A generalization of the Multiple-try Metropolis algorithm for Bayesian estimation and model selection

Silvia Pandolfi
University of Perugia, IT

Francesco Bartolucci
University of Perugia, IT

Nial Friel
University College Dublin, IE

Abstract

We propose a generalization of the Multiple-try Metropolis (MTM) algorithm of Liu et al. (2000), which is based on drawing several proposals at each step and randomly choosing one of them on the basis of weights that may be arbitrary chosen. In particular, for Bayesian estimation we also introduce a method based on weights depending on a quadratic approximation of the posterior distribution. The resulting algorithm cannot be reformulated as an MTM algorithm and leads to a comparable gain of efficiency with a lower computational effort. We also outline the extension of the proposed strategy, and then of the MTM strategy, to Bayesian model selection, casting it in a Reversible Jump framework. The approach is illustrated by real examples.

1 INTRODUCTION

A well known method for Bayesian estimation is the Metropolis Hastings (MH) algorithm proposed by Metropolis et al. (1953) and modified by Hastings (1970) . The algorithm allows us to generate a Markov chain whose stationary distribution is equal to the target distribution $\pi(\boldsymbol{x})$ which, in the Bayesian context, corresponds to the posterior distribution of the model parameters. The same strategy may be adopted for Bayesian model choice and leads to the Reversible Jump (RJ) algorithm, see Green (1995). This algorithm generates a reversible Markov chain which jumps between parameter spaces of different models, which may have different dimensions.

The MH algorithm was extended by Liu et al. (2000) to the case of Multiple-try Metropolis (MTM). The extended method consists of proposing, at each step, a fixed number of moves and then selecting one of them with a certain probability. A larger portion of the explored sample space and a better mixing result. It is also worth noting that, in order to attain the detailed balance condition, MTM uses selection probabilities which are proportional to the product of the target, the proposal, and a λ-function which has to be nonnegative and symmetric. Moreover, at least to our knowledge, a Multiple-try extension of the RJ algorithm has not been proposed, although this extension may represent a natural way to overcome typical problems of this algorithm.

In this paper, we propose a generalization of the MTM scheme in which the selection probabilities are defined so that minimal constraints are necessary to attain the detailed balance condition. These constraints are much weaker than those required in the original MTM algorithm. In principle, any mathematical function giving valid probabilities may be adopted, even if this choice is crucial for the efficiency in the estimation of the target distribution. The connection between the two algorithms is illustrated in detail. We also introduce an useful choice of the selection probabilities which is based on a quadratic approximation of the target distribution. This choice leads to a considerable gain of efficiency over the traditional MH algorithm, while being much less computing demanding than the MTM algorithm.

We also introduce a Generalized Multiple-try version of the RJ algorithm, so as to increase the efficiency even from a Bayesian model selection prospective. In this case, the acceptance probability includes a component that depends on the Jacobian of the transformation between different parameter spaces.

The paper is structured as follows. In the following Section we briefly review the MH, RJ, and MTM algorithms. In Section 3 we introduce our generalization of the MTM strategy for Bayesian estimation. In Section 4 we outline the extension to Bayesian model selection.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

The proposed approach is illustrated in Section 5 by some applications.

2 PRELIMINARIES

We first introduce basic concepts about the MH and RJ algorithms and then we outline the MTM methodology.

2.1 METROPOLIS HASTINGS AND REVERSIBLE JUMP ALGORITHM

The MH algorithm is a Markov Chain Monte Carlo method that can be used to generate random samples from a target distribution $\pi(\boldsymbol{x})$ for which direct sampling is cumbersome. The basic idea of the algorithm is to construct an ergodic Markov chain in the state space of $\boldsymbol{x}$ that has the target distribution as its stationary distribution.

This algorithm may be seen as a form of generalized rejection sampling, where the next state of the chain $\boldsymbol{x}_{t+1}$ is drawn from a proposal distribution $T(\boldsymbol{x}_t, \cdot)$ and the candidate point $\boldsymbol{x}_{t+1} = \boldsymbol{y}$ is accepted with probability

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x}_t)}{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t, \boldsymbol{y})}\right\}.$$

In this way, the generated chain is reversible, with invariant/stationary distribution $\pi(\boldsymbol{x})$, because it satisfies the *detailed balance condition*, i.e.

$$\pi(\boldsymbol{y})P(\boldsymbol{y}, \boldsymbol{x}) = \pi(\boldsymbol{x})P(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $P(\boldsymbol{y}, \boldsymbol{x})$ is the transition kernel from $\boldsymbol{y}$ to $\boldsymbol{x}$.

A potential problem of the above algorithm is that draws are often highly correlated. Therefore, the estimates based on the generated sample tend to have high variance. Moreover, the algorithm may have slow convergence, since it may be trapped in a local mode of the target function. Another challenge is the choice of an efficient proposal. In fact, it is often the case that a small step-size in the proposal transition results in a slow convergence of the corresponding Markov chain, whereas a large step-size results in a very low acceptance rate (Liu, 2001).

2.2 REVERSIBLE JUMP ALGORITHM

In the Bayesian model choice context, the RJ algorithm introduced by Green (1995) uses the MH paradigm to build a suitable reversible chain which is able to jump between models with different parameter space dimensions.

