathcal{X}$ can then be written $\mathcal{X} = \cup_k(\{k\} \times \boldsymbol{\Theta}_k)$. Furthermore, for the remainder of this chapter we suppose that there are only a finite number of models K_{max}, and denote the model space by $\mathcal{M} = \{1, 2, \ldots, K_{max}\}$. This requirement is frequently imposed for practical implementation of reversible jump algorithms. In order to simplify the following we introduce the following extra notation, $\mathcal{M}^{\leftarrow j} := \{1, 2, \ldots, K_{max} - j\} \subseteq \mathcal{M}$, where $j \in \mathcal{M}$.

Suppose now that the current state of the Markov chain is $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$. We

concentrate on reversible jump algorithms which proceed in the following manner. Begin by proposing a new model k' with probability, $\rho_{k,k'}$. Having selected a new model k', generate random numbers $\boldsymbol{u}$ from a probability distribution with known density $g_{k,k'}$, and propose a new parameter vector $\boldsymbol{\theta}'_{k'}$ given by $(\boldsymbol{\theta}'_{k'}, \boldsymbol{u}') = \boldsymbol{t}_{k,k'}(\boldsymbol{\theta}_k, \boldsymbol{u})$ where $\boldsymbol{t}_{k,k'}$ is some differentiable function and $\boldsymbol{u}'$ is the vector of random number values that would have to be generated from a known distribution $g_{k',k}$ in order to move in the reverse direction. The proposed state $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$ is then accepted as the new state of the chain with probability $\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\{1, A(\boldsymbol{x}, \boldsymbol{x}')\}$, where

$$A(\boldsymbol{x}, \boldsymbol{x}') = \frac{\pi(\boldsymbol{k'}, \boldsymbol{\theta}'_{k'})\rho_{k',k}g_{k',k}(\boldsymbol{u}')}{\pi(k, \boldsymbol{\theta}_k)\rho_{k,k'}g_{k,k'}(\boldsymbol{u})} \left| \frac{\partial(\boldsymbol{\theta}'_{k'}, \boldsymbol{u}')}{\partial(\boldsymbol{\theta}_k, \boldsymbol{u})} \right|.$$

We now focus on the case where the proposed new model k' is independent of the current model k, i.e $\rho_{k,k'} = \psi^{k'}$. Suppose further that for each value of k' it is somehow possible to choose the distribution $g_{k,k'}$ and the function $\boldsymbol{t}_{k,k'}$ to provide a good approximation of the conditional distribution $\pi(\boldsymbol{\theta}'_{k'}|k')$. (In the next chapter we introduce a generic reversible jump sampler that is designed for such purposes). Then, for each k', an appealing choice of the probability of jumping to model k' is the marginal posterior probability of model k', i.e. $\psi^{k'} = \pi(k') \;\; k' \in \mathcal{M}$.

In practice we are unlikely to know the posterior model probabilities $\pi(k)$. Indeed, for many reversible jump problems these might be exactly the quantities that we most wish to make inference about. Fortunately, such a situation is ideal for the application of adaptive sampling. If we can show that starting with any choice of initial model jumping probabilities $\{\psi_0^k : k \in \mathcal{M}\}$, we can iteratively update these probabilities, so that the resulting reversible jump chain remains ergodic and $\psi_m^k \to \pi(k)$ as $m \to \infty$ $(\forall k \in \mathcal{M})$, then we have a desirable adaptive

reversible jump sampler.

The first of our two adaptive methods, achieves adaptation through regeneration. Recall from section 4.2 that in order to apply adaptation by regeneration it must be possible to find a proper atom B in our state space $\mathcal{X}$. While Gilks *et al.* (1998) and Sahu and Zhigljavsky (2003) present methods for doing this for various Metropolis-Hastings algorithms, we move away from these approaches and develop a new approach which we now introduce. Our method has similarities to the perfect simulation methods described in Brooks *et al.* (2002) and Møller and Nicholls (2004). In addition, a similar adaptive approach is independently proposed by Brockwell and Kadane (2004).

In order to create a proper atom B for the generic reversible jump problem we augment the state space by introducing a null model $k = 0$, which consists of a single point mass on the point $\theta_0 = 0$. Having defined this new model, we create a Markov chain over $\tilde{\mathcal{X}}$, where $\tilde{\mathcal{X}} = \mathcal{X} \cup (\{0\} \times \{0\})$, with stationary distribution given by

$$\tilde{\pi}(k, \boldsymbol{\theta}_k) = \begin{cases} \frac{1}{K_{max}+1} & (k, \boldsymbol{\theta}_k) \in \{0\} \times \{0\}, \\ \frac{K_{max}}{K_{max}+1} \pi(k, \boldsymbol{\theta}_k) & (k, \boldsymbol{\theta}_k) \in \bigcup_{j \in \mathcal{M}} (j \times \boldsymbol{\Theta}_j), \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

An important feature of the distribution $\tilde{\pi}$ is that the conditional distribution $\tilde{\pi}(k, \boldsymbol{\theta}_k | k > 0)$ is just the original target distribution $\pi(k, \boldsymbol{\theta}_k)$. Thus, if we can create a Markov chain with $\tilde{\pi}$ as its stationary distribution, then conditioning on the event $k > 0$, (i.e. considering only those $\boldsymbol{X}_n$ such that $k_n > 0$) we can make inference about the distribution π. A Markov chain with $\tilde{\pi}$ as its equilibrium distribution can easily be achieved by modifying the above reversible jump algorithm. In particular, if the current state of the chain is $\boldsymbol{X} = (k, \boldsymbol{\theta}_k)$,

we propose a jump to model k' with probability $\tilde{\psi}^{k'}$, where

$$\tilde{\psi}^{k'} = \begin{cases} \frac{1}{K_{max}+1} & k' = 0 \\ \frac{K_{max}}{K_{max}+1}\psi^{k'} & k' \in \mathcal{M} \\ 0 & otherwise. \end{cases}$$

If a jump to the null model $k' = 0$, is proposed then we deterministically set $\boldsymbol{\theta}'_{k'} = 0$. For jumps to other values of k' we use the above framework, generating random numbers $\boldsymbol{u}$ from a distribution $g_{k,k'}$ and using transformations $\boldsymbol{t}_{k,k'}$. We note that since the null model has dimension zero, if $k = 0$ and $k' \neq 0$, it is necessary to generate a random vector $\boldsymbol{u}$ of dimension $n_{k'}$. The acceptance probabilities remain unchanged for jumps where both k and k' are in $\mathcal{M}$. If the chain is currently in the null model $k = 0$, the proposed new state $\boldsymbol{X}' = (k', \boldsymbol{\theta}'_{k'})$ is accepted with probability 1 if $k' = 0$ and probability $\min\{1, A((0,0), \boldsymbol{x}')\}$ if $k' \in \mathcal{M}$ where,

$$A((0,0), \boldsymbol{x}') = \pi(k, \boldsymbol{\theta}_k) \times \frac{1}{\psi^{k'} g_{0,k'}(\boldsymbol{u})} \times |J|,$$

and again J is the Jacobian of the transformation $\boldsymbol{t}_{0,k'}$. As for the standard reversible jump algorithm, the reverse move is accepted with probability $\min\{1, A(\boldsymbol{x}', (0,0))\}$, where $A(\boldsymbol{x}', (0,0)) = [A((0,0), \boldsymbol{x}')]^{-1}$.

The null model $k = 0$ is a proper atom of the modified Markov chain. Every time the chain visits the null model, the chain regenerates and the transition kernel can be modified using the entire history of the chain without contravening the ergodicity of the resulting output.

By creating this augmented Markov chain, we have a natural vehicle for applying the regeneration principle for adapting the transition kernel of the chain. However, we have not yet considered specifically how we adapt these proposals

to achieve the original aim of making ψ^k as similar to $\pi(k)$ as possible. Recall that at the i^{th} regeneration time T_i we can use the whole history of the chain $(\boldsymbol{X}_0, \ldots, \boldsymbol{X}_{T_i-1})$ to define the new kernel $\mathcal{K}_i$. As the Markov chain is ergodic and has stationary distribution $\tilde{\pi}$, it is reasonable to expect that if we condition on $k > 0$ then for each $j \in \mathcal{M}^{\leftarrow 1}$, the estimate $p_n^j = \frac{1}{N_n} \sum_{i=1}^{T_n-1} \mathbf{1}_{\{k_i=j\}}$ tends to $\pi(k)$ as $n \to \infty$, where $N_n = \sum_{i=1}^{T_n-1} \mathbf{1}_{\{k_i>0\}}$ is the total number of iterations minus the number of visits to the null model. This motivates the following adaptive algorithm, which we term algorithm A.

Define the initial transition kernel $\mathcal{K}_0$ as described for the augmented reversible jump algorithm above, with model proposal probabilities $\psi_0^j = \frac{1}{K_{max}}$ for all $j \in \mathcal{M}$. Start the Markov chain in the null model and proceed using the adaptive transition kernel that is updated at every return to the null model according to the following rule. At the i^{th} regeneration time T_i, define the new transition kernel as being identical to $\mathcal{K}_0$, except for the model proposal probabilities, which become

$$\psi_n^j = \psi_{n-1}^j + \frac{1}{N_n} \left[\sum_{i=T_{n-1}}^{T_n-1} \mathbf{1}_{\{k_i=j\}} - (N_n - N_{n-1})\psi_{n-1}^j \right], \quad \text{for } j \in \mathcal{M}. \tag{4.18}$$

Despite having no formal proof of the ergodicity of our sampler, the numerical evidence (see chapter 5) seems to support this case. If indeed the resulting chain is ergodic, it is clear that the adaptive proposal parameters ψ^j will automatically tend towards the model probabilities $\pi(j)$ as required, thus convergence is guaranteed.

Although algorithm A appears to provide an appealing way to use adaptation in a reversible jump setting, the approach suffers from a substantial limitation. In many instances it may not be possible to implement the sampler. For standard

reversible jump algorithms, the target distribution $\pi(k, \boldsymbol{\theta}_k)$ is only required to be known up to a constant of proportionality. However, for our augmented sampler above, which samples from the adjusted target distribution $\tilde{\pi}(k, \boldsymbol{\theta}_k)$ this requirement is not sufficient. Proposed moves to (or from) the null model are accepted with a probability that only has π appearing in either the denominator (or numerator). Thus the constant of proportionality of π does not cancel, and therefore it must be known for the chain to have the correct stationary distribution. In many real instances, particularly when π is a Bayesian posterior distribution, this constant of proportionality is not known and cannot be easily computed.

The severity of requiring the normalising constant will prevent the method from being applicable for all but the most simple problems. It may be possible to develop ways in which we might employ various approximations to the normalising constant, perhaps even doing so in an adaptive fashion. This possibility might warrant further research, although we note that the stationary distribution of the resulting sampler would then only be an approximation to the target distribution. Rather than pursue this approach here, we now introduce a new adaptive reversible jump algorithm, based on the theory behind diminishing adaptation. This algorithm does not require an adjustment of the target distribution and thus calculation of the constant of proportionality remains unnecessary.

Let the Markov chain $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \ldots$ be the result of the algorithm where all details are as for the non-adaptive reversible jump algorithm with the exception that at iteration $n+1$, if the current state of the chain is $\boldsymbol{X}_n = (k, \boldsymbol{\theta}_k)_n$, a move to model $k' \in \mathcal{M}^{\leftarrow 1}$ is attempted with probability $\psi_n^{k'}$, and a move to model

K_{max} is attempted with probability $1 - \sum_{j=1}^{K_{max}-1} \psi_n^j$ (i.e. ψ^j is replaced by ψ_n^j, for $j \in \mathcal{M}^{\leftarrow 1}$ and $\psi^{K_{max}}$ is replaced by $1 - \sum_{j=1}^{K_{max}-1} \psi_n^j$).

Our diminishing adaptation algorithm, which we call algorithm B, is designed to be a specific example of the general adaptive algorithm presented in Andrieu *et al.* (2004) and Andrieu and Moulines (2004). Restricting attention to the case where $\psi^j > 0$ for $j = 1, 2, \ldots, K_{max} - 1$ and $0 < \sum_{j=1}^{K_{max}-1} \psi^j < 1$, we define the adaptive parameter space Ψ as the open subset of $\mathbb{R}^{K_{max}-1}$ within these boundaries. We define a family $\{\mathcal{C}_l : l \in \mathbb{N}\}$ of compact subsets of Ψ, such that $\mathcal{C}_l$ is the closed set defined by the equations $\psi^j \geq \frac{1}{10(l+1)}$, for $j = 1, 2, \ldots, K_{max}-1$, and $\frac{1}{10(l+1)} \leq \sum_{j=1}^{K_{max}-1} \psi^j \leq 1 - \frac{1}{10(l+1)}$. Clearly we have

$$\bigcup_{l \geq 0} \mathcal{C}_l = \Psi \text{ and } \mathcal{C}_l \subset \text{int}(\mathcal{C}_{l+1}).$$

Having defined such subsets we set $\kappa_0 = 0$ and choose some initial proposal probabilities $\boldsymbol{\psi}_0 = (\psi_0^1, \psi_0^2, \ldots, \psi_0^{K_{max}-1}) \in \Psi_0 = \mathcal{C}_0$. At iteration $n + 1 \geq 1$, the adaptive proposal parameter vector $\boldsymbol{\psi_n} = (\psi_n^1, \psi_n^2, \ldots, \psi_n^{K_{max}-1})$ is first used to determine the new state $X_{n+1} = (k, \boldsymbol{\theta}_k)_{n+1}$ of the Markov chain. Next, the adaptive parameters are then themselves updated according to the following iterative relationships,

$$\tilde{\psi}_{n+1}^j = \psi_n^j + \left(\frac{1}{n+2}\right)^{(2/3)} (\mathbf{1}_{\{k_{n+1}=j\}} - \psi_n^j) \text{ for } j \in \mathcal{M}^{\leftarrow 1}; \tag{4.19}$$

$$\boldsymbol{\psi}_{n+1} = \begin{cases} \tilde{\boldsymbol{\psi}}_{n+1} & \tilde{\boldsymbol{\psi}}_{n+1} \in \Psi_n \text{ and } |\tilde{\boldsymbol{\psi}}_{n+1} - \boldsymbol{\psi}_n| \leq \left(\frac{1}{n+2}\right)^{0.51} \\ \boldsymbol{\psi}_0 & \text{otherwise;} \end{cases} \tag{4.20}$$

$$\kappa_{n+1} = \begin{cases} \kappa_n & \tilde{\boldsymbol{\psi}}_{n+1} \in \Psi_n \text{ and } |\tilde{\boldsymbol{\psi}}_{n+1} - \boldsymbol{\psi}_n| \leq \left(\frac{1}{n+2}\right)^{0.51} \\ \kappa_n + 1 & \text{otherwise; and} \end{cases} \tag{4.21}$$

$$\Psi_{n+1} = \mathcal{C}_{\kappa_{n+1}}. \tag{4.22}$$

To remain consistent throughout this thesis, the notation of algorithm B is a

modification of that for the general algorithm in Andrieu *et al.* (2004) so that $\boldsymbol{\psi}$, Ψ and $\{\mathcal{C}_l,\ l \geq 0\}$ replace the terms θ, $\mathcal{K}$ and $\{\mathcal{K}_q,\ q \geq 0\}$ respectively. Our choice of values $\gamma_n = \left(\frac{1}{n+1}\right)^{(2/3)}$ and $\epsilon_n = \left(\frac{1}{n+1}\right)^{0.51}$ is made to satisfy the assumption (**A4**) of Andrieu *et al.* (see section 4.2).

In order to apply the convergence results of these papers to our algorithms it remains necessary to show that our choice of the function

$$\boldsymbol{H}(\boldsymbol{\psi}, \boldsymbol{x}) = (\mathbf{1}_{\{k=1\}} - \psi^1, \mathbf{1}_{\{k=2\}} - \psi^2, \ldots, \mathbf{1}_{\{k=K_{max}-1\}} - \psi^{K_{max}-1}) \tag{4.23}$$

satisfies the remaining assumptions that these results require. For convenience, we will denote $\psi^{K_{max}} = 1 - \sum_{j=1}^{K_{max}-1} \psi^j$, but emphasise that this is *not* a free parameter like the other ψ^j parameters.

The mean field, $\boldsymbol{h}(\boldsymbol{\psi})$, is given by

$$\begin{aligned}
\boldsymbol{h}(\boldsymbol{\psi}) &:= \sum_{k=1}^{K_{max}} \int_{\boldsymbol{\theta}_k \in \boldsymbol{\Theta}_k} \boldsymbol{H}(\boldsymbol{\psi}, (k, \boldsymbol{\theta}_k)) \pi(k, \boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k, \\
&= \sum_{k=1}^{K_{max}} \int_{\boldsymbol{\theta}_k \in \boldsymbol{\Theta}_k} (\mathbf{1}_{\{k=1\}} - \psi^1, \ldots, \mathbf{1}_{\{k=K_{max}-1\}} - \psi^{K_{max}-1}) \pi(k, \boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k, \\
&= \sum_{k=1}^{K_{max}} (\mathbf{1}_{\{k=1\}} - \psi^1, \ldots, \mathbf{1}_{\{k=K_{max}-1\}} - \psi^{K_{max}-1}) \pi(k), \\
&= (\pi^1 - \psi^1, \pi^2 - \psi^2, \ldots, \pi^{K_{max}-1} - \psi^{K_{max}-1}),
\end{aligned} \tag{4.24}$$

where $\pi^j = \pi(j)$, is the marginal posterior probability of model j, for $j \in \mathcal{M}^{\leftarrow 1}$. The following proposition demonstrates that for a suitable choice of function w, the adaptive procedure B satisfies the assumptions (**A1**) specified by Andrieu *et al.* (2004) (and detailed in section 4.2).

Proposition 4.3. *Define the function* $w : \Psi \to [0, \infty)$ *by*

$$w(\boldsymbol{\psi}) = \sum_{j=1}^{K_{max}} \log\left(\frac{\pi^j}{\psi^j}\right)\pi^j.$$

Then, for all $\boldsymbol{\psi} \in \Psi$, $\langle \nabla w(\boldsymbol{\psi}), h(\boldsymbol{\psi})\rangle \leq 0$ *with equality if and only if* $\psi^j = \pi^j$ *for all* $j \in \mathcal{M}^{\leftarrow 1}$.

Proof. Recalling that $\psi^{K_{max}} = 1 - \sum_{j=1}^{K_{max}-1} \psi^j$, simple differentiation yields

$$\nabla w(\boldsymbol{\psi}) = \left(\frac{\pi^{K_{max}}}{\psi^{K_{max}}} - \frac{\pi^1}{\psi^1}, \frac{\pi^{K_{max}}}{\psi^{K_{max}}} - \frac{\pi^2}{\psi^2}, \ldots, \frac{\pi^{K_{max}}}{\psi^{K_{max}}} - \frac{\pi^{K_{max}-1}}{\psi^{K_{max}-1}}\right).$$

This gives

$$\begin{aligned}
\langle \nabla w(\boldsymbol{\psi}), \boldsymbol{h}(\boldsymbol{\psi})\rangle &= -\sum_{j=1}^{K_{max}-1}\left[(\pi^j - \psi^j)\left(\frac{\pi^j}{\psi^j} - \frac{\pi^{K_{max}}}{\psi^{K_{max}}}\right)\right] \\
&= -\left(\sum_{j=1}^{K_{max}-1}\left[(\pi^j - \psi^j)\left(\frac{\pi^j}{\psi^j}\right)\right] - \frac{\pi^{K^{max}}}{\psi^{K_{max}}}[\psi^{K_{max}} - \pi^{K_{max}}]\right) \\
&= -\sum_{j=1}^{K_{max}}\left[(\pi^j - \psi^j)\left(\frac{\pi^j}{\psi^j}\right)\right] \\
&= -\left(\sum_{j=1}^{K_{max}}\frac{(\pi^j)^2}{\psi^j} - 1\right) \\
&= -\left(\mathbb{E}_\psi\left[\left(\frac{\pi}{\psi}\right)^2\right] - 1\right) \\
&= -\left(\mathrm{Var}_\psi\left[\frac{\pi}{\psi}\right] + \left(\mathbb{E}_\psi\left[\frac{\pi}{\psi}\right]\right)^2 - 1\right) \\
&= -\mathrm{Var}_\psi\left[\frac{\pi}{\psi}\right] \leq 0.
\end{aligned}$$

It is also clear that the left hand side equals zero iff $\psi^j = \pi^j$ for all $j \in \mathcal{M}^{\leftarrow 1}$ which concludes the proof. □

An immediate consequence of proposition 4.3, is that with this choice of $w(\boldsymbol{\psi})$, the assumptions (**A1**) of Andrieu *et al.* are met, with any choice of $M_0 > 0$ and $M_1 \in (M_0, \infty)$. The next proposition confirms that the adaptive algorithm satisfies assumption (**A2**) in Andrieu *et al.*.

Proposition 4.4. *Let the function $\boldsymbol{H}(\boldsymbol{\psi}, (k, \boldsymbol{\theta}_k))$ be as in equation 4.23. For any value of $\boldsymbol{\psi} \in \Psi$, define $\mathcal{K}_{\boldsymbol{\psi}}$ to be the transition kernel corresponding to the RJMCMC algorithm with the model jumping proposal vector given by $\boldsymbol{\psi}$. Then,*

1. *for all $\boldsymbol{\psi} \in \Psi$, $\mathcal{K}_{\boldsymbol{\psi}}$ has the stationary distribution π, i.e. $\pi\mathcal{K}_{\boldsymbol{\psi}} = \pi$; and*
2. *for all $\boldsymbol{\psi} \in \Psi$, $\tilde{H}(\boldsymbol{\psi}) := \sum_{k \in \mathcal{M}} \int_{\boldsymbol{\Theta}_k} |\boldsymbol{H}(\boldsymbol{\psi}, (k, \boldsymbol{\theta}_k))| \pi(k, \boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k < \infty$.*

Proof.

1. By design, the reversible jump algorithm is reversible and has π as its stationary distribution (see e.g. Green, 1995).
2. It is easily seen that,

$$\begin{aligned} \tilde{H}(\boldsymbol{\psi}) &\leq \sum_{k \in \mathcal{M}} \pi(k) \int_{\boldsymbol{\Theta}_k} \left[\sum_{j=1}^{K_{max}-1} |\mathbf{1}_{\{k=j\}} - \psi^j| \right] \pi(\boldsymbol{\theta}_k | k) \mathrm{d}\boldsymbol{\theta}_k \\ &< \sum_{k \in \mathcal{M}} \pi(k) \int_{\boldsymbol{\Theta}_k} [K_{max} - 1] \, \pi(\boldsymbol{\theta}_k | k) \mathrm{d}\boldsymbol{\theta}_k \\ &= K_{max} - 1 < \infty \, . \end{aligned}$$

□

The next stage in studying the convergence of the chains resulting from our adaptive algorithm B is to demonstrate that assumptions (**A3**) are satisfied. Furthermore, if these assumptions can be met, then from the results of Andrieu and Moulines (2004), the resulting chain $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ would be known to be ergodic. As discussed in section 4.2, due to the difficulty of demonstrating assumptions (**A3**) directly, we might instead hope to show that our algorithm meets the alternative and more easily verifiable drift conditions (**DRI**).

Unfortunately, taking this next step for the adaptive reversible jump chain is not easy. The problem is that as indicated in section 4.2, demonstrating such conditions hold is equivalent to demonstrating the geometric ergodicity of a relatively general reversible jump algorithm, with general stationary distribution π. Making progress in this direction remains an area that requires considerable

future work. Any study into this area would be likely to require several technical results and we do not explore this avenue of research. Rather, we appeal to the results in section 5.5 to support our belief that indeed such an adaptive scheme does have the necessary ergodicity properties, and provide incentive for future studies to help us better understand such algorithms.

4.7 Conclusions and Improvements

The area of adaptive sampling is a fast growing discipline within theoretical and computational statistics. While the review of the area in the early sections of the chapter is not intended to be exhaustive, it is hoped that we have provided a basic summary of the main results and papers in the area. Perhaps more importantly, we hope to have demonstrated the large amounts of further research required in the area. Such efforts are needed to explore the necessity of the sufficient conditions that are currently used for theoretical results, with the ultimate aim of providing more easily verifiable conditions. Just as we wish to take the RJMCMC sampler out of the domain of the MCMC expert, so we must wish to take the adaptive algorithm into a position where non-experts can readily apply such tools. Indeed, to achieve the first of these aims we must not underestimate the importance of the latter.

This chapter has concentrated in particular on the AAP algorithm proposed by Atchadé and Rosenthal. The research presented in sections 4.3 to 4.5 demonstrates the considerable and involved nature of a study into the theoretical properties of such algorithms. While little progress was made in checking the assumption underlying the convergence of the AAP algorithm, we hope that further research will allow the validity of the assumption to be ascertained either

way. If this is not possible, conditions for those target distributions for which the algorithm will converge would be the next best solution. Without further evidence, the potential user should remain cautious.

In spite of our call for caution, the numerical results that we present in section 4.5 and our experience to date (wherein we have always observed that the algorithm has behaved as expected, see for example chapter 5) suggest that the underlying assumption does indeed hold for at least most reasonable target distributions. More importantly, since the resulting samples appear to be ergodic, the algorithm remains a valid inferential tool.

Although checking the theoretical properties of new adaptive samplers remains a difficult and challenging task, the creation and implementation of intuitive new samplers, is relatively easy. Once the need for an adaptive sampler has been identified, the design of an algorithm to achieve this aim is not difficult to achieve. Once the design stage is complete, it is easy to look at the algorithm in practice and ascertain whether it is performing as desired. This is the approach that motivated our research in section 4.6.

