Reversible jump Riemann Manifold Hamiltonian Monte Carlo

J. Wyse†, N. Friel* and M. Girolami‡

† School of Computer Science and Statistics, Trinity College Dublin, Ireland.
* School of Mathematical Sciences, University College Dublin, Ireland.
‡ Department of Statistics, University College London, UK.

February, 2012

Abstract

This paper explores how the recently introduced Riemann manifold Hamiltonian Monte Carlo (RMHMC) methodology can be extended to allow one to explore Bayesian model uncertainty. In essence, RMHMC provides an automatic scaling of the proposal distribution by adapting to the geometry of the posterior distribution. A natural next step in the development of this methodology is to understand how this adaption can be exploited in situations where there is model uncertainty. The algorithm which we describe illustrates how RMHMC can be embedded in a reversible jump MCMC framework. We hope that this newly introduced methodology will serve as a starting point for research in this direction. We apply our algorithm to logistic regression model choice and to the analysis of finite mixture distributions with an unknown number of components.

1 Introduction

Since the introduction of reversible jump Markov chain Monte Carlo (RJMCMC), in the seminal paper by Green (1995), it has been possible for practitioners to evaluate the relative evidence for a collection of plausible models via a principled stochastic model search. This work continues to have an impact in the area of Bayesian model selection. More recently, Girolami & Calderhead (2011) presented a novel framework, Riemann Manifold Hamiltonian Monte Carlo (RMHMC), which, broadly speaking, adapts to the geometry of the posterior distribution resulting in an efficient Monte Carlo mechanism. The main contribution of this paper is to examine how RMHMC can be extended to a trans-dimensional MCMC setting, thereby combining the within model sampling of RMHMC with the between model sampling capability of RJMCMC.

Hamiltonian (or Hybrid) Monte Carlo (HMC) (Duane et al. 1987) is aims to produce more efficiently mixing Markov chains by allowing for larger moves than would be achieved by a typical random walk Metropolis algorithm. This is achieved by introducing an auxiliary
“momentum” and forming a Hamiltonian for the problem under consideration. The Hamiltonian is formed by summing the negative log likelihood (potential energy) with the inner product of the momentum under some metric (usually termed the mass matrix). Hamilton’s equations are then numerically integrated with respect to time producing paths along isocontours of total energy (the Hamiltonian) in position-momentum phase space. Such a procedure can potentially propose large moves in the position or parameter vector as the momentum changes, thus leading to a more efficient exploration of the support of the posterior.

The paper by Girolami & Calderhead (2011) significantly built on Hamiltonian Monte Carlo literature already in existence by proposing an automatic scaling for the mass matrix, whose value is crucial to the efficacy of HMC methods. They proposed a framework whereby the mass matrix is position dependent and can potentially exploit the local curvature information of the log posterior. It was demonstrated by Girolami & Calderhead (2011) that this approach led to a significant gain in efficiency over HMC tuned by pilot runs. They termed their approach Riemann manifold (RM)HMC, the RM indicating the use of a position-specific metric of a Riemann manifold to propose moves.

Here we propose a novel algorithm which uses the reversibility and volume preserving properties of the leapfrog integrator employed by Girolami & Calderhead (2011), to extend the ideas of RMHMC to a model selection setting. The proposed approach has a distinct RJMCMC flavour, and can be seen as a combination of the ideas of Green (1995) with those of Girolami & Calderhead (2011). Moreover, the presented algorithm can also be employed in the typical HMC setting where a position specific mass matrix is not used, and gives a slightly more general form of this algorithm also. We compare the new idea with traditional RJ algorithms on two benchmark examples.

The remainder of this paper is organized as follows. In Section 2 we review RJMCMC and RMHMC algorithms. Section 3 formalizes the combination of these two approaches and demonstrates the validity of the resulting algorithm. In Section 4 we demonstrate the methods on two examples, variable selection in logistic regression and Gaussian finite mixtures with an unknown number of components. Section 5 explores sensitivity to step-size and number of steps of the leapfrog integrator through a small toy changepoint example. Section 6 concludes with a discussion.

2 Reversible jump and RMHMC algorithms

2.1 RJMCMC

Suppose that $M_1, \ldots, M_J$ are all plausible models for data $y = (y_1, \ldots, y_n)$. Each of the models $M_k$ has a parameter vector $\theta_k$ and we write the posterior of the data conditioning on model $M_k$ as

$$
\pi(\theta_k | y, M_k) \propto \pi(y | \theta_k, M_k) \pi(\theta_k | M_k)
$$

where the first term on the right corresponds to the likelihood of the data under model $M_k$ and the second term is the prior for the parameter vector $\theta_k$ of model $M_k$.

To incorporate the uncertainty in our choice of model, we can take a prior for each of the
models \( \pi(M_1) \ldots \pi(M_J) \). The posterior probability of model \( M_k \) is then given by

\[
\pi(M_k | y) = \frac{\int \pi(\theta_k | y, M_k) \pi(M_k) \, d\theta_k}{\sum_{j=1}^{J} \int \pi(\theta_j | y, M_j) \pi(M_j) \, d\theta_j}
\]

These posterior probabilities are usually not available in closed form and so must be approximated using some analytic or Monte carlo technique.

Denote the support of model \( M_k \) by \( \Theta_k \subseteq \mathbb{R}^{d_k} \) where \( d_k \) is the dimension of the vector \( \theta_k \). Then a reversible jump algorithm generates samples from the union of the supports of

\[
\pi(M_k, \theta_k | y) \propto \pi(\theta_k | y, M_k) \pi(\theta_k | M_k) \pi(M_k)
\]

\( k = 1, \ldots, J \):

\[
Z = \bigcup_{k=1}^{J} \{ k \} \times \{ \Theta_k | M_k \}.
\]

That is, each sample in a RJMCMC scheme consists of an indicator denoting the current model, and a sample of parameters from that model. This is achieved by constructing a reversible irreducible Markov chain whose transition probabilities correctly account for changes in the dimension of \( \theta \). We outline briefly this construction below, but refer the reader to Green (1995) for more technical details on these moves.

