Stat Comput
DOI 10.1007/s11222-011-9301-9

Interacting multiple try algorithms with different proposal distributions

Roberto Casarin · Radu Craiu · Fabrizio Leisen

Received: 3 November 2010 / Accepted: 14 November 2011

Abstract We introduce a new class of interacting Markov chain Monte Carlo (MCMC) algorithms which is designed to increase the efficiency of a modified multiple-try Metropolis (MTM) sampler. The extension with respect to the existing MCMC literature is twofold. First, the sampler proposed extends the basic MTM algorithm by allowing for different proposal distributions in the multiple-try generation step. Second, we exploit the different proposal distributions to naturally introduce an interacting MTM mechanism (IMTM) that expands the class of population Monte Carlo methods and builds connections with the rapidly expanding world of adaptive MCMC. We show the validity of the algorithm and discuss the choice of the selection weights and of the different proposals. The numerical studies show that the interaction mechanism allows the IMTM to efficiently explore the state space leading to higher efficiency than other competing algorithms.

Keywords Interacting Monte Carlo · Markov chain Monte Carlo · Multiple-try Metropolis · Population Monte Carlo · Simulated annealing

R. Casarin
Advanced School of Economics, Venice, Italy

R. Casarin
University Ca' Foscari of Venice, Venice, Italy

R. Craiu (✉)
University of Toronto, Toronto, Canada
e-mail: craiu@utstat.toronto.edu

F. Leisen
Universidad Carlos III de Madrid, Madrid, Spain

1 Introduction

Markov chain Monte Carlo (MCMC) algorithms are now essential for the analysis of complex statistical models. In the MCMC universe, one of the most widely used class of algorithms is represented by the Metropolis-Hastings (MH) (Metropolis et al. 1953; Hastings 1970) sampler and its variants. An important generalization of the standard MH formulation is given by the multiple-try Metropolis (MTM) (Liu et al. 2000). While in the MH formulation one accepts or rejects a single proposed move, the MTM is designed so that the next state of the chain is selected among multiple proposals. The multiple-proposal setup can be used effectively to explore the sample space of the target distribution and subsequent developments have taken advantage of this added flexibility. For instance, Craiu and Lemieux (2007) and Bédard et al. (2010) propose to use antithetic and quasi-Monte Carlo samples to generate the proposals and to improve the efficiency of the algorithm while Pandolfi et al. (2010a, 2010b) apply the multiple-proposal idea to a trans-dimensional setup and combine Reversible Jump MCMC with MTM.

This work further generalizes the MTM algorithm presented in Liu et al. (2000) in two directions. First, we show that the original MTM transition kernel can be modified to allow for different proposal distributions in the multiple-try generation step while preserving the ergodicity of the chain. The extension of the original MTM algorithm offers flexibility in constructing transition kernels for target distributions that have a complex geometry or may require different proposals across the sample space. Choosing the proposal distributions is an important challenge which is addressed here by adapting ideas used within the population Monte Carlo class of algorithms.

The population Monte Carlo procedures (Mengersen and Robert 2003; Cappé et al. 2004; Del Moral et al. 2006;

Published online: 07 December 2011

Del Moral 2004; Jasra et al. 2007; Campillo et al. 2009) have been designed to address the inefficiency of classical MCMC samplers in complex applications involving multi-modal and high dimensional target distributions (Pritchard et al. 2000; Heard et al. 2006). Its formulation relies on a number of MCMC processes that are run in parallel while learning from one another about the geography of the target distribution.

A second contribution of the paper is finding reliable generic methods for constructing the proposal distributions for the MTM algorithm.We propose an interacting MCMC sampling design for the MTM that preserves the Markovian property. More specifically, in the proposed interacting MTM (IMTM) algorithm, we allow the distinct proposal distributions to use the information produced by a population of auxiliary chains. We infer that the resulting performance of the MTM is tightly connected to the performance of the chains' population. In order to maximize the latter, we propose and compare via simulations a number of strategies that can be used to tune the auxiliary chains.

In the next section we discuss the IMTM algorithm, propose a number of alternative implementations and prove their ergodicity. In Sect. 3 we focus on some special cases of the IMTM algorithm and in Sect. 4 the performance of the methods proposed is demonstrated with simulations and real examples. We end the paper with a discussion of future directions for research.

2 Interacting Monte Carlo chains for MTM

We begin by describing the MTM and its extension for using different proposal distributions.

2.1 Multiple-try Metropolis with different proposal distributions

Suppose that of interest is sampling from a distribution π that has support in $\mathcal{Y} \subset \mathbf{R}^d$ and is known up to a normalizing constant. Assuming that the current state of the chain is x, the updating rule for the MTM algorithm of Liu et al. (2000) is described in Algorithm 1.

Note that while the MTM uses the same distribution to generate all the proposals, it is possible to extend this formulation to different proposal distributions without altering the ergodicity of the associated Markov chain.

Let $T_j(\cdot|x)$, with $j = 1, \dots, M$, be a set of proposal distributions for which $T_j(y|x) > 0$ if and only if $T_j(x|y) > 0$. Define

$$w_j(x, y) = \pi(x)T_j(y|x)\lambda_j(x, y), \quad j = 1, \dots, M$$

where $\lambda_j(x, y)$ is a nonnegative symmetric function in x and y that can be chosen by the user. The only requirement is that $\lambda_j(x, y) > 0$ whenever $T(x, y) > 0$. Then the MTM algorithm with different proposal distributions is given in Algorithm 2.