As mentioned at the beginning of section 4.6, as we do not prove that assumption (**A3**) holds, we have no guarantee of the theoretical convergence or ergodicity properties of our diminishing adaptation reversible jump algorithm. However, by demonstrating that assumptions (**A1**) and (**A2**) hold and noting that the algorithm is designed to satisfy (**A4**), it is clear that we have made progress towards this goal. Moreover, the compelling practical evidence that we present in chapter 5 provides the incentive for further research into this area.

We close this chapter by noting that for a truly automatic reversible jump sampler the ideals behind adaptive sampling must be embraced. In the next chapter we describe our work in this direction, showing how we harness the usefulness of adaptive samplers to work towards our ultimate aim of an automatic reversible jump sampler.

Chapter 5

Towards Automatic Reversible Jump MCMC

In this chapter we bring together some of the ideas discussed up until this point and present our research into an automatic RJMCMC sampler. The chapter acts as a focal point for the thesis, with the work within best reflecting the thesis title. However, the chapter does not stand alone and in particular we adopt many of the adaptive sampling approaches presented in chapter 4.

5.1 Introduction

In order to exploit the potential of the reversible jump technique, scientists must be able to implement reversible jump samplers with minimum input or specialist MCMC knowledge. As such, the quest to produce a generic, automatic RJMCMC sampler remains an important goal. This chapter is devoted to extending the recent research in this direction, taking advantage of adaptive tools. The newly proven properties of these adaptive methods, discussed in detail in chapter 4, make the techniques ideal candidates for inclusion within an automatic sampler.

The main criteria which we feel are important for achieving a successful automatic sampler are the ease of use of the sampler and its broad applicability. Importantly, we should not expect to develop a sampler that performs uniformly well for all problems. In some problems the state space of our stochastic model will be considerably more complicated than that in others. This is borne out by the fact that in some cases (for example see the image analysis problems in Al-Awadhi *et al.*, 2004), even MCMC samplers that are designed by specialist statisticians for the specific problem perform very badly. For such problems it is unrealistic to expect a sampler that has been designed for broad applicability across many problems to perform as well as one that has been carefully developed and tuned for the specific problem considered. Nonetheless, we do recognise that any tool that we develop must be useful in practice. A sampler that takes a prohibitively long time to run or that produces erratic results is perhaps more dangerous than no automatic sampler at all.

The generic model jumping formulation of a reversible jump problem (discussed in chapter 2) means that construction of an automatic sampler that will perform well on a wide range of problems is realistic. In the remainder of this chapter we introduce and discuss our steps in this direction. We begin in section 5.2 by reviewing the first steps towards an automatic sampler introduced by Green (2003). The review provides motivation for a significant extension of this sampler. Our new extended sampler, which we call the *AutoMix* sampler, is introduced in section 5.3.1. In sections 5.3.2 and 5.3.3 we review some of the components that are required for the AutoMix sampler. We then present the full AutoMix algorithm before applying the sampler to some examples in section 5.5 and discussing potential improvements in section 5.6.

5.2 Automatic Reversible Jump MCMC: A Review

The consideration of automatic reversible jump techniques is a relatively new idea and therefore literature on the subject is limited. In chapter 2 we noted the recent work by Brooks *et al.* (2003b), on order methods and the saturated state space approach to reversible jump. To develop successful automatic samplers we must first understand how to design efficient proposals. This is the main issue that the methods introduced in Brooks *et al.* attempt to address and it is clear from this work that these authors recognise the importance of increasing the ease of RJMCMC implementation.

Brooks *et al.* make some important steps in the direction of automatic reversible jump MCMC, for example advocating the use of order methods to optimally scale the random number proposal distributions g. While these methods offer exciting progress, they still contain limitations. One such example is the assumption that the transformation function $\boldsymbol{t}$ is assumed fixed. This issue requires further research before automatic samplers based upon such approaches could be envisaged.

This section concentrates on the review of an alternative automatic sampler proposed by Green (2003). We refer to Green's sampler as the *AutoRJ* sampler throughout this chapter. In common with Brooks *et al.*, Green recognises the potential of achieving a more automated approach to sampler design but employs very different techniques to make progress towards this goal. Despite the simplicity of the AutoRJ sampler, the performance demonstrated for the few real examples that the author considers is surprisingly good. The AutoMix sampler is heavily based upon Green's AutoRJ sampler but includes considerable

modification in order to turn it into a successful automatic tool.

In order to introduce Green's sampler we return to the generic model jumping problem of section 2.3. For each model k suppose we are given a fixed n_k-vector $\boldsymbol{\mu}_k$, and a fixed $n_k \times n_k$-matrix B_k, where $\boldsymbol{\mu}_k$ is some rough estimate of the mean of $\boldsymbol{\theta}_k$ and $B_k B_k^T$ is some rough estimate of the covariance matrix of $\boldsymbol{\theta}_k$. Suppose the Markov chain is currently in state $(k, \boldsymbol{\theta}_k)$. The AutoRJ sampler proceeds by first choosing a new model k' with probability $\rho_{k,k'}$. Given that we have proposed a move to model k', our proposed new parameter state $\boldsymbol{\theta}'_{k'}$ depends upon the dimension $n_{k'}$ as follows:

$$\boldsymbol{\theta}'_{k'} = \begin{cases} \boldsymbol{\mu}_{k'} + B_{k'}[R_{k,k'} B_k^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k)]_1^{n_{k'}} & n_{k'} < n_k \\ \boldsymbol{\mu}_{k'} + B_{k'} R_{k,k'} B_k^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k) & n_{k'} = n_k \\ \boldsymbol{\mu}_{k'} + B_{k'} R_{k,k'} \begin{pmatrix} B_k^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k) \\ \boldsymbol{u} \end{pmatrix} & n_{k'} > n_k \end{cases} . \tag{5.1}$$

Here $[\ldots]_1^m$ denotes the first m components of a vector, $\boldsymbol{u}$ is an $(n_{k'} - n_k)$-vector of random numbers drawn from density $g_{k,k'}$, and $R_{k,k'}$ is an orthogonal matrix of order $\max\{n_k, n_{k'}\}$, satisfying $R_{k,k'} = R_{k',k}^T$, where $R_{k',k}$ is the orthogonal matrix used for a proposal from model k' to model k. Although any density $g_{k,k'}$ is possible, Green suggests that $\boldsymbol{u}$ are independent standard Normal or Student t distributed random variables. The motivation behind the inclusion of the matrix $R_{k,k'}$ is that by taking $R_{k,k'}$ as a random permutation matrix, moves to models with lower dimensions are no longer deterministic. Alternatively, including a fixed orthogonal matrix $R_{k,k'}$ might improve the acceptance rate if some of the conditional distributions $\pi(\boldsymbol{\theta}_k|k)$ are skewed. We observe, however, that for nested models, including $R_{k,k'}$ may not be a good idea.

As shown by Green, denoting $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ and $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$, the acceptance

probability corresponding to this proposal is given by $\min\{1, A(\boldsymbol{x}, \boldsymbol{x}')\}$, where $A(\boldsymbol{x}, \boldsymbol{x}')$ is given by

$$A(\boldsymbol{x}, \boldsymbol{x}') = \frac{\pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}|B_{k'}|}{\pi(k, \boldsymbol{\theta}_k)\rho_{k,k'}|B_k|} G_{k,k'}(\boldsymbol{u}) \ .$$

Here $|B_{k'}|/|B_k|$ is the Jacobian, and $G_{k,k'}(\boldsymbol{u})$ is given by

$$G_{k,k'}(\boldsymbol{u}) = \begin{cases} g_{k,k'}(\boldsymbol{u}) & n_{k'} < n_k \\ 1 & n_{k'} = n_k \\ [g_{k,k'}(\boldsymbol{u})]^{-1} & n_{k'} > n_k \end{cases} \ .$$

Since the matrices $R_{k,k'}$ and $R_{k',k}$ are orthogonal they each have a determinant of ± 1 and so make no contribution in the acceptance probability.

Green motivates the sampler by considering the special case where for each model k, $\pi(\boldsymbol{\theta}_k|k)$ is a Normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $B_k B_k^T$. If also the random numbers $\boldsymbol{u}$ are standard normals then $A(\boldsymbol{x}, \boldsymbol{x}')$ would simplify to

$$A(\boldsymbol{x}, \boldsymbol{x}') = \frac{\pi(k')\rho_{k',k}}{\pi(k)\rho_{k,k'}} \ . \tag{5.2}$$

This corresponds exactly to the acceptance probability of the standard MCMC algorithm with proposal $\rho_{k,k'}$ that explores the model space $\mathcal{M}$. Green observes that if also $\rho_{k,k'} = \pi(k')$ then the proposals would automatically be in detailed balance and there would be no need to compute the acceptance probability, as all moves would always be accepted.

In the more general case, when the conditionals are not Normal, the AutoRJ sampler uses proposals that approximate the distributions $\pi(\boldsymbol{\theta}_k|k)$ by roughly matching the first two moments. Godsill (2003) observes the intuitive appeal of proposals that approximate $\pi(\boldsymbol{\theta}_k|k)$ and notes that using such proposals agrees

with the ideas behind the higher order methods of Brooks *et al.* (2003b).

Godsill also discusses a similar sampler whereby the new proposed state vector is given by $\boldsymbol{\theta}'_{k'} = \boldsymbol{\mu}_{k'} + B_{k'}\boldsymbol{v}$ where $\boldsymbol{v}$ is an $n_{k'}$ vector of random numbers. The difference here is that the current state $\boldsymbol{\theta}_k$ plays no role in the proposed new state. The merits of such a scheme are discussed, and Godsill highlights that in the Gaussian special case, where the conditionals are Normal and $\boldsymbol{v}$ are standard Normal random variables, the method is identical to Green's and the acceptance ratio is exactly equal to equation 5.2.

An important feature of the AutoRJ sampler is the need for a vector $\boldsymbol{\mu}_k$ and matrix B_k to be available for each value of k. Green suggests that this information can be provided by the user at the run time of the sampler. Alternatively, the AutoRJ sampler provides the facility for a more automatic approach, avoiding the need for the user to be able to estimate these parameters. If no estimates are provided by the user, the sampler automatically carries out initial pre-RJMCMC within-model random walk Metropolis runs, for each k. Conditional on a value of k, these RWM runs result in an MCMC sample from $\pi(\boldsymbol{\theta}_k|k)$, from which sensible estimates of $\boldsymbol{\mu}_k$ and $B_k B_k^T$ can be derived.

Green demonstrates his sampler on two variable-dimension problems: a logistic regression model choice problem and a point process change point problem. In both examples posterior inference agrees with previously designed problem-specific RJMCMC samplers. This is particularly surprising given the multimodal nature of the posterior in the second problem and indicates that this simple automatic sampler might be more widely applicable than we would intuitively expect.

In spite of the encouraging performance for the problems presented, Green does not consider the AutoRJ sampler as anything more than a simple first attempt towards an automatic sampler. In particular, one can envisage conditional distributions $\pi(\boldsymbol{\theta}_k|k)$ (for example severely multimodal distributions) for which any distribution characterised by a single position vector and a single matrix of scale parameters will be a poor approximation. In such cases it seems highly likely that the proposals used by the AutoRJ sampler will no longer perform adequately.

Another criticism of the AutoRJ sampler is that the sampler is not fully automatic. In particular, the sampler is likely to perform best if the user provides reasonable estimates of the parameters $\boldsymbol{\mu}_k$ and B_k for each value of k under consideration. Clearly this information may not necessarily be readily available to a non-expert user. Whilst the possibility of initial within-model RWM runs to estimate these parameters might appear to overcome this problem, there is unfortunately a hidden drawback. In order for the RWM to work sufficiently well to get reliable estimates of the parameters, the RWM must itself be suitably parameterised in terms of a suitable spread parameter which must be specified by the user. For the non-expert, choosing this spread parameter may not be easy. Even if the user is sufficiently confident to choose appropriate spread parameters, it might be necessary for there to be some initial tuning to ensure that the parameters chosen are suitable.

In the remainder of this chapter we introduce the AutoMix sampler which we design to build upon the sampler above and address these issues.

5.3 The AutoMix Sampler

We begin this section with a brief descriptive summary of our proposed solutions to the above issues. Having discussed the broad approach of the AutoMix sampler, sections 5.3.1 to 5.3.3 provide full details of our approach.

Consider first the case where the AutoRJ proposals do not provide a good approximation of the conditionals. The approach that we adopt for the AutoMix sampler is to generalise the type of distribution that we use within our proposals. An obvious alternative to using only a single vector $\boldsymbol{\mu}_k$ and matrix $B_k B_k^T$ to characterise $\pi(\boldsymbol{\theta}_k|k)$ is to use a proposal distribution consisting of a mixture of distributions, where each of the components of the mixture is characterised by such a vector and matrix. For our sampler we concentrate on mixtures of Normal distributions. By using such a proposal we should hope to better cover the important features of the conditionals $\pi(\boldsymbol{\theta}_k|k)$. This also provides an easy way of dealing more robustly with multimodality in the conditionals.

Although fitting a mixture may seem an obvious generalisation, there are several important issues that arise. Firstly we must address how to fit these mixtures to the conditionals. Another problem is how to decide upon the number of components in each mixture that we fit. In addition, the cost of fitting mixtures of several components will obviously be considerably more than the cost associated with estimating a single mean vector and covariance matrix as in the AutoRJ sampler. This suggests that for problems in which the AutoRJ sampler works sufficiently well, we should allow the user the option of using this sampler (modified to achieve full automaticity) as a special case. In section 5.3.3 we discuss further the idea of fitting mixtures and consider a particular method

that is appropriate for the AutoMix sampler.

To address the second criticism of the AutoRJ sampler, that it is not fully automatic, we appeal to the ideas presented on adaptive sampling in chapter 4. In order to remove the need for the user to specify either the mixture parameters or any other parameters (such as RWM parameters) we propose the use of such adaptive sampling techniques. This will allow the sampler to improve its performance as it progresses without needing to be directed by the user. Section 5.3.2 provides further details of our use of adaptive tools within the AutoMix sampler.

5.3.1 Sampler Outline

For the remainder of this chapter we work in terms of the generic model jumping problem, as introduced in section 2.3, with the further assumption that $\mathcal{M}$ is finite, i.e. there are only $K_{max} < \infty$ models, indexed by $1, 2, \ldots, K_{max}$. Our task is to make inference about the joint distribution π of $(k, \boldsymbol{\theta}_k)$. To be consistent with previous literature and for ease of presentation, we set $\boldsymbol{\Theta}_k \subseteq \mathbb{R}^{n_k}$ and (in an abuse of notation) assume $\pi(\cdot)$ denotes a joint density across $\bigcup_k (\{k\} \times \boldsymbol{\Theta}_k)$. With this formulation in place we are able to introduce the AutoMix sampler.

We begin our presentation by remarking that our automatic sampler is more than just a simple RJMCMC sampler. In fact, the AutoMix sampler can be thought of as a multistage process, with the final stage being the conventional reversible jump sampler. The preceding stages come from the automatic nature of the sampler and are analogous to the tuning process of a simple reversible jump sampler.

In the AutoMix sampler, there are three stages. These stages are: (1) an initial stage to adapt within-model RWM samplers and to provide the second stage with appropriate data (samples from $\pi(\boldsymbol{\theta}_k|k)$ for each value of k); (2) a stage to fit a Normal mixture distribution to the conditional target distributions $\pi(\boldsymbol{\theta}_k|k)$; and (3) a RJMCMC sampler based upon the parameters derived in the preceding sections. In some instances, if the sampler has been run previously the first two stages may not be required. However, we defer discussion of this idea until section 5.4 where we present a summary of the full sampler.

In this section we give a mathematical formulation for the third stage of the sampler, the RJMCMC part of the process as we are familiar with it. We assume that the previous stages have occurred and in particular, for each value of k, we have a Normal mixture distribution M_k which will provide the basis for our reversible jump proposals. For the mixture M_k, we assume that there are L_k components with weights λ_k^l, $l = 1, \ldots, L_k$. Component l of this mixture has mean vector $\boldsymbol{\mu}_k^l$ and covariance matrix $B_k^l(B_k^l)^T$. Denoting the density of the Normal mixture M_k (with respect to the n_k-dimensional Lebesgue measure) by f_{M_k}, we can write

$$f_{M_k}(\boldsymbol{\theta}_k) = \sum_{l=1}^{L_k} \lambda_k^l f_l(\boldsymbol{\theta}_k|\boldsymbol{\mu}_k^l, B_k^l), \tag{5.3}$$

where $f_l(\cdot|\cdot)$ denotes the density of component l.

We proceed in a similar way to the automatic sampler proposed by Green.

Suppose the Markov chain is currently in state $(k, \boldsymbol{\theta}_k)$. The AutoMix sampler proceeds by first assigning $\boldsymbol{\theta}_k$ to one of the L_k components, l_k, of the mixture M_k.

This allocation takes account of the relative densities of each of the components by assigning according to the distribution with p.m.f. given by

$$\mathbb{P}(l_k = l) = p_{k,\boldsymbol{\theta}_k}(l) = \frac{\lambda_k^l f_l(\boldsymbol{\theta}_k|\boldsymbol{\mu}_k^l, B_k^l)}{\sum_{j=1}^{L_k} \lambda_k^j f_j(\boldsymbol{\theta}_k|\boldsymbol{\mu}_k^j, B_k^j)},\ l = 1, \ldots, L_k. \tag{5.4}$$

Note that l_k is not a state variable like k or $\boldsymbol{\theta}_k$, but simply a random variable used in the process of proposing a new state.

Having allocated $\boldsymbol{\theta}_k$ to component l_k we then propose a new model k'. When in model k the probability of choosing a new model k' is given by $\rho_{k,k'}$. Given that we have proposed a move to model k', we then choose a new component $l'_{k'}$ in the mixture $M_{k'}$ according to the mixture component probabilities (i.e. $\mathbb{P}(l'_{k'} = l') = \lambda_{k'}^{l'}$ for $l' = 1, \ldots, L_{k'}$). Again, $l'_{k'}$ is just another random variable used in the proposal mechanism.

Having made the above proposals, our proposed new parameter state $\boldsymbol{\theta}'_{k'}$ depends upon the dimension $n_{k'}$ as follows:

$$\boldsymbol{\theta}'_{k'} = \begin{cases} \boldsymbol{\mu}_{k'}^{l'_{k'}} + B_{k'}^{l'_{k'}}[R_{k,k'}(B_k^{l_k})^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k})]_1^{n_{k'}} & n_{k'} < n_k \\ \boldsymbol{\mu}_{k'}^{l'_{k'}} + B_{k'}^{l'_{k'}} R_{k,k'}(B_k^{l_k})^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k}) & n_{k'} = n_k \\ \boldsymbol{\mu}_{k'}^{l'_{k'}} + B_{k'}^{l'_{k'}} R_{k,k'} \begin{pmatrix} (B_k^{l_k})^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k}) \\ \boldsymbol{u} \end{pmatrix} & n_{k'} > n_k \end{cases}. \tag{5.5}$$

As for the sampler introduced by Green (2003), $[\ldots]_1^m$ denotes the first m components of a vector, $\boldsymbol{u}$ is an $(n_{k'} - n_k)$-vector of random numbers drawn from density $g_{k,k'}$ (which we shall usually take to be that of independent standard Normal or Student t distributions), and $R_{k,k'}$ is an orthogonal matrix of order $\max\{n_k, n_{k'}\}$, satisfying $R_{k,k'} = R_{k',k}^T$, where $R_{k',k}$ is the orthogonal matrix used for a proposal from model k' to model k. The inclusion of the matrix $R_{k,k'}$ is less intuitive for the AutoMix sampler because the down moves are already

randomised by the choice of mixture components. Furthermore, any skew should now be accounted for by mixture M_k. However, we include $R_{k,k'}$ so that the AutoRJ sampler is a special case of our new sampler. In practice, at run-time the user has the choice of whether to include $R_{k,k'}$ (in particular as a random perturbation matrix) or not (see section 5.4.2 for details).

Intuitively, the proposal described above for the AutoMix sampler is a straightforward generalisation of the AutoRJ sampler. When in state $(k, \boldsymbol{\theta}_k)$, the AutoRJ sampler proposes a new state $(k', \boldsymbol{\theta}'_{k'})$ as follows. First, a new model k' is proposed. Next, thinking of $\pi(\boldsymbol{\theta}|k)$ as being approximated by some distribution parameterised by vector $\boldsymbol{\mu}_k$ and matrix B_k (for example a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $B_k B_k^T$), the state vector is standardised. If the proposed new model k' is of greater dimension than model k, standard random variables are generated and appended to the state vector and the new standardised vector is (possibly) permuted using a matrix $R_{k,k'}$. If the proposed new model is of the same dimension as the existing model the standardised state vector is (possibly) permuted using $R_{k,k'}$. Finally, if the model k' is of smaller dimension than model k the standardised vector is permuted and the last $n_k - n'_k$ elements are discarded. The final part of the proposal, is to unstandardise the standardised vector using the vector $\boldsymbol{\mu}_{k'}$ and matrix $B_{k'}$ corresponding to the approximation of the model k' conditional distribution.

The AutoMix moves are almost identical to those of the AutoRJ sampler but differ because our proposals are mixture distributions, M_k. Therefore, we must additionally choose which component of the mixture M_k we are going to use to standardise our state vector, $\boldsymbol{\theta}_k$, and which component of the mixture $M_{k'}$ we are going to use to unstandardise the resulting vector to get $(k', \boldsymbol{\theta}'_{k'})$.

Letting $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ and $\boldsymbol{x}' = (k, \boldsymbol{\theta}'_{k'})$, the acceptance probability corresponding to the AutoMix proposal is given by

$$\alpha_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}') = \min\{1, A_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}'), \} \tag{5.6}$$

where $A_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}')$ is given by

$$A_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\pi(k', \boldsymbol{\theta}'_{k'}) p_{k', \boldsymbol{\theta}'_{k'}}(l'_{k'}) \rho_{k',k} \lambda_k^{l_k} |B_{k'}^{l'_{k'}}|}{\pi(k, \boldsymbol{\theta}_k) p_{k, \boldsymbol{\theta}_k}(l_k) \rho_{k,k'} \lambda_{k'}^{l'_{k'}} |B_k^{l_k}|} G_{k,k'}(\boldsymbol{u}) \ . \tag{5.7}$$

Here $|B_{k'}^{l'_{k'}}|/|B_k^{l_k}|$ is the Jacobian, $p_{k', \boldsymbol{\theta}'_{k'}}(l'_{k'})$ is derived as in equation 5.4 with $(k, \boldsymbol{\theta}_k)$ replaced by $(k', \boldsymbol{\theta}'_{k'})$ and $G_{k,k'}(\boldsymbol{u})$ is given by

$$G_{k,k'}(\boldsymbol{u}) = \begin{cases} g_{k,k'}(\boldsymbol{u}) & n_{k'} < n_k \\ 1 & n_{k'} = n_k \\ [g_{k,k'}(\boldsymbol{u})]^{-1} & n_{k'} > n_k \end{cases} \ .$$

Again, since the matrices $R_{k,k'}$ and $R_{k',k}$ are orthogonal they make no contribution in the acceptance probability.

This acceptance probability is very similar to the acceptance probability for the AutoRJ sampler. The only generalisation arises from ensuring that we take into account which components of the mixtures M_k and M'_k are used and how these components were chosen (for the move and its reverse). The following proposition shows that using this acceptance probability the resulting chain is reversible, with invariant distribution π

Proposition 5.1. *Consider a state space $\mathcal{X} = \cup_{k \in \mathcal{M}}(\{k\} \times \boldsymbol{\Theta}_k)$. Let π be a distribution over $\mathcal{X}$, with density denoted (in an abuse of notation) by $\pi(\cdot)$. Let $(\boldsymbol{X}_n)$ be the Markov chain resulting from the AutoMix kernel. Then $(\boldsymbol{X}_n)$ is reversible with invariant distribution π.*

Proof. To prove this result we must show that the AutoMix transition kernel satisfies the integrated detailed balance equations, with respect to the distribution π. Let $\mathcal{C}' = \{k'\} \times \mathcal{D}'$, where $k' \in \mathcal{M}$ and $\mathcal{D}' \subset \boldsymbol{\Theta}_{k'}$. Let the current state of the Markov chain be $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$. The transition kernel of the AutoMix sampler can be written

$$\mathcal{K}(\boldsymbol{x}, \mathcal{C}') = \mathcal{Q}(\boldsymbol{x}, \mathcal{C}') + \mathbf{1}_{\{\boldsymbol{x} \in \mathcal{C}'\}} \mathcal{R}(\boldsymbol{x}). \tag{5.8}$$

Here, $\mathcal{Q}$ is the contribution of accepted moves defined as

$$\mathcal{Q}(\boldsymbol{x}, \mathcal{C}') = \begin{cases} \rho_{k,k'} \mathcal{Q}_1(\boldsymbol{\theta}_k, \mathcal{D}') & n_k < n_{k'} \\ \rho_{k,k'} \mathcal{Q}_2(\boldsymbol{\theta}_k, \mathcal{D}') & n_k \geq n_{k'} \end{cases} \tag{5.9}$$

where,

$$\begin{aligned} \mathcal{Q}_1(\boldsymbol{\theta}_k, \mathcal{D}') &= \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \int_{\boldsymbol{\theta}'_{k'} \in \mathcal{D}'} p_{k,\boldsymbol{\theta}_k}(l) \lambda_{k'}^{l'} g_{k,k'}(\boldsymbol{u}) \alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}\boldsymbol{u} \quad \text{and} \\ \mathcal{Q}_2(\boldsymbol{\theta}_k, \mathcal{D}') &= \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \mathbf{1}_{\{\boldsymbol{\theta}'_{k'} \in \mathcal{D}'\}} p_{k,\boldsymbol{\theta}_k}(l) \lambda_{k'}^{l'} \alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}') \ , \end{aligned} \tag{5.10}$$

recalling that $\boldsymbol{\theta}'_{k'}$ is defined by the transformation from $\boldsymbol{\theta}_k$ (and possibly $\boldsymbol{u}$) in equation 5.5 and $\alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}')$ is the probability of accepting a move from $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ to $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$ defined by equations 5.6 and 5.7.