Suppose we are currently in model \( M_k \) with parameter \( \theta_k \), and we wish to propose to move to model \( M_l \) where we assume (wlog) \( d_l > d_k \). To propose the parameter \( \theta_l \) in model \( M_l \) it is necessary to generate additional variables \( u \in \mathbb{R}^s \) and \( v \in \mathbb{R}^t \) and set up an invertible bijection

\[
f : \mathbb{R}^{d_k} \times \mathbb{R}^s \to \mathbb{R}^{d_l} \times \mathbb{R}^t
\]

such that \( (\theta_l, v) = f(\theta_k, u) \) and \( d_k + s = d_l + r \). The last condition here is the “dimension matching” requirement. The acceptance probability of such a move is then given by \( \min(1, R) \) where

\[
R = \frac{\pi(M_l, \theta_l | y) p_{lk} g(v)}{\pi(M_k, \theta_k | y) p_{kl} q(u)} \left| \frac{\partial(\theta_l, v)}{\partial(\theta_k, u)} \right|
\]

(1)

where \( p_{kl} \) is the probability of proposing a move to model \( l \) when currently in model \( k \), \( q(\cdot) \) and \( g(\cdot) \) are the densities of \( u \) and \( v \) respectively. The last term is a Jacobian for \( f \). This is crucial to correct the acceptance probability for volume when moving between spaces of different dimensions. The reverse of this move, from model \( M_l \) to \( M_k \) would have acceptance probability \( \min(1, R^{-1}) \).

It can be seen that the user has great freedom in their choice of \( f \) when implementing the RJ algorithm in practice. Clearly \( f \) should be chosen so that jumps between models occur frequently. This will happen when the proposed \( \theta_l \) is in an area of high posterior mass in \( \Theta_l \). However, it is obvious that choosing such an \( f \) to achieve this is not at all an easy feat, especially in a more general setting. Previous research has focussed on choosing \( f \) to maximize the acceptance probability (1) (Brooks, Giudici & Roberts 2003) or to draw the entire \( \theta_l \) from a proposal tuned from a pilot run (Green 2003). Our proposed approach is to try to harness the fast exploration abilities of RMHMC within both models, in the hope that a short RMHMC run within the proposed model will guide us to an area of high posterior mass in \( \Theta_l \). The next section reviews the RMHMC algorithm. Then in Section 3 we describe our approach to combining the two methodologies towards the aim of model selection.
2.2 RMHMC

First off we give a brief review of the Hamiltonian Monte Carlo (HMC) method and following this we discuss RMHMC. The reader is referred to Neal (2010) for an extensive review of HMC and Girolami & Calderhead (2011) for a detailed explanation of RMHMC. In the interests of clarity and brevity we suppress the conditioning on a model $M_k$ in this section. It is to be understood that we only explore the support of a single model—this is generalized to multiple models in Section (3). Thus we call the parameter vector $\theta$ and say it has dimension $d$. To be precise, we require $\theta \in \mathbb{R}^d$ to allow for unconstrained sampling from the posterior. This is always possible by reparameterizing any continuous parameters that are constrained to a particular interval in $\mathbb{R}$ to the entire real line.

HMC sampling uses an auxiliary variable termed the momentum $p \in \mathbb{R}^d$ to help exploration of $\pi(\theta|y)$. The momentum comes from a $d$-variate Gaussian distribution with zero mean and variance-covariance matrix $T$. Focus then shifts to the joint density of $\theta$ and $p$. Noting that $p$ is independent of $\theta$, $\pi(\theta,p|y) = \pi(\theta|y)\pi(p)$. If we denote the log of the posterior by $L(\theta) = \log \pi(\theta|y)$, the negative joint log density of $\theta$ and $p$ is $H(\theta,p) = -L(\theta) + 1/2 \log\{(2\pi)^d|T|\} + 1/2 p^T T^{-1} p$.

The physical analogy of this is as a Hamiltonian for a particle system, where the negative log likelihood represents the potential energy and the other two terms the kinetic. The second term on the right is included to ensure the momentum can be marginalized out without requiring the introduction of any other corrections to the posterior. Hamilton’s equations for this system are

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p} = T^{-1} p,$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial \theta} = \nabla_\theta L(\theta).$$

The solution flow for Hamilton’s equations $(\theta(t), p(t)) = \Phi_t(\theta(0), p(0))$ preserve the total energy of the system, preserve the volume element i.e. $d\theta(t) dp(t) = d\theta(0) dp(0)$, and are time reversible. See Leimkuhler & Reich (2004) for details.

A sampling scheme can be constructed by proposing $p \sim N(0, T)$ and numerically integrating Hamilton’s equations either forward or backwards in time using a Stormer-Verlet leapfrog integrator with a user specified step size $h$ for a given number of time steps:

$$p(t + h/2) = p(t) + h \nabla_\theta L(\theta(t))/2$$

$$\theta(t + h) = \theta(t) + h T^{-1} p(t + h/2)$$

$$p(t + h) = p(t + h/2) + h \nabla_\theta L(\theta(t + h))/2.$$ 

This results in a proposal $(\theta^*, p^*)$. The numerical integrator will introduce some error in that total energy is not completely preserved. Thus an additional Metropolis acceptance step will ensure convergence to the correct distribution. The proposed point is accepted with probability $\min(1, A)$ where

$$A = \exp\{-H(\theta^*, p^*) + H(\theta, p)\}.$$ 

4
Although the HMC algorithm has the potential to be a very powerful tool for inference, its performance is highly dependent on the choice of mass matrix $T$. Neal (1993), Neal (1996), Neal (2010) and Liu (2001) propose ways in which this can be tuned. However, these methods of tuning $T$ usually require knowledge that is not available a priori e.g. the marginal variances of the target density, and thus tuning of $T$ must be based on pilot runs.

Girolami & Calderhead (2011) overcome some of these issues by using a position specific mass matrix $G(\theta)$ which can take into account the local structure about the point from which a proposal is being made. Then proposals can be adapted automatically, taking large jumps where possible and smaller jumps where required. Practitioners are free in their choice of $G(\theta)$. Future research may make it possible to recommend a particular $G(\theta)$ for particular problems. However, this is an area where to date there has been little work. In this paper, we opt only to use the form of mass matrix employed by Girolami & Calderhead (2011) in their explorations and take

$$G(\theta) = E_{y|\theta} \left\{ -\frac{\partial^2}{\partial \theta \partial \theta^T} L(\theta) \right\}, \quad (2)$$

the expected Fisher information for $\theta$ plus the information from the prior.