Algorithm 1 Multiple-try Metropolis algorithm (MTM)

```
1. Draw $M$ trial proposals $y_1, \dots, y_M$ from the proposal distribution $T(\cdot|x)$. Compute $w(y_j, x)$ for each $j \in \{1, \dots, M\}$, where $w(y, x) = \pi(y)T(x|y)\lambda(y, x)$, and $\lambda(y, x)$ is a symmetric function of $x, y$.
2. Select $y$ among the $M$ proposals with probability proportional to $w(y_j, x)$, $j = 1, \dots, M$.
3. Draw $x_1^*, \dots, x_{M-1}^*$ variates from the distribution $T(\cdot|y)$ and let $x_M^* = x$.
4. Accept $y$ with generalized acceptance probability
   $$\rho = \min\left\{1, \frac{w(y_1, x) + \cdots + w(y_M, x)}{w(x_1^*, y) + \cdots + w(x_M^*, y)}\right\}.$$
```

Algorithm 2 MTM with different proposal distributions

```
1. Draw independently $M$ proposals $y_1, \dots, y_M$ such that $y_j \sim T_j(\cdot|x)$. Compute $w_j(y_j, x)$ for $j = 1, \dots, M$.
2. Select $Y = y$ among the trial set $\{y_1, \dots, y_M\}$ with probability proportional to $w_j(y_j, x)$, $j = 1, \dots, M$. Let $J$ be the index of the selected proposal. Then draw $x_j^* \sim T_j(\cdot|y)$, $j \neq J$, $j = 1, \dots, M$ and let $x_J^* = x$.
3. Accept y with probability
   $$\rho = \min\left\{1, \frac{w_1(y_1, x) + \cdots + w_M(y_M, x)}{w_1(x_1^*, y) + \cdots + w_M(x_M^*, y)}\right\}$$
   and reject with probability $1 - \rho$.
```

It should be noted that Algorithm 2 is a special case of the interacting MTM presented in the next section and that the proof of ergodicity for the associated chain follows closely the proof given in Appendix for the interacting MTM and, therefore, it is not given here.

2.2 General construction

Undoubtedly, Algorithm 2 offers additional flexibility in organizing the MTM sampler. This section introduces generic methods for using a population of MCMC chains to define the proposal distributions.

Consider a population of N chains, $X^{(i)} = \{X_n^{(i)}\}_{n \in \mathbb{N}}$ and $i = 1, \dots, N$. For full generality we assume that the ith chain has MTM transition kernel with M_i different proposals $\{T_j^{(i)}\}_{1 \le j \le M_i}$ (if we set $M_i = 1$ we imply that the chain has a MH transition kernel). The interacting mechanism allows each proposal distribution to possibly depend on the values of the chains at the previous step. Formally, if $\Xi_n = \{x_n^{(i)}\}_{i=1}^N$ is the vector of values taken at iteration

Algorithm 3 Interacting multiple try algorithm (IMTM)

```
• For $i = 1, \dots, N$
    1. Let $x = x_n^{(i)}$; for $j = 1, \dots, M_i$ draw $y_j \sim T_j^{(i)}(\cdot|\tilde{f}_n^{(i)}(x))$ independently and compute
       $w_j^{(i)}(y_j, x) = \pi(y_j) T_j^{(i)}(x|\tilde{f}_n^{(i)}(y_j)) \lambda_j^{(i)}(y_j, x)$.
    2. Select $J \in \{1, \dots, M_i\}$ with probability proportional to $w_j^{(i)}(y_j, x)$, $j = 1, \dots, M_i$ and set $y = y_J$.
    3. For $j = 1, \dots, M_i$ and $j \neq J$ draw $x_j^* \sim T_j^{(i)}(\cdot|\tilde{f}_n^{(i)}(y))$, let $x_J^* = x_n^{(i)}$ and compute
       $w_j^{(i)}(x_j^*, y) = \pi(x_j^*) T_j^{(i)}(y|\tilde{f}_n^{(i)}(x_j^*)) \lambda_j^{(i)}(x_j^*, y)$.
    4. Set $x_{n+1}^{(i)} = y$ with probability
       $\rho_i = \min\left\{1, \frac{w_1^{(i)}(y_1, x) + \dots + w_{M_i}^{(i)}(y_{M_i}, x)}{w_1^{(i)}(x_1^*, y) + \dots + w_{M_i}^{(i)}(x_{M_i}^*, y)}\right\}$,
       and $x_{n+1}^{(i)} = x_n^{(i)}$ with probability $1 - \rho_i$.
```

$n \in \mathbb{N}$ by the population of chains, then we allow each proposal distribution used in updating the population at iteration $n+1$ to depend on Ξ_n. The mathematical formalization is used in the description of Algorithm 3. One expects that the chains in the population are spread throughout and, thus, offer a good representation of the sample space $\mathcal{Y}$.

The first step in Algorithm 3 suggests that each proposal distribution used in each parallel MTM chain is allowed to depend on the current states of all the chains in the population. However, this general formulation for the IMTM, though correct in theory, can be difficult to tune efficiently in a given practical problem. Before we move to discuss implementations that simplify and enhance the practical application of the IMTM algorithm, we prove below that the chain underlying Algorithm 3 is ergodic to π.

In order to give a representation of the IMTM transition kernel let us introduce the following notation. Denote the map $\tilde{f}_n^{(i)}(z) = (x_n^{(1:i-1)}, z, x_n^{(i+1:N)})^T$ and let $T^{(i)}(y_{1:M_i}|x) = \prod_{k=1}^{M_i} T_k^{(i)}(y_k|\tilde{f}_n^{(i)}(x))$ and $T_{-j}^{(i)}(y_{1:M_i}|x) = \prod_{k \neq j}^{M_i} T_k^{(i)}(y_k|\tilde{f}_n^{(i)}(x))$, where $dy_{1:M_i} = \prod_{k=1}^{M_i} dy_k$ and $dy_{-j} = \prod_{k \neq j}^{M_i} dy_k$.