The contribution $\mathcal{R}$ in equation 5.8 corresponds to rejected jumps. As it cancels out of the detailed balance equations (see below) we do not define $\mathcal{R}$ explicitly but note that it is analogous to the rejection contribution of standard Metropolis-Hastings kernels.

Let $\mathcal{C} = \{k\} \times \mathcal{D}$, where $k \in \mathcal{M}$ and $\mathcal{D} \subset \boldsymbol{\Theta}_k$. Letting $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ and $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$, detailed balance requires that for all Borel sets $\mathcal{C}$, $\mathcal{C}'$,

$$\int_{\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}) \mathcal{K}(\boldsymbol{x}, \mathcal{C}') = \int_{\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}') \mathcal{K}(\boldsymbol{x}', \mathcal{C}).$$

This holds if and only if, for all Borel sets $\mathcal{C}$, $\mathcal{C}'$,

$$\begin{aligned} &\int_{\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}) \mathcal{Q}(\boldsymbol{x}, \mathcal{C}') + \int_{\mathcal{C} \cap \mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}) \mathcal{R}(\boldsymbol{x}) \\ &\qquad = \int_{\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}') \mathcal{Q}(\boldsymbol{x}', \mathcal{C}) + \int_{\mathcal{C}' \cap \mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}') \mathcal{R}(\boldsymbol{x}') \\ \Leftrightarrow \quad &\int_{\mathcal{C}} \pi(\mathrm{d}\boldsymbol{x}) \mathcal{Q}(\boldsymbol{x}, \mathcal{C}') = \int_{\mathcal{C}'} \pi(\mathrm{d}\boldsymbol{x}') \mathcal{Q}(\boldsymbol{x}', \mathcal{C}). \end{aligned} \tag{5.11}$$

This will hold if, for each (k, k') pair and for all Borel sets $\mathcal{D} \subset \boldsymbol{\Theta}_k$, $\mathcal{D}' \subset \boldsymbol{\Theta}_{k'}$,

$$\int_{\mathcal{D}} \pi(k, \boldsymbol{\theta}_k) \mathcal{Q}(\boldsymbol{\theta}_k, \mathcal{D}') \mathrm{d}\boldsymbol{\theta}_k = \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \mathcal{Q}(\boldsymbol{\theta}'_{k'}, \mathcal{D}) \mathrm{d}\boldsymbol{\theta}'_{k'}. \tag{5.12}$$

Consider first the case $n_k < n_{k'}$. Substituting the expression for $\mathcal{Q}$ (equation 5.9) into equation 5.12 we need,

$$\int_{\mathcal{D}} \pi(k, \boldsymbol{\theta}_k) \rho_{k,k'} \mathcal{Q}_1(\boldsymbol{\theta}_k, \mathcal{D}') \mathrm{d}\boldsymbol{\theta}_k = \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} \mathcal{Q}_2(\boldsymbol{\theta}'_{k'}, \mathcal{D}) \mathrm{d}\boldsymbol{\theta}'_{k'}. \tag{5.13}$$

Substituting the expression for $\mathcal{Q}_1$ (equations 5.10) into the left-hand side of equation 5.13 we have,

$$\text{L.H.S.} = \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \int_{(\boldsymbol{\theta}_k, \boldsymbol{\theta}'_{k'}) \in \mathcal{D} \times \mathcal{D}'} \pi(k, \boldsymbol{\theta}_k) \rho_{k,k'} p_{k,\boldsymbol{\theta}_k}(l) \lambda_{k'}^{l'} g_{k,k'}(\boldsymbol{u}) \alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}\boldsymbol{\theta}_k \mathrm{d}\boldsymbol{u}. \tag{5.14}$$

Using the definition of $\alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}')$ in equations 5.6 and 5.7, the integrand can be rewritten as

$$\min \left\{ \pi(k, \boldsymbol{\theta}_k) \rho_{k,k'} p_{k,\boldsymbol{\theta}_k}(l) \lambda_{k'}^{l'} g_{k,k'}(\boldsymbol{u}), \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} p_{k',\boldsymbol{\theta}'_{k'}}(l') \lambda_k^l \frac{|B_{k'}^{l'}|}{|B_k^l|} \right\},$$

and rewriting this as

$$\pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} p_{k',\boldsymbol{\theta}'_{k'}}(l') \lambda_k^l \alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x}) \frac{|B_{k'}^{l'}|}{|B_k^l|},$$

equation 5.14 becomes,

$$\text{L.H.S.} = \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \int_{(\boldsymbol{\theta}_k, \boldsymbol{\theta}'_{k'}) \in \mathcal{D} \times \mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} p_{k',\boldsymbol{\theta}'_{k'}}(l') \lambda_k^l \alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x}) \frac{|B_{k'}^{l'}|}{|B_k^l|} \mathrm{d}\boldsymbol{\theta}_k \mathrm{d}\boldsymbol{u}.$$

Since $|B_{k'}^{l'}|/|B_k^l|$ is the determinant of the Jacobian of the transformation from $(\boldsymbol{\theta}_k, \boldsymbol{u})$ to $\boldsymbol{\theta}'_{k'}$, this can be rewritten as

$$\begin{aligned}
\text{L.H.S.} &= \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \int_{(\boldsymbol{\theta}_k, \boldsymbol{\theta}'_{k'}) \in \mathcal{D} \times \mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} p_{k',\boldsymbol{\theta}'_{k'}}(l') \lambda_k^l \alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x}) \mathrm{d}\boldsymbol{\theta}'_{k'} \\
&= \sum_{l=1}^{L_k} \sum_{l'=1}^{L_{k'}} \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} \mathbf{1}_{\{\boldsymbol{\theta}_k \in \mathcal{D}\}} p_{k',\boldsymbol{\theta}'_{k'}}(l') \lambda_k^l \alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x}) \mathrm{d}\boldsymbol{\theta}'_{k'} \\
&= \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'}) \rho_{k',k} \mathcal{Q}_2(\boldsymbol{\theta}'_{k'}, \mathcal{D}) \mathrm{d}\boldsymbol{\theta}'_{k'}.
\end{aligned}$$

Since this is the right-hand side of equation 5.13 the result holds.

Considering next the case $n_k > n_{k'}$, we note that we can swap the labels of k and k'. The result then follows immediately from the above arguments.

Finally we must consider the case of $n_k = n_{k'}$. Substituting the expression for $\mathcal{Q}$ (equation 5.9) into equation 5.12 we need,

$$\int_{\mathcal{D}} \pi(k, \boldsymbol{\theta}_k)\rho_{k,k'}\mathcal{Q}_2(\boldsymbol{\theta}_k, \mathcal{D}')\mathrm{d}\boldsymbol{\theta}_k = \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}\mathcal{Q}_2(\boldsymbol{\theta}'_{k'}, \mathcal{D})\mathrm{d}\boldsymbol{\theta}'_{k'}. \tag{5.15}$$

Following the same arguments as above (noting that when $n_k = n_{k'}$, $|B^{l'}_{k'}|/|B^l_k|$ is the determinant of the Jacobian of the transformation from $\boldsymbol{\theta}_k$ to $\boldsymbol{\theta}'_{k'}$) we get,

$$\begin{aligned}
\text{L.H.S.} &= \sum_{l=1}^{L_k}\sum_{l'=1}^{L_{k'}} \int_{(\boldsymbol{\theta}_k, \boldsymbol{\theta}'_{k'})\in\mathcal{D}\times\mathcal{D}'} \pi(k, \boldsymbol{\theta}_k)\rho_{k,k'}p_{k,\boldsymbol{\theta}_k}(l)\lambda^{l'}_{k'}\alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}')\mathrm{d}\boldsymbol{\theta}_k \\
&= \sum_{l=1}^{L_k}\sum_{l'=1}^{L_{k'}} \int_{(\boldsymbol{\theta}_k, \boldsymbol{\theta}'_{k'})\in\mathcal{D}\times\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}p_{k',\boldsymbol{\theta}'_{k'}}(l')\lambda^{l}_{k}\alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x})\frac{|B^{l'}_{k'}|}{|B^l_k|}\mathrm{d}\boldsymbol{\theta}_k \\
&= \sum_{l=1}^{L_k}\sum_{l'=1}^{L_{k'}} \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}\mathbf{1}_{\{\boldsymbol{\theta}_k\in\mathcal{D}\}}p_{k',\boldsymbol{\theta}'_{k'}}(l')\lambda^{l}_{k}\alpha_{l',l}(\boldsymbol{x}', \boldsymbol{x})\mathrm{d}\boldsymbol{\theta}'_{k'} \\
&= \int_{\mathcal{D}'} \pi(k', \boldsymbol{\theta}'_{k'})\rho_{k',k}\mathcal{Q}_2(\boldsymbol{\theta}'_{k'}, \mathcal{D})\mathrm{d}\boldsymbol{\theta}'_{k'}.
\end{aligned}$$

Since this is the right-hand side of equation 5.15, this completes the proof. □

An attractive feature of the AutoMix sampler is that the trans-dimensional jump outlined above also provides a method for improving within-model mixing. In particular, if we allow $k' = k$, then we can make significant changes in state vector by moving between different components of M_k. Allowing $k' = k$ is easily achieved by suitable choice of the function $\rho_{k,k'}$. We defer discussion of our particular choice of $\rho_{k,k'}$ until later in the chapter after we have discussed the use of adaptive samplers in the AutoMix sampler.

The AutoMix sampler does not only rely upon the trans-dimensional move illustrated above. At each sweep the sampler also attempts within-model

proposals. In the problems we have implemented, we have found that when used in addition to the trans-dimensional move, simple random walk Metropolis ensures adequate mixing. At each sweep we do single update random walk Metropolis for each of the components $\boldsymbol{\theta}_k$. To improve mixing, at every tenth sweep we replace the single update RWM moves by a block update RWM. This move attempts to update all n_k components of $\boldsymbol{\theta}_k$ simultaneously. Discussion of the RWM parameters and a full summary of the AutoMix algorithm is deferred until we have provided more details about the interesting aspects of the AutoMix approach. We commence this discussion by considering the use of adaptation in the AutoMix sampler.

5.3.2 Adaptation in the AutoMix Sampler

The design of an automatic reversible jump sampler provides an excellent vehicle for demonstrating the exciting potential of adaptive sampling techniques as discussed in chapter 4. The adoption of adaptive techniques is a major feature of the AutoMix sampler and in this section we discuss how we utilise this powerful feature. Adaptive techniques are made available for use at the first and third stages of the AutoMix sampler.

We begin by looking at the first stage of the AutoMix sampler, the within-model RWM automatic tuning runs that we use to sample from the conditional distributions $\pi(\boldsymbol{\theta}_k|k)$ for each value of k. As mentioned in section 5.2, the use of pre-RJMCMC RWM runs was a feature of Green's AutoRJ sampler, but such runs required the underlying RWM parameters to be supplied by the user. Employing an adaptive RWM method in the AutoMix sampler allows the proposal parameters to tend towards optimal values (according to some criteria)

without any user intervention.

An important question is which particular adaptive algorithm to employ. As discussed in chapter 4 this depends upon what criteria we use to adapt our proposal parameters. While other choices are possible, we choose the adaptive acceptance probability (AAP) algorithm introduced by Atchadé and Rosenthal (2003) and discussed in depth in chapter 4.

Recall from chapter 4 that the AAP algorithm is an adaptive MCMC algorithm, based on a RWM kernel, which updates the variance parameter σ^2 of the kernel at each iteration. The value σ^2 is updated with the aim of achieving a user defined optimal value for the expected acceptance probability. We modify the AAP algorithm for use in the multivariate case (recall for the AutoMix sampler model k has n_k components) by doing componentwise RWM, and finding the optimal value of σ^2 for each component. A further slight generalisation of our implementation of the AAP algorithm is that the user has the option of using either a Normal or t distribution as the proposal for the algorithm.

The appeal of the AAP algorithm is its simplicity. The aim of the adaptive algorithm is easy to understand and at each iteration the adaptation is easily performed with minimal computational cost. This feature is very important since we must apply the adaptive algorithm repeatedly, for each of the n_k parameters of the conditional distribution $\pi(\boldsymbol{\theta}_k|k)$ and for each of the K_{max} models in $\mathcal{M}$. In order that our AutoMix sampler runs within a reasonable amount of time, our adaptive methods can not afford to be expensive.

A concern about the use of the AAP algorithm is the unresolved question raised

in chapter 4 about the convergence of the algorithm for all target distributions π. Recall that we recommended caution in the general use of the algorithm, due to the inability to verify one of the assumptions underlying the convergence of the adaptive parameters. However, we have experimented extensively with this algorithm and to date have found no counter example of a target distribution π for which the algorithm does not behave as we expect. Thus it appears that there is empirical evidence for the validity of the AAP algorithm for most target distributions that we encounter.

As the AAP algorithm is ergodic, then whether or not the adaptive RWM parameters have converged does not affect the results of fitting mixtures to the resulting samples. Therefore, discovering that for some target distributions π, the use of the AAP algorithm did not result in convergence of the adaptive proposal parameters should have no impact on the second stage of the AutoMix sampler. However, the AutoMix sampler uses the final value of the adaptive parameters for within-model RWM in the reversible jump third stage. If the AAP algorithm did not converge, this RWM would not be optimal (in terms of achieving our chosen value for the expected acceptance probability). Should subsequent research demonstrate that the AAP algorithm does not converge in certain cases, it may be advisable to alter our adaptive approach to avoid the above situation from occurring.

Even if we are confident that the AAP algorithm will converge, an important question is how to decide when the parameters *have* sufficiently converged. This has implications for the run length of the first stage. More RWM sweeps are likely to lead to better convergence of the adaptive parameters. Moreover, longer runs will result in a bigger RWM sample which may result in a mixture

distribution that better fits the conditional distribution $\pi(\boldsymbol{\theta}_k|k)$. However, both the mixture distributions and the final values of adaptive RWM parameters are only used in our reversible jump proposals. Therefore we must not invest too much time trying to find a proposal distribution that is only slightly better than the proposal distribution that would have resulted from a RWM run of half the length. As such, the AutoMix sampler uses fixed length RWM runs, giving the user the run time option to override the default number of sweeps.

The use of adaptive sampling need not be limited to just the first stage of the AutoMix sampler. In particular, the AutoMix sampler also allows the user the opportunity to implement the reversible jump adaptive methods described in section 4.6.

In the context of the algorithm introduced in section 5.3.1, if the user decides to employ adaptive methods in stage 3 of the AutoMix sampler, then at iteration n of the reversible jump sampler we have $\rho_{k,k'} = \psi_n^{k'}$ for all k. In words, the probability of jumping to model k' is $\psi_n^{k'}$, independent of the current state. The proposal probability parameters are updated at each iteration according to algorithm B defined in equations 4.19 to 4.22 and described in section 4.6. We choose algorithm B over algorithm A because the limited applicability of algorithm A (discussed in chapter 4) means it is not appropriate for inclusion within a generic sampler that must be applicable for a variety of problems. To demonstrate that for some problems algorithm A works equally well, we provide an illustration of the AutoMix sampler with adaptive algorithm A for the toy example detailed in section 5.5.1.

Although we have so far been unable to verify any ergodicity or convergence

properties of the adaptive algorithm B, the empirical evidence for these properties is convincing (see section 5.5). However, we recognise that some users may wish not to employ theoretically unproven methods and so we allow the user to dictate whether or not the AutoMix sampler should employ this adaptive algorithm B in the reversible jump stage. If the user selects not to use the adaptive regime, the probability of jumping to model k' at iteration n is taken to be $\psi^{k'} = 1/K_{max}$.

The adoption of adaptive techniques gives the AutoMix sampler a large advantage over Green's AutoRJ sampler. The methods not only allow us to overcome a restriction of the original sampler, but also offer us a way of improving the efficiency of the reversible jump stage of the sampler. Although we do not use adaptive techniques in the second stage of the AutoMix sampler it is certainly possible to imagine a way in which such methods could also be useful there. We return to this possibility briefly in section 5.6.

5.3.3 Fitting Normal Mixtures in the AutoMix Sampler

The performance of Green's AutoRJ sampler depends upon the proposal parameters $\boldsymbol{\mu}_k$ and B_k for each k. For the AutoMix sampler, approximating the conditionals $\pi(\boldsymbol{\theta}_k|k)$ by a Normal mixture M_k (for each k) is equally important for the success of the reversible jump algorithm.

In the AutoRJ sampler, the user is able to supply parameters $\boldsymbol{\mu}_k$ and B_k for each of the models, avoiding the pre-RJMCMC tuning runs. In the AutoMix sampler this possibility is also allowed, so that the parameters of the mixture distributions can be taken into account if they are known to the user. In such instances the AutoMix sampler can skip the first two stages and begin

immediately at the reversible jump stage. However, for most real problems this will not be the case and so in this section we describe how the AutoMix sampler fits a mixture of Normals to the MCMC samples from the conditionals $\pi(\boldsymbol{\theta}_k|k)$ that were generated by stage 1.

One benefit of the original sampler over the AutoMix sampler is that estimating a mean vector and covariance matrix is a much simpler task than fitting a mixture of Normals. Moreover, the estimation of these parameters is a very quick process. In fact, in the case of the original sampler, for each k, the estimates $\boldsymbol{\mu}_k$ and B_k can easily be estimated online during the RWM run. (Note that this is not adaptive sampling because $\boldsymbol{\mu}_k$ and B_k play no role in the proposal of the RWM).

In spite of the ease and speed of determining the $\boldsymbol{\mu}_k$ and B_k for each k, the replacement of this approach by using a mixture of Normals is easily motivated (see section 5.3). An important question to consider is how we estimate the number of components L_k that make up the mixture in model k. Too few components will result in proposals that miss important features of the conditionals. On the other hand choosing a mixture which always had too many components would result in the reversible jump sampler being incredibly inefficient. The only restriction that we impose is that for each mixture M_k, the number of components L_k must be finite.

The subject of how best to fit a Gaussian mixture distribution to empirical data has been one of considerable research over the years. Many alternative methods exist and we do not attempt any kind of a review here. An extensive review

of the topic is provided by McLachlan and Peel (2000). For a concise introduction, with many references we direct the reader to Figueiredo and Jain (2002).

Bayesian methods provide a natural way of fitting mixture distributions where the number of components and the component parameters (weights, means and covariance matrices) are estimated simultaneously. In particular, methods exist that use MCMC methods to fit the mixture such as Neal (1992), Richardson and Green (1997) or Dellaportas and Papageorgiou (2004). Unfortunately, despite being natural candidates for fitting mixture distributions, such methods are typically too expensive to be repeatedly employed to approximate the conditional distributions $\pi(\boldsymbol{\theta}_k|k)$ for every k. Furthermore, using any MCMC scheme, especially a reversible jump scheme such as those proposed by Richardson and Green or Dellaportas and Papageorgiou, seems perverse in the context of our overall purpose of trying to implement a generic RJMCMC sampler.

Methods for fitting mixtures using classical statistical criteria also have drawbacks. Many such algorithms first find the maximum likelihood mixture parameters (component weights, means and covariance matrices) for a given number of components, and then repeat the process for a different number of components. The final mixture is selected according to some criterion (such as the Bayesian Information Criterion). Since the purpose of fitting mixtures in stage 2 of the AutoMix sampler is merely to produce proposal distributions for the reversible jump stage, using this multistage approach seems unnecessarily expensive.

One approach that overcomes the above difficulties is a relatively new method introduced by Figueiredo and Jain (2002). By framing the problem as an

information theory problem, the authors describe a sensible scheme for fitting mixtures based on the Minimum Message Length Criterion. The scheme that they propose uses a component-wise EM algorithm to simultaneously fit the number of components and the component parameters to result in a mixture distribution that best fits the data.

The algorithm achieves the simultaneous estimation of the number of components by annihilating components which exhibit little support from the data. The annihilation of components occurs naturally as part of the iterative process. However, as Figueiredo and Jain emphasise in the original paper, after the algorithm converges it may be necessary to restart it, by forcing the least probable component to have zero weight and renormalising the weights of the remaining components. This step is required because the evolution of the cost function at the natural component annihilation times means that the algorithm may find only a local minimum. This re-initialisation step must be repeated until the number of components is equal to a user specified minimum (which by default is taken to be 1). By selecting the mixture that corresponds to the minimum of the naturally occurring local minima a globally best fitting mixture is found.

The algorithm proposed by Figueiredo and Jain sidesteps two of the common problems associated with the EM approach. Firstly, the component annihilation permitted in the Figueiredo and Jain algorithm ensures that the algorithm doesn't converge to the boundary of the parameter space (i.e. to mixtures where certain components have zero weight). Secondly, the algorithm does not suffer from a sensitivity to initialisation, which is a drawback of many EM based algorithms. This robustness to initialisation is a result of choosing the initial

number of components to be much larger than the true number in the mixture, and initialising the mean vectors to take the values of randomly selected data points throughout the parameter space. By spreading the initial components across the range of the data, the algorithm is able to escape from local minima in the cost function that is being minimised.

The mixture fitting algorithm proposed by Figueiredo and Jain converges to the optimal mixture in a reasonably small number of iterations. This is illustrated in the examples provided within the original paper. Our own testing of the algorithm found that it was robust and produced reliable results in a short period of time, leading us to adopt it as our method for fitting Normal mixture distributions for the AutoMix sampler. We do not claim that our choice of method for fitting mixtures is the best and employing some other method may improve the AutoMix sampler. It is possible that other mixture algorithms may exist which are equally as robust but which take less time to run. This might be an area for future research.

5.4 AutoMix Summary

In order to make the AutoMix sampler fully transparent we now provide full details of the algorithm, demonstrating how the sampler brings the three stages together and how the sampler depends upon user input. In section 5.4.2 we discuss some of the computational and implementation issues for the AutoMix sampler.

5.4.1 The Full Algorithm

We outline the algorithm in an enumerated fashion, maintaining the numbering of stages that we introduced in section 5.3.1. A summary of the run-time user options, indicated by sans-serif font in the text, is tabulated in table 5.1. We discuss these options further below. We return to the functions that must be supplied by the user in the next section.

Run-time flag	Summary
m	Controls the mode of the sampler
n	Controls stage 1 RWM run length
N	Controls stage 3 RJ run length
s	Controls the random number seed
a	Controls whether adaptive sampling is used in stage 3 RJ
p	Controls whether permutation is used in stage 3 RJ
t	Controls whether Normal or t-distribution variables are used
f	Controls output filenames

Table 5.1: A summary of the run-time options of the AutoMix algorithm.

(0) **Preliminary Stage**

(a) Read in command line user options: m, a, s, n, N, t, p and f. If not input use default values

(b) Initial file handling, filenames specified by f

(c) Initialise random number generator using seed s

(d) Read in number of models K_{max} from user supplied function

(e) Read in dimension n_k of each model $k = 1, \ldots, K_{max}$ from user supplied function

(1) **First Stage** - If m=1 skip this stage

(a) Set $k = 1$

(b) Read in initial conditions for model k from user supplied function

(c) Perform single update adaptive acceptance probability RWM (if $\mathsf{t}= d$ use a t-distributed proposal with d degrees of freedom, otherwise use a Normal proposal). Do for $\max\{\mathsf{n}, 100000, 10000 \times n_k\}$ sweeps. Use target acceptance prob of 0.25.

(d) Store (sub-sampled) RWM output $\boldsymbol{\theta}_k^1, \ldots, \boldsymbol{\theta}_k^{1000n_k}$

(e) Store vector of adapted RWM scale parameters $\boldsymbol{\sigma_k}$. Go to 2a.

(2) **Second Stage** - If m=1 skip this stage

(a) If m=0 go to 2b otherwise go to 2f

(b) Fit mixture to $\boldsymbol{\theta}_k^1, \ldots, \boldsymbol{\theta}_k^{1000n_k}$, using the Figueiredo and Jain (2002) algorithm

(c) Store mixture parameters for model k, including number of components L_k, weight of components, mean vectors and covariance matrices

(d) Write mixture parameters to file for possible use in future runs

(e) If $k = K_{max}$ go to 3a, otherwise set $k = k + 1$ and go to 1b

(f) Estimate parameters $\boldsymbol{\mu}_k$ and B_k from $\boldsymbol{\theta}_k^1, \ldots, \boldsymbol{\theta}_k^{1000n_k}$ to apply AutoRJ approximation

(g) If $k = K_{max}$ go to 3a, otherwise set $k = k + 1$ and go to 1b

(3) **Third Stage**

(a) Perform RJMCMC for N sweeps using mechanism described in section 5.3.1. If a=1 use adaptive model jumping proposal (using algorithm B, as described in section 4.6) otherwise do not use adaptation. Use permutation matrix R if p=1. Use Normal random variables $\boldsymbol{u}$

if t=0, otherwise use Student t random variables. In addition to reversible jump updates, do within-model single update RWM for each of the components of $\boldsymbol{\theta}_k$ at each sweep. Attempt a within-model block update RWM (for all components of $\boldsymbol{\theta}_k$ simultaneously) at every 10^{th} sweep. For both RWM moves, use the scale parameters $\boldsymbol{\sigma}_k$ from the adaptive RWM in stage 1.