Let $\{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$ denote the set of available models and, for model $\mathcal{M}_m$, let Θ_m be the parameter space, whose elements are denoted by $\boldsymbol{\theta}_m$. In simulating from the target distribution, the RJ algorithm moves both within and between models. Moreover, the move from the current state of Markov chain $(m, \boldsymbol{\theta}_m)$ to a new state has to be performed so as to ensure that the detailed balance condition holds. For this aim, Green (1995) introduced a set of auxiliary variables, such that all states of the chain have the same dimension. In particular, a jump between models $\mathcal{M}_i$ and $\mathcal{M}_j$ is achieved supplementing each of the parameter spaces Θ_i and Θ_j with artificial spaces in order to create a bijection between them and to impose a dimension matching condition; see also Brooks et al. (2003).

The probability of acceptance for the move from $\mathcal{M}_i$ to $\mathcal{M}_j$ is computed involving the Jacobian of the transformation from the current value of the parameters to the new value. The acceptance probability also includes the probability of choosing the jump and the density distribution of the auxiliary variables.

2.3 MULTIPLE-TRY METROPOLIS ALGORITHM

The MTM method enables one to propose multiple trial points from the proposal distribution in order to make large step-size jumps, without lowering the acceptance rate. The selection probabilities of the proposed points are calculated, and the selected point is then accepted, according to a modified MH ratio.

Let $T(\boldsymbol{x}, \boldsymbol{y})$ be an arbitrary proposal function satisfying the condition $T(\boldsymbol{x}, \boldsymbol{y}) > 0 \iff T(\boldsymbol{y}, \boldsymbol{x}) > 0$ and let $\lambda(\boldsymbol{x}, \boldsymbol{y})$ be an arbitrary non-negative symmetric function. Suppose the current state of Markov chain is $\boldsymbol{x}_t$. The MTM strategy performs the following steps:

Step 1: Draw k independent trial proposals $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ from $T(\boldsymbol{x}_t, \cdot)$.

Step 2: Select a point $\boldsymbol{y}$ from $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ with probability proportional to

$$w(\boldsymbol{y}_j, \boldsymbol{x}_t) = \pi(\boldsymbol{y}_j)T(\boldsymbol{y}_j, \boldsymbol{x}_t)\lambda(\boldsymbol{y}_j, \boldsymbol{x}_t), \tag{2}$$

where $j = 1, \ldots, k$.

Step 3: Draw $\boldsymbol{x}^*_1, \ldots, \boldsymbol{x}^*_{k-1}$ from the distribution $T(\boldsymbol{y}, \cdot)$ and set $\boldsymbol{x}^*_k = \boldsymbol{x}_t$.

Step 4: Accept $\boldsymbol{y}$ with probability

$$\alpha = \min\left\{1, \frac{w(\boldsymbol{y}_1, \boldsymbol{x}_t) + \ldots + w(\boldsymbol{y}_k, \boldsymbol{x}_t)}{w(\boldsymbol{x}^*_1, \boldsymbol{y}) + \ldots + w(\boldsymbol{x}^*_k, \boldsymbol{y})}\right\}. \tag{3}$$

Liu et al. (2000) proved that such a Metropolis scheme satisfies the detailed balance condition and therefore defines a reversible Markov chain with $\pi(\boldsymbol{x})$ as its stationary distribution.

Several special cases of this algorithm are possible, the most interesting of which is when $\lambda(\boldsymbol{x}, \boldsymbol{y}) = \{T(\boldsymbol{x}, \boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x})\}^{-1}$. In this case, $w(\boldsymbol{x}, \boldsymbol{y})$ corresponds to the importance weight of $\boldsymbol{x}$ when the sampling distribution is $T(\boldsymbol{y}, \boldsymbol{x})$ and the target is $\pi(\boldsymbol{x})$. We refer to this version of the algorithm as MTM-inv. Another interesting choice is $\lambda(\boldsymbol{x}, \boldsymbol{y}) = 1$, which leads to the MTM-I algorithm.

The efficiency of the MTM method relies on the calibration between the proposal step size, the number of trials k, and the landscape of the target distribution $\pi(\boldsymbol{x})$.

3 GENERALIZED MTM METHOD

The key innovation of the Generalized MTM (GMTM) algorithm is that it uses selection probabilities that are not constrained as in (2). These probabilities are easily computed, so as to increase the number of trials without loss of efficiency. We also show that replacing the target distribution by a quadratic approximation results in performance very similar to the MTM algorithm, with less computational effort.

3.1 THE ALGORITHM

Let $w^*(\boldsymbol{y}, \boldsymbol{x})$ be an arbitrary function satisfying $w^*(\boldsymbol{y}, \boldsymbol{x}) > 0$. Let $\boldsymbol{x}_t$ be the current state of Markov chain at iteration t. The GMTM algorithm performs the following step:

Step 1: Draw k trials $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ from a proposal distribution $T(\boldsymbol{x}_t, \cdot)$.

Step 2: Select a point $\boldsymbol{y}$ from the set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ with probability given by

$$p_{\boldsymbol{y}} = \frac{w^*(\boldsymbol{y}, \boldsymbol{x}_t)}{\sum_{j=1}^k w^*(\boldsymbol{y}_j, \boldsymbol{x}_t)}.$$

Step 3: Draw realizations $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{k-1}^*$ from the distribution $T(\boldsymbol{y}, \cdot)$ and set $\boldsymbol{x}_k^* = \boldsymbol{x}_t$.