The transition kernel associated to the population of chains is then

$$K(\Xi_n, \Xi_{n+1}) = \prod_{i=1}^{N} K_i(x_n^{(i)}, x_{n+1}^{(i)}), \tag{1}$$

where

$$K_i(x, y) = \sum_{j=1}^{M_i} A_j^{(i)}(x, y) T_j^{(i)}(y|x) + \left(1 - \sum_{j=1}^{M_i} B_j^{(i)}(x)\right) \delta_x(y) \tag{2}$$

is the transition kernel used to run the ith chain of the population and

$$A_j^{(i)}(x, y) = \int_{\mathcal{Y}^{2(M_i-1)}} \tilde{w}_j^{(i)}(y, x) \rho_j^{(i)}(x, y) T_{-j}^{(i)}(x_{1:M_i}^*|y) \times T_{-j}^{(i)}(y_{1:M_i}|x) dx_{-j}^* dy_{-j},$$

$$B_j^{(i)}(x) = \int_{\mathcal{Y}^{2(M_i-1)+1}} \rho_j^{(i)}(x, y) T_{-j}^{(i)}(x_{1:M_i}^*|y) \times T^{(i)}(y_{1:M_i}|x) dx_{-j}^* dy_{1:M_i}.$$

In the above equations $\tilde{w}_j^{(i)}(y_j, x) = w_j^{(i)}(y_j, x) / (w_j^{(i)}(y, x) + \bar{w}_{-k}^{(i)}(y_{1:M_i}|x))$, with $j = 1, \dots, M_i$ and $\bar{w}_{-j}^{(i)}(y_{1:M_i}|x) = \sum_{k \neq j}^{M_i} w_k^{(i)}(y_k, x)$, are the normalized weights used in the selection step of the IMTM algorithm and

$$\rho_j^{(i)}(x, y) = \min\left\{1, \frac{w_j^{(i)}(y, x) + \bar{w}_{-j}^{(i)}(y_{1:M_i}|x)}{w_j^{(i)}(x, y) + \bar{w}_{-j}^{(i)}(x_{1:M_i}^*|y)}\right\}$$

is the generalized MH ratio associated to a MTM algorithm.

The validity of the IMTM algorithm relies upon the detailed balance condition.

Theorem 1 *The transition density $K_i(x_n^{(i)}, x_{n+1}^{(i)})$ associated to the ith chain of the IMTM algorithm satisfies the conditional detailed balanced condition.*

Proof See Appendix. □

Since each transition $K_i(x_n^{(i)}, x_{n+1}^{(i)})$, $i = 1, \dots, N$, has $\pi(x)$ as stationary distribution and satisfies the conditional detailed balance condition, the joint transition $K(\Xi_n, \Xi_{n+1}) = \prod_{i=1}^{N} K_i(x_n^{(i)}, x_{n+1}^{(i)})$ has $\pi(x)^N$ as a stationary distribution.

3 Practical implementation

Note that at each IMTM iteration the computational complexity is $\mathcal{O}(\sum_{i=1}^{N} M_i)$. This can become burdensome when the number of chains, N, and the number of proposals, M_i, are simultaneously large so one needs to decide on a strategy for choosing the number of chains and proposals. We distinguish two possible tactics in designing the interaction mechanism. The first one uses a small number of chains, say $5 \le N \le 20$, and a number of proposals equal to the number of chains, i.e. $M_i = N$, for all $1 \le i \le N$. In this way all the chains can interact at each iteration of the algorithm and many search directions can be included among the proposals.

A second strategy is to use a higher number of chains, e.g. $N = 100$, in order to possibly have, at each iteration, a good approximation of the target or a much higher number of search directions for a good exploration of the sample space. This design is common in Population Monte Carlo or Interacting MCMC methods. Clearly, when a high number of chains is used within IMTM, it is necessary to set $M_i < N$, possibly $M_i = 1$ for each auxiliary chain.

Generally, while we would like to see the number of chains, N, increase with the target's dimension, it is reasonable to assume that the choose of the number of chains that are run in parallel depends on the available computational power (e.g., number of CPU's, server memory, etc.). The number of proposals, M_i, should not be too small compared to N. Based on our experiments, we recommend using $M_i/N \in [5\%, 20\%]$.

In this section we discuss a few strategies to built the M_i proposals for each chain and in the simulation section we compare the two strategies outlined above.

3.1 Parsing the population of auxiliary chains

When N is large, we may not want to use all the chains at each iteration of the IMTM. One approach that turned out to be successful in our applications produces the proposals using a random subset of the chains' population. For ease of description, assume that $M_i = M < N$, for all chains, $1 \le i \le N$. Then, when updating the i-th chain of the population, we sample the random indices $I_1, \ldots, I_{M-1}$ from the uniform distribution $\mathcal{U}\{1, \ldots, N\}$ and we let $I_M = i$. Then the M proposals used for chain i will be allowed to depend only on the current states of those chains with indices $I_1, \ldots, I_M$. Using the notation introduced and letting $I_n^{(i)} = (I_1, \ldots, I_M)$ then the M proposals used for chain i at time n are sampled using $T_j^{(i)}(y|x_n^{(I_1)}, \ldots, x_n^{(I_M)})$, for all $j = 1, \ldots, M$. Our simulation experiments showed a good performance when we used a relatively simpler version in which the jth proposal depends only on the current state of chain I_j, i.e., it is sampled using $T_j^{(i)}(\cdot|x_n^{(I_j)})$, for all $j = 1, \ldots, M$. One can see that the interweaving of the chains is performed by allowing the proposals used in chain i to be sampled conditional not only on the current state of the chain, $x_n^{(i)}$, but also on the current states of those chains whose indices are sampled at random and stored in $I_n^{(i)}$.

Another important issue directly connected to the practical implementation of the IMTM is the choice of $\lambda_j^{(i)}(x, y)$. Previously suggested forms for the function $\lambda_j^{(i)}(x, y)$ (Liu et al. 2000) are:

(a) $\lambda_j^{(i)}(x, y) = 2\{T_j^{(i)}(x|y) + T_j^{(i)}(y|x)\}^{-1}$,

(b) $\lambda_j^{(i)}(x, y) = \{T_j^{(i)}(x|y)T_j^{(i)}(y|x)\}^{-\alpha}$, $\alpha > 0$.