(b) Write output $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ to file, where $\boldsymbol{X}_i = (k, \boldsymbol{\theta}_k)_i$

(c) Report summary statistics

The choice of 0.25 is motivated by its closeness to 0.239, the asymptotically optimal acceptance probability for a RWM sampler (Roberts *et al.*, 1997). Although this optimal value only applies for target distributions meeting certain conditions it seems a satisfactory guide for our sampler.

The ability to switch the 'mode' of the full AutoMix algorithm by using the run-time parameter m is worthy of further comment. Firstly, the user has the option to skip the first two stages of the algorithm. If this option is chosen then the sampler must instead be supplied with a file containing mixture parameters for each of the k models. Since the sampler stores the mixture parameters every time it is run for a particular problem, the first two stages only need to be run once for a particular problem. Subsequent runs can use the mixture parameters from this initial run. The benefit of this facility is that it is the first two stages of the AutoMix sampler that are the most expensive parts of the algorithm. Thus once the mixtures have been fitted for the models under consideration, future runs of the AutoMix algorithm (with m=1) take considerably less time to run.

A third mode of the sampler is also available by setting the parameter m=2. When the user selects this mode, our version of Green's AutoRJ sampler is implemented. Equivalently, the mixture distributions are constrained to have only one component and only $\boldsymbol{\mu}_k$ and B_k must be estimated for each model. As reported in the earlier sections of this chapter, estimating these parameters is considerably less time consuming than fitting a mixture distribution and in many instances still results in a sampler that performs well. Thus, for simple problems, the user may prefer to choose this mode.

It is possible to imagine other modes of the sampler, for example a mode where only the first stage is omitted. This would be possible, for example, if the user was able to use the output (conditional on a particular model k) from a previous full RJMCMC run in stage 3, to feed back into stage 2 for future runs. RJMCMC based samples from the conditional $\pi(\boldsymbol{\theta}_k|k)$ might provide a better sample than the RWM samples resulting from stage 1 and thus the resulting mixture might approximate the conditional much better. However, to avoid over complication we do not include this option, but consider it for future versions of the sampler.

5.4.2 Implementation Issues

The AutoMix algorithm is implemented as a C program. In order to apply the sampler to a particular problem, the user needs only to supply a file which contains four functions relevant to the statistical model about which the user wishes to make inference.

The first function tells the AutoMix sampler the value of K_{max}, i.e. the number

of models in the model space $\mathcal{M}$ of the specific problem. The second function that must be supplied by the user returns the dimension n_k of each of the models under consideration.

Having supplied the number of models and dimension of each model, the user must also supply appropriate initial conditions (which may be selected to be random) for the pre-RJMCMC RWM runs. This is achieved through a third function, which when given the model index k, returns appropriate initial conditions. The user requires no specialist knowledge to supply this information but must merely ensure that the parameter vector returned lies within the parameter space $\boldsymbol{\Theta}_k$ of model k.

By far the most important function that the user must supply evaluates the log of the target density $\pi(k, \boldsymbol{\theta}_k)$, up to an additive constant, at any point $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$. The format of the function is intuitive and should cause few problems for the user to implement.

As indicated in section 5.4.1, the user controls how the program is executed by the use of run-time flags (see table 5.1 for a summary). In addition to choices such as the number of sweeps in both the RWM and reversible jump stages, the flags also control the various options described in the previous section. For example, flags control whether or not the stages 1 and 2 of the sampler should be carried out, whether Normal or t-distributed random variables should be used, whether or not adaptation should be done in stage 3 and whether or not the random permutation matrix R should be used in the reversible jump proposals. If invalid choices are made, such as to skip the first and second stage of the AutoMix sampler without providing the mixture parameters to be used for the

third stage, the sampler reverts to default behaviour.

Finally we comment on the output of the AutoMix sampler. Upon completion, the sampler outputs a number of files which include: a log file containing important run statistics; a file containing the sampled values of the model index k; for each model $k = 1, \ldots, K_{max}$ a file containing the sampled values of $\boldsymbol{\theta}_k|k$; a file containing the mixture parameters; a file summarising the adaptation in stage 1 of the algorithm; and a file containing the log-posterior and log-likelihood chains. Additionally, the sampler produces a file containing summaries of the autocorrelation, a file detailing the cost function of the mixture fitting stage and a file with the (potentially) adaptive reversible jump model jumping probabilities.

The AutoMix sampler and examples of user files (for the examples considered in section 5.5) are available from the author's web page[1]. Additionally, users can register their email address at this site to ensure that they receive updates of the software.

5.5 Examples

In this section we look at a selection of problems that reversible jump techniques can be applied to, starting from a very simple toy example and progressing to some more realistic problems. The section highlights the implementation of the sampler in practice, demonstrating where it performs well and where it has limitations that may require future research.

[1] http://www.davidhastie.me.uk/AutoMix

5.5.1 A Toy Example

Consider the following simple toy problem to which we can apply the AutoMix sampler. Suppose we have only two models, $k = 1$ and $k = 2$, the first with weight 0.3 and the second with weight 0.7. Suppose that the first model has dimension $n_1 = 1$ and is a mixture of two Normal distributions, the first component having a weight of 0.2, a mean of -3.0 and a variance of 4.0. The second component has a weight of 0.8, a mean of 2.0 and a variance of 1.0. The second model is a mixture of three equally weighted two-dimensional Normals in a boomerang shape. The first component has a mean vector of (0,3) and covariance matrix given by $\begin{pmatrix} 4 & 0 \\ 0 & 0.5 \end{pmatrix}$, the second a mean of (-4,1) and covariance matrix given by $\begin{pmatrix} 2 & 1.5 \\ 1.5 & 2 \end{pmatrix}$ and the third a mean of (4,1) and covariance matrix given by $\begin{pmatrix} 2 & -1.5 \\ -1.5 & 2 \end{pmatrix}$. The conditional distributions $\pi(\boldsymbol{\theta}_1|k = 1)$ and $\pi(\boldsymbol{\theta}_2|k = 2)$ are illustrated by the solid lines in figures 5.1 and 5.2.

In practice we would not apply a technique as complex as reversible jump to obtain a sample from such a distribution. Indeed for the above problem such a sample could be undertaken directly. However, using a distribution whose shape and properties are familiar to us allows us to confirm that the AutoMix sampler works as intended. Furthermore, a simple distribution such as the above allows us to verify the performance of each of the stages of the AutoMix sampler.

For each model we choose n= 100000 for the first stage RWM, and N= 100000 for the reversible jump phase. We employ adaptation at the reversible jump stage (set a=1) but do not use permutation (set p=0). We report results for a

variety of other user options. Reported results are typical of repeated runs with identical settings.

Figures 5.1 and 5.2 show the samples resulting from a single run of the AutoMix sampler, conditioning upon the model. For both conditionals, the sample (recorded as a histogram in figure 5.1 and a scatter plot in figure 5.2) can be seen to represent the conditional target distribution (indicated by the solid blue lines). Additionally, by averaging over 4 repeated runs of the AutoMix sampler with identical user options but from random initial states, we can estimate probabilities of model 1 and model 2 as 0.2997 and 0.7003 respectively. The Monte Carlo error is estimated by blocking (see Green, 1999) to be less than 0.001. These plots and summary statistics indicate that applying the AutoMix sampler results in a good sample from the target distribution.

Perhaps more interestingly for this example, we can study the performance of the first two stages of the AutoMix sampler. We begin by considering the fitting of mixtures for this problem. We performed repeated independent runs of the sampler with the same settings but from random initial states. In more than 90% of runs for this problem, a two component mixture is fitted to the RWM sample from $\pi(\boldsymbol{\theta}_1|k = 1)$. In such cases the mixture fitted has very similar parameters to the mixture that we are trying to approximate. We highlight a typical example of the mixture fitted for model 1 by the dashed red line in figure 5.1. For model 2, the number of components in the fitted mixture is slightly more variable, although the modal number (across repeated runs) is 3, which is the number of components in the true mixture $\pi(\boldsymbol{\theta}_2|k = 2)$. When more than 3 components are fitted, there are typically extra components with little weight and the mixture is still a good approximation of $\pi(\boldsymbol{\theta}_2|k = 2)$. Figure 5.3

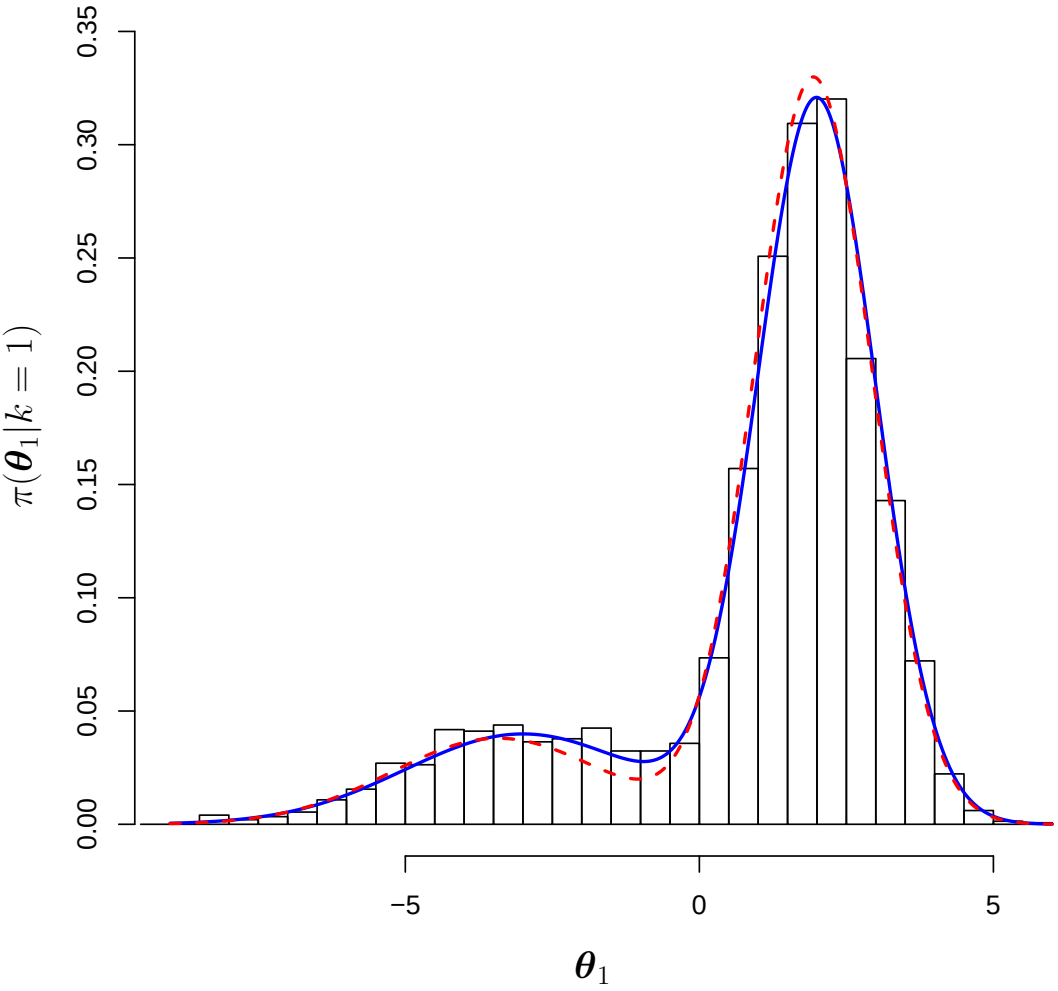

Figure 5.1: Histogram of MCMC sample of $\pi(\boldsymbol{\theta}_1|k=1)$. The true target density is shown in a solid (blue) line. The mixture fitted in stage 2 of the AutoMix sampler is shown in a dashed (red).

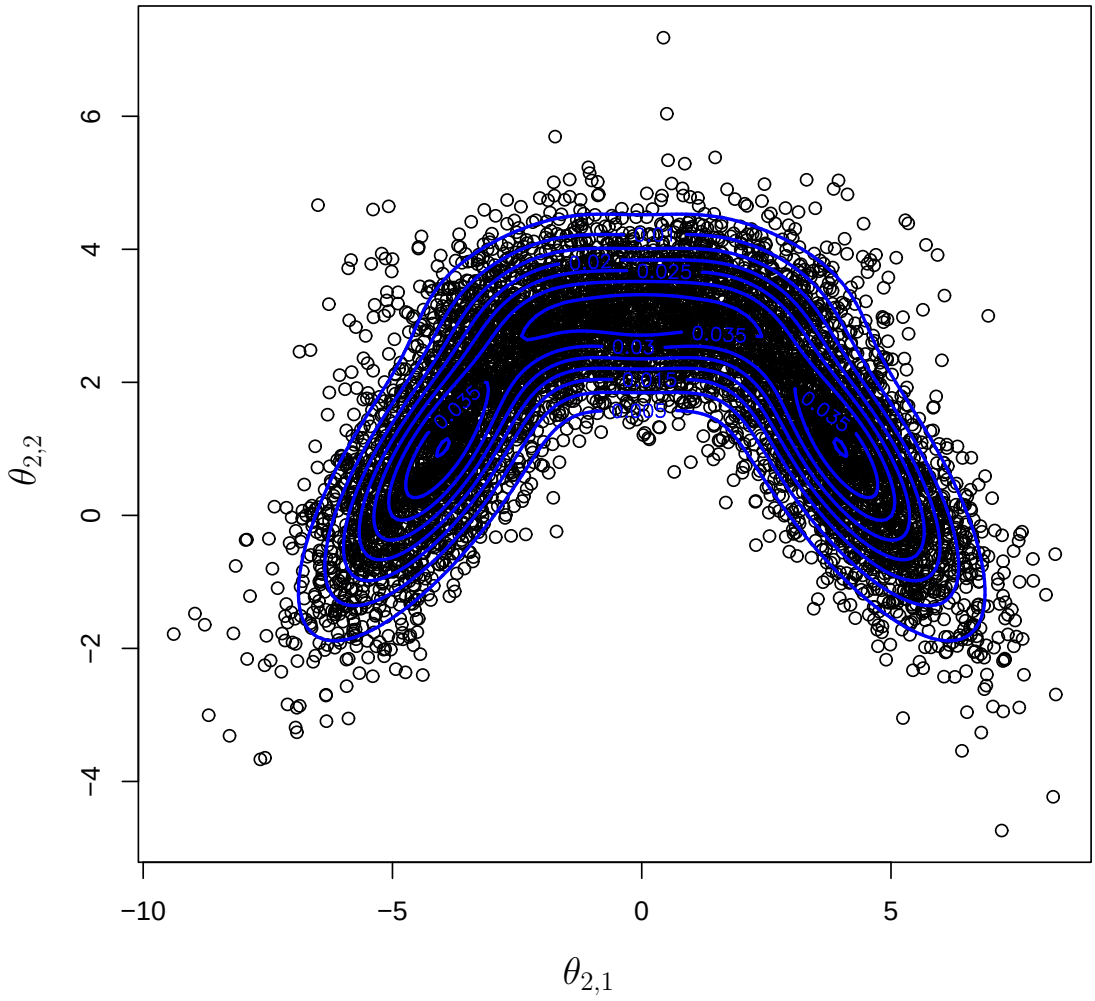

Figure 5.2: Scatter plot of MCMC sample of $\pi(\boldsymbol{\theta}_2|k = 2)$. Contours of target distribution are shown in blue.

shows contour plots of the fitted mixture (red lines) and the target conditional distribution (blue lines) for 4 separate runs of the sampler.

It is also possible to analyse the adaptation in the RWM pre-reversible jump stage of the AutoMix sampler. For the single component of $\boldsymbol{\theta}_1$ in model 1 and the two components of $\boldsymbol{\theta}_2$ in model 2, we can look at the evolution of the RWM scaling parameters and average acceptance probability. Figure 5.4 shows this progression over the RWM runs. These plots indicate that the adaptive acceptance probability algorithm appears to converge satisfactorily for this problem.

Using adaptation in the third stage of the AutoMix sampler the average acceptance probability for trans-dimensional jumps for this toy problem is approximately 94%. If no adaptation is employed in the reversible jump stage (i.e. setting a=0) this acceptation probability drops to about 78%. Enabling permutation has no effect on these acceptance rates. The evolution of the model jumping proposal probabilities ψ^1 and ψ^2 for the adaptive case is shown in figure 5.5. As can be seen from these plots, the proposals converge towards the target model probabilities as desired. For comparison, we also demonstrate the alternative adaptive reversible jump algorithm A (described in section 4.6) which is possible for this simple example. This alternative version converges much quicker than the standard algorithm B and results in adaptive parameters with considerably less variance. The resulting samples appear to represent the target distribution π and can be displayed in plots almost identical to those in figures 5.1 and 5.2. This suggests that algorithm A is also ergodic. However, because of its limited applicability we do not include algorithm A in the general algorithm.

If we re-run the AutoMix sampler using t distributed random variables (with 1,2

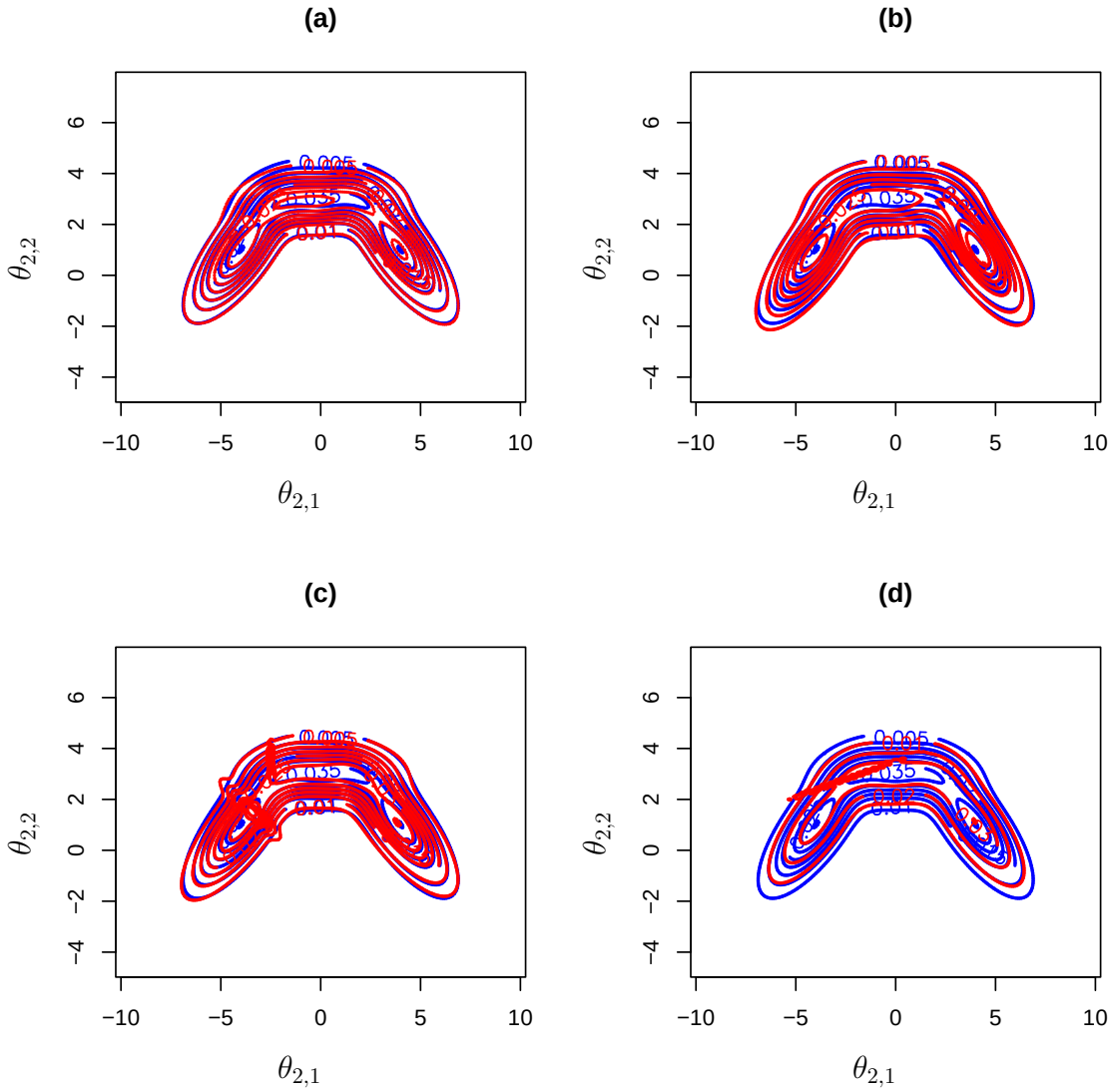

Figure 5.3: Contour plots of fitted mixtures (red lines) and target conditional distribution $\pi(\boldsymbol{\theta}_2|k=2)$ (blue lines) for 4 runs of the AutoMix sampler: (a) run 1; (b) run 2; (c) run 3; and (d) run 4.

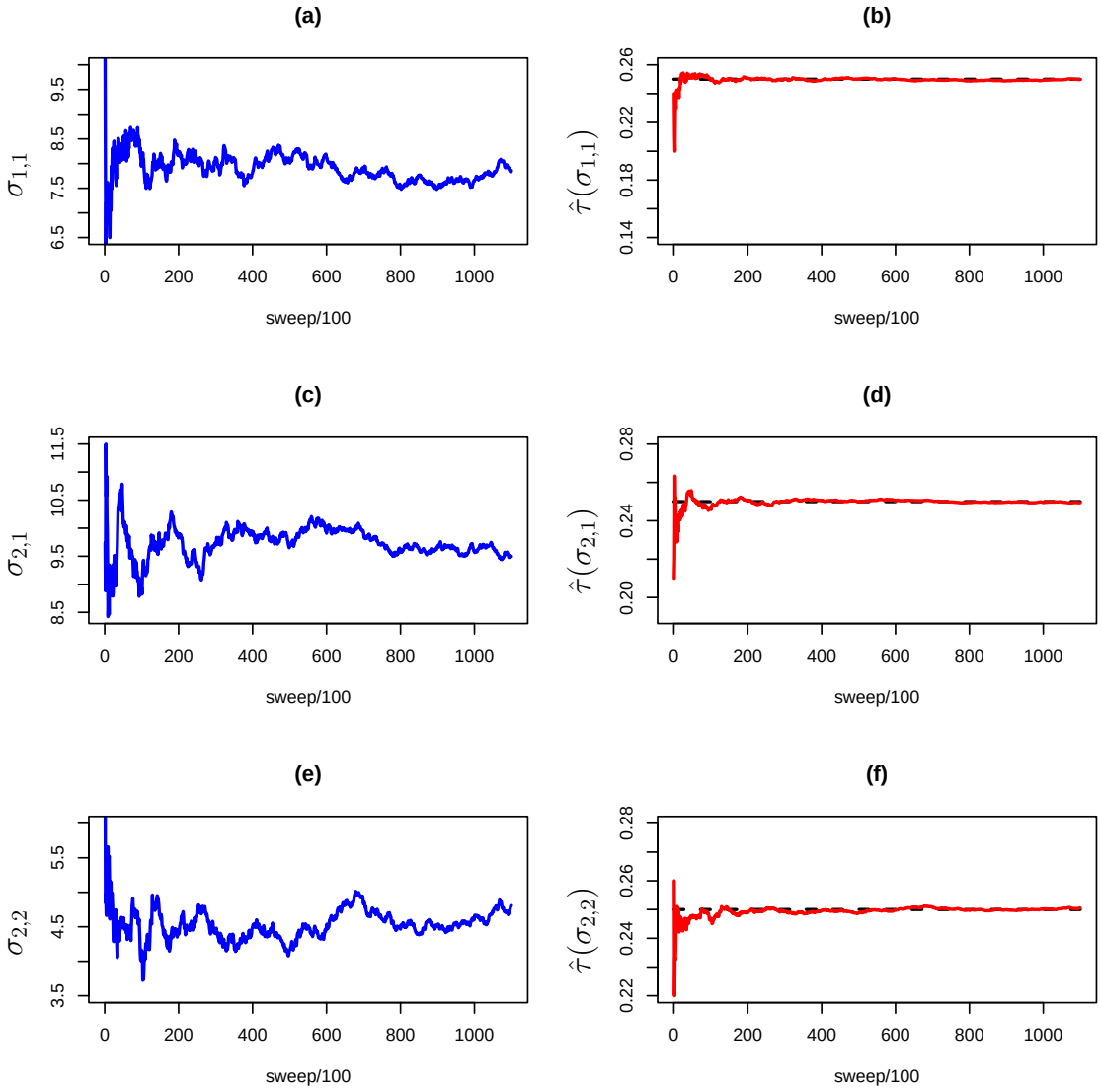

Figure 5.4: Evolution of RWM parameters and target functions using AAP algorithm described in chapter 4: (a) RWM scaling parameter $\sigma_{1,1}$ for model 1; (b) average acceptance probability $\hat{\tau}(\sigma_{1,1})$ of RWM for model 1; (c) RWM scaling parameter $\sigma_{2,1}$ for component 1 of model 2; (d) average acceptance probability $\hat{\tau}(\sigma_{2,1})$ of RWM for component 1 of model 2; (e) RWM scaling parameter $\sigma_{2,2}$ for component 2 of model 2; and (f) average acceptance probability $\hat{\tau}(\sigma_{2,2})$ of RWM for component 2 of model 2.