Using the position specific mass matrix changes the expression for the Hamiltonian to

$$H(\theta, p) = -L(\theta) + \frac{1}{2} \log\{(2\pi)^d|G(\theta)|\} + \frac{1}{2} p^T G(\theta)^{-1} p.$$ 

Hamilton’s equations are also modified, since the Hamiltonian is no longer separable. These read

$$\frac{d\theta_i}{dt} = \frac{\partial H}{\partial p_i} = \{G(\theta)^{-1} p\}_i,$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial \theta_i} = \frac{\partial L(\theta)}{\partial \theta_i} - \frac{1}{2} \text{tr} \left\{ G(\theta)^{-1} \frac{\partial G(\theta)}{\partial \theta_i} \right\} + \frac{1}{2} p^T G(\theta)^{-1} \frac{\partial G(\theta)}{\partial \theta_i} G(\theta)^{-1} p.$$ 

Note that the $i$ subscripts here refer to vector elements and not model indexes. Girolami & Calderhead (2011) use a generalized leapfrog algorithm to do approximate integration of these non-separable Hamilton equations through time. The case of a non-separable Hamiltonian is more delicate since the leapfrog integrator must still be volume preserving and reversible. These two qualities of the integrator are, in fact, crucial for the work we present here. The leapfrog scheme is:

$$p(t + h/2) = p(t) - \frac{h}{2} \nabla_\theta H(\theta(t), p(t + h/2)),$$ 

$$\theta(t + h) = \theta(t) + \frac{h}{2} \left[ \nabla_p H(\theta(t), p(t + h/2)) + \nabla_p H(\theta(t + h), p(t + h/2)) \right]$$

$$p(t + h) = p(t + h/2) - \frac{h}{2} \nabla_\theta H(\theta(t + h), p(t + h/2)).$$ 

When the Hamiltonian is non-separable the equations (3) and (4) are defined implicitly and we solve these following Girolami & Calderhead (2011) by using a fixed point iterations run to convergence. Should a more versatile or efficient way of solving these implicit equations
come to light in the future this should be employed, since, as noted by Cornebise (ref needed) in his discussion of Girolami & Calderhead (2011), fixed point iterations can be unstable. If we encounter any such behaviour here, the move is rejected outright.

The next Section describes how we combine the ideas of RJMCMC and RMHMC to form an extension of the classic reversible jump algorithm which we term reversible jump (RJ)RMHMC.

3 Reversible jump RMHMC

The main aim of this work is to provide a coherent valid extension of the classic RJMCMC algorithm by utilizing the power of RMHMC methods. It turns out that such an extension is relatively straightforward via a novel acceptance step and exploiting the fact that the leapfrog integrator is reversible and volume preserving. What we present below is what we imagine to be a vanilla RJRMHMC, and there is most likely significant potential for this approach to be extended as the methodology around RMHMC is explored and extended further.

A key to designing between model moves which exploit the auxiliary variables introduced by HMC is to ensure that the moves are time reversible. That is, if it is possible to transition from model \( M_k \) to \( M_l \) in one iterate, it must also be possible to move from \( M_l \) to \( M_k \) in one iterate, and the probability of both these events must be calculable.

When proposing moves without changing the model, the approach is as outlined in Section 2.2. Sometimes it may be more efficient to update some subset of the parameters using a Gibbs step if reparameterization to the real space is taxing (as for the mixture weights in Section ). It is possible to create hybrid algorithms which update particular subsets of the parameter vector using different MCMC kernels. Usually arriving at the optimal “bag of tricks” in terms of mixing or some diagnostic will require trial and error. It is valid, sensible and pragmatic to exploit RMHMC for subsets of the parameter vector where it works best and other MCMC updates where they work best. Not all unknown parameters need be prescribed the same sampling approach.

We now describe in detail how we propose a move to model \( M_l \) from model \( M_k \). Suppose the current system configuration is \((\theta_k, p_k)\). We begin by taking \( N_k \) leapfrog steps with step-size \( h_k \) from the current configuration. This brings us to \((\tilde{\theta}_k, \tilde{p}_k)\). Then generate \( u \sim q(\cdot) \), and set \((\tilde{\theta}, v) = f(\tilde{\theta}_k, u)\). Note that we are now in model \( M_l \). We generate a momentum for model \( M_l, \tilde{p}_l \sim N(0, G(\tilde{\theta}_l)) \), and take \( N_l \) leapfrog steps from \((\tilde{\theta}_l, \tilde{p}_l)\) with step-size \( h_l \). The result of these steps is \((\tilde{\theta}_l, p_l)\), this will be our proposed point in model \( M_l \). The mechanics of this move are shown in Figure 1. The motivation behind this scheme is that the additional leapfrog steps within model \( M_l \) should transport us to a configuration of high posterior mass. The leapfrog steps within model \( M_k \) are necessary to ensure that the move is reversible as we will now explain.

Since the generalized leapfrog integrator is time reversible, it is possible to move back to our starting position by simply negating the step-size and taking the same number of steps. Thus to go from \((\tilde{\theta}_l, p_l)\) to \((\tilde{\theta}_l, \tilde{p}_l)\) we simply take \( N_l \) steps with step-size \(-h_l\). The values of \( \tilde{\theta}_k \) and \( u \) can then be obtained by generating \( v \sim q(\cdot) \) and applying the inverse of \( f \) i.e. \((\tilde{\theta}_k, u) = f^{-1}(\tilde{\theta}_l, v)\). Now generate \( \tilde{p}_k \sim N(0, G(\tilde{\theta}_k)) \) and take a further \( N_k \) leapfrog steps
with step-size $-h_k$ from here. This brings us back to $(\theta_k, p_k)$.