Little guidance is offered in the existent literature regarding the choice of λ and, to our knowledge, in most applications of the original MTM algorithm the default choice is $\lambda = 1$.

Here we propose to include in the construction of λ the information provided by the population of chains. Therefore, we suggest to modify the above functions to

(a′) $\lambda_j^{(i)}(x, y) = 2\nu_j \{T_j^{(i)}(x|y) + T_j^{(i)}(y|x)\}^{-1}$,

(b′) $\lambda_j^{(i)}(x, y) = \nu_j \{T_j^{(i)}(x|y)T_j^{(i)}(y|x)\}^{-\alpha}$, $\alpha > 0$,

where the factor ν_j is

$$\nu_j = \frac{1}{N}\left[1 + \sum_{i=1}^{N} \mathbf{1}_{\{j\}}(J_{n-1}^{(i)})\right], \qquad j = 1, \ldots, M, \tag{3}$$

and $J_{n-1}^{(i)}$ is the index of the proposal selected in the ith chain update at iteration $n-1$. It can be seen that the $\{\nu_j\}_{1\le j\le M}$ capture the behaviour of the auxiliary chains at the previous iteration. More precisely, ν_j will be relatively larger for those proposal distributions $T_j(\cdot|\cdot)$ whose samples have been selected as the potential next states for the chains in the population at iteration $n-1$. The modifications proposed for $\lambda(\cdot, \cdot)$ would increase the use of those proposal distributions favoured by the population of chains at previous iteration. Since ν_j depends only on samples generated at the previous step by the population of chains, the ergodicity of the IMTM chain is preserved. In the simulation section we compare the performance of IMTM coupled with either (a′) or (b′) when $\alpha = 1$.

3.2 Annealed IMTM

Our belief in IMTM's improved performance is underpinned by the assumption that the population of Monte Carlo chains is spread throughout the sample space. This can be partly achieved by initializing the chains using draws from a distribution overdispersed with respect to π (see also Jennison 1993; Gelman and Rubin 1992; Craiu and Meng 2005) and partly by modifying the stationary distribution for some of the chains in the population. Specifically, we consider the sequence of annealed distributions $\pi_t = \pi^t$ with $t \in \{\xi_1, \xi_2, \ldots, \xi_N\}$, where $1 = \xi_1 > \xi_2 > \cdots > \xi_n$, for instance $\xi_t = 1/t$. When t, s are close temperatures, π_t is similar to π_s, but $\pi = \pi_1$ may be much harder to sample from than π_{ξ_N}, as has been long recognized in the simulated annealing and simulated tempering literature (see Marinari and Parisi 1992; Geyer and Thompson 1994;

Neal 1994). Therefore, it is likely that some of the chains designed to sample from $\pi_1, \dots, \pi_N$ have good mixing properties, making them suitable candidates for the population of MCMC samplers needed to run the IMTM. Recent theoretical work by Atchadé et al. (2011) has build, in an adaptive setup, connections between the temperature ladder and the optimal scaling problem. While an extension of their study to IMTM is beyond the scope of this paper, in the simulation section we compare three different methods for constructing the temperature ladder $1 = \xi_1 > \xi_2 > \dots > \xi_n$.

We consider the Monte Carlo population made of the $N-1$ chains having $\{\pi_2, \dots, \pi_N\}$ as stationary distributions. However, the use of MTM for *each* auxiliary chain may be redundant since for smaller ξ_i's the distribution π_i is easy to sample from. For this reason, in our simulations we shall use the AIMTM in which the chain that is ergodic to π has an IMTM transition kernel and each auxiliary chain is a MH chain (i.e., with $M = 1$) with target π_i, $2 \le i \le N$. The AIMTM is described in Algorithm 4. In practice, we always use the current state of the chain ergodic to π ($\xi = 1$) among the states used for generating one of the proposals (e.g., in Algorithm 4 we automatically set $I_1 = 1$ and let $I_2, \dots, I_M$ be sampled at random). The recommended value of M for the chain of interest is such that $M/N \in [5\%, 20\%]$.

An additional gain could be obtained if the auxiliary chains' transition kernels are modified using adaptive MCMC strategies (see also Chauveau and Vandekerkhove 2002, for another example of adaption for interacting chains). However, letting the auxiliary chains adapt indefinitely results in complex theoretical justifications for the IMTM which go beyond the scope of this paper and will be presented elsewhere. Our recommendation is to use finite adaptation for the auxiliary chains prior to the start of the IMTM. One could take advantage of multi-processor computing units and use parallel programming to increase the computational efficiency of this approach.

The adaptation of $\lambda_j^{(i)}$, through the weights ν_j defined in (3), should be used cautiously in this case. The aim of the annealing procedure is to allow the higher temperatures chains to explore widely the sample space and to improve the mixing of the MTM chain. Using ν_j in the context of annealed IMTM could arbitrarily penalize some of the higher temperature proposals and reduce the effectiveness of the annealing strategy. For this reason we do not consider using adaptive λ's for AIMTM.