Step 4: Define

$$p_{\boldsymbol{x}_t} = \frac{w^*(\boldsymbol{x}_t, \boldsymbol{y})}{\sum_{j=1}^k w^*(\boldsymbol{x}_j^*, \boldsymbol{y})}.$$

Step 5: The transition from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1} = \boldsymbol{y}$ is accepted with probability

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x}_t)p_{\boldsymbol{x}_t}}{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t, \boldsymbol{y})p_{\boldsymbol{y}}}\right\}. \qquad (4)$$

For the proof that the detailed balance condition is attained see the Appendix (Theorem 5.1).

In order to show that the MTM algorithm is a special case of the GMTM algorithm, consider that (4) may be rewritten as

$$\alpha = \min\left\{1, \frac{\sum_j w^*(\boldsymbol{y}_j, \boldsymbol{x}_t)}{\sum_j w^*(\boldsymbol{x}_j^*, \boldsymbol{y})} \frac{\pi(\boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x}_t)w^*(\boldsymbol{x}_t, \boldsymbol{y})}{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t, \boldsymbol{y})w^*(\boldsymbol{y}, \boldsymbol{x}_t)}\right\},$$

so that:

1. if $w^*(\boldsymbol{y}_j, \boldsymbol{x}_t) = \pi(\boldsymbol{y}_j)T(\boldsymbol{y}_j, \boldsymbol{x}_t)$, then the acceptance ratio above reduces to that characterizing the MTM-I scheme;

2. if $w^*(\boldsymbol{y}_j, \boldsymbol{x}_t) = \dfrac{\pi(\boldsymbol{y}_j)}{T(\boldsymbol{x}_t, \boldsymbol{y}_j)}$, then the MTM-inv algorithm results.

Our main interest is to explore situations where the selection probabilities are easy to compute so as to increase the efficiency of the GMTM algorithm.

3.2 QUADRATIC APPROXIMATION OF THE TARGET DISTRIBUTION

Even if in principle one could choose any mathematical function giving valid probabilities $p_{\boldsymbol{y}}$, we aim at using selection probabilities so that the resulting algorithm is more effective than the standard MTM.

We propose to use a quadratic approximation of the target distribution given by

$$\begin{aligned} \pi^*(\boldsymbol{y}) &= \pi(\boldsymbol{x}_t)A(\boldsymbol{y}, \boldsymbol{x}_t) \\ A(\boldsymbol{y}, \boldsymbol{x}_t) &= e^{\boldsymbol{s}(\boldsymbol{x}_t)'(\boldsymbol{y}-\boldsymbol{x}_t)+\frac{1}{2}(\boldsymbol{y}-\boldsymbol{x}_t)'\boldsymbol{D}(\boldsymbol{x}_t)(\boldsymbol{y}-\boldsymbol{x}_t)}, \end{aligned}$$

where $\boldsymbol{s}(\boldsymbol{x})$ and $\boldsymbol{D}(\boldsymbol{x})$ are, respectively, the first and second derivatives of $\log \pi(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. The selection probabilities are then proportional to

$$w^*(\boldsymbol{y}, \boldsymbol{x}_t) = \pi^*(\boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x}_t)\lambda(\boldsymbol{y}, \boldsymbol{x}_t).$$

In particular, using the MTM-inv version, the selection probability simplifies to

$$p_{\boldsymbol{y}} = \frac{A(\boldsymbol{y}, \boldsymbol{x}_t)}{T(\boldsymbol{x}_t, \boldsymbol{y})} \Big/ \sum_j \frac{A(\boldsymbol{y}_j, \boldsymbol{x}_t)}{T(\boldsymbol{x}_t, \boldsymbol{y}_j)} \qquad (5)$$

Then, after some simple algebra, we find an expression that, contrary to the MTM algorithm, does not require to compute the target distribution for each proposed value, saving much computing time.

4 GENERALIZATION OF THE RJ ALGORITHM

We extend the GMTM strategy to the Bayesian model selection problem in order to overcome some of the typical drawbacks of the RJ algorithm. The latter usually requires accurate tuning of the jump proposals in order to promote mixing among models. The extension consists of proposing, at each step, a fixed number of moves, so as to improve the performance of the algorithm and to increase the efficiency from a Bayesian model selection perspective.

4.1 THE GENERALIZED MULTIPLE-TRY RJ ALGORITHM

Suppose the Markov chain currently visits model $\mathcal{M}_i$ with parameters $\boldsymbol{\theta}_i$ and let $w^*(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_i)$ be an arbitrary function satisfying $w^*(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_i) > 0$. The proposed strategy (GMTRJ) is based on the following:

Step 1: Choose a subset of models $\mathcal{M}_{\mathcal{A}} = \{M_s : s \in \mathcal{A}\}$ for some index set $\mathcal{A} = \{a_1, \ldots, a_k\}$ of size k from which to propose trials.

Step 2: Draw parameters $\boldsymbol{\theta}_{a_1}, \ldots, \boldsymbol{\theta}_{a_k}$ of models $\mathcal{M}_{a_1}, \ldots, \mathcal{M}_{a_k}$, respectively, with proposal density $T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_1}), \ldots, T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_k})$.

Step 3: Choose $\boldsymbol{\theta}_{a_j}$ from $\{\boldsymbol{\theta}_{a_1}, \ldots, \boldsymbol{\theta}_{a_k}\}$ with probability given by

$$p_{\boldsymbol{\theta}_{a_j}} = \frac{w^*(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_i)}{\sum_{h=1}^{k} w^*(\boldsymbol{\theta}_{a_h}, \boldsymbol{\theta}_i)}.$$

Step 4: Choose a subset of models $\mathcal{M}_{\mathcal{B}} = \{m_t : t \in \mathcal{B}\}$ for some index set $\mathcal{B} = \{b_1, \ldots, b_k\}$ where $b_k = i$.