The acceptance probability of the move from $M_k$ to $M_l$ will include terms for the intermediate “∼” stages of the proposal mechanism. Let $\phi_d(x; \mu, \Sigma)$ represent the density of a $d$-variate Gaussian vector $x$ with mean $\mu$ and variance-covariance matrix $\Sigma$. Then the move should be accepted with probability $\min(1, B)$ where

$$B = \frac{\exp\{-H_l(\tilde{\theta}_l, \tilde{p}_l)\} \phi_{dk}(\tilde{p}_k; 0, G(\tilde{\theta}_k)) g(v) p_{lk}}{\exp\{-H_k(\tilde{\theta}_k, \tilde{p}_k)\} \phi_{dk}(\tilde{p}_k; 0, G(\tilde{\theta}_k)) q(u) p_{kl} \left| \frac{\partial(\tilde{\theta}_l, v)}{\partial(\tilde{\theta}_k, u)} \right|}. \quad (6)$$

What is of note here is that the Jacobian of $f$ is evaluated for the intermediate states where we switch models. We also have terms for the momenta which are generated before performing the leapfrog steps within each of the models. The (opposite) proposed move from model $M_l$ to $M_k$ will be accepted with probability $\min(1, B^{-1})$.

The proposal scheme as presented above allows considerable of flexibility to the user. Here we give some general remarks on the scheme.

1. It is not necessary to have the same step-size or number of steps for the generalized leapfrog steps across all models. For example, as the size of the parameter space grows, a larger number of steps at a smaller step-size may prove more efficient.

2. The positive-definite position specific mass matrix $G(\theta_k)$ can have different forms for different models $M_k$. For example, the full expected Fisher information could be used
for smaller models, while in models with a larger number of parameters it may be useful to use a simplified scheme where the off diagonal entries are ignored. This could potentially save on the usual $O(d_k^3)$ computation which would be required to find the Cholesky decomposition and the subsequent inversion.

3. It is possible to randomize the step directions. That is, we could start with step-size $-h_k$ in model $M_k$ instead of necessarily starting with a positive step-size. An extra term should be included in the acceptance probability (6) to account for this.

4. RJRMHMC can exploit previous work done on designing efficient RJMCMC schemes easily. It is simply a case of using these previously developed proposal strategies at the intermediate step, when changing from $(\tilde{\theta}_k, p_k)$ to $(\tilde{\theta}_l, p_l)$. This should enhance the mixing properties the model search algorithm.

5. As in HMC, RMHMC algorithms can show sensitivity to the step-size and number of leapfrog steps. There is no way to automatically choose these free parameters in existence currently. We defer further discussion of this to Section 5.

In the next section we demonstrate the RJRMHMC approach to model selection on two benchmark examples for trans-dimensional MCMC methods. The first is variable selection for logistic regression models and the second is Gaussian finite mixtures with an unknown number of components.

4 Examples

4.1 Variable selection in Logistic regression

Consider a vector of binary outcomes $y = (y_1, \ldots, y_n)$. For each outcome we have a record of each of a set of potential predictors say $x_1, \ldots, x_P$ and relate these to $y$ using a classic logistic regression model. The task is to find subsets $G \subseteq \{1, \ldots, P\}$ giving the indexes of "good" predictors for the outcome $y$. Exhaustive search and ranking of these subsets is usually not possible unless $P$ is a small number, since the total number of subsets is $2^P$. Thus in most realistic situations it will be necessary to resort to some kind of stochastic model search strategy. See for example Brooks, Friel & King (2003). Here we demonstrate how the methods presented in Section 3 can be used for such a search.

A useful strategy in variable selection tasks is the introduction of an indicator vector $\gamma = (\gamma_1, \ldots, \gamma_P)$ where $\gamma_j = 1$ only if $x_j$ is in the model and is zero otherwise. This translates the model search problem into finding high posterior probability $\gamma$ vectors. For a given $\gamma$, the logistic regression model may be stated as

$$\log \left( \frac{p_i}{1 - p_i} \right) = x_{\gamma, i}^T \beta$$

where $p_i = \Pr\{y_i = 1\}$, $x_{\gamma, i}$ is a vector of the covariate values on the $i^{th}$ individual which have a corresponding entry in $\gamma$ equal to one, and $\beta$ is a $d_\gamma \times 1$ vector of regression coefficients for these covariates where $d_\gamma = \sum_{j=1}^P \gamma_j$. Here we always assume that $\gamma_1 = 1$ and first entry
of $x_{\gamma,i}$ is 1, to always have a constant in the model. We assume a zero mean multivariate Gaussian prior on $\beta_\gamma$ with a variance-covariance matrix equal to $\tau^2$ times $I_{d_\gamma \times d_\gamma}$, the $d_\gamma \times d_\gamma$ identity matrix.

Following Section 7 of Girolami & Calderhead (2011) the position specific mass matrix using the expected Fisher information is

$$ G(\beta_\gamma) = X_\gamma^T D_\gamma X_\gamma + \frac{1}{\tau^2} I_{d_\gamma \times d_\gamma} $$

where $X_\gamma = (x_{\gamma,1} \, x_{\gamma,2} \ldots \, x_{\gamma,n})^T$, $D_\gamma$ is an $n \times n$ diagonal matrix with $(i, i)$ entry

$$ D_{\gamma,ii} = \frac{\exp\{x_{\gamma,i}^T \beta_\gamma\}}{(1 + \exp\{x_{\gamma,i}^T \beta_\gamma\})^2}. $$

The partial derivative of $G(\beta_\gamma)$ with respect to any member $\beta_{\gamma,j}$ of $\beta_\gamma$ is given by

$$ \frac{\partial G(\beta_\gamma)}{\partial \beta_{\gamma,j}} = X_\gamma^T D_\gamma C_{\gamma,j} X_\gamma $$

where $C_{\gamma,j}$ is an $n \times n$ diagonal matrix with $(i, i)$ entry

$$ C_{\gamma,j} = \begin{cases} 1 - 2 \frac{\exp\{x_{\gamma,i}^T \beta_\gamma\}}{1 + \exp\{x_{\gamma,i}^T \beta_\gamma\}} \end{cases} x_{\gamma,i}^j $$

where $x_{\gamma,i}^j$ is the $j$th entry of $x_{\gamma,i}$.