Note that although the AIMTM requires additional computation effort, one can take advantage of the samples produced by *all* the auxiliary chains in the population to obtain a Monte Carlo approximation of a quantity of interest. For example, suppose we are interested in computing

Algorithm 4 Annealed IMTM algorithm (AIMTM)

```
- For $i = 1$
    1. Let $x = x_n^{(i)}$ and sample $I_1, \dots, I_M$ from $\mathcal{U}\{1, \dots, N\}$.
    2. For $j = 1, \dots, M$ draw $y_j \sim T_j^{(i)}(\cdot|x_n^{(I_j)})$ independently and
        (a) If $I_j \neq 1$ set
            $$w_j^{(i)}(y_j, x) = \pi(y_j) T_j^{(i)}(x|x_n^{(I_j)}) \lambda_j^{(i)}(y_j, x).$$
        (b) If $I_j = 1$ set
            $$w_j^{(i)}(y_j, x) = \pi(y_j) T_j^{(i)}(x|y_j) \lambda_j^{(i)}(y_j, x).$$
    3. Select $J \in \{1, \dots, M\}$ with probability proportional to $w_j^{(i)}(y_j, x)$, $j = 1, \dots, M$ and set $y = y_J$.
    4. Let $x_J^* = x_n^{(i)}$ and for $j = 1, \dots, M$, $j \neq J$,
        (a) If $I_j \neq 1$ draw $x_j^* \sim T_j^{(i)}(\cdot|x_n^{(I_j)})$,
        (b) If $I_j = 1$ draw $x_j^* \sim T_j^{(i)}(\cdot|y)$.
    5. Compute $w_j^{(i)}(x_j^*, y)$ using the same rule as in 2.
    6. Set $x_{n+1}^{(i)} = y$ with probability $\rho_i$, where $\rho_i$ is the generalized MH ratio of the IMT algorithm and $x_{n+1}^{(i)} = x_n^{(i)}$ with probability $1 - \rho_i$.
- For $i = 2, \dots, N$ we perform the usual MH update using proposal distribution $T^{(i)}$ for chain $i$.
    1. Let $x = x_n^{(i)}$ and update the proposal function $T^{(i)}(\cdot|x)$.
    2. Draw $y \sim T^{(i)}(\cdot|x)$ and compute
        $$\rho_i = \min\left\{1, \frac{\pi(y)^{\xi_i} T^{(i)}(x|y)}{\pi(x)^{\xi_i} T^{(i)}(y|x)}\right\}.$$
    3. Set $x_{n+1}^{(i)} = y$ with probability $\rho_i$ and $x_{n+1}^{(i)} = x_n^{(i)}$ with probability $1 - \rho_i$.
```

$$\mathcal{I} = \int_{\mathcal{Y}} h(x)\pi(x)dx,$$

where h is a test function. It is possible to approximate $\mathcal{I}$ using

$$\mathcal{I}_{NT} = \frac{1}{T}\sum_{n=1}^{T} \frac{1}{\bar{\zeta}} \sum_{j=1}^{N} h(x_n^{(j)})\zeta_j(x_n^{(j)}),$$

where $x_n^{(i)}$ is the output of the i-th chain ergodic to target π^{ξ_i} at time n, for all $n = 1, \dots, T$ and all $i = 1, \dots, N$, $\zeta_j(x) = \pi(x)/\pi^{\xi_j}(x)$ are the importance weights and $\bar{\zeta} = \sum_{j=1}^{N} \zeta_j(x_n^{(j)})$.

4 Simulation results

4.1 Beta mixture model

Mixture models have been used to capture heterogeneity in the data in many applications. The Bayesian inference for such models presents computational challenges. Specifically, the Bayesian analysis of k-component mixture model leads to a posterior distribution that is invariant with respect to permutation of the parameter labels and exhibits $k!$ modes. Sampling from the posterior is therefore a challenging problem which rarely can be solved successfully by the conventional single-chain MCMC methods. More efficient sampling algorithms are thus needed. As emphasized by Jasra et al. (2007) in the context of Bayesian mixture models, population Monte Carlo methods allow to sample efficiently from the posterior distribution.

We consider here a Bayesian mixture of normals that was previously used by Jasra et al. (2005, 2007) for comparing the performance of different population Monte Carlo methods. Let $y_1, \ldots, y_n$ be n i.i.d. samples with density

$$\sum_{h=1}^{K} \tau_h f(y|\mu_h, \eta_h^{-1}), \tag{4}$$

where K is the number of mixture components and $f(y_i|\mu_h, \eta_h^{-1})$ is the density of a normal distribution with location parameter μ_h and precision parameter η_h. The weights $\tau_h \geq 0$, $h = 1, \ldots, K$ of the mixture are such that $\sum_{h=1}^{K} \tau_h = 1$. We assume the following priors (see also Jasra et al. 2005; Richardson and Green 1997).

$$\begin{aligned} &\mu_j \sim \mathcal{N}(\xi, \kappa^{-1}),\\ &\eta_j \sim \mathcal{G}a(\alpha, \beta),\\ &\tau_{1:k-1} \sim \mathcal{D}ir(\delta), \end{aligned} \tag{5}$$

where $\mathcal{N}(\xi, \kappa^{-1})$, $\mathcal{G}a(\alpha, \beta)$ and $\mathcal{D}ir(\delta)$ are, respectively, the normal distribution with location ξ and precision κ, the gamma distribution with shape parameter α and scale parameter β and the symmetric Dirichlet distribution, with parameter δ.

We will use the problem of sampling from the posterior distribution defined by the model above as a benchmark for comparing the IMTM methods proposed in this paper with other population Monte Carlo algorithms based on MH kernels. We assume we have available a dataset of 100 (simulated) samples from an equally weighted, (i.e. $\tau_j = 1/K$ for $j = 1, \ldots, K$) normal mixture with $K = 4$ components with true means, $(\mu_1, \mu_2, \mu_3, \mu_4)^T = (-3, 0, 3, 6)^T$, and equal standard deviations $\eta_j^{-1/2} = 0.55$, $1 \leq j \leq 4$.

The algorithms being compared below are the following:

MH A population of Monte Carlo algorithms in which all the N parallel chains have random walk MH (RWMH) kernels in which the jth Gaussian proposal distribution has covariance $\sigma_j^2\mathbf{I}$ where $\sigma_j = 0.01 + 0.59 * j/N$ for all $1 \leq j \leq N$ such that the acceptance rates obtained for the population of chains are between 10–60%.

MH1 A population of Monte Carlo algorithms in which each of the N parallel chains run a RWMH algorithm whose proposal distribution is a mixture of 4 normal densities. The standard deviations of the proposals are divided equally between 0.01 and 0.3.