Step 5: Draw parameters $\boldsymbol{\theta}_{b_1}, \ldots, \boldsymbol{\theta}_{b_{k-1}}$ of $\mathcal{M}_{b_1}, \ldots, \mathcal{M}_{b_{k-1}}$, with proposal density $T(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_{b_1}), \ldots, T(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_{b_{k-1}})$ and set $\boldsymbol{\theta}_{b_k} = \boldsymbol{\theta}_i$.

Step 6: Define

$$p_{\boldsymbol{\theta}_i} = \frac{w^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_j})}{\sum_{h=1}^{k} w^*(\boldsymbol{\theta}_{b_h}, \boldsymbol{\theta}_{a_j})}.$$

Step 7: Accept the move from $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_{a_j}$ with probability

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{\theta}_{a_j}) T(\boldsymbol{\theta}_{a_j}, \boldsymbol{\theta}_i) p_{\boldsymbol{\theta}_i}}{\pi(\boldsymbol{\theta}_i) T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_j}) p_{\boldsymbol{\theta}_{a_j}}} |J(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_j})|\right\}.$$

It is possible to prove that the GMTRJ algorithm satisfies detailed balance; see Theorem 5.2 in Appendix.

Similarly to the previous algorithm, even in this case it is possible to consider some special cases of the GMTRJ scheme:

1. if $w^*(\boldsymbol{\theta}_{a_h}, \boldsymbol{\theta}_i) = \pi(\boldsymbol{\theta}_{a_h}) T(\boldsymbol{\theta}_{a_h}, \boldsymbol{\theta}_i)$, then we have the MTRJ-I scheme;
2. if $w^*(\boldsymbol{\theta}_{a_h}, \boldsymbol{\theta}_i) = \dfrac{\pi(\boldsymbol{\theta}_{a_h})}{T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_h})}$, then we have the MTRJ-inv scheme;
3. if $w^*(\boldsymbol{\theta}_{a_h}, \boldsymbol{\theta}_i) = \dfrac{\pi^*(\boldsymbol{\theta}_{a_h})}{T(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{a_h})}$, where $\pi^*(\boldsymbol{\theta}_{a_h})$ is the quadratic approximation of the target distribution, then the GMTRJ scheme results.

5 SOME APPLICATIONS

We tested our methods by three examples about Bayesian estimation and model selection. The first example is about a logistic regression model and is illustrated in some detail. The other two are about a linear and a latent class model and, for reason of space, are only briefly explained.

5.1 LOGISTIC REGRESSION ANALYSIS

To illustrate the GMTM method we estimated a logistic regression model for the number of survivals in a sample of 79 subjects suffering from a certain illness. The patient condition, A, and the received treatment, B, are the explanatory factors. See Dellaportas et al. (2002) for details.

We considered the model formulated as

$$Y_{ij} \sim Bin(n_{ij}, p_{ij}), \qquad \text{logit}(p_{ij}) = \mu + \mu_i^A + \mu_j^B + \mu_{ij}^{AB}$$

where, for $i, j = 1, 2$, Y_{ij}, n_{ij} and p_{ij} are, respectively, the number of survivals, the total number of patients and the probability of survival for the patients with condition i who received treatment j. The parameter vector of the model is $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3) = (\mu, \mu_2^A, \mu_2^B, \mu_{22}^{AB})$. As in Dellaportas et al. (2002), we used the prior $N(0, 8)$ for any of these parameters, which by assumption are also a priori independent.

In this framework, we compared the following algorithms in terms of efficiency in drawing samples from the posterior distribution of $\boldsymbol{\beta}$: MH, MTM-I, MTM-inv, and GMTM (based on the quadratic approximation illustrated in Section 3.2). In all algorithms, the proposal distribution used to update the parameters is $\boldsymbol{\beta}_{t+1} \sim N(\boldsymbol{\beta}_t, \sigma_p^2)$ and as initial value of the Markov chain we used $\boldsymbol{\beta} = \mathbf{0}$. We evaluated the results with three different values of σ_p (0.5,0.75,2), three different numbers of trials k (10,50,100) and on the basis of 1,000,000 draws, with a burn-in of 50,000 iterations.

The above algorithms were compared in terms of acceptance rate and efficiency of the estimator of the posterior expected value of $\boldsymbol{\beta}$. For each algorithm, the acceptance rate is reported in Table 1.

Table 1: Acceptance rates for the logistic example

		$\sigma_p = 0.5$	$\sigma_p = 0.75$	$\sigma_p = 2$
	MH	13.98	5.15	0.20
$k = 10$	MTM-I	55.56	32.63	1.85
	MTM-inv	54.93	32.65	1.88
	GMTM	49.63	29.15	1.83
$k = 50$	MTM-I	71.94	62.69	8.20
	MTM-inv	77.72	63.80	8.23
	GMTM	62.00	51.15	7.40
$k = 100$	MTM-I	74.74	71.29	14.68
	MTM-inv	83.94	73.91	14.69
	GMTM	64.55	57.03	12.70

We observe that, for all algorithms, the acceptance rate is strongly affected by the parameter values of the proposal distribution, but the MTM and GMTM algorithms always outperform the MH algorithm. As for the comparison between the MTM and GMTM strategies, it is important to note that the latter leads to an acceptance rate which is slightly lower than that of the first, but this reduction is more than compensated by the saved computing time due to the use of the quadratic approximation illustrated in Section 3.2.