4.1.1 Pima Indian dataset

The Pima Indians data records instances of diabetes and a range of possible diabetes indicators for $n = 532$ Pima Indian women aged 21 years or over. Seven potential predictors of diabetes are recorded for this group; number of pregnancies (NP); plasma glucose concentration (PGC); diastolic blood pressure (BP); triceps skin fold thickness (TST); body mass index (BMI); diabetes pedigree function (DP) and age (AGE). This gives 129 potential models (including a model with only a constant term). We took $\tau = 10$ for this data. The results of model choice algorithms, especially estimated Bayes factors can be quite sensitive to these specifications however. We do not discuss this further here, but refer the reader to Friel & Wyse (2012), Kass & Raftery (1995) for further discussion of these issues.

The RJRMHMC algorithm was implemented as follows. At each iteration a within model update was carried out. This was done using RMHMC with a step-size of 0.5 and a randomly chosen number of steps between one and six. Following this, a covariate was either proposed to be removed or added to the model at random. We chose the probability of the add move to be $p_{\text{add}} = 0.5$ except when all the variables were in(out) of the model in which case we deterministically propose respectively remove(add) moves. In each part of the add/remove scheme we employed six leapfrog steps with step-size 0.5. This appeared to give satisfactory mixing for this application. When adding a covariate to the model, this was chosen at random from those available. It’s value at the intermediate stage of the sampling scheme was drawn from a $N(0, \sigma^2_{\beta})$ distribution. This was simply appended to $\tilde{\beta}_\gamma$ to give $\hat{\beta}_\gamma$, making
the Jacobian term in expression (6) equal to one. When removing a covariate, this was chosen at random from those already in the model.

Suppose there are $P$ potential predictors. Then a move from a model with $\{\gamma, \beta_\gamma\}$ to $\{\gamma', \beta_{\gamma'}\}$ where $d_{\gamma'} = d_\gamma + 1$ is accepted with probability which is the minimum of 1 and

$$\frac{\exp\{-H_{\gamma'}(\beta_{\gamma'}, p_{\gamma'})\} \phi_{d_{\gamma'}}(\tilde{p}_{k}; 0, G(\beta_{\gamma'}))}{\exp\{-H_{\gamma}(\beta_\gamma, p_\gamma)\} \phi_{d_\gamma}(\tilde{p}_{k}; 0, G(\beta_\gamma)) \phi(u; 0, \sigma_{\beta^2})} \frac{1}{d_\gamma} P - d_{\gamma'}.$$

The probability of the reverse move is calculated similarly.

In addition to the RJRMHMC algorithm we also ran a classic reversible jump algorithm. The proposed parameter $\beta_{\gamma'}$ was constructed similarly to the RJRMHMC case, by generating $u$ from a univariate Gaussian and using this as the value of the coefficient to be added. We performed a considerably longer run of the RJMCMC algorithm (5,000,000 iterations) than the RJRMHMC algorithm (500,000) although these two runs took roughly the same CPU time. The length of the RJMCMC run makes it feasible to assume inferences derived from the chain as “close to the truth”, and compare inferences derived from the RJRHMHM algorithm with these.

As a simple diagnostic, we examine traceplots of the posterior probability of inclusion of variables in the model over the progression of the Markov chain. For any variable this
Figure 3: Distribution of $\gamma$ vectors (converted from base 2) for RJRMHMC (left) and RJMCMC (right) runs.

Table 1: Probability of inclusion of each covariate for the Pima Indians data for RJRMHMC and RJMCMC runs.

<table>
<thead>
<tr>
<th>Probability</th>
<th>NP</th>
<th>PGC</th>
<th>BP</th>
<th>TST</th>
<th>BMI</th>
<th>DP</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RJRMHMC</td>
<td>0.926</td>
<td>1.000</td>
<td>0.014</td>
<td>0.019</td>
<td>0.997</td>
<td>0.955</td>
<td>0.138</td>
</tr>
<tr>
<td>RJMCMC</td>
<td>0.929</td>
<td>1.000</td>
<td>0.013</td>
<td>0.020</td>
<td>0.997</td>
<td>0.946</td>
<td>0.143</td>
</tr>
</tbody>
</table>

Table 2: Cross tabulation of Bayes factors for RJRMHMC and RJMCMC runs for ranked models in order of appearance in the chain. The entry in column 1 row 2 is $BF_{12}$.  

<table>
<thead>
<tr>
<th>Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12.54</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>13.49</td>
<td>1.08</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>23.69</td>
<td>1.89</td>
<td>1.76</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>57.82</td>
<td>4.61</td>
<td>4.28</td>
<td>2.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12.26</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>13.21</td>
<td>1.08</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>20.43</td>
<td>1.67</td>
<td>1.55</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>54.88</td>
<td>4.47</td>
<td>4.15</td>
<td>2.69</td>
</tr>
</tbody>
</table>

RJRMHMC     RJMCMC
is computed at iteration \( t \) as \( \# \text{ times in model} \leq \text{iteration } t)/t \). This is a reasonable
diagnostic to get an impression of speed of convergence. Figure 2 shows traceplots for
three of the variables (NP, BP and TST) in the Pima dataset. It can be seen that both
chains take some time to converge to a stable but that RJRMHMC seems to be performing
well after 450,000 post burn-in iterations. However, the boost in performance compared to
RJMCMC appears to be marginal and it comes at a much higher cost per iteration. Our
general observation when experimenting with the step-size for between model moves was
that careful tuning of the step-size or the number of steps taken could lead to marginal gains
in the acceptance probability of these moves for this data. This probability rarely got over
2.5%. The acceptance probability for RJMCMC was always roughly 2.2%. So there are no
notable gains from this perspective.