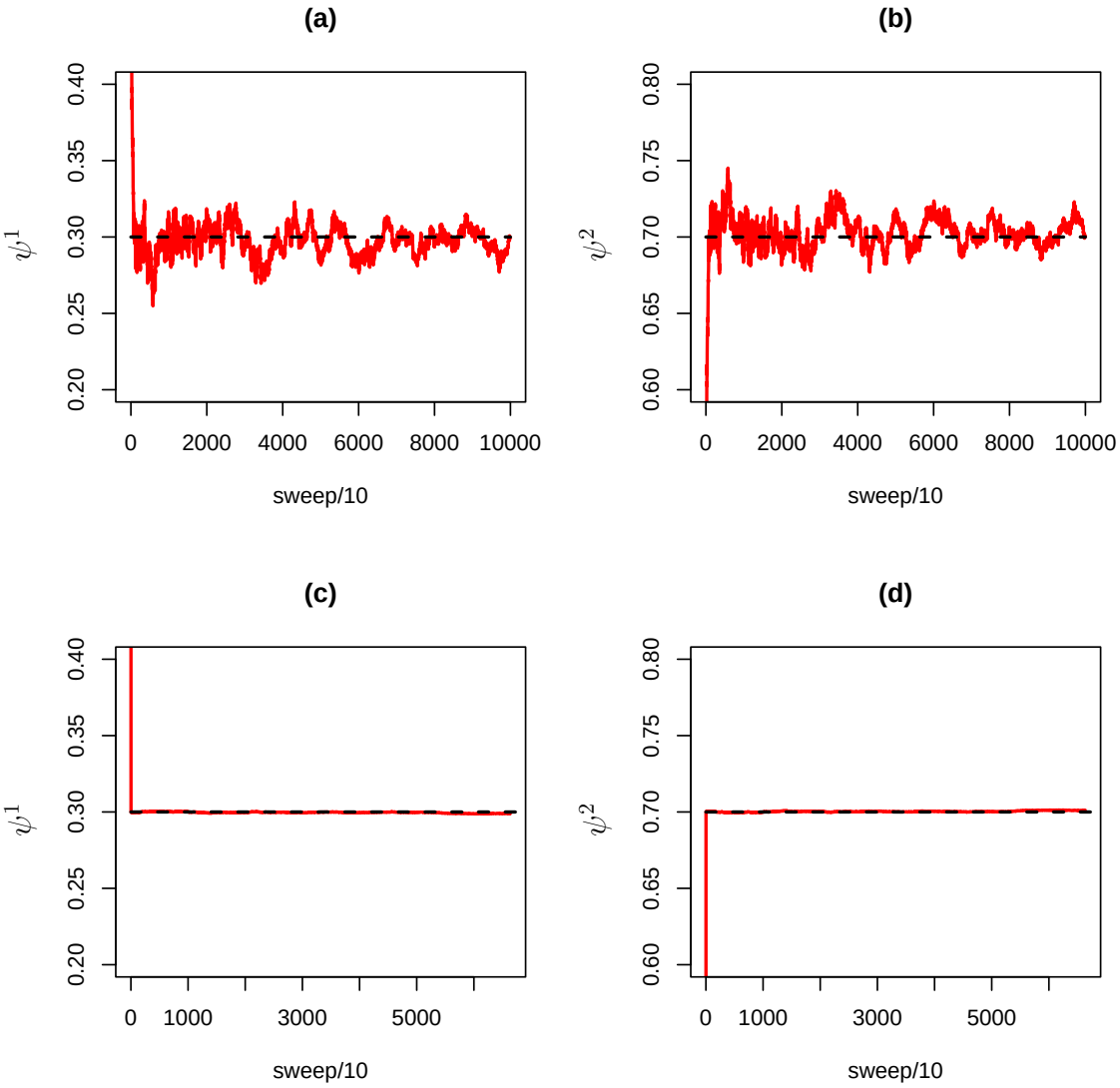

Figure 5.5: Evolution of reversible jump proposal probabilities: (a) ψ^1 (probability of proposing a jump to model 1), adapted using diminishing adaptation algorithm B, described in section 4.6; (b) ψ^2 (probability of proposing a jump to model 2), adapted using algorithm B; (c) ψ^1 (probability of proposing a jump to model 1), adapted using adaptation through regeneration algorithm A, described in section 4.6; and (d) ψ^2 (probability of proposing a jump to model 2), adapted using algorithm A.

or 5 degrees of freedom) $\boldsymbol{u}$ instead of Gaussian variables (i.e. setting t=1,2, or 5), we see very little departure from the results reported above. Employing our equivalent of Green's AutoRJ algorithm (using the adaptive RWM in the first stage), the acceptance probability decreases to 77% when a=1 and 67% otherwise.

The integrated autocorrelation time for the model index chain, ranged between 3.75 and 4.75 (calculated using Sokal's method, Green and Han, 1992) for our version of the AutoRJ sampler.[2] If instead the AutoMix sampler fitted a mixture to each of the conditionals this decreased to between 1.11 and 1.15, a range that is remarkably low for a dependent sample. Nonetheless, the run time for the mixture fitting AutoMix sampler was around 125 seconds[3], whereas this time was typically less than 20 seconds for our version of the AutoRJ sampler. By not doing adaptation at the reversible jump stage less than 10 seconds were saved.

Multiplying the run time and the autocorrelation time gives a measure of efficiency (with low values being more efficient). For the AutoMix sampler this value is approximately $1.13 \times 125 = 141.25$, whereas for the AutoRJ sampler the value is $4.25 \times 20 = 85$. For this simple problem, fitting mixtures is less efficient than just estimating parameters $\boldsymbol{\mu}_k$ and B_k for each k. However, we remark that this conclusion is only based upon measuring autocorrelation time for the model index k and we expect the mixture fitting feature of the AutoMix sampler to also reduce autocorrelation in the $\boldsymbol{\theta}_k$. Furthermore, if the AutoMix sampler is run in mode m=1 after a mixture has been fitted in a previous run, then the run

[2]We note that this autocorrelation time has no meaning in terms of posterior inference (since k is categorical rather than ordinal). However, the measure still provides an indication of trans-dimensional performance, with a low autocorrelation time being a desirable property.

[3]All times quoted are from running the AutoMix algorithm on a machine with a 1600MHz Intel Pentium M processor. All programs were compiled using the gcc GNU compiler, with optimisation level 3.

time decreases to below 20 seconds, whereas the decrease is minimal if a single Normal had been fitted in the previous run.

We end this section by noting that the high acceptance probability and good performance for the mixture version of the AutoMix sampler should come as no surprise, since by design the conditional distributions of this simple posterior are both mixture distributions. Therefore if our fitted mixtures are good approximations to these mixtures then our reversible jump sampler is almost sampling directly from the target distribution. In the following sections we look at more complicated examples which provide a more robust test of the AutoMix algorithm.

5.5.2 A Change Point Example

The second example to which we apply the AutoMix sampler is a change point analysis for a point process. In particular we consider the model when such a process is applied to the coal mine disaster data used in the original reversible jump paper by Green (1995). This stochastic model assumes that coal mining disasters occur as a Poisson process with a piecewise constant rate. The object of the analysis is to estimate how many changes in rate occur (the number of change points), the times at which they occur and the constant rate values between changes.

Recall from section 2.3 that this problem is easily framed as a generic model jumping problem by letting the model index k correspond to the number of change points. Thus, for model k, the parameters $\boldsymbol{\theta}_k$ consist of $k+1$ rates and k change points or times, meaning that model k has $2k+1$ parameters in total.

The coal mining change point analysis was used as a specific non-trivial example to demonstrate the original automatic sampler proposed in Green (2003). Consistent with Green (2003) we restrict our attention to models 1 to 6. In other words we impose the condition that the number of change points must be between 1 and 6 (inclusive). Otherwise we adopt the original settings and prior parameter used by Green (1995).

As noted by Green (2003) the original model probabilities reported by Green (1995) are not reliable due to the subsequent discovery that the sampler requires longer to converge than allowed in this original paper. Thus we compare our model to the more recent results presented by Green (2003) which the author confirms match the results obtained from running the Green (1995) algorithm for sufficiently many sweeps. In particular, we compare the model probabilities to the values (0.058,0.251,0.294,0.236,0.117,0.044) reported in Green (2003). Also of interest is the performance of the AutoMix sampler in comparison to both the problem-specific and generic samplers presented by Green (1995) and Green (2003) respectively.

Green reports a run time of 28 seconds (on an 800MHz PC) to do 1 million sweeps for the automatic sampler. In fact, to accurately obtain the reported model probabilities, Green's sampler must be run for 10 million sweeps (noting that the sampler only uses a sub-sample taken at every tenth sweep). Testing the run time of the exact AutoRJ algorithm (kindly provided by the author) on the same computer as the AutoMix sampler runs were performed, the run time is 2 minutes and 29 seconds. However, this forgets that this original 'automatic' sampler has been provided with good scaling parameters

for the initial within-model RWM stage. By applying adaptive RWM, the AutoMix sampler does not require such specialist user input and so in this respect is much more automatic. Thus it is fairer to compare the full mixture fitting AutoMix sampler against the AutoMix equivalent to Green's sampler.

To run the AutoMix mixture fitting sampler for 1 million sweeps (N= 1000000) takes around 11 minutes. The resulting posterior model probabilities are (0.058,0.250,0.296,0.234,0.118,0.044) which are almost identical to those reported by Green. The posterior densities for the change points and rate parameters, for models $k \in \{1, 2, 3\}$ are shown in figures 5.6 and 5.7. These plots demonstrate similar properties to the equivalent plots presented in Green (1995). The integrated autocorrelation in the model index chain is estimated to be approximately 38, which is considerably less than the figures quoted by Green (2003) for his automatic sampler (118) or the original problem-specific sampler (67.8).

A subsequent run of the AutoMix sampler in mode m=1, (i.e. using the mixture parameters from a previous run), again with 1 million sweeps, takes around 9.5 minutes. As we would expect the resulting autocorrelation time and posterior probabilities are unchanged.

Very similar posterior model probabilities arise from running the AutoMix sampler when only a single $\boldsymbol{\mu}_k$ and B_k is estimated for each model. Such a run, with 1 million sweeps, also takes just under 9.5 minutes, and the autocorrelation increases to 84. Multiplying run times and autocorrelation, the relative efficiency of the mixture fitting version to the AutoRJ version is approximately $(84 \times 565)/(38 \times 660) \simeq 189\%$, meaning it is better to use the mixture fitting

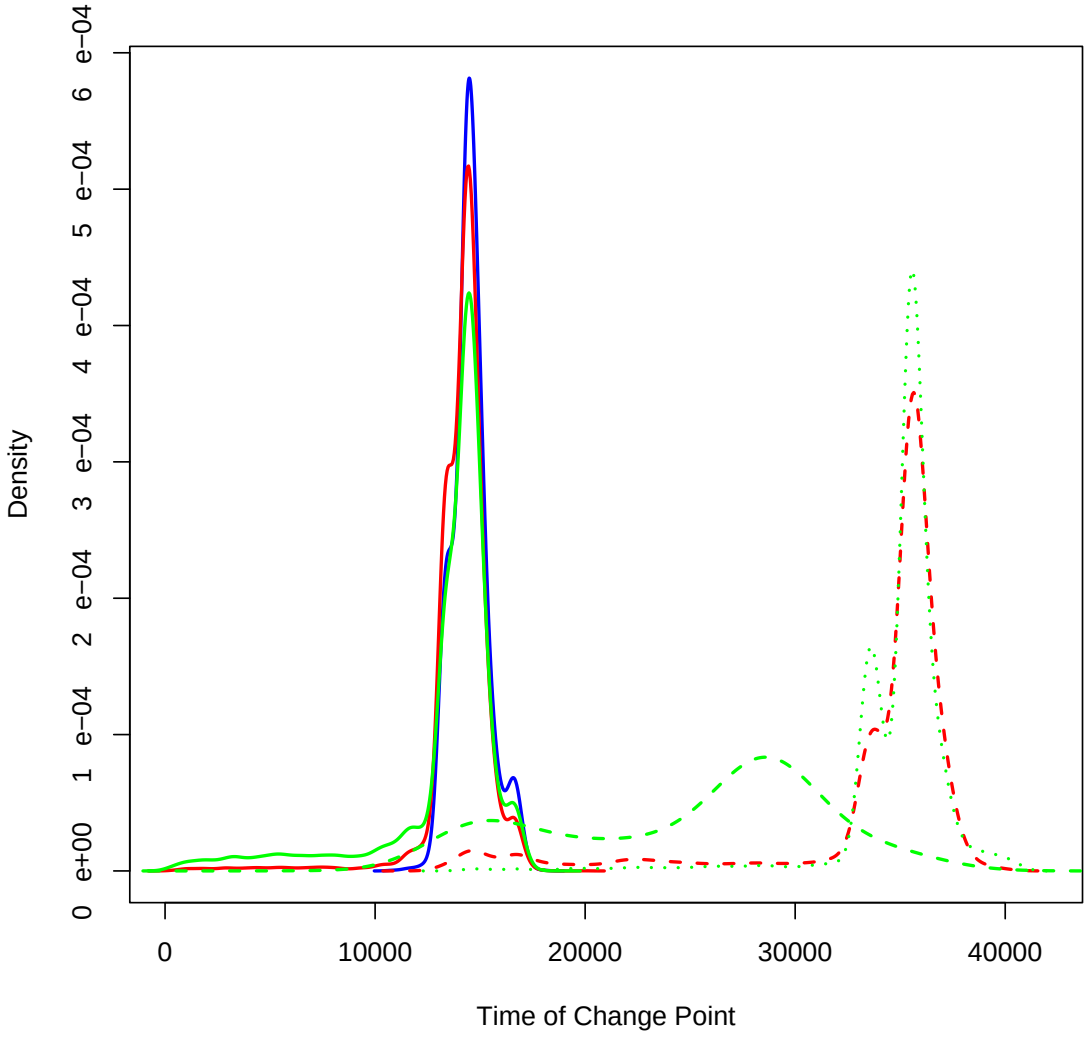

Figure 5.6: Estimated marginal posterior densities for change points for models with 1 (blue line), 2 (red lines), and 3 (green lines) change points. For models with more than one change point different marginal densities are denoted by different line styles.

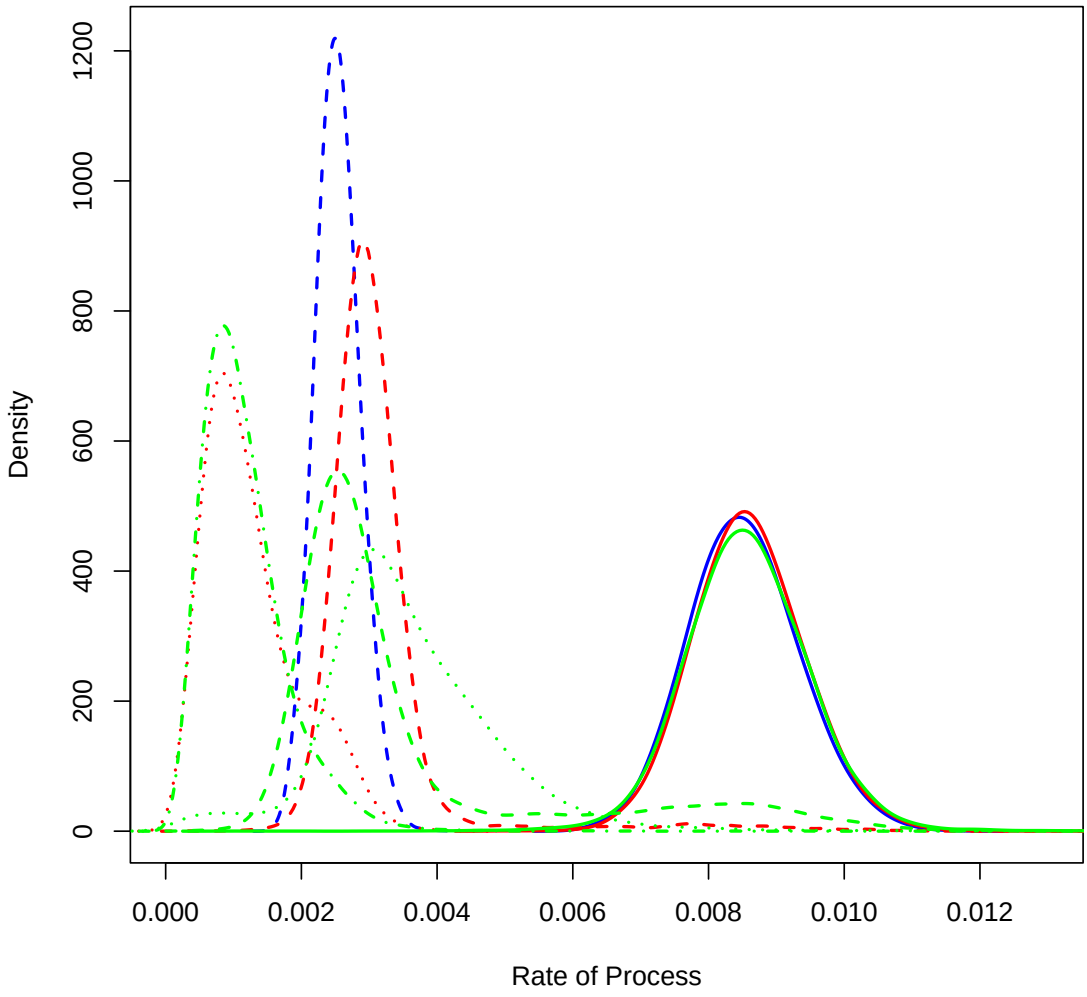

Figure 5.7: Estimated marginal posterior densities for rates of coal mine disasters for models with 1 (blue lines), 2 (red lines), and 3 (green lines) change points. Different marginal densities are denoted by different line styles.

version for this problem.

Before leaving our discussion of run times it is perhaps worth commenting about the large difference in run time between Green's sampler and the equivalent AutoMix sampler. As mentioned above some of the difference lies in the adaptation of the AutoMix sampler that is not a feature of Green's sampler. However, this does not account for a run time of just under 4 times that of the AutoRJ sampler. Much of this time is down to the extra generality included in the design of the AutoMix sampler, in particular the extra output files produced by the AutoMix sampler to allow more detailed and varied analysis. Nonetheless, we appreciate that the code of the AutoMix sampler could no doubt be considerably optimised if it was recoded by a specialist in C programming.

The trans-dimensional acceptance rate of the AutoMix sampler when fitting mixtures is approximately 26% which is in fact higher than the 21% that was achieved by Green's original problem-specific sampler. The acceptance rate decreases to 21% if no adaptation is done in the third phase. Interestingly, running our version of the AutoRJ sampler, the acceptance rate actually increases to about 28%. However, this rate is artificially high. This is because reversible jump moves that propose the current model as the new model and then accept the move, contribute to the trans-dimensional acceptance rate. While this seems reasonable for the full AutoMix sampler (as the use of mixture proposals allows moves between different parameter states of the same model), for the AutoRJ version the proposed new state is identical to the current state and therefore such proposals are always accepted.

The AutoMix results quoted above are all from runs when permutation does

not take place in the reversible jump move. If permutation is enabled, the full AutoMix sampler only achieves reversible jump acceptance rates of 16%. For such runs the associated autocorrelation time is 32, which is less than when permutation is not enabled.

The higher acceptance rates that occur when permutation is not used require further comment. Further investigation suggests that the phenomenon occurs specifically for permutation between the small scale rate parameters (where all posterior mass is on the range 0 to 0.015) and the large scale change point times (where almost all posterior mass lies between 5000 and 45000). To test this hypothesis we ran the sampler on the same problem but with rescaled time units, so that the rate parameters and change point times were on approximately the same scale. The posterior inference was unchanged but the sampler achieved 34% acceptance rates (with adaptation in the RJ phase). This did not alter when permutation was employed.

Comparison of the original and rescaled problems reveals that the AAP algorithm used in the within-model random walk phase is sensitive to scaling. Returning to the unscaled problem and looking at all models, the adaptive algorithm converges with no problem for the rate parameters. However, for the change point times the algorithm demonstrates exceptionally slow convergence, to the extent that by the end of the RWM phase the average acceptance rate for these components of $\boldsymbol{\theta}_k$ remains significantly different from the target value of 0.25. Figure 5.8 shows this phenomenon for the model with 1 change point. For the change point component, the average acceptance probability for the RWM jumps only decreases from 93% to 53%. In comparison, figure 5.9 shows that the problem does not occur for the rescaled problem. This sensitivity appears

to result in the fitted mixtures from the second stage approximating some conditionals much worse than others. This seems to explain why permutation reduces acceptance rates in the original problem.

Looking more closely at the AAP algorithm we see that the sensitivity to scale occurs because (contrary to the claims of Atchadé and Rosenthal) the algorithm is sensitive to the choice of σ_0, the initial value of the adaptive RWM scale parameter. This sensitivity is likely to be partly attributable to the fact that using the algorithm recommended by the authors (see section 4.2, equations 4.1 and 4.2), the increments of the adaptive parameter depend upon σ_0. In the case of the change point problem, for the components of $\boldsymbol{\theta}_k$ that correspond to change point times the value of σ_0 is too small for rapid convergence.

Solving the issue of sensitivity to σ_0 may automatically improve the acceptance rates for the AutoMix sampler. The question of how to find an automatic value of σ_0 that is suitable for the problem under study remains a subject for further research. However, we note that despite the problems of the AAP algorithm, the flexibility of the AutoMix sampler to enable or disable various features (such as permutation) means that for this problem, this generically designed sampler achieves better acceptance rates and lower autocorrelation times than any previous MCMC sampler.

5.5.3 A Return to the Rb9 Problem

In this section we demonstrate how the AutoMix sampler can be used to make inference about the Rb9 problem that we introduced in detail in chapter 3. This section provides a contrast to the earlier implementation of reversible jump for

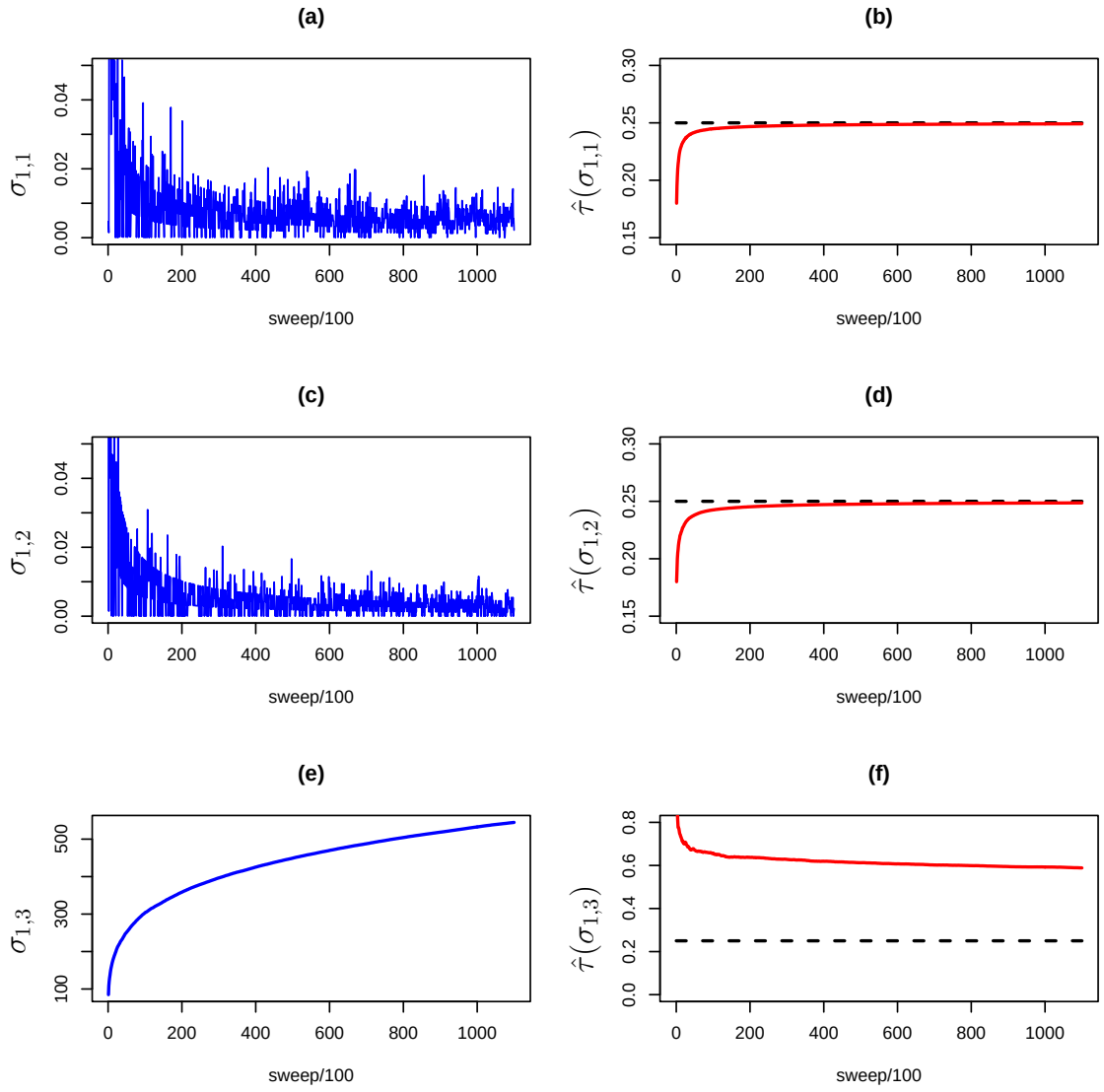

Figure 5.8: Evolution of RWM parameters and target functions using AAP algorithm, for the original change point problem: (a) RWM scaling parameter $\sigma_{1,1}$ for component 1 of model 1; (b) average acceptance probability $\hat{\tau}(\sigma_{1,1})$ of RWM for component 1 of model 1; (c) RWM scaling parameter $\sigma_{1,2}$ for component 2 of model 1; (d) average acceptance probability $\hat{\tau}(\sigma_{1,2})$ of RWM for component 2 of model 1; (e) RWM scaling parameter $\sigma_{1,3}$ for component 3 of model 1; and (f) average acceptance probability $\hat{\tau}(\sigma_{1,3})$ of RWM for component 3 of model 1.