The efficiency of the algorithms is measured on the basis of the ratio $R = \sigma_a^2/\sigma^2$, where σ^2 is the Monte Carlo variance of the mean estimator and σ_a^2 is the asymptotic variance of the same estimator based on the draws generated by the algorithm of interest. In particular, σ_a^2 is computed on the basis of the autocorrelation between these draws. For each parameter in $\boldsymbol{\beta}$, the results are given in Table 2 for $\sigma_p = 0.5, 0.75, 2$ and $k = 50$. Moreover, in Figure 1 we show some autocorrelation plots for β_1 when $\sigma_p = 2$ and $k = 50$.

It is clear that the GMTM algorithm can reach results similar to the MTM algorithm with less computational effort. Especially, when the value of σ_p is not adequate, the MH sampler performs poorly and the resulting draws are highly correlated. In these situations, the Multiple-try strategy is more effective and, when the number of trials becomes large, the GMTM algorithm allows us to obtain an important gain of efficiency after adjusting for the computing time.

In the same framework, we also considered the problem of Bayesian model selection, closely following the example in Dellaportas et al. (2002). In particular, we considered five possible models: $\mathcal{M}_1$ (intercept); $\mathcal{M}_2$ (intercept + A); $\mathcal{M}_3$ (intercept + B); $\mathcal{M}_4$ (intercept + A + B); $\mathcal{M}_5$ (intercept + A + B + $A.B$). The last model, also termed as full model, is the same considered above and based on the parameter vector $\boldsymbol{\beta}$ having four elements. We assumed again a $N(0, 8)$ prior distribution for $\boldsymbol{\beta}$, whereas the proposal distribution $N(\boldsymbol{\beta}_t, \sigma_p^2)$ was also used to jump from one model to another.

Table 2: Values (divided by 1000) of R for the logistic example with $k = 50$

	β_0	β_1	β_2	β_3
$\sigma_p = 0.5$				
MH	0.0217	0.0227	0.0206	0.0219
MH*	0.6450	0.6771	0.6127	0.6514
MTM-I	0.0079	0.0069	0.0075	0.0080
MTM-inv	0.0038	0.0040	0.0039	0.0039
GMTM	0.0058	0.0062	0.0060	0.0058
MTM-I*	2.3664	2.0696	2.2438	2.3878
MTM-inv*	0.8848	0.9282	0.9067	0.9024
GMTM*	0.7738	0.8227	0.7933	0.7762
$\sigma_p = 0.75$				
MH	0.0493	0.0460	0.0431	0.0393
MH*	1.4384	1.3432	1.2591	1.1483
MTM-I	0.0054	0.0048	0.0052	0.0055
MTM-inv	0.0041	0.0039	0.0041	0.0038
GMTM	0.0062	0.0067	0.0062	0.0061
MTM-I*	1.3037	1.1661	1.2558	1.3192
MTM-inv*	0.9841	0.9380	0.9699	0.9190
GMTM*	0.8568	0.9184	0.8593	0.8454
$\sigma_p = 2$				
MH	0.8030	0.8293	0.7832	0.8144
MH*	23.5490	24.3220	22.9690	23.8860
MTM-I	0.0251	0.0254	0.0282	0.0243
MTM-inv	0.0268	0.0284	0.0273	0.0260
GMTM	0.0261	0.0280	0.0328	0.0263
MTM-I*	5.9598	6.0312	6.6819	5.7700
MTM-inv*	6.5457	6.9318	6.6629	6.3516
GMTM*	3.4431	3.6968	4.3265	3.4667

*Computing time taken into account.

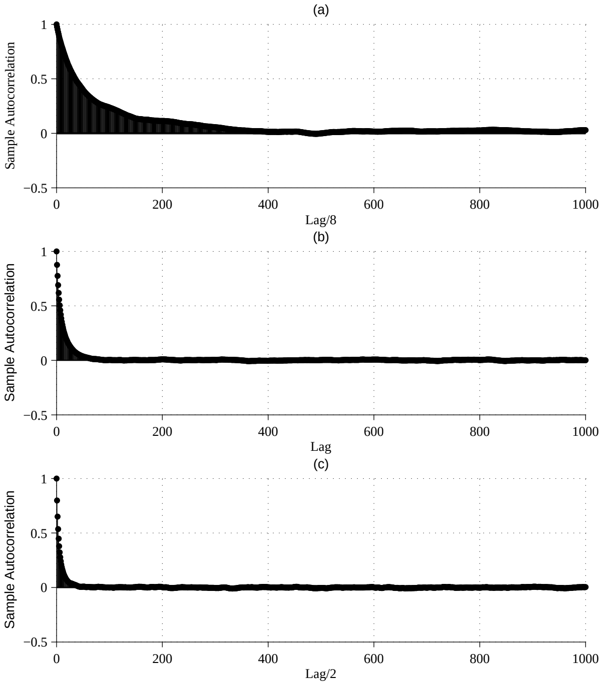

Figure 1: Autocorrelation plot for β_1 with $\sigma_p = 2$ and $k = 50$ (computing time taken into account): (a) MH, (b) MTM-inv, (c) GMTM.