As another measure of performance we investigate how well the RJRMHMC algorithm
estimated the Bayes factor between the two most popular models from the RJMCMC output.
We begin by converting each \( \gamma \) vector in the output from its binary representation to an
integer using \( \sum_{j=0}^{P-1} 2^j \gamma_j \) (we let \( P \) counts the constant term). Each model is converted to
a unique integer. Figure 3 shows the (scaled) frequency distribution of this quantity for
the RJRMHMC and RJMCMC run. It can be seen that both give very similar frequency
distributions. Table 2 shows cross tabulations of Bayes factors between the top five models
output from the two runs. It can be seen that RJRMHMC gives an accurate estimate of the
Bayes factors suggesting it is representing the posterior of \( \gamma \) well.

It is worth noting that the within model mixing for RJRMHMC is superior to RJMCMC
as a result of using RMHMC moves within model. We envisage that the within model moves
should usually mix better for RMHMC that just a simple vanilla Metropolis algorithm. This
was demonstrated in many contexts by Girolami & Calderhead (2011). For example, here
we noticed high acceptance rates of over 90% for the within model RMHMC moves.

### 4.2 Finite mixture models with unknown number of components

Stathopoulos & Girolami (2011) demonstrated how RMHMC outlined in Section 2.2 could be
used for the analysis of Gaussian finite mixture models for a given number of components.
Here attention is turned to Gaussian finite mixtures where the number of components is
unknown. This is a largely documented problem with many papers on trans-dimensional
MCMC using it as a benchmark example. Three of the most notable and related papers
in MCMC methodology dealing with this are those of Richardson & Green (1997), Nobile
& Fearnside (2007) and Stephens (2000). We now demonstrate following Section 3 how
RMHMC can be incorporated into the algorithm of Richardson & Green (1997) and inves-
tigate whether this leads to any notable gains in efficiency.

The assumption is that data \( y = (y_1, \ldots, y_n) \) are iid from some mixture of Gaussian
distributions \( y_i \sim \pi(\cdot|\psi, k) \) and

\[
\pi(\cdot|\psi, k) = \sum_{g=1}^{k} \omega_g \phi(\cdot|\mu_g, \sigma_g^2)
\]

where \( \sum_{g=1}^{k} \omega_g = 1 \) and \( \psi \) is the parameter vector containing \( (\omega_g, \mu_g, \sigma_g^2)_{g=1,\ldots,k} \). This dis-
tribution can be expressed as a sum over latent variables \( z = (z_1, \ldots, z_n) \) which give the
component (or constituent Gaussian density) from which each data point arises:

$$\pi(y|\psi, k) = \sum_z \prod_i \nu_{g} \phi(y_i|\mu, \sigma^2_g)^{z_i=g}.$$ 

Analysis using MCMC usually focuses on this completed likelihood which can be conveniently sampled using a Gibbs sampling scheme for all parameters

$$\pi(y|z, \psi, k) = \prod_{g=1}^{k} \omega_{g} \times \prod_{g=1}^{k} \nu_{g} \prod_{i=1}^{n} \phi(y_i|\mu, \sigma^2_g)^{z_i=g}$$

where \( n_g \) is the number of data points assigned to component \( g \). Some authors have worked with the incomplete data, for example Stathopoulos & Girolami (2011). Here, we use complete data, and show that the position specific mass matrix for the unknown parameters using expected Fisher information may be expressed in a simple diagonal form. We use RMHMC steps to update mean and transformed variance parameters, and use Gibbs updates for the memberships \( z \) and the component weights \( \omega \). These Gibbs updates are documented in e.g. Diebolt & Robert (1994).

We follow Richardson & Green (1997) in guiding our prior elicitation, since their study into prior assumptions revealed that the output of RJMCMC algorithms for mixtures can show sensitivity to certain parameters. We take the prior on the \( \mu_g \) to be \( N(\xi, \kappa^{-1}) \) and on the \( \sigma_g^{-2} \) to be \( \text{Gamma}(\alpha, \beta) \) where the Gamma distribution has mean \( \alpha/\beta \) and variance \( \alpha/\beta^2 \). Let \( R = y_{\text{max}} - y_{\text{min}} \) denote the range of the data. Then Richardson & Green (1997) recommend setting \( \kappa \) to be some small multiple of \( 1/R^2 \). Also, they assume a hyperprior for \( \beta \) which is \( \Gamma(g, r) \) taking \( g = 0.2 \) and \( r = 10/R^2 \). The value of \( \xi \) is taken to be the midpoint of the range of the data \( y_{\text{min}} + R/2 \). The prior on the component weights \( \omega \) is taken to be a uniform Dirichlet \((1, 1, \ldots, 1)\) over the \( k-1 \)-dimensional simplex. The prior on \( k \) was taken to be uniform on \( \{1, \ldots, 30\} \).

In this problem we must transform the component variances \( \sigma_g^2 \) to allow for unconstrained sampling from the posterior. We do this by using the parameterization \( \gamma_g = \log \sigma_g^2 \). This allows \( \gamma_g \) to take values from the entire real line. The transformation is corrected for by inclusion of Jacobian terms in the posterior.

Consider now a component \( g \) with \( n_g \) members. The likelihood contribution from this component will only involve the data currently assigned to component \( g \), so when averaging over the data in order to obtain the expected Fisher information we pick up only these terms. This leads to the position specific mass matrix for the \((\mu_g, \gamma_g)\) parameters in component \( g \) being

$$G_g = \begin{pmatrix} n_g/\sigma_g^2 + 1/\nu^2 & 0 \\ 0 & n_g/2 + \beta/\sigma_g^2 \end{pmatrix}.$$ 

Thus the mass matrix for the entire set of means and transformed variances is then \( G(\theta_k) = \text{BlockDiagonal}(G_1, \ldots, G_k) \).

The within model sampling scheme consists of (i) updating the latent \( z \) giving the component memberships using a Gibbs update, (ii) sample the weights using a Gibbs step from their Dirichlet full conditional form (iii) update the means and transformed variances using
Figure 4: Traceplots of the number of components post burn-in, data with overlain top ten most probable samples from the RJ-RMHMC run and cumulative probability plots of the number of components.
RMHMC with the metric tensor $G(\theta_g)$ given above and (iv) update $\beta$ from its full conditional distribution using a Gibbs step

$$\beta \sim \text{Gamma} \left( g + k\alpha, r + \sum_{j=1}^{k} \sigma_j^{-2} \right).$$

Within each sweep there is then an additional step which proposes either to split or merge a component (equi-probably except at the extremes of one component or $k_{\text{max}}$ components.