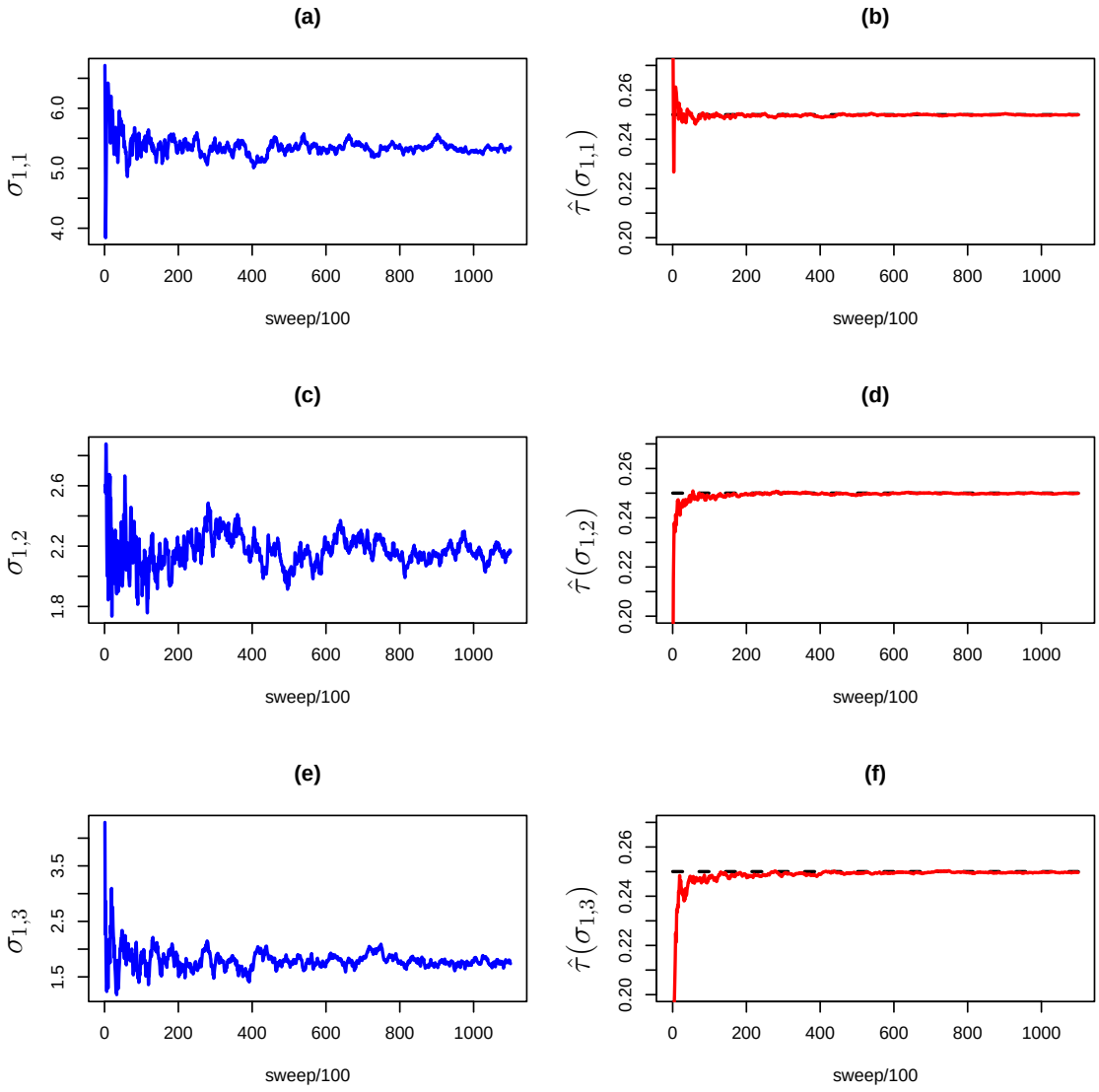

Figure 5.9: Evolution of RWM parameters and target functions using AAP algorithm, for the rescaled change point problem: (a) RWM scaling parameter $\sigma_{1,1}$ for component 1 of model 1; (b) average acceptance probability $\hat{\tau}(\sigma_{1,1})$ of RWM for component 1 of model 1; (c) RWM scaling parameter $\sigma_{1,2}$ for component 2 of model 1; (d) average acceptance probability $\hat{\tau}(\sigma_{1,2})$ of RWM for component 2 of model 1; (e) RWM scaling parameter $\sigma_{1,3}$ for component 3 of model 1; and (f) average acceptance probability $\hat{\tau}(\sigma_{1,3})$ of RWM for component 3 of model 1.

this problem and highlights how the automatic approach completely removes the complicated design phase required for problem-specific samplers. Throughout the section we consider only the Rb9 problem with prior A (see section 3.2).

Our experience has found that using the AutoMix sampler to consider many models results in an infeasibly long run time. Therefore, for illustration we apply the AutoMix sampler to a reduced problem where the aim is to choose between one of the ten models listed in table 5.2. The ten models that we restrict our attention to are such that the first 9 have high posterior probability (see section 3.4) and the last has a special meaning which has been of interest in other literature (Newton and Hastie, 2004).

Having selected a subset of 10 models, all that is required is a single user file containing the basic C functions as described in section 5.4.2. Such functions take very little time to create assuming familiarity with writing C code. Once such a file has been written the AutoMix sampler can be immediately applied.

To run the AutoMix sampler with 100000 stage 1 Normal random walk iterations per model and 100000 adaptive iterations in the reversible jump stage takes around 15 minutes. We do not enable permutation. The associated autocorrelation time for the model index chain is approximately 1.35. The reversible jump acceptance probability is 84%. Considering the arguments presented in section 5.5.2, this suggests that the mixtures fitted to the conditionals $\pi(\boldsymbol{\theta}_k|k)$ fit equally well in all directions. This is supported by a third stage RWM acceptance rate of 25% for all runs, showing that the first stage algorithm appears to have converged. This appears to be confirmed by figure 5.10 which shows the evolution of the adaptive parameters for a typical model.

sub-model index		l pattern					k pattern				Posterior probability	
$m = (l, k)$	$i =$	1	2	3	4	$i =$	1	2	3	4	Chapter 3	AutoMix
(12 , 5)		1	2	2	3		1	0	0	0	0.233	0.239
(12 , 9)		1	2	2	3		1	0	0	1	0.228	0.232
(12 , 13)		1	2	2	3		1	0	1	1	0.091	0.084
(12 , 14)		1	2	2	3		1	1	0	1	0.077	0.078
(14 , 5)		1	2	3	3		1	0	0	0	0.052	0.053
(14 , 9)		1	2	3	3		1	0	0	1	0.096	0.095
(12 , 20)		1	2	2	3		1	0	0	2	0.087	0.086
(15 , 5)		1	2	3	4		1	0	0	0	0.065	0.063
(15 , 9)		1	2	3	4		1	0	0	1	0.063	0.062
(15 , 16)		1	2	3	4		1	1	1	1	0.008	0.008

Table 5.2: Posterior probabilities for 10 sub-models of the Rb9 problem. Probabilities are calculated using the problem-specific sampler introduced in chapter 3 and the AutoMix sampler.

Running the AutoMix sampler with the same settings as above, but estimating a single $\boldsymbol{\mu}_k$ and B_k for each model instead of fitting a mixture reduces the acceptance probability to 80% and increases the autocorrelation to around 1.85, but takes just under 6.5 minutes to run. These statistics suggest that it is not necessary to fit mixtures for this problem.

Table 5.2 compares the posterior model probabilities resulting from the mixture fitting AutoMix sampler (averaged across two separate runs, with Monte Carlo error estimated by blocking to be less than 0.005) and those obtained by renormalising the model probabilities that occurred from the problem-specific sampler. These probabilities are clearly very similar. Figure 5.11 shows that samples from the marginal distributions (across models) of the mean parameters $\boldsymbol{\lambda}$ are similar to those shown in figure 3.2 in section 3.4.

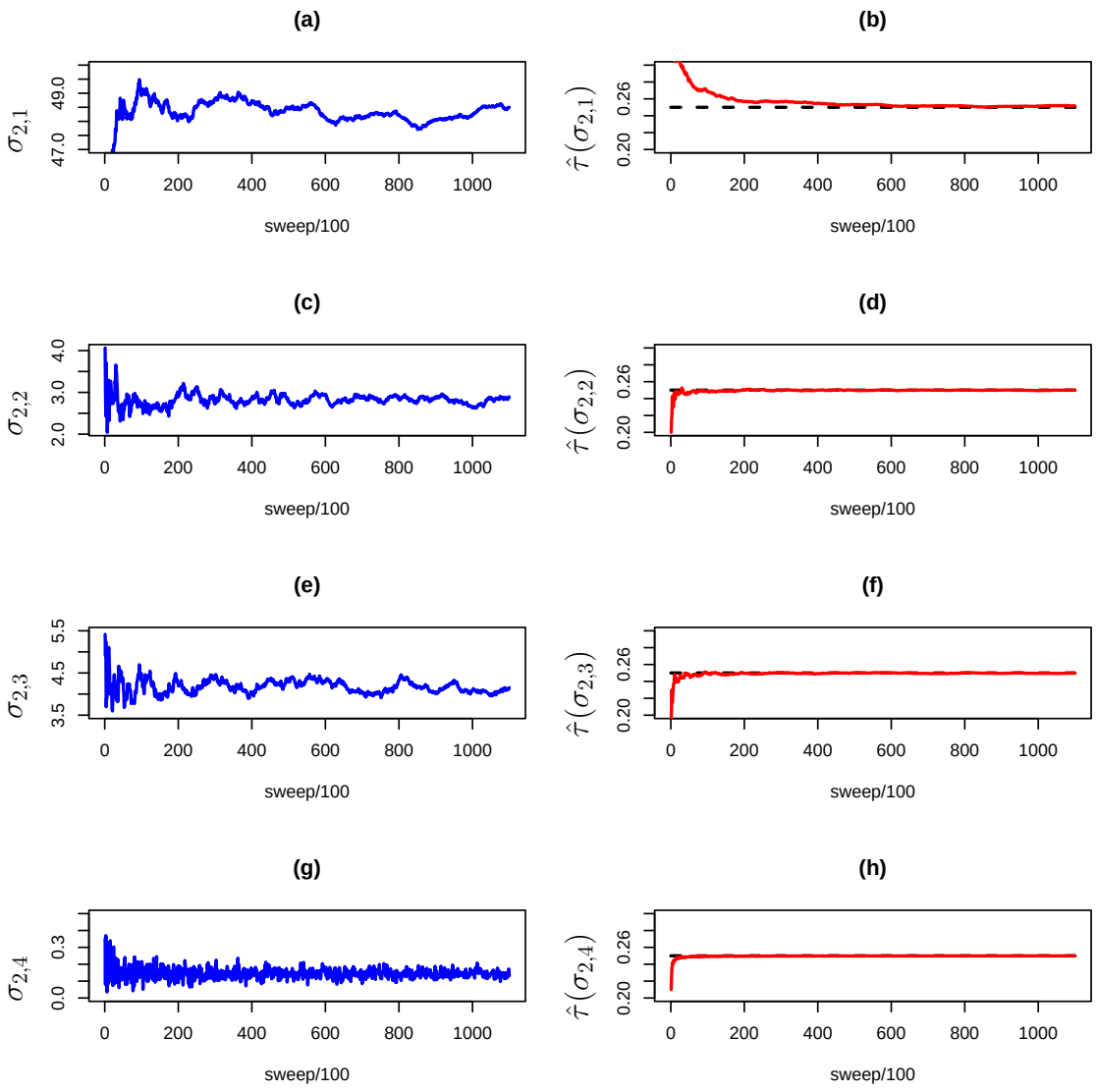

Figure 5.10: Evolution of model 2 RWM parameters and target functions using AAP algorithm, for the Rb9 problem: (a) RWM scaling parameter $\sigma_{2,1}$ for component 1 of model 2; (b) average acceptance probability $\hat{\tau}(\sigma_{2,1})$ of RWM for component 1 of model 2; (c) RWM scaling parameter $\sigma_{2,2}$ for component 2 of model 2; (d) average acceptance probability $\hat{\tau}(\sigma_{2,2})$ of RWM for component 2 of model 2; (e) RWM scaling parameter $\sigma_{2,3}$ for component 3 of model 2; (f) average acceptance probability $\hat{\tau}(\sigma_{2,3})$ of RWM for component 3 of model 2; (g) RWM scaling parameter $\sigma_{2,4}$ for component 4 of model 2; and (h) average acceptance probability $\hat{\tau}(\sigma_{2,4})$ of RWM for component 4 of model 2.

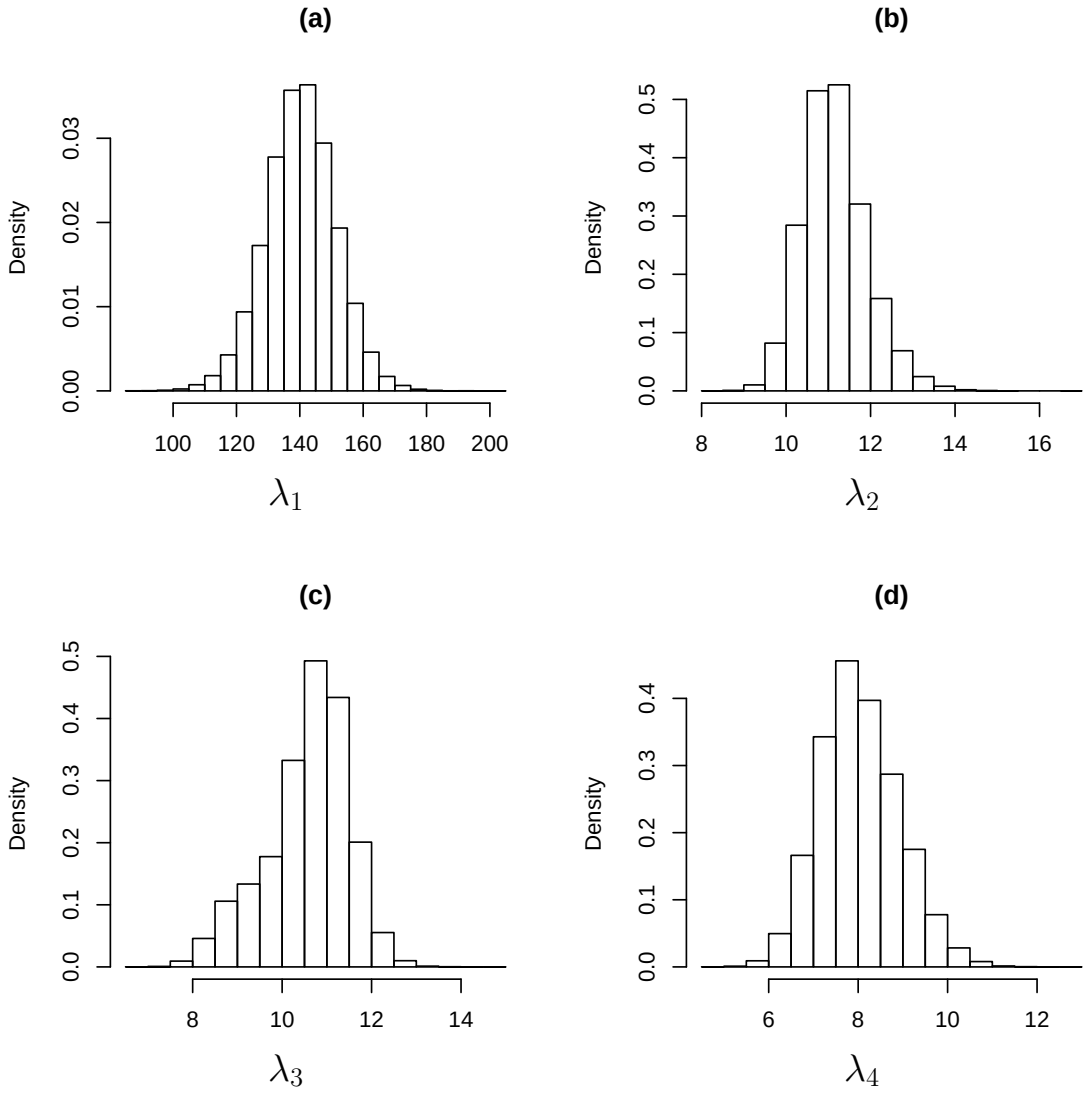

Figure 5.11: Histograms of marginal posterior samples of the components of $\boldsymbol{\lambda}$ (prior A): (a) +/+; (b) Rb9 trans; (c) Rb9 cis; and (d) Rb9/Rb9.

Comparing trans-dimensional acceptance rates between the problem-specific sampler and the AutoMix sampler is not very instructive because the problem-specific sampler is allowed to propose moves into a much larger model space, which includes models with almost zero posterior probability. Nonetheless, the reversible jump acceptance rate for the AutoMix sampler is very impressive. We can also conclude from this section that implementation is considerably easier for the AutoMix sampler than for the sampler specifically designed for the Rb9 problem. However, we emphasise that to apply the AutoMix sampler we had to reduce the number of models under consideration.

5.5.4 An AIDS Clinical Trial

In this section we apply the AutoMix sampler to an example arising from an AIDS clinical trial introduced by Abrams *et al.* (1994). This problem was studied by Han and Carlin (2001) to provide a comparison of various statistical methods for problems with a trans-dimensional nature. We choose this problem specifically because Han and Carlin report considerable difficulty in implementing a reversible jump sampler, identifying the main difficulty as the tuning that is required to achieve satisfactory performance. As shown below, the AutoMix sampler has no difficulty in making inference about this problem, avoiding any need for tuning runs. Furthermore, we show that the AutoMix sampler does not require one of the levels of marginalisation that Han and Carlin have to impose to get their sampler to work. Before reporting the performance of the AutoMix sampler, we briefly summarise the problem as presented by Han and Carlin.

We consider the clinical trial presented by Abrams *et al.*, involving a longitudinal study consisting of 467 patients, with the purpose of making inference about

two possible treatments. The trial involved measuring each patient's CD4 lymphocyte count at 0, 2, 6, 12, and 18 months. Also recorded was an indicator of the patients' treatment group, either didanosine (ddI) or zalcitabine (ddC), and an indicator of whether the patient had a positive AIDS diagnosis at the beginning of the study. Importantly, there are many missing observations due to death or loss of patient during the study. Indeed, only 5% of patients have CD4 counts for all 5 times.

Following the approach of Han and Carlin, we look at the square root of the CD4 counts, denoting the random variable for the i^{th} patient at the j^{th} monitoring point as $Y_{i,j}$, for $i = 1, 2, \ldots, 467$ and $j = 1, \ldots, s_i$, where s_i is the number of observations made for patient i. The two models that we compare, which we label 1 and 2, are both mixed effects models with the difference that the first permits a change point in the observed CD4 counts at the 2 month stage. In particular, the first model accommodates the possibility of an increase in the CD4 count which is the desired clinical effect of the two drugs.

Mathematically, if we denote the i^{th} patient's observation vector by $\boldsymbol{Y}_i = (Y_{i,1}, Y_{i,2}, \ldots, Y_{i,s_i})^T$, then model 1 can be written as

$$\boldsymbol{Y}_i = X_i\boldsymbol{\alpha} + W_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i, \;\; \boldsymbol{\beta}_i \sim^{i.i.d.} \mathrm{N}(\mathbf{0}, V), \;\; \boldsymbol{\epsilon}_i \sim^{i.i.d.} \mathrm{N}(\mathbf{0}, \sigma^2 I_{s_i}), \tag{5.16}$$

and model 2 can be written as

$$\boldsymbol{Y}_i = \tilde{X}_i\tilde{\boldsymbol{\alpha}} + \tilde{W}_i\tilde{\boldsymbol{\beta}}_i + \tilde{\boldsymbol{\epsilon}}_i, \;\; \tilde{\boldsymbol{\beta}}_i \sim^{i.i.d.} \mathrm{N}(\mathbf{0}, \tilde{V}), \;\; \tilde{\boldsymbol{\epsilon}}_i \sim^{i.i.d.} \mathrm{N}(\mathbf{0}, \tilde{\sigma}^2 I_{s_i}). \tag{5.17}$$

In the first model W_i is an $s_i \times 3$ design matrix, with j^{th} row given by $(1, t_{i,j}, \max\{0, t_{i,j} - 2\})$, where $t_{i,j} \in \{0, 2, 6, 12, 18\}$ is the time of the j^{th} observation made on patient i. The matrix X_i is given by $[W_i|(z_{i,1}W_i)|(z_{i,2}W_i)]$,

where $z_{i,1} = 1$ if the i^{th} patient is in the ddI treatment group and 0 otherwise and $z_{i,2} = 1$ if the patient has a positive AIDS diagnosis at the beginning of the trial and 0 otherwise. Similarly, for model 2, $\tilde{W}_i$ is an $s_i \times 2$ design matrix, with j^{th} row given by $(1, t_{i,j})$ and $\tilde{X}_i = [\tilde{W}_i|(z_{i,1}\tilde{W}_i)|(z_{i,2}\tilde{W}_i)]$. For further details on the statistical models the reader is directed to Han and Carlin (2001), Chib and Carlin (1999) or Carlin and Louis (2000).

Approaching the problem from a Bayesian perspective, we have a simple two model choice problem, for which reversible jump is ideally suited. Before proceeding however, we simplify the likelihood by the approach advocated by Chib and Carlin (1999) and marginalise the two models over the random effects, giving

$$\boldsymbol{Y}_i \sim \mathrm{N}(X_i\boldsymbol{\alpha}, W_i V W_i^T)$$

under model 1 and

$$\boldsymbol{Y}_i \sim \mathrm{N}(\tilde{X}_i\tilde{\boldsymbol{\alpha}}, \tilde{W}_i\tilde{V}\tilde{W}_i^T)$$

under model 2. This marginalisation is used by Han and Carlin (2001) for the marginal likelihood method that they advocate in this paper (see Han and Carlin for a review). However, to obtain a working reversible jump algorithm Han and Carlin also recommend marginalising over $\boldsymbol{\alpha}$ and $\tilde{\boldsymbol{\alpha}}$. We do not require this additional level of marginalisation for the AutoMix sampler, thus allowing a more direct comparison with the marginal likelihood method.

In order to progress we must specify the prior distribution for $\boldsymbol{\theta}_1 = (\alpha_1, \ldots, \alpha_9, V_{1,1}^{-1}, V_{2,1}^{-1}, V_{2,2}^{-1}, V_{3,1}^{-1}, V_{3,2}^{-1}, V_{3,3}^{-1}, \sigma^2)$ and $\boldsymbol{\theta}_2 = (\tilde{\alpha}_1, \ldots, \tilde{\alpha}_6, \tilde{V}_{1,1}^{-1}, \tilde{V}_{2,1}^{-1}, \tilde{V}_{2,2}^{-1}, \tilde{\sigma}^2)$. We adopt identical priors to those used by Han and Carlin which are themselves motivated by Carlin and Louis (2000). In particular, we emphasise that we put priors (and therefore make posterior

inference) on the inverse of the covariance matrices V and $\tilde{V}$. This was originally done to achieve conjugacy of the full conditional distributions. Although the AutoMix sampler does not take advantage of this conjugacy, we retain the same prior for direct comparison.

With the statistical model choice problem formulated it is easy to apply the AutoMix sampler to make inference. Again, this requires little more than a simple C function to evaluate the log of the posterior distribution (up to an additive constant) at any point in the sample space. We run the sampler with n= 100000 and N= 100000, enabling adaptation but not permutation in the reversible jump phase.

Running the sampler takes just under 5 hours and results in an autocorrelation time for the model index chain of between 1.8 and 2.5. If the sampler is subsequently run with m=1, the run time decreases to just over an hour. No run or autocorrelation times are reported for the methods run by Han and Carlin (2001) so we are not able to compare. The long run times for this simple two model example are caused mainly by the expensive evaluation of the likelihood and possibly could be reduced by looking for simplifications. We have not investigated this further. The time per evaluation is compounded by the dimensions of the two models (16 and 9 respectively). Since all RWM is carried out in a componentwise fashion in the AutoMix sampler, increasing dimensions becomes increasingly expensive.