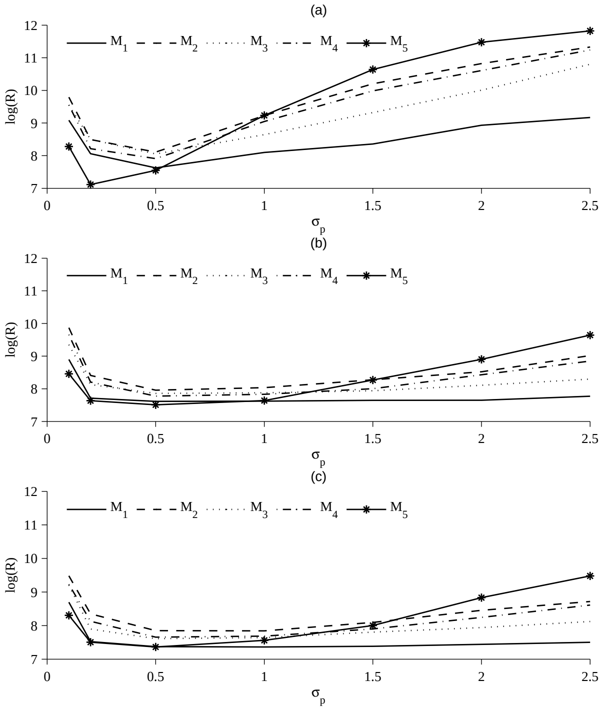

Figure 2: Values of the index R as σ_p increases with $k = 50$ (computing time taken into account): (a) RJ, (b) MTRJ-inv, (c) GMTRJ

In this case, we applied the following algorithms: RJ, MTRJ-inv, and GMTRJ. We used a larger set of values of σ_p (0.1,0.2,0.5,1,1.5,2,2.5), and, for the last two algorithms, three different numbers of trials ($k = 10, 50, 100$). All the Markov chains were initialized from the full model, with $\boldsymbol{\beta} = \mathbf{0}$, and their moves were restricted to adjacent models (which increase or decrease the model dimension by one). In the MTRJ-inv and GMTRJ algorithms, the Multiple-try strategy is only applied in drawing the parameter values. For all algorithms, every iteration consists of one GMTM step, in order to update the parameters of the current model, and one RJ, MTRJ-inv, or GMTRJ step, in order to jump from a model to another. Finally, each Markov chain ran for 1,000,000 iterations, discarding the first 50,000 as burn-in.

Limited to the case of $k = 50$ trials, the results of the above comparison are reported in Figure 2, which shows how the index R (computed by an adjusted formula which takes into account the permanence in the same model and corrected for the computing time) behaves as σ_p increases. In Figure 3 we also report the efficiency ratios, corrected again for the computing time.

We observe that, for most values of σ_p there is a consistent gain of efficiency of the MTRJ-inv and GMTRJ algorithms with respect to the RJ algorithm. This gain of efficiency corresponds to an increase of the acceptance rate, which is higher in both the MTRJ-inv and the GMTRJ algorithms with respect to the RJ algorithm. Overall, the proposed algorithm GMTRJ outperforms the other two algorithms, when the computing time is properly taken into account.

5.1.1 Linear regression analysis

In this experiment, we considered a linear regression problem based on a collection of possible predictor variables, among which we have to choose the ones to be included. In particular, we defined a normal-gamma prior system for which we can calculate posterior model probability exactly, and so we can compare the performance of the MTRJ algorithm to the standard RJ algorithm (Bernardo and Smith, 1994). Here, 50 independent datasets were generated with four predictor variables, three of which were used to simulate the response data. Therefore, there are $2^4 - 1 = 15$ possible models containing at least one variable.

The MTRJ algorithm was applied to each dataset, where trials to every model including the current model were proposed. The parameter values from the proposed models were drawn from their full-conditional distributions. The RJ algorithm was also applied to each dataset, where a move to a model which increased or decreased the model dimension

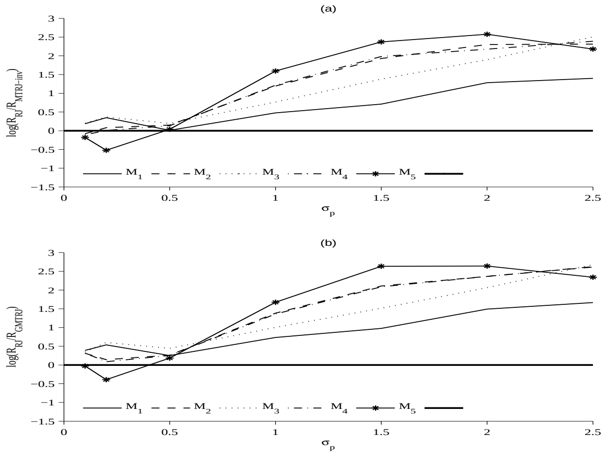

Figure 3: Efficiency ratio as σ_p increases with $k = 50$ (computing time taken into account): (a) RJ versus MTRJ-inv, (b) RJ versus GMTRJ.

by one was proposed. Again, parameter values were drawn from full-conditional distributions. For a fair comparison, the computing time was chosen to be the same for both algorithms.

In order to compare the performance of each algorithm, we computed a weighted distance between the exact posterior model probabilities and those estimated from each algorithm. Overall, 40 out of 50 of the datasets yielded a smaller weighted distance between the exact and estimated MTRJ model probabilities, than between the exact and estimated RJ probabilities. This result indicates that mixing of the first is improved with respect to the second.

5.1.2 Latent class analysis

We considered the same latent class model and the same data considered by Goodman (1974), which concern the responses to four dichotomous items of a sample of 216 respondents. These items were about the personal feeling toward four situations of role conflict; then there are four response variables collected in the vector $\boldsymbol{Y} = (Y_1, Y_2, Y_3, Y_4)$.