MH2 A population of Monte Carlo algorithms in which each of the N transition kernels is a mixture of four RWMH kernels with same standard deviations as those defined for MH2.

The above algorithms do not allow interaction between the parallel chains which is arguably less flexible than the IMTM setup. Therefore we include in our comparison the above three algorithms to which we apply the cross-over interaction introduced by Liang and Wong (2001). The different chains of the population have the same target thus the acceptance-probability of the cross-over move is one.

MH.c.o The MH algorithm described above with cross-over moves.

MH1.c.o The MH1 algorithm described above with cross-over moves.

MH2.c.o The MH2 algorithm described above with cross-over moves.

The six algorithms described above are compared with the following IMTM samplers:

IMTM-TA An IMTM algorithm with N chains defined as in Sect. 3.1 and using $\lambda_j^{(i)}(x, y) = 2\{T_j^{(i)}(x|y) + T_j^{(i)}(y|x)\}^{-1}$ weights. The jth proposal uses $T_j^{(i)}(y|x) = N(x, \sigma_j^2\mathbf{I})$ where $\sigma_j = 0.01 + 0.59 * j/M$ for all $1 \leq j \leq M, 1 \leq i \leq N$.

IMTM-TA-a The same algorithm as IMTM-TA but with adaptive weights $\lambda_j^{(i)}(x, y) = 2\nu_j\{T_j^{(i)}(x|y) + T_j^{(i)}(y|x)\}^{-1}$ where ν_j is defined as in (3).

IMTM-IS An IMTM algorithm identical to IMTM-TA but using $\lambda_j^{(i)}(x, y) = \{T_j^{(i)}(x|y)T_j^{(i)}(y|x)\}^{-1}$ weights.

IMTM-IS-a The same algorithm as IMTM-IS but with adaptive weights $\lambda_j^{(i)}(x, y) = \nu_j\{T_j^{(i)}(x|y)T_j^{(i)}(y|x)\}^{-1}$ where ν_j is defined as in (3).

The comparison is made with respect to the estimation of the marginal means $\mu_1, \ldots, \mu_4$. We consider $T = 100{,}000$ samples obtained with $N = 100$ parallel chains for the MH, MH1, MH2, MH.c.o, MH1.c.o and MH2.c.o algorithms. For all the IMTM algorithms we sampled $T = 10{,}000$ draws from running $N = 100$ chains each with $M = 10$ proposals.

We observed from all the simulation experiments that IMTM-TA and IMTM-IS have similar performances, so we

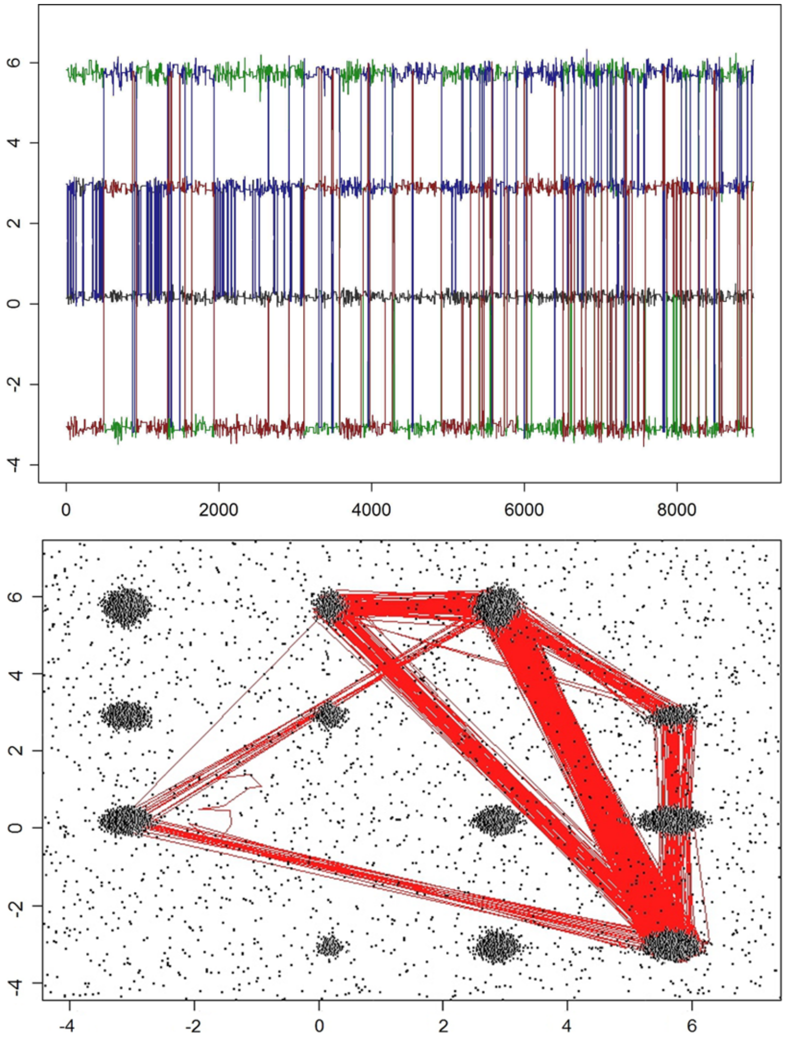

Fig. 1 *Top panel*: Trace plots generated using 9,000 samples obtained for $\mu_1, \ldots, \mu_4$ from one of the IMTM-TA chains. *Bottom panel*: The dots represent the projection of the values sampled by the IMTM-TA population of chains on the (μ_i, μ_j) planes, with $i \neq j$. The trajectory of one of the chains is projected on the plane (μ_1, μ_2)

present the graphical results only for IMTM-TA. A typical output of the IMTM-TA algorithm is given in the top panel of Fig. 1 which shows, for one of the chains in the population, the traces for each of the four coordinates sampled, μ_1, μ_2, μ_3 and μ_4. We notice that the chain is able to switch frequently between the different modes of the posterior distribution and this compares favourably with the MH, MH1 and MH2 algorithms, with and without cross-over. The MH chains rarely switch between modes as can be seen also in Fig. 1 of Jasra et al. (2007).