Since the posterior of the labels suffers from non-identifiability it is necessary to impose a constraint to ensure that we only sample from one of the $k!$ identical modes of the mixture model. We do this by using a constraint on the component means and imposing $\mu_1 < \mu_2 < \cdots < \mu_k$. Any updates from (iii) above which do not satisfy this constraint are rejected. This is also the case for the split and merge moves described below.

We construct a reversible moves to propose the splitting or combining of components as follows. Suppose the current model has $g$ components and we are proposing to add a component. Firstly we take $h_k$ steps in the current model. Choose a component $g_\star$ at random to split. Form new means and log-variances and weights by generating random variables

$$u_1 = \text{be}(2,2) \quad u_2 = \text{be}(2,2) \quad u_3 = \text{be}(1,1)$$

and setting

$$\tilde{\omega}_{g_1} = u_1 \tilde{\omega}_{g_\star} \quad \tilde{\omega}_{g_2} = (1 - u_1) \tilde{\omega}_{g_\star}$$

$$\tilde{\mu}_{g_1} = \tilde{\mu}_{g_\star} - u_2 \exp\{\tilde{\gamma}_{g_\star}\} \sqrt{\tilde{\omega}_{g_2}} / \tilde{\omega}_{g_1}$$

$$\tilde{\mu}_{g_2} = \tilde{\mu}_{g_\star} + u_2 \exp\{\tilde{\gamma}_{g_\star}\} \sqrt{\tilde{\omega}_{g_1}} / \tilde{\omega}_{g_2}$$

$$\tilde{\gamma}_{g_1} = \log u_3 + \log(1 - u_2^2) + \tilde{\gamma}_{g_\star} - \log u_1$$

$$\tilde{\gamma}_{g_2} = \log(1 - u_3) + \log(1 - u_2^2) + \tilde{\gamma}_{g_\star} - \log(1 - u_1).$$

Resample the members of $g_\star$, allocating to either component $g_1$ or $g_2$. Generate $p_{k+1} \sim N(0, G(\theta_{k+1}^\star))$ and subsequently take $h_{k+1}$ steps in the new model. If at any time the means violate the ordering constraint $\mu_{g_1} < \mu_{g_2}$ or the ordering with the remaining means, either upon forming the new model or after the $h_{k+1}$ steps the move is rejected outright. Otherwise we have the proposed model parameters $(\theta_{k+1}, p_{k+1})$. This move is then reversible, with a similar attention paid to the ordering constraints on the way back. This component splitting move is identical to Richardson & Green (1997) except that we have extra leapfrog steps. Here we did not allow for births and deaths of empty components as in Richardson & Green (1997). The acceptance probability must contain the Jacobian for the bijection forming the new parameters. This is

$$\frac{(1 + u_2^2) |\mu_{g_2} - \mu_{g_1}| \tilde{\omega}_{g_\star}}{u_3(1 - u_3) u_2(1 - u_2^2)}$$

for the transformation used here.
<table>
<thead>
<tr>
<th>Galaxy data</th>
<th>Within model</th>
<th>Split</th>
<th>Combine</th>
</tr>
</thead>
<tbody>
<tr>
<td>RJRMHMC</td>
<td>92.75 (0.51)</td>
<td>12.09 (0.72)</td>
<td>12.14 (0.78)</td>
</tr>
<tr>
<td>RJMCMC</td>
<td></td>
<td>Gibbs (–)</td>
<td>10.30 (0.73)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enzyme data</th>
<th>Within model</th>
<th>Split</th>
<th>Combine</th>
</tr>
</thead>
<tbody>
<tr>
<td>RJRMHMC</td>
<td>94.37 (0.23)</td>
<td>8.44 (0.33)</td>
<td>8.66 (0.36)</td>
</tr>
<tr>
<td>RJMCMC</td>
<td></td>
<td>Gibbs (–)</td>
<td>7.42 (0.24)</td>
</tr>
</tbody>
</table>

Table 3: Investigation into improvements in mean percentage acceptance rates and standard errors in parentheses for each of the algorithms run ten times for 10000 iterations.

4.3 Galaxy and Enzyme datasets

We applied the above RJRMHMC algorithm to the Enzyme and Galaxy data used in Richardson & Green (1997). The algorithm was run for 200,000 iterations using 100,000 as a burn-in. Convergence was monitored by observing the trace of the cumulative probability plots of $k$. These are shown in Figure 4, and demonstrate convergence after 100,000 iterations. Overall acceptance rates for births and deaths were close to 9% for the Enzyme and 12% for the Galaxy data. These are both 1% higher than an RJMCMC algorithm (without births and deaths of empty components). The drawback with the approach suggested here however is that it requires tuning of the stepsize for the generalized Leapfrog steps when moving between models. Generally the stepsize has to be set low for between model moves and can show sensitivity. Setting the within model move step sizes was not such a problem. In both of the examples here this was set to 0.5, with random number of up to six Leapfrog steps taken. One possible explanation for sensitivity to the step-size in between model moves is that the ordering constraint on means prevents wide exploration of the model space. Thus step-sizes which are too large tend to overshoot and lead to rejecting moves. Unfortunately it is not possible at this point to give guidance on choosing the stepsize for between model moves.

To investigate the difference in acceptance rates between the RJRMHMC and RJMCMC, we carried out a small simulation experiment, running ten chains each for ten thousand post burn-in iterations for both algorithms. Average acceptance rates for both the Enzyme and Galaxy data sets are shown in Table 3. The suggestion from this table is that there appears to be some potential for improvements in model jump frequency in reversible jump schemes already in existence by careful tuning of the step-size for between model moves in RJRMHMC.