Parameters of the model are the class weights π_c and the conditional probabilities of success $\lambda_{j|c}$, where $c = 1, \ldots, C$, with C denoting the number of classes. On the basis of these parameters, the probability of a response configuration $\boldsymbol{y}$, with elements y_j, is given by

$$P(\boldsymbol{y}) = \sum_{c=1}^{C} \pi_c \prod_{j=1}^{4} \lambda_{j|c}^{y_j} (1 - \lambda_{j|c})^{1-y_j}.$$

A priori, we assumed a Dirichlet distribution for the parameter vector $(\pi_1, \ldots, \pi_C)$ and independent Beta distributions for the parameters $\lambda_{j|c}$. Finally, for C we assumed a uniform distribution between 1 and C_{max}.

The objective of the analysis was the inference about the number of classes (C), and the parameters $\lambda_{j|c}$ and π_c. At this aim we exploited the approach of Richardson and Green (1997), which consists of a RJ strategy where the moves are restricted to models with one more or one less component. In this approach, we associated to each subject in the sample an *allocation variable* z_i equal to c when subject i belongs to latent class c. The a priori distribution of each z_i depends on the class weights π_c; see also Cappe et al. (2003). We considered two different pair of dimension-changing moves: split-combine and birth-death. The first one consists of a random choice between splitting an existing component into two or combining two components into one, whereas the second one consists of adding a new empty component or deleting an existing one. At every iteration, split-combine and birth-death moves are preceded by a Gibbs move aimed at updating the parameters of the current model sampling from the full conditional distribution.

In this application, we compared the standard RJ algorithm with the MTRJ-inv algorithm. Within the split-combine move, the Multiple-try strategy consists of choosing, at each step, among a fixed number k of different components to split or to combine. Moreover, the birth-death move is carried out by selecting among k new components to add or among k empty components to delete.

We ran each Markov chain for 1,000,000 sweeps following a burn-in of 200,000 iterations; moreover, we set $C_{max} = 10$. From the output of the algorithm, it results that MTRJ-inv algorithm has a higher acceptance rate with respect to the RJ algorithm (around 1.9% for split-combine and 4.8% for birth-death in RJ - around 5% for split-combine and 9.7% for birth-death in MTRJ-inv algorithm with $k = 5$), lowering the autocorrelation of the chain. This has obvious implications on the efficiency in estimating the posterior probability of each model.

Appendix

Theorem 5.1 *The GMTM algorithm satisfies detailed balance.*

It is sufficient to prove that

$$\pi(\boldsymbol{x}_t) P(\boldsymbol{x}_t, \boldsymbol{y}) = \pi(\boldsymbol{y}) P(\boldsymbol{y}, \boldsymbol{x}_t),$$

where $P(\boldsymbol{x}_t, \boldsymbol{y})$ is the transition probability of the Markov chain from state $\boldsymbol{x}_t$ to $\boldsymbol{y}$. Suppose that $\boldsymbol{x}_t \neq \boldsymbol{y}$, noting that the $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ are exchangeable; it holds that

$\pi\left(\boldsymbol{x}_t\right) P\left(\boldsymbol{x}_t, \boldsymbol{y}\right)$

$$
\begin{aligned}
&= k\pi\left(\boldsymbol{x}_t\right) T(\boldsymbol{x}_t,\boldsymbol{y}) p_{\boldsymbol{y}} \int\ldots\int T(\boldsymbol{x}_t,\boldsymbol{y}_1)\ldots T(\boldsymbol{x}_t,\boldsymbol{y}_{k-1}) \\
&\quad \min\left\{1, \frac{\pi(\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x}_t)p_{\boldsymbol{x}_t}}{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t,\boldsymbol{y})p_{\boldsymbol{y}}}\right\} T(\boldsymbol{y},\boldsymbol{x}_1^*)\ldots T(\boldsymbol{y},\boldsymbol{x}_{k-1}^*) \\
&\quad d\boldsymbol{y}_1\ldots d\boldsymbol{y}_{k-1}\, d\boldsymbol{x}_1^*\ldots d\boldsymbol{x}_{k-1}^* \\
&= k \int\ldots\int T(\boldsymbol{x}_t,\boldsymbol{y}_1)\ldots T(\boldsymbol{x}_t,\boldsymbol{y}_{k-1}) \\
&\quad \min\left\{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t,\boldsymbol{y})\, p_{\boldsymbol{y}}, \pi(\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x}_t)\, p_{\boldsymbol{x}_t}\right\} \\
&\quad T(\boldsymbol{y},\boldsymbol{x}_1^*)\ldots T(\boldsymbol{y},\boldsymbol{x}_{k-1}^*) d\boldsymbol{y}_1\ldots d\boldsymbol{y}_{k-1} d\boldsymbol{x}_1^*\ldots d\boldsymbol{x}_{k-1}^* \\
&= k\pi\left(\boldsymbol{y}\right) T(\boldsymbol{y},\boldsymbol{x}_t) p_{\boldsymbol{x}_t} \int\ldots\int T(\boldsymbol{x}_t,\boldsymbol{y}_1)\ldots T(\boldsymbol{x}_t,\boldsymbol{y}_{k-1}) \\
&\quad \min\left\{1, \frac{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t,\boldsymbol{y})p_{\boldsymbol{y}}}{\pi(\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x}_t)p_{\boldsymbol{x}_t}}\right\} T(\boldsymbol{y},\boldsymbol{x}_1^*)\ldots T(\boldsymbol{y},\boldsymbol{x}_{k-1}^*) \\
&\quad d\boldsymbol{y}_1\ldots d\boldsymbol{y}_{k-1}\, d\boldsymbol{x}_1^*\ldots d\boldsymbol{x}_{k-1}^* \\
&= \pi\left(\boldsymbol{y}\right)\ P\left(\boldsymbol{y},\boldsymbol{x}_t\right),
\end{aligned}
$$

as required.