In order to give an alternative representation of the raw output of the population of chains we follow Früwirth-Schnatter (2006) and present in Fig. 2 the samples produced by each algorithm. The bottom panel in Fig. 2 has been produced by projecting the samples on all the planes (μ_i, μ_j) with $i \neq j$ (in total we have $K(K-1) = 12$ such planes) and then superimposing all the plots into a single one. As discussed in Früwirth-Schnatter (2006) the number of simulation clusters in this graphical representation, for a K-components mixture, is $K(K-1) = 12$, that is equal to 12 in our example. In the same panel sthe solid line shows one of the chains' trajectory.

In Fig. 2 we show samples produced by the other algorithms considered in the comparison. The six populations of chains, MH, MH1, MH2, with and without cross-over, are able to visit different modes of the posterior. Note that the samples from the population of MH chains are usually not evenly distributed across the different posterior modes. Moreover the single chains of the population of the MH algorithms usually visit only one of the clusters and are not able to visit the other clusters. In each panel the line represents the path followed by one of the chains. One can easily notice the difficulty of the MH, MH1 or MH2 chains without cross-over to explore the posterior surface. The lines shown in the right-side panels crystallize the effect of the cross-over moves on the mixing property of the population of interacting chains. Each chain is now able to visit many modes and this results in improved efficiency for the class of MH algorithms considered here. However, one can notice the superiority of the IMTM-TA algorithm from the paths shown

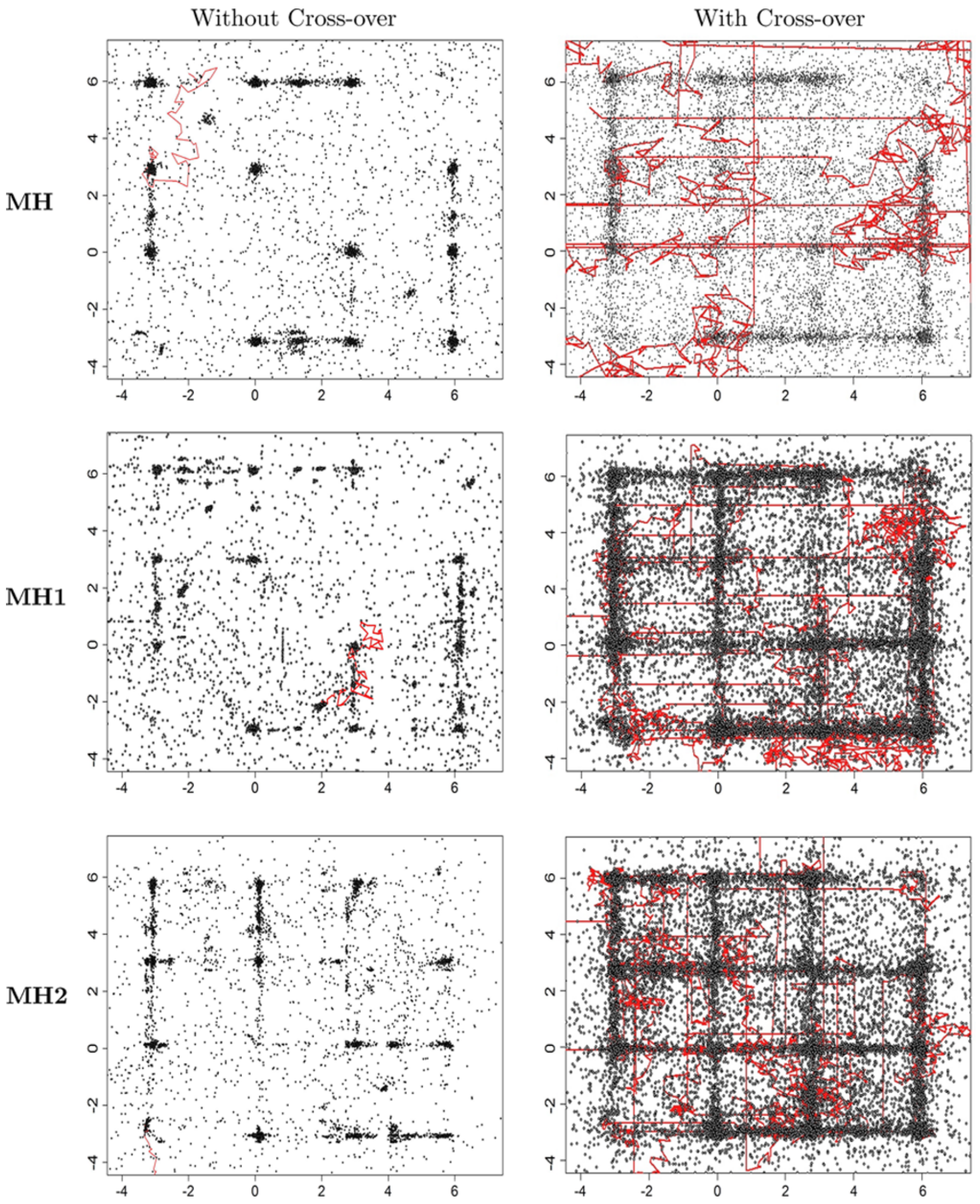

Fig. 2 *The dots* represent the projection of the values sampled by the population of MH chains considered in the simulation on the (μ_i, μ_j) planes, with $i \neq j$. *The plots* illustrate the samples obtained without cross-over (*left column*) and with cross-over (*right column*) for the MH (*top row*), MH1 (*middle row*) and MH2 (*bottom row*). *The lines* show the trajectory of one of the chains

in the bottom panel of Fig. 1 where it is clear that the chain visits many modes of the posterior distribution.