We examined gain in efficiency in estimation of the posterior distribution of $k$ by running the RJRMHMC and RJMCMC algorithms many times on the Galaxy dataset. One hundred independent runs each of 200,000 RJMCMC, 20,000 RJMCMC and 20,000 RJRMHMC iterations were carried out. Figure 5 shows the posterior distribution of $k$ over the 100 runs. It is supposed that the 200,000 iteration RJMCMC plot should be the most accurate, so we can compare the other two with this to give a rough idea of the gains in estimation by using RMHMC. It appears that although it is more precise, RMHMC tends to place more of the posterior mass on seven as opposed to six components. It also appears to visit models with three and four components much less than RJMCMC.
Figure 5: Results from three experiments involving 100 independent runs of RJRMHMC and RJMCMC algorithms. The top panel is 100 runs of RJMCMC for 200,000 iterations, the middle panel gives RJMCMC for 20,000 iterations and the bottom panel RJRMHMC for 20,000 iterations.
5 Sensitivity to step-size parameter

It is apparent from the results presented in the previous sections that the RJRMHMC algorithm may be sensitive to the choice of step-size $h$ used for the leapfrog steps. Here, we attempt to examine this a little more closely and try to gauge the impact this has on the efficiency of the algorithm in general. We use a very simple simulated changepoint example, where the position of the changepoint is known. We experiment with some different step-sizes and number of steps and look at the corresponding proposals and their acceptance probabilities.

We generated $n = 200$ independent data points from the model

$$y_i \sim \begin{cases} \mathcal{N}(-2, 1) & i \leq 100 \\ \mathcal{N}(+2, 1) & i > 100 \end{cases}$$

obtaining the data shown in Figure 6. Here we will entertain two possible models for the data. The first assumes a constant mean and the second has a changepoint at 100. We call these models $M_0$ and $M_1$ and they may be stated:

$$M_0 : \; y_i \sim \mathcal{N}(\mu, 1)$$

$$M_1 : \; y_i \sim \begin{cases} \mathcal{N}(\mu_1, 1) & i \leq 100 \\ \mathcal{N}(\mu_2, 1) & i > 100 \end{cases}$$

We consider the sensitivity of proposals from $M_0$ to $M_1$ for a range of step-sizes and numbers of steps.

Firstly we assume no change and simulate from model $M_0$. Note that we need to simulate both parameter values $\mu$ and momenta $p$ in this step. Since the position specific mass is constant in this situation, that is, $G = n + \beta$ where a priori $\mu$ is assumed $\mathcal{N}(0, \beta^{-1})$
the parameters and momenta may be generated independently from their Gaussian full
conditionals:
\[
\mu^{(t)} \sim N \left( \frac{1}{n} \sum_{i=1}^{n} y_i, (n + \beta)^{-1} \right) \quad \sigma^{(t)} \sim N(0, n + \beta)
\]
for \( t = 1, \ldots, m \). Here we took \( \beta = 0.25 \).

For each \( t = 1, \ldots, m \) we then propose a move to \( M_1 \) from initial point \( (\mu^{(t)}, \sigma^{(t)}) \) in \( M_0 \) by taking \( N \) leapfrog steps in each model at stepsize \( h \). At the intermediate stage we generate \( u \sim \text{Uniform}[0, 2] \) and set \( \tilde{\mu}_1 = \mu - u \) and \( \tilde{\mu}_2 = \mu + u \). The corresponding proposed values of the means in \( M_1, (\mu_1^{(t)}, \mu_2^{(t)}) \), are shown in Figure 6 for a range of values of \( h \) and \( N \).

The results of the experiment are shown in Figure 7. It can be seen that it is difficult to predict the behaviour of the proposal mechanism for different step-sizes and there does not appear to be any intuitive way to do this. For example the \((N, h)\) pairs \((10, 1), (5, 0.5), (7, 0.5), (7, 0.3)\) and \((10, 0.3)\) all produce similar results for the same starting points, but all operate over different effective time intervals (roughly \( N \times h \)). We can also observe that for \( N = 10 \) and \( h = 0.1 \) we obtain much better proposals that with just three steps less \((N = 7)\). This choice also gives quite similar results to the case \( N = 5, h = 1 \) which could suggest that the latter traverses the iso-contour of \( H \) completely.

The conclusion of this small study is that, as yet, there is little guidance to be offered in choosing either the step-size or number of steps for RJRMHMC, and thus this must be tuned in some way. We note also that this is a more general problem, and not necessarily related to the dynamics introduced by Girolami & Calderhead (2011), since the above example had constant position specific mass matrices and thus did not require use of the generalized leapfrog scheme needed for the other examples.

6 Discussion

The main outcome of this paper was to demonstrate that the ideas of Green (1995) and Girolami & Calderhead (2011) can be combined to give a model selection algorithm in the same vein as RJMCMC. The new algorithm RJRMHMC uses a novel acceptance step to ensure that sampling is from the correct distribution.

The proposed approach was applied to two benchmark examples. The conclusion of the experiments was that with careful tuning of the step-size parameter in RJRMHMC for between model moves, it may well be possible to increase the efficiency of existing RJMCMC algorithms. In the Gaussian finite mixtures example, we observed an improvement over the algorithm of Richardson & Green (1997) in terms of acceptances of jumps between models. Variable selection for the Pima Indians data did not show such improvements in mixing across models. However, the posterior model space appeared to be represented well by a short (relative to RJMCMC) run of RJRMHMC, and Bayes factors well estimated.

Many open issues were highlighted in the paper. Most notably the choice of the step-size and number of steps for the leapfrog scheme on the toy changepoint model in Section 5. An automatic algorithm or rule of thumb guideline for choosing these free parameters would be ideal, but as yet there is little guidance in this direction. This would certainly be the most important issue to tackle in further development of this algorithm.
Figure 7: Proposed means for changepoint example. Blue corresponds to $\mu_1$ and green to $\mu_2$. Step-sizes are $h = 1, 0.5, 0.3, 0.1$ (rows) and number of steps are $N = 5, 7, 10$ (columns).
Overall, the algorithm presented here should improve as the theory around RMHMC and HMC grows. As is, it offers an extension to the existing methodology in these areas. Following the paper of Girolami & Calderhead (2011) which gave a significant boost to this research area it is not too pessimistic to hope that there will be a flurry of work surrounding these methods in the coming years which may, overall, lead to fixes of some of the issues that plague MCMC algorithms in general. Hopefully, this will bring us one step closer to the ultimate Monte Carlo algorithm.

References