It is possible to prove that the GMTRJ algorithm satisfies detailed balance in an entirely similar manner to the previous theorem.

Theorem 5.2 *The GMTRJ algorithm satisfies detailed balance.*

The GMTRJ algorithm involves transitions to states of variable dimension, and consequently the detailed balance condition is now written as

$$\pi(\boldsymbol{\theta}_i)P(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_j}) = \pi(\boldsymbol{\theta}_{a_j})P(\boldsymbol{\theta}_{a_j},\boldsymbol{\theta}_i)|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_j})|.$$

Suppose that $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_{a_j}$, noting that $\boldsymbol{\theta}_{a_1},\ldots,\boldsymbol{\theta}_{a_k}$ are exchangeable, it holds that

$$
\begin{aligned}
&\pi\left(\boldsymbol{\theta}_i\right) P\left(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k}\right) \\
&\quad = \quad k\,\pi\left(\boldsymbol{\theta}_i\right) T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k}) p_{\boldsymbol{\theta}_{a_k}} \\
&\qquad\quad \int\ldots\int T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_1})\ldots T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_{k-1}}) \\
&\qquad\quad \min\left\{1, \frac{\pi(\boldsymbol{\theta}_{a_k})T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i)\ p_{\boldsymbol{\theta}_i}}{\pi(\boldsymbol{\theta}_i)T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})\ p_{\boldsymbol{\theta}_{a_k}}}|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})|\right\} \\
&\qquad\quad T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_1})\ldots T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_{k-1}}) \\
&\qquad\quad d\boldsymbol{\theta}_{a_1}\ldots d\boldsymbol{\theta}_{a_{k-1}}\ d\boldsymbol{\theta}_{b_1}\ldots d\boldsymbol{\theta}_{b_{k-1}} \\
&\quad = \quad k \int\ldots\int T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_1})\ldots T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_{k-1}}) \\
&\qquad\quad \min\left\{\pi(\boldsymbol{\theta}_i)T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})p_{\boldsymbol{\theta}_{a_k}}, \pi(\boldsymbol{\theta}_{a_k}) \times \right. \\
&\qquad\quad \left. \times T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i)p_{\boldsymbol{\theta}_i}|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})|\right\} \\
&\qquad\quad T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_1})\ldots T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_{k-1}}) \\
&\qquad\quad d\boldsymbol{\theta}_{a_1}\ldots d\boldsymbol{\theta}_{a_{k-1}}\ d\boldsymbol{\theta}_{b_1}\ldots d\boldsymbol{\theta}_{b_{k-1}} \\
&\quad = \quad k\,\pi\left(\boldsymbol{\theta}_{a_k}\right) T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i) p_{\boldsymbol{\theta}_i}|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})| \\
&\qquad\quad \int\ldots\int T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_1})\ldots T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_{k-1}}) \\
&\qquad\quad \min\left\{1, \frac{\pi(\boldsymbol{\theta}_i)T(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})\ p_{\boldsymbol{\theta}_{a_k}}}{\pi(\boldsymbol{\theta}_{a_k})T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i)\ p_{\boldsymbol{\theta}_i}} \frac{1}{|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})|}\right\} \\
&\qquad\quad T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_1})\ldots T(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_{b_{k-1}}) \\
&\qquad\quad d\boldsymbol{\theta}_{a_1}\ldots d\boldsymbol{\theta}_{a_{k-1}}\ d\boldsymbol{\theta}_{b_1}\ldots d\boldsymbol{\theta}_{b_{k-1}} \\
&\quad = \quad \pi\left(\boldsymbol{\theta}_{a_k}\right)\ P\left(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i\right)|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})|,
\end{aligned}
$$

as required. Note that $|J(\boldsymbol{\theta}_{a_k},\boldsymbol{\theta}_i)| = 1/|J(\boldsymbol{\theta}_i,\boldsymbol{\theta}_{a_k})|$.

References

Bernardo, J. M. and Smith, A. F. (1994). *Bayesian Theory.* Wiley.

Brooks, S., Giudici, P., and Roberts, G. O. (2003). Efficient construction of reversible jump markov chain monte carlo proposal distributions. *Journal fo the Royal Statistical Society, Series B*, 65:3–55.

Cappe, O., Robert, C. P., and Ryden, T. (2003). Reversible jump, birth-and-death and more general continuous time markov chain monte carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 65(3):679–700.

Dellaportas, P., Forster, J., and Ntzoufras, I. (2002). On bayesian model and variable selection using mcmc. *Statistics and Computing*, 12:27–36.

Goodman, L. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61(2):215–231.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Journal of Chemical Physics*, 21:1087–1091.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing.* Springer, New-York.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The multiple-try method and local optimization in metropolis sampling. *Journal of American Statistical Association*, 95:121–134.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091.

Richardson, S. and Green, P. (1997). On bayesian analysis of mixture with an unknown number of components. *Journal fo the Royal Statistical Society, Series B*, 59:731–792.