The efficiency improvement is also obvious from the autocorrelation functions (ACF) shown in Fig. 3. For each method included in the comparison, the curves shown are obtained by averaging the ACF estimates over the N chains of the population and over 10 replicates. The MH with cross-over are more efficient then the parallel MH algorithms but still less efficient than the IMTM algorithms.

The results in Table 1 show that the IMTM algorithms are generally able to produce more efficient estimates than the MH class of algorithms considered in the comparison. The cross-over moves bring the efficiency of the MH, MH1 and MH2 closer to that of the IMTM samplers, especially when the number of parallel chains is large ($N = 100$). However

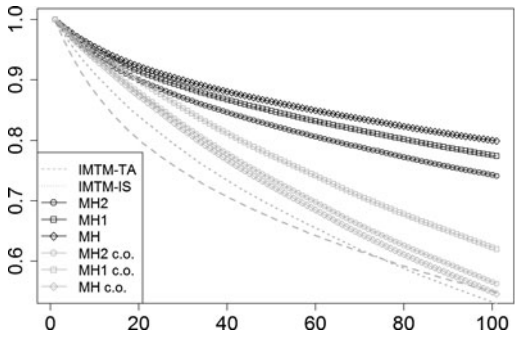

Fig. 3 Autocorrelation functions for the methods considered. *The curves* are obtained by averaging over the population of chains used for each algorithm and over the 10 replicated runs of each algorithm

Table 1 Estimates of $\mu_1, \dots, \mu_4$ and the corresponding standard errors (between brackets). For the parallel MHs, without and with crossover (c.o.), we considered alternatively $N = 20$ and $N = 100$ chains for a total of $T = 100{,}000$ samples. For the IMTMs, without and with adaptation (IMTM-TA-a and IMTM-IS-a), we consider $T = 10{,}000$ draws obtained from $N = 100$ chains and $M = 10$ different proposals. Reported values are obtained by averaging over 10 replicated runs for each algorithms. The Mean Square Error (MSE), averaged over the parameters, is reported for each algorithm

	$N = 100$					$N = 20$				
	1	2	3	4	MSE	1	2	3	4	MSE
MH	0.81	0.42	2.08	1.06	18.83	0.39	0.69	0.67	2.28	26.76
	(4.22)	(4.37)	(4.39)	(4.10)		(5.35)	(5.16)	(6.02)	(3.15)	
MH1	0.72	0.21	0.62	0.91	5.42	0.10	0.17	0.66	0.78	7.35
	(2.12)	(2.09)	(2.14)	(2.19)		(2.47)	(1.89)	(2.49)	(2.91)	
MH2	0.99	1.89	1.47	1.01	3.30	0.11	2.80	0.42	0.37	5.09
	(1.57)	(1.73)	(1.87)	(1.89)		(1.99)	(1.71)	(1.98)	(1.85)	
MH c.o.	1.87	1.09	1.91	1.66	7.89	1.74	1.11	1.01	1.75	11.02
	(2.52)	(2.79)	(2.88)	(2.92)		(3.14)	(3.12)	(3.58)	(3.33)	
MH1 c.o.	0.65	0.21	1.59	1.46	2.77	0.51	0.22	1.83	1.12	3.51
	(1.86)	(1.35)	(1.24)	(1.35)		(1.48)	(1.91)	(1.27)	(1.91)	
MH2 c.o.	1.11	1.69	1.27	1.26	2.17	0.59	1.68	0.97	1.14	2.26
	(1.33)	(1.34)	(1.76)	(1.29)		(1.43)	(1.16)	(1.36)	(1.58)	
IMTM-IS	1.40	1.52	1.37	1.42	1.05	1.36	1.39	1.61	1.69	1.42
	(1.01)	(0.98)	(1.22)	(0.87)		(0.98)	(1.20)	(1.12)	(1.42)	
IMTM-IS-a	1.37	1.44	1.58	1.54	0.49	1.31	1.71	1.35	1.72	1.18
	(0.83)	(0.56)	(0.71)	(0.64)		(0.81)	(0.97)	(1.23)	(1.24)	
IMTM-TA	1.31	1.46	1.53	1.61	0.52	1.29	1.21	1.70	1.32	0.89
	(0.38)	(1.06)	(0.48)	(0.73)		(1.34)	(1.05)	(0.31)	(0.59)	
IMTM-TA-a	1.56	1.39	1.60	1.37	0.47	1.63	1.75	1.61	1.44	0.85
	(0.48)	(0.91)	(0.76)	(0.42)		(0.76)	(0.86)	(1.02)	(0.97)	

when we reduce the number of chains (e.g. $N = 20$) the performance of the MH algorithms (with and without crossover) is clearly inferior to that of the IMTM algorithms. Interestingly, the IMTM-TA and IMTM-IS perform similarly whether we choose to adapt the weights λ_j or not.

4.1.1 Comparison in the presence of annealing

The performance of the MH, MH1 and MH2 populations can be improved by combining them with an annealing procedure. Our interest, here, lies in comparing AIMTM with the algorithms MH, MH1, MH2 which are modified to incorporate an annealing-based strategy. We consider once again two variants of the AIMTM defined by the choice of weights λ_j. Specifically, we consider AIMTM-TA and AIMTM-IS which use Algorithm 4 with, respectively, the same λ's as IMTM-TA and IMTM-IS.

We also consider the uniform, logarithmic and power tempering schemes that were also suggested by Jasra et al. (2007):

$$\xi_i = \xi_{i-1} - \frac{1}{N},$$

$$\xi_i = \log(\xi_{i-1} + 1)/\log(Q), \quad Q > 0,$$

$$\xi_i = (\xi_{i-1} - Q)^{\psi}, \quad \psi > 0,\ Q \in (0, 1),$$

where $\xi_1 = 1$ and $i = 2, \dots, N$. The three tempering schemes are denoted, respectively, M1, M2 and M3. For the logarithmic scheme M2 we consider $Q = 2.25$ and for the power scheme M3 we set $Q = 0