Although the AutoMix sampler is expensive to run, it performs well for this problem. Figure 5.12 shows the trace plots of the log posterior and log likelihood values during the reversible jump phase of the algorithm, providing evidence

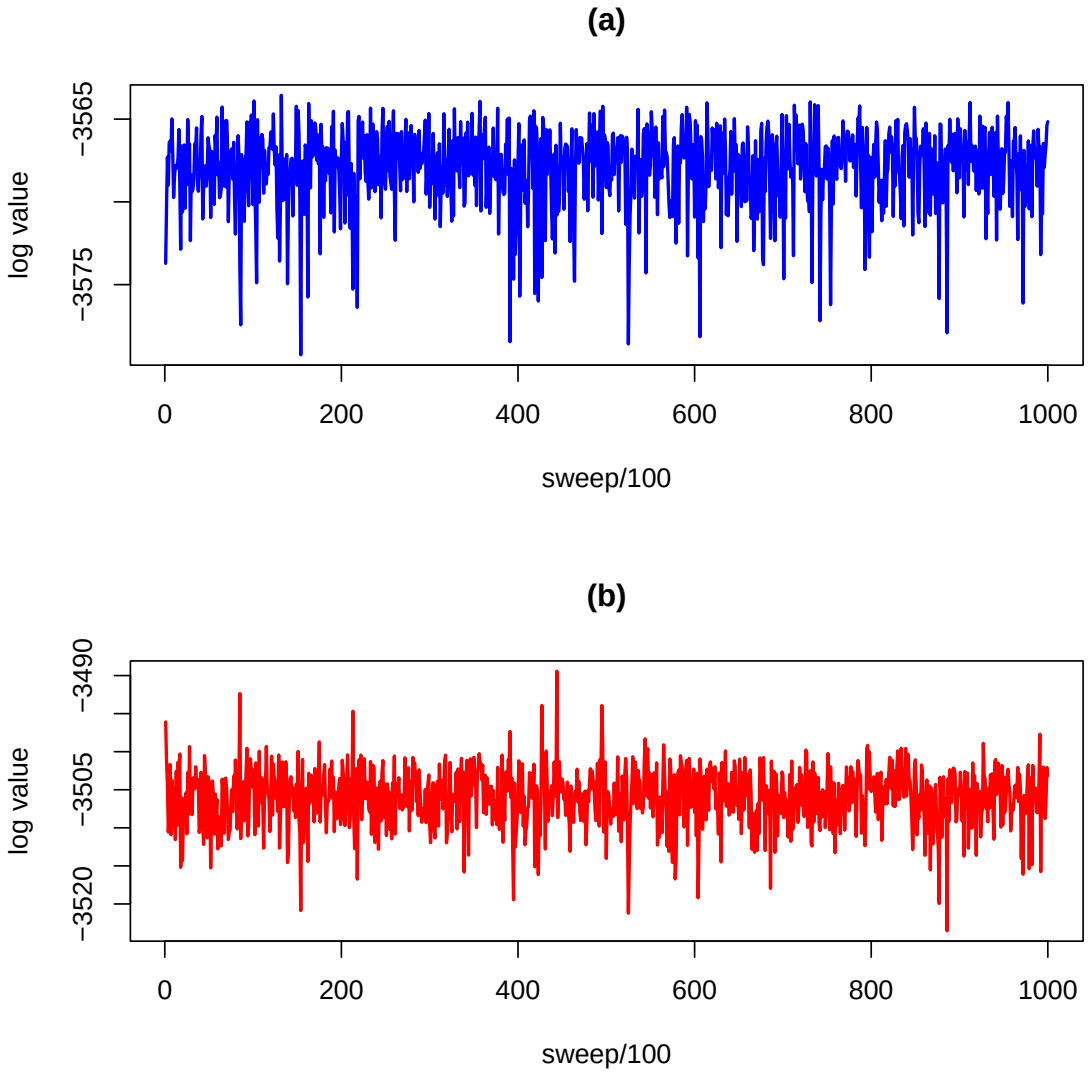

Figure 5.12: Trace plots from the AutoMix sampler: (a) log posterior; and (b) log likelihood.

of the apparent good mixing of the AutoMix sampler. Once again the trans-dimensional acceptance rate of the chain is high, at around 86%.

Figure 5.13 shows the convergence of the adaptive reversible jump model jumping parameters. As can be seen from these plots, the adaptive probabilities are reinitialised several times during the run. As discussed in chapter 4, this reprojection is necessary whenever the adaptive parameters escape a compact set (which is also allowed to adapt over the course of the run). This is a feature of the general diminishing adaptation algorithm introduced by Andrieu and Moulines (2004) and Andrieu *et al.* (2004) and ensures the convergence of the parameters. For this particular problem reprojection occurs because of the small posterior probability of model 1 (see below).

The AAP algorithm in the first phase also performs well, meaning that the average acceptance rate of the RWM in the third phase is 25%. The convergence of the RWM is illustrated in figure 5.13 for two parameters from each model.

Most importantly, the sampler allows posterior inference to be made about the above model. In particular, the average posterior model probabilities (across 2 independent MCMC runs, with Monte Carlo errors estimated by blocking of less than 0.001) are 0.012865 for model 1 and 0.987135 for model 2. This gives an estimate of 76.73 for the Bayes factor of model 2 over model 1 (see for example Kass and Wasserman, 1995 or chapter 3), meaning there is strong posterior evidence that the simpler model 2 better explains the data. This estimate of the Bayes factor agrees with the estimates of 75.71 and 76.74 reported by Han and Carlin for the reduced reversible jump sampler and the marginal likelihood methods that they compare. We emphasise again that the

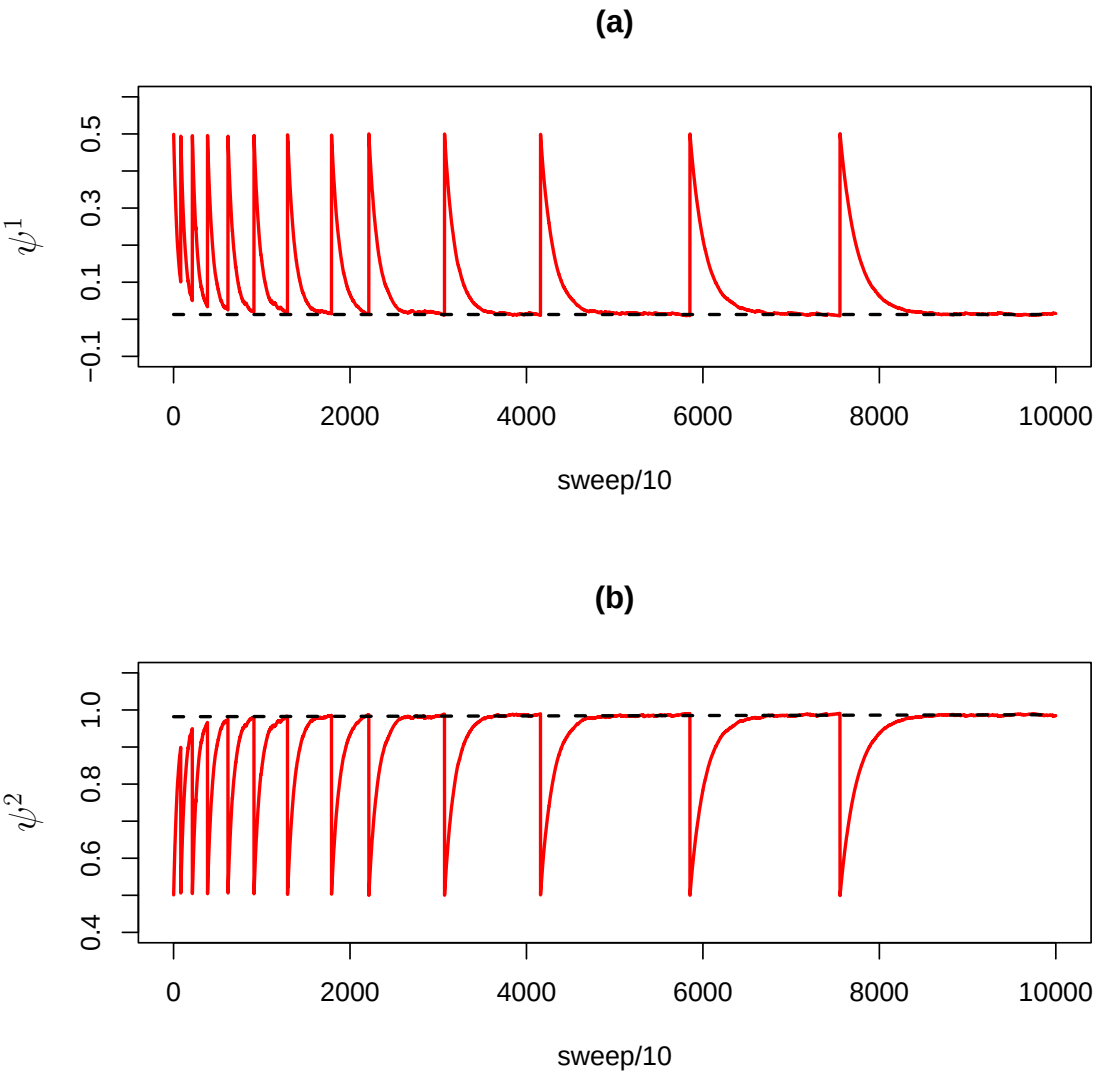

Figure 5.13: Evolution of reversible jump proposal probabilities: (a) ψ^1 (probability of proposing a jump to model 1), adapted using diminishing adaptation algorithm B, described in section 4.6; and (b) ψ^2 (probability of proposing a jump of model 2), adapted using algorithm B.

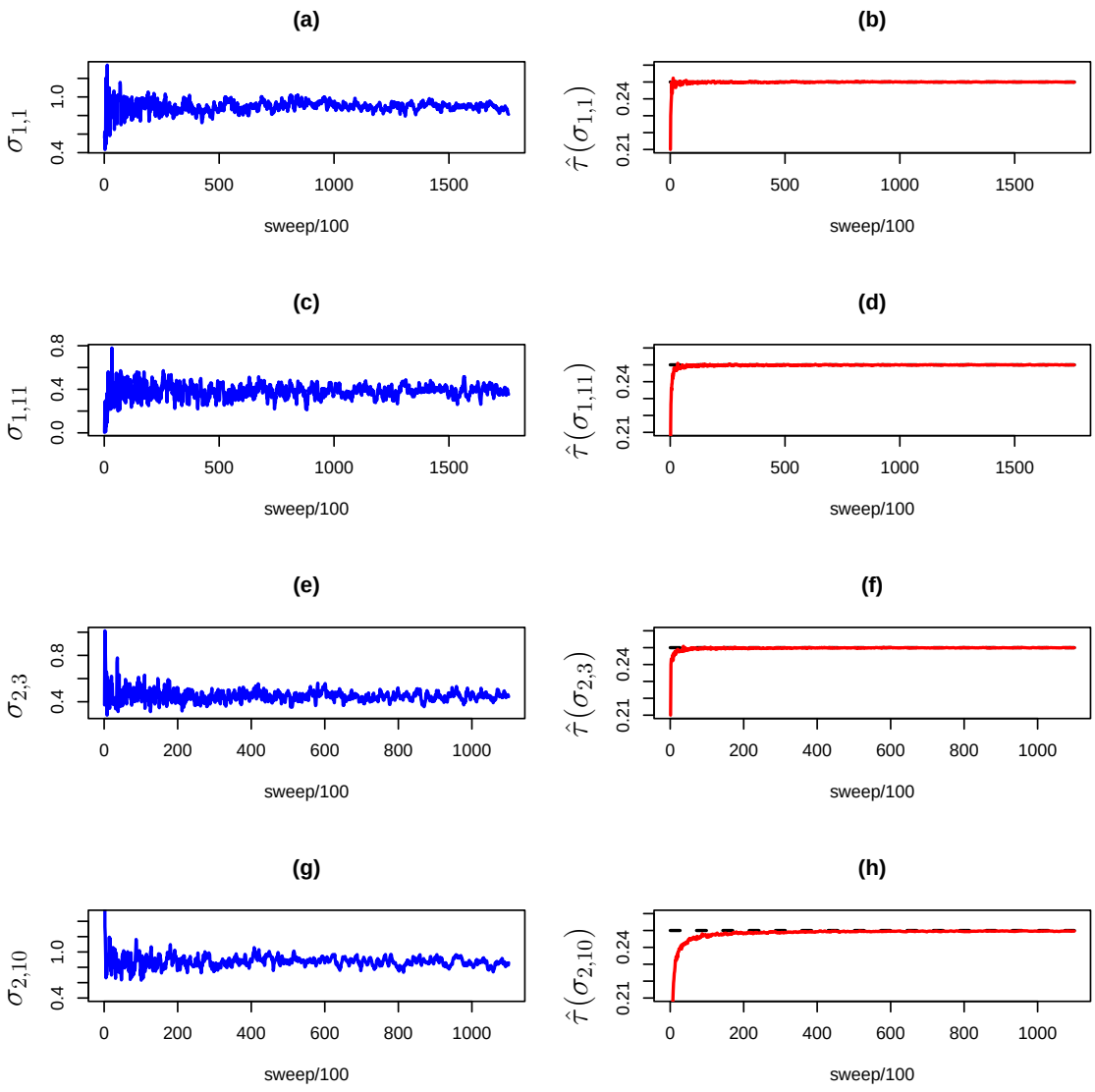

Figure 5.14: Evolution of typical RWM parameters and target functions using AAP algorithm, for the AIDS clinical trial problem: (a) RWM scaling parameter $\sigma_{1,1}$ for component 1 of model 1; (b) average acceptance probability $\hat{\tau}(\sigma_{1,1})$ of RWM for component 1 of model 1; (c) RWM scaling parameter $\sigma_{1,11}$ for component 11 of model 1; (d) average acceptance probability $\hat{\tau}(\sigma_{1,11})$ of RWM for component 11 of model 1; (e) RWM scaling parameter $\sigma_{2,3}$ for component 3 of model 2; (f) average acceptance probability $\hat{\tau}(\sigma_{2,3})$ of RWM for component 3 of model 2; (g) RWM scaling parameter $\sigma_{2,10}$ for component 10 of model 2; and (h) average acceptance probability $\hat{\tau}(\sigma_{2,10})$ of RWM for component 10 of model 2.

AutoMix sampler works on the same model space as Han and Carlin's marginal likelihood method and does not require the extra marginalisation that they need to implement a reversible jump sampler. Furthermore, the AutoMix estimate of the Bayes factor does not require good prior estimates of the model probabilities as Han and Carlin need for their problem-specific reversible jump sampler. Although for the purposes of comparison with Han and Carlin we are most interested in model choice, we confirm that the posterior distributions for the model parameters are also consistent with previous findings reported by Carlin and Louis (2000).

5.6 Conclusions and Improvements

The AutoMix sampler that we have introduced in this chapter is an automatic tool that satisfies our design criteria of broad applicability and efficiency. By adopting innovative adaptive techniques, the process of sampler design is removed and tuning is achieved automatically within the run of the sampler, eliminating the need for specialist experience of this difficult process. The examples presented in section 5.5 demonstrate the potential of the AutoMix sampler, showing good performance and inference consistent with previous problem-specific samplers. For problems such as the change point problem and the Rb9 problem the AutoMix sampler appears to perform equally well as the problem-specific samplers with which it was compared. More impressively, for the AIDS problem considered in section 5.5.4, the AutoMix sampler provides a way of using reversible jump to make successful inference in a way that has not been previously achieved by a problem-specific sampler.

Future research needs to be carried out to find ways of making the sampler run

faster so that it can be feasibly applied to problems with a large number of models or to models with high dimensions. A reduction in the run time may be achieved through the use of different statistical methods, for example within the mixture fitting stage. One such possibility that may offer substantial run time benefits is the combination of stages 1 and 2 of the sampler by fitting mixtures online. Such a method might involve updating the mean and covariance of existing mixture components in a similar way to the algorithm proposed by Haario *et al.* (2001) (reviewed in chapter 4), with an additional step to allocate the chain to a particular component. How to estimate the number of components online might be less obvious.

More simply, the run time could probably be shortened through computational methods, for example by optimising the existing code. Other avenues might also be considered, for example the use of parallel computing facilities where available. Certainly, the expensive within-model stages 1 and 2 of the sampler could all be run concurrently.

An important question that we have not attempted to address in this chapter is how to adequately compare two reversible jump samplers. The assessment of performance of a reversible jump sampler is in itself a very difficult task. However, even if we decide upon a measure or a set of criteria for assessing the performance of a sampler or comparing two samplers, we must be careful how we interpret such measures. In particular, we should not expect an automatic sampler to perform better than a sampler that has been specifically designed and tuned for a certain problem. Indeed, making such a comparison misses the unquantifiable (scientific) benefit of an automatic sampler, that a reversible jump expert is not required to make inference about each individual problem.

Similarly, any conventional measure of efficiency (for example taking account of measures such as the autocorrelation in the resulting sample and time taken to run the sampler) might often favour the specifically designed algorithm. This disregards the fact that an automatic sampler involves considerably less expense and time during the design phase. Unfortunately, such features are not easily measurable. A subjective indication of ease of use may only be measured from user feedback in time. While applicability can be gauged to a certain extent by looking at the performance of the sampler for various different types of problems, this is not a satisfactory quantitative measure.

Although in general we might not expect an automatic sampler to perform better than a specifically developed sampler, we recognise that an automatic sampler must still meet realistic requirements. A generic sampler that will work well in theory, yet in practice takes an infeasibly long time to converge or explores only a small proportion of the sample space is clearly not useful. Indeed, designing such a sampler would be more harmful than good, because an automatic sampler should allow the non-expert to have confidence in the output. A bad automatic sampler will mean that flawed inference may be trusted by the non-expert. We must therefore have confidence that any automatic sampler performs sufficiently well for the cases we test. We should also expect it to perform equally as well on similar untested cases. The AutoMix sampler appears to meet the first of these needs at least.

Any sampler that we develop will not be universally applicable for all statistical models. This may be because certain assumptions are made that preclude some statistical models. For example, for the AutoMix sampler, we make

the assumption that for each model k the parameter space $\boldsymbol{\Theta}_k$ is a subset of $\mathbb{R}^{n_k}$. Adding a discrete parameter to certain models might not be difficult if the parameter was independent of the continuous parameters but would be more difficult otherwise. However, even amongst models for which an automatic sampler is applicable, there will be some problems for which automatic samplers will perform badly. Due to the diversity of models that the sampler could potentially encounter this is not surprising. Indeed, for some models, reversible jump may not even be the best way to proceed. Although a totally generic sampler is an unrealistic goal, it is important that when an automatic sampler does not perform well the user is provided with sufficient information to determine this failure. For the AutoMix sampler some indication is given, but it is not really targeted at allowing the non-specialist to ascertain bad performance. This is an area where further research might be usefully channelled.

One area that might prove problematic in the AutoMix sampler is the reliance on random walk Metropolis in the first stage within-model runs. Another area of the AutoMix sampler that could be developed is the use of adaptive tools. We discuss these issues further in chapter 6. Here, we note only that as research progresses into MCMC methods (both adaptive and otherwise) so new improved methods will evolve. We must keep assessing whether such methods are suitable for the AutoMix sampler.

In conclusion, we feel that the AutoMix sampler makes considerable progress towards automatic reversible jump MCMC. The sampler has a wide applicability and in many cases can considerably simplify the process of making inference through reversible jump. It is hoped that this sampler will provide the building blocks and motivation for further research to move closer to realising the

potential of RJMCMC.

Chapter 6

Areas For Future Work

With the introduction of tools such as the AutoMix sampler, the possibility of automatic RJMCMC methods becomes a little closer. The motivation for the design and implementation of automatic samplers has been addressed in some detail in earlier chapters of this thesis and we hope that work will continue in the area of automatic sampler design. While it is likely that future research will result in innovative samplers that are unrelated to the AutoMix sampler there remain considerable possibilities for research into the extension and improvement of some or all of the components that make up the AutoMix sampler. In this chapter we bring together some of the features of the AutoMix sampler that might benefit from future research.

As introduced in chapter 5, the AutoMix sampler is made up of three distinct stages. We begin by considering an interesting modification to the third stage reversible jump proposal mechanism that we have not explored in this thesis. The idea is a generalisation of that suggested by Godsill (2003) in relation to Green's original AutoRJ sampler.

As described in the previous chapter, given that the Markov chain is currently

in state $(k, \boldsymbol{\theta}_k)$, the AutoMix sampler first allocates $\boldsymbol{\theta}_k$ to a component l_k of the mixture M_k and then uses this mixture component to standardise $\boldsymbol{\theta}_k$. Next, a new model k' and a $M_{k'}$ mixture component $l'_{k'}$ are proposed. The proposed new state vector $\boldsymbol{\theta}'_{k'}$ is then derived by transforming the standardised vector using the mean and covariance matrix of component $l'_{k'}$, with the aid of random numbers $\boldsymbol{u}$ if required. The proposed state $(k', \boldsymbol{\theta}'_{k'})$ is then accepted with the appropriate acceptance probability (see section 5.3.1).

A simpler alternative to this mechanism proceeds as follows. Suppose as before we select a new model k' with probability $\rho_{k,k'} = \psi^{k'}$ and a component $l'_{k'}$ of the mixture $M_{k'}$ with probability $\lambda_{k'}^{l'_{k'}}$. However, we now discard the current state vector $\boldsymbol{\theta}_k$ without allocating it to a component l_k. Instead, the proposed new state vector $\boldsymbol{\theta}'_{k'}$ is set equal to a $n_{k'}$-vector of random variables drawn from component $l'_{k'}$. Mathematically, if $\boldsymbol{u}_{k'}$ was a vector of random variables from some standard distribution $g_{k'}$, then $\boldsymbol{\theta}'_{k'} = \boldsymbol{\mu}_{k'}^{l'_{k'}} + B_{k'}^{l'_{k'}} \boldsymbol{u}_{k'}$. We observe that this move is actually a transition from $(\boldsymbol{\theta}_k, \boldsymbol{u}_{\boldsymbol{k'}})$ to $(\boldsymbol{\theta}'_{k'}, \boldsymbol{u}_{\boldsymbol{k}})$, where $\boldsymbol{u}_k$ are the random numbers that would be required to move in the reverse direction. Therefore, in addition to the transition from $\boldsymbol{u}_{k'}$ to $\boldsymbol{\theta}'_{k'}$, the move involves the selection of a component l_k from the mixture M_k (with probability $\lambda_k^{l_k}$) so that the transition $\boldsymbol{u}_k = [B_k^{l_k}]^{-1}(\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k})$ is also well defined.

Under such a scheme the proposed state $\boldsymbol{x'} = (k', \boldsymbol{\theta}'_{k'})$ would then be accepted with probability

$$\alpha_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x'}) = \min\{1, A_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x'})\}, \tag{6.1}$$

where

$$A_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x'}) = \frac{\pi(k', \boldsymbol{\theta}'_{k'})\psi^k \lambda_k^{l_k} \lambda_{k'}^{l'_{k'}} g_k(\boldsymbol{u}_k)|B_{k'}^{l'_{k'}}|}{\pi(k, \boldsymbol{\theta}_k)\psi^{k'} \lambda_{k'}^{l'_{k'}} \lambda_k^{l_k} g_{k'}(\boldsymbol{u}_{k'})|B_k^{l_k}|} = \frac{\pi(k', \boldsymbol{\theta}'_{k'})\psi^k g_k(\boldsymbol{u}_k)|B_{k'}^{l'_{k'}}|}{\pi(k, \boldsymbol{\theta}_k)\psi^{k'} g_{k'}(\boldsymbol{u}_{k'})|B_k^{l_k}|} \ . \tag{6.2}$$

This new sampler might be seen as a trans-dimensional analogue of the independence sampler (whereas the original AutoMix sampler is more like a trans-dimensional version of the random walk Metropolis sampler). The potential effect of discarding any information that was contained in $\boldsymbol{\theta}_k$ is not immediately apparent. In the limited case of problems involving nested models it seems intuitive that the information contained in $\boldsymbol{\theta}_k$ would prove some useful information about good values of $\boldsymbol{\theta}'_{k'}$. In such cases, the original sampler may be better than the alternative independence version of the sampler (although, as mentioned in chapter 5, only if the random permutation was disabled).

One potential benefit of the independence based sampler is that all moves (in particular *down* moves that decrease the dimension of the state vector) become random. Although the original AutoMix sampler achieves randomised down moves, such moves are only random in the sense that $\boldsymbol{\theta}'_{k'}$ takes a random choice of one out of a finite number of candidate values. For the independence version the number of possible values for the proposed new vector $\boldsymbol{\theta}'_{k'}$ is infinite.

The question of the relative merits of the two samplers warrants further investigation, perhaps through a comparison for a variety of problems. However, we note that the independence version of the AutoMix sampler and the original sampler are really just opposite ends of a spectrum of such samplers that use the mixture based mechanism in the reversible jump phase. It is easy to imagine samplers which are between the original and independence versions, where a (possibly random) number of components of $\boldsymbol{\theta}_k$ are retained to form $\boldsymbol{\theta}'_{k'}$. Assessing this collection of samplers for a variety of problems seems an

interesting area for further work.

Before leaving this topic we note that in analogy with the standard MCMC independence sampler it is relatively straightforward to establish sufficient conditions for the target distribution π so that the AutoMix independence sampler detailed above is uniformly ergodic. The following theorem formalises these conditions.

Theorem 6.1. *Let $\mathcal{M}$ be a finite set of models. For each model $k \in \mathcal{M}$ let $\boldsymbol{\Theta}_k \subseteq \mathbb{R}^{n_k}$ be the associated parameter space. Define $\mathcal{X} = \cup_{k\in\mathcal{M}}(\{k\} \times \boldsymbol{\Theta}_k)$. Let π be a distribution over $\mathcal{X}$ with density $\pi(\cdot,\cdot)$, such that for each $k \in \mathcal{M}$, $\pi(k,\cdot)$ is absolutely continuous with respect to the n_k-dimensional Lebesgue measure.*

For each model k suppose there exists an associated mixture distribution M_k, consisting of $L_k < \infty$ components, where each component l is characterised by a weight $\lambda_k^l > 0$, a n_k-vector $\boldsymbol{\mu}_k^l$ and an $n_k \times n_k$ invertible matrix B_k^l.

Assume further that there exists a constant ε, such that for each $k \in \mathcal{M}$ there exists a component l_k of the mixture M_k satisfying

$$\varepsilon\pi(k,\boldsymbol{\theta}_k) \leq g_k([B_k^{l_k}]^{-1}\{\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k}\}), \quad \forall \boldsymbol{\theta}_k \in \boldsymbol{\Theta}_k, \tag{6.3}$$

where $g_k(\cdot)$ is the joint probability density (with respect to the n_k-dimensional Lebesgue measure) of the random number distribution g_k, detailed in the AutoMix independence algorithm.

Then the Markov chain $(\boldsymbol{X}_n)$, resulting from the AutoMix independence sampler with $\psi^k > 0$ for all $k \in \mathcal{M}$, is uniformly ergodic.

Proof. Since the chain $(\boldsymbol{X}_n)$ is aperiodic, we demonstrate the uniform ergodicity by showing that the whole state space is a small set for the transition kernel corresponding to the above algorithm.

Consider an arbitrary set $A \in \mathcal{B}(\mathcal{X})$. For such an A, we can write $A = \cup_{k'\in\mathcal{M}'}(\{k'\} \times A_{k'})$, where $\mathcal{M}' \subset \mathcal{M}$ and $A_{k'} \subset \boldsymbol{\Theta}_{k'}$ for all $k' \in \mathcal{M}'$. Define the collection of transition functions $\boldsymbol{t}_k^l(\boldsymbol{u}_k) = \boldsymbol{\mu}_k^l + B_k^l\boldsymbol{u}_k$, for $k \in \mathcal{M}$, $l = 1,\ldots,L_k$. Finally, define the collection of sets $\{U_k^l : k \in \mathcal{M}', l = 1,\ldots,L_k\}$ by the relation $A_k = \boldsymbol{t}_k^l(U_k^l)$.

For any $\boldsymbol{x} \in \mathcal{X}$, we can write $\boldsymbol{x} = (k, \boldsymbol{\theta}_k)$ for some $k \in \mathcal{M}$ and $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}_k$. Then,

$$\mathcal{K}(\boldsymbol{x}, A) \geq \sum_{k' \in \mathcal{M}'} \psi^{k'} \sum_{l'=1}^{L_{k'}} \sum_{l=1}^{L_k} \int_{U_{k'}^{l'}} \lambda_{k'}^{l'} \lambda_k^l g_{k'}(\boldsymbol{u}_{k'}) \alpha_{l,l'}(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}\boldsymbol{u}_{k'}, \tag{6.4}$$

where $\boldsymbol{x}' = (k', \boldsymbol{\theta}'_{k'})$ such that $\boldsymbol{\theta}'_{k'} = \boldsymbol{t}_{k'}^{l'}(\boldsymbol{u}_{k'})$ for a given $l' \in \{1, 2, \ldots, L_{k'}\}$.

Choosing l_k and $l'_{k'}$ as two components of the mixtures M_k and $M_{k'}$ respectively for which assumption 6.3 holds, equation 6.4 implies

$$\mathcal{K}(\boldsymbol{x}, A) \geq \sum_{k' \in \mathcal{M}'} \psi^{k'} \int_{U_{k'}^{l'_{k'}}} \lambda_{k'}^{l'_{k'}} \lambda_k^{l_k} g_{k'}(\boldsymbol{u}_{k'}) \alpha_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}\boldsymbol{u}_{k'}. \tag{6.5}$$

Substituting the expression for $\alpha_{l_k, l'_{k'}}(\boldsymbol{x}, \boldsymbol{x}')$ from equations 6.1 and 6.2 into equation 6.5 gives

$$\mathcal{K}(\boldsymbol{x}, A) \geq \sum_{k' \in \mathcal{M}'} \int_{U_{k'}^{l'_{k'}}} \min \left\{ \frac{\psi^{k'} \lambda_{k'}^{l'_{k'}} \lambda_k^{l_k} g_{k'}(\boldsymbol{u}_{k'})}{\pi(k', \boldsymbol{\theta}'_{k'}) |B_{k'}^{l'_{k'}}|}, \frac{\psi^k \lambda_k^{l_k} \lambda_{k'}^{l'_{k'}} g_k(\boldsymbol{u}_k)}{\pi(k, \boldsymbol{\theta}_k) |B_k^{l_k}|} \right\} \pi(k', \boldsymbol{\theta}'_{k'}) |B_{k'}^{l'_{k'}}| \mathrm{d}\boldsymbol{u}_{k'}.$$

Noting that $\boldsymbol{u}_k = [B_k^{l_k}]^{-1}\{\boldsymbol{\theta}_k - \boldsymbol{\mu}_k^{l_k}\}$, $\boldsymbol{u}_{k'} = [B_{k'}^{l'_{k'}}]^{-1}\{\boldsymbol{\theta}'_{k'} - \boldsymbol{\mu}_{k'}^{l'_{k'}}\}$ and $\mathrm{d}\boldsymbol{\theta}'_{k'} = |B_{k'}^{l'_{k'}}| \mathrm{d}\boldsymbol{u}_{k'}$, assumption 6.3 yields

$$\mathcal{K}(\boldsymbol{x}, A) \geq \sum_{k' \in \mathcal{M}'} \int_{A_{k'}} \varepsilon \min \left\{ \frac{\psi^{k'} \lambda_{k'}^{l'_k} \lambda_k^{l_k}}{|B_{k'}^{l'_{k'}}|}, \frac{\psi^k \lambda_k^{l_k} \lambda_{k'}^{l'_{k'}}}{|B_k^{l_k}|} \right\} \pi(k', \boldsymbol{\theta}'_{k'}) \mathrm{d}\boldsymbol{\theta}'_{k'}.$$

Defining

$$\tilde{\varepsilon} = \varepsilon \min_{k \in \mathcal{M}} \left(\min_{k' \in \mathcal{M}'} \left[\min \left\{ \frac{\psi^{k'} \lambda_{k'}^{l'_k} \lambda_k^{l_k}}{|B_{k'}^{l'_{k'}}|}, \frac{\psi^k \lambda_k^{l_k} \lambda_{k'}^{l'_{k'}}}{|B_k^{l_k}|} \right\} \right] \right),$$

we have

$$\begin{aligned} \mathcal{K}(\boldsymbol{x}, A) &\geq \tilde{\varepsilon} \sum_{k' \in \mathcal{M}'} \int_{A_{k'}} \pi(k', \boldsymbol{\theta}'_{k'}) \mathrm{d}\boldsymbol{\theta}'_{k'} \\ &= \tilde{\varepsilon} \pi(A). \end{aligned} \tag{6.6}$$

Since the set A is arbitrary, $\mathcal{X}$ is a small set. □

While this result does not necessarily mean that the independence version of the AutoMix sampler will be better than the original in practical terms, the result may be useful to establish theoretical properties if adaptive versions of the new

sampler were used.

The reversible jump stage of the AutoMix sampler is only one aspect that would benefit from further research. In particular, there may be more efficient ways to complete the computationally expensive first two stages involved in fitting the mixtures for each of the models under consideration. The componentwise EM algorithm that is currently used for the mixture fitting second stage (see chapter 5) appears quite robust but is a computationally expensive phase of the algorithm. Other methods for fitting mixtures may be equally stable but may take less time to run. If alternative methods for mixture fitting are explored, it should be remembered that the mixture is only serving as a proposal distribution. Thus we may be willing to consider methods that result in mixtures that fit the conditionals slightly less well but which take considerably less time to run. Our chosen method is only one out of several possibilities and finding a suitable solution to this trade off remains an interesting challenge.

Another issue that requires further consideration is the robustness of using random walk Metropolis (RWM) to sample from the conditionals $\pi(\cdot|k)$ in the first stage of the AutoMix sampler. An interesting discussion of the practical implications of theoretical results for the RWM algorithm is presented by Roberts (2003). Within this discussion, Roberts highlights simple examples where RWM can fail, leading to samplers that behave badly in the tails of the target distribution. The option to use heavier tailed t-distributed proposals for the RWM, is one way that the AutoMix sampler seeks to avoid the problems discussed. However, even this might not be enough (for example Roberts notes that simple transformations to the RWM can often make considerable further improvements to the properties of the sampler) and perhaps there are

alternatives to RWM that ought to be considered. One example where it is easy to imagine that RWM would fail is where the conditionals $\pi(\cdot|k)$ are multimodal with well separated modes. However, many alternatives to RWM would also have difficulties with properly sampling from such a distribution. Interestingly the third stage of the AutoMix sampler mechanism provides a very natural way of jumping between modes, but only after relying on some other method in the first stage to provide a sample that represents those modes.

As with the mixture fitting in the second stage of the algorithm, there is a trade off between the sophistication of such alternatives and the time taken to apply such methods. However, the importance of producing a representative sample of the conditionals $\pi(\cdot|k)$ during the first stage should not be underestimated. If these conditionals are not adequately sampled, then the mixture that we fit will not approximate the conditionals as desired. Returning to the case where the conditionals were multimodal with well separated modes, if the RWM missed some of these modes, then the mixtures fitted in the second stage would also not account for these modes and therefore it would be likely that these modes would not be well represented in the final reversible jump sample. In effect, the AutoMix sampler is currently only as robust as the RWM in the first stage.

The use of adaptive methods in the AutoMix sampler to scale the RWM offers a significant improvement over standard RWM. However, the use of Atchadé and Rosenthal's adaptive acceptance probability (AAP) algorithm is not without problems and further efforts are required to advance the method. As detailed in chapter 4, theoretical work is required to establish the validity of the assumption underlying the convergence of the AAP algorithm, that the expected acceptance

probability,

$$\tau(\sigma) = \int_{\mathbb{R}\times\mathbb{R}} \alpha(x, x')q_\sigma(x, x')\pi(x)\mathrm{d}x\mathrm{d}x',$$

is a decreasing function of σ, where σ^2 is the variance of the Normal proposal distribution q_σ centred at x. Our efforts demonstrate that this is not a trivial task, although numerical results appear to provide evidence that the result holds at least for most reasonable target distributions π.

On a practical level, attention needs to be given to the sensitivity of the AAP algorithm to the initial choice of the adaptive parameter σ_0. This phenomenon was noticed for the change point problem studied in chapter 5, where the value of σ_0 used by the AutoMix sampler was considerably too small for the components of the state vector corresponding to change point times. For these components, the adaptation in the RWM phase of the AutoMix sampler did not converge in a reasonable time. This appeared to result in mixtures being fitted in stage 2 that did not approximate the conditionals particularly well.

For the change point problem, the problem was alleviated by rescaling the data so that all components of the state vector were on a similar scale. While this method significantly improved the adaptative performance, encouraging users to rescale the data for problems which appear to demonstrate similar symptoms lessens the automatic nature of the AutoMix sampler. Although it would be possible to extend the AutoMix sampler so that rescaling could be done automatically, this would add one further level of complexity to an already complicated sampler. It seems more sensible to focus extra research on alleviating the sensitivity of the AAP algorithm, perhaps considering the possibility of including some mechanism to adapt the scaling parameter σ more rapidly in an initial phase of the adaptation.

In addition to such research, we advocate the continuation of work on new adaptive algorithms which may provide alternative tools for use in the AutoMix sampler. In particular, adaptive methods designed specifically for RJMCMC are required. Although such methods were introduced in chapter 4 we did not directly verify the required drift conditions of Andrieu *et al.* (2004) or Andrieu and Moulines (2004) that guarantee the ergodicity of the resulting chain. This area clearly requires further attention. Finally on this subject, we note that work into the verification of such conditions might prove easier in the case of the alternative independence version of the AutoMix sampler given its uniform ergodicity in the non-adaptive case (see earlier in this chapter).

Another interesting possibility for the AutoMix sampler is to consider allowing the option to replace stages 2 and 3 with a method for estimating the marginal likelihoods (and therefore the posterior model probabilities) based on the conditional samples from stage 1. Various methods for computing such estimates are considered by several authors including, Meng and Wong (1996), Chib and Jeliazkov (2001), Han and Carlin (2001), Meng and Schilling (2002) and Mira and Nicholls (2004). For many problems such methods are not straightforward and indeed might be considerably harder to implement than reversible jump. However, for some problems such an option might offer a viable alternative. Alternatively, the AutoMix sampler might in future take advantage of the very recent work by Bartolucci *et al.* (2004), wherein the authors suggest a method for using reversible jump output to compute improved estimates of the Bayes factors. If nothing more, estimating posterior model probabilities by different methods will provide confidence in the precision of the estimates.

Realistically, it is hard to imagine an automatic sampler that performs universally well for all problems. Useful diagnostics that allow users to recognise the reliability of their inference are an essential part of any automatic sampler. This is a feature of the AutoMix sampler which could benefit from more attention, although this has much to do with the more general lack of such diagnostics for the assessment or comparison of reversible jump chains. To have confidence in new automatic methods it is important that progress is made in this area.

We have learned from our experience of automatic sampler design that we should not expect automatic samplers to do too much. The tools that we develop should probably be targeted at people with a sound statistical background but not necessarily an expertise in MCMC. For the majority of problems some statistical knowledge can probably be assumed and will have already been important in the formulation and understanding of a stochastic model. By making this assumption we narrow the target users of the automatic tools that we design and focus our energies with these particular users in mind.

Perhaps our most important recommendation for automatic sampler design is to make sure that the goals remain realistic. By keeping things simple, with clear, achievable targets, the area of automatic RJMCMC will progress in the confidence that such tools will offer scientists myriad opportunities.

Bibliography

Abrams, D. I., Goldman, A. I., Launer, C., Korvick, J. A., Neaton, J. D., Crane, L. R., Grodesky, M., Wakefield, S., Muth, K., Kornegay, S., Cohn, D. L., Harris, A., Luskin-Hawk, R., Markowitz, N., Sampson, J. H., Thompson, M., Deyton, L., and Terry Beirn Community Programs for Clinical Research on AIDS (1994). Comparative trial of Didanosine and Zalcitabine in patients with Human Immunodeficiency Virus who are intolerant of or have failed Zidovudine therapy. *New England Journal of Medicine*, **330**, 657–662.

Al-Awadhi, F., Hurn, M., and Jennison, C. (2004). Improving the acceptance rate of reversible jump MCMC proposals. *Statistics and Probability Letters*, **69**, 189–198.

Andrieu, C. and Moulines, A. (2004). On the ergodicity properties of some adaptive MCMC algorithms. Technical report (submitted), University of Bristol.

Andrieu, C., Moulines, E., and Priouret, P. (2004). Stability of stochastic approximation under verifiable conditions. *Accepted for publication in SIAM Journal on Control and Optimization*.

Andrieu, C. and Robert, C. P. (2001). Controlled MCMC for optimal sampling. Technical report, University of Bristol and CEREMADE, Université Paris-Dauphine.

Atchadé, Y. F. and Rosenthal, J. S. (2003). On adaptive Markov chain Monte Carlo algorithms. Technical report, University of Montreal and University of Toronto.

Bartolucci, F., Scaccia, L., and Mira, A. (2004). Efficient Bayes factor estimation from the reversible jump output. Technical report, Università di Urbino, Università di Perugia and Università dell'Insubria.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society, B*, **55**, 25–37.

Brockwell, A. E. and Kadane, J. B. (2004). Identification of regeneration times in MCMC simulation, with application to adaptive schemes. *Accepted for publication in Journal of Computational and Graphical Statistics*.

Brooks, S. P. (1998). Markov chain Monte Carlo method and its application. *The Statistician*, **47**, 69–100.

Brooks, S. P., Fan, Y., and Rosenthal, J. S. (2002). Perfect forward simulation via simulated tempering. Technical report, University of Cambridge.

Brooks, S. P. and Giudici, P. (2000). MCMC convergence assessment via two-way ANOVA. *Journal of Computational and Graphical Statistics*, **9**, 266–285.

Brooks, S. P., Giudici, P., and Philippe, A. (2003a). Nonparametric convergence assessment for MCMC model selection. *Journal of Computational and Graphical Statistics*, **12**, 1–22.

Brooks, S. P., Giudici, P., and Roberts, G. O. (2003b). Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions (with discussion). *Journal of the Royal Statistical Society, B*, **65**, 3–56.

Cappé, O., Robert, C. P., and Rydén, T. (2003). Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *Journal of the Royal Statistical Society, B*, **65**, 679–700.

Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, B*, **57**, 473–484.

Carlin, B. P. and Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis (2nd ed.)*. Chapman & Hall/CRC , Boca Raton, Florida.

Chib, S. and Carlin, B. P. (1999). On MCMC sampling in hierarchical longitudinal models. *Statistics and Computing*, **9**, 17–26.

Chib, S. and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, **96**, 270–281.

Christensen, O. F., Roberts, G. O., and Rosenthal, J. S. (2004). Scaling limits for the transient phase of local Metropolis-Hastings algorithms. *Accepted for publication in Journal of the Royal Statistical Society, B.*

Consonni, G. and Veronese, P. (1995). A Bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, **90**, 935–944.

Dellaportas, P., Forster, J. J., and Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing*, **12**, 27–36.

Dellaportas, P. and Papageorgiou, I. (2004). Multivariate mixtures of normals with unknown number of components. Technical report, Athens University of Economics and Business.

Drinkwater, N. R. and Klotz, J. H. (1981). Statistical methods for the analysis

of tumor multiplicity data. *Cancer Research*, **41**, 113–119.

Ehlers, R. S. and Brooks, S. P. (2002). Efficient construction of reversible jump MCMC proposals for autoregressive time series models. Technical report, University of Cambridge.

Erland, S. (2003). *On adaptivity and Eigen-decompositions of Markov chains*. Ph. D. thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology.

Fan, Y. and Brooks, S. P. (2000). Bayesian modelling of prehistoric corbelled domes. *Journal of the Royal Statistical Society, D*, **49**, 339–354.

Figueiredo, M. A. T. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 381–396.

Gamerman, D. (1997). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman & Hall/CRC, New York.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis (2nd ed.)*. Chapman & Hall/CRC, New York.

Gilks, W. R., Roberts, G. O., and Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association*, **93**, 1045–1054.

Godsill, S. J. (2001). On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, **10**, 230–248.

Godsill, S. J. (2003). Proposal densities and product-space methods. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, Oxford Statistical Science Series, No. 27, pp. 199–203. Oxford

University Press, Oxford.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

Green, P. J. (1999). A primer on Markov chain Monte Carlo. In O. E. Barndorff-Nielsen, D. R. Cox, and C. Klüppelberg (Eds.), *Complex Stochastic Systems*, Monographs on Statistics and Applied Probability, No. 87, pp. 1–62. Chapman & Hall/CRC, Boca Raton, Florida.

Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, Oxford Statistical Science Series, No. 27, pp. 179–198. Oxford University Press, Oxford.

Green, P. J. and Han, X.-L. (1992). Metropolis methods, Gaussian proposals and antithetic variables. In A. Frigessi, P. Barone, and M. Piccioni (Eds.), *Stochastic Models, Statistical Methods and Algorithms in Image Analysis*, Lecture Notes in Statistics, No. 74, pp. 142–164. Springer-Verlag, Berlin.

Green, P. J. and Mira, A. (2001). Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika*, **88**, 1035–1053.

Green, P. J. and O'Hagan, A. (1998). Model choice with MCMC on product spaces without using pseudo-priors. Technical report, University of Nottingham.

Green, P. J. and Richardson, S. (2002). Hidden Markov models and disease mapping. *Journal of the American Statistical Association*, **97**, 1055–1070.

Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, **14**, 375–395.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, **7**, 223–242.

Haigis, K. M. and Dove, W. F. (2003). A Robertsonian translocation suppresses a somatic recombination pathway to loss of heterozygosity. *Nature Genetics*, **33**, 33–39.

Han, C. and Carlin, B. P. (2001). Markov chain Monte Carlo methods for computing Bayes factors: a comparative review. *Journal of the American Statistical Association*, **96**, 1122–1132.

Hardy, R. G., Meltzer, S. J., and Jankowski, J. A. (2000). ABC of colorectal cancer: molecular basis for risk factors. *British Medical Journal*, **321**, 886–889.

Hastie, D. (2003). Discussion of paper by Brooks *et al. Journal of the Royal Statistical Society, B*, **65**, 49–50.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Heikkinen, J. (2003). Trans-dimensional Bayesian non-parametrics with spatial point processes. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, Oxford Statistical Science Series, No. 27, pp. 203–206. Oxford University Press, Oxford.

Holden, L. (1998). Adaptive chains. Technical report SAND/11/98, Norwegian Computing Center.

Kass, R. E. and Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*, **90**, 928–934.

King, R. and Brooks, S. P. (2002). Bayesian model discrimination for multiple

strata capture-recapture data. *Biometrika*, **89**, 785–806.

Kokoska, S. M. (1987). The analysis of cancer chemoprevention experiments. *Biometrics*, **43**, 525–534.

Li, Y. Q., Roberts, S. A., Paulus, U., Loeffler, M., and Potten, C. S. (1994). The crypt cycle in mouse small intestinal epithelium. *Journal of Cell Science*, **107**, 3271–3279.

Liu, J. S., Liang, F., and Wong, W. H. (2001). A theory for dynamic weighting in Monte Carlo computation. *Journal of the American Statistical Association*, **96**, 561–573.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New York.

McLeish, D. L. (1975). A maximal inequality and dependent strong laws. *Annals of Probability*, **3**, 829–839.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, **11**, 552–586.

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, **6**, 831–860.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1091.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*. Springer-Verlag, New York.

Mira, A. (1998). *Ordering, Slicing and Splitting Monte Carlo Markov Chains*. Ph. D. thesis, School of Statistics, University of Minnesota.

Mira, A. and Nicholls, G. K. (2004). Bridge estimation of the probability density at a point. *Statistica Sinica*, **14**, 603–612.

Møller, J. and Nicholls, G. K. (2004). Perfect simulation for sample based inference. *Accepted for publication in Statistics and Computing*.

Moolgavkar, S. H. and Knudson, A. G. (1981). Mutation and cancer: a model for human carcinogenesis. *Journal of the National Cancer Institute*, **66**, 1037–1052.

Moser, A. R., Dove, W. F., Roth, K. A., and Gordon, J. I. (1992). The Min (multiple intestinal neoplasia) mutation: its effect on gut epithelial cell differentiation and interaction with a modifier system. *Journal of Cell Biology*, **116**, 1517–1526.

Neal, R. M. (1992). Bayesian mixture modeling. In C. R. Smith, G. J. Erickson, and P. O. Neudorfer (Eds.), *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis, Seattle 1991*, pp. 197–211. Kluwer Academic Publishers, Dordrecht.

Neal, R. M. (2003). Slice sampling (with discussion). *Annals of Statistics*, **31**, 705–767.

Neal, R. M. (2004). Improving asymptotic variance of MCMC estimators: non-reversible chains are better. Technical report, University of Toronto.

Newton, M. A. and Hastie, D. I. (2004). Assessing Poisson variation of intestinal tumour multiplicity in Min mice carrying a Robertsonian translocation. Technical report, University of Wisconsin.

O'Hagan, A. (1994). *Kendall's Advanced Theory of Statistics, Volume 2B: Bayesian Inference*. Arnold, London.

Pasarica, C. and Gelman, A. (2004). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. Technical report, Columbia University.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, 607–612.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, **9**, 223–252.

Ramachandran, S., Fryer, A. A., Lovatt, T., Smith, A., Lear, J., Jones, P. W., and Strange, R. C. (2002). The rate of increase in the numbers of primary sporadic basal cell carcinomas during follow up is associated with age at first presentation. *Carcinogenesis*, **23**, 2051–2054.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, B*, **59**, 731–792.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of mathematical statistics*, **22**, 400–407.

Robert, C. P. and Casella, G. (2002). *Monte Carlo Statistical Methods*. Springer, New York.

Roberts, G. O. (2003). Linking theory and practice of MCMC. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, Oxford Statistical Science Series, No. 27, pp. 145–166. Oxford University Press, Oxford.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied*

Probability, **7**, 110–120.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society, B*, **60**, 255–268.

Roberts, G. O. and Rosenthal, J. S. (1999). Convergence of slice sampler Markov chains. *Journal of the Royal Statistical Society, B*, **61**, 643–660.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, **16**, 351–367.

Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation Structure, blocking and parametrization for the Gibbs sampler. *Journal of the Royal Statistical Society, B*, **59**, 291–317.

Sahu, S. K. and Zhigljavsky, A. A. (2003). Self-regenerative Markov chain Monte Carlo with adaptation. *Bernoulli*, **9**, 395–422.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**, 461–464.

Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, B*, **55**, 3–23.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods. *Annals of Statistics*, **28**, 40–74.

Tierney, L. (1998). A note on Metropolis-Hastings kernels for general state spaces. *Annals of Applied Probability*, **8**, 1–9.

Tierney, L. and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, **18**, 2507–2515.

Tjelmeland, H. and Hegstad, B. K. (2001). Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics*, **28**, 205–223.

Waagepetersen, R. and Sorensen, D. (2001). A tutorial on reversible jump MCMC with a view toward QTL-mapping. *International Statistical Review*, **69**, 49–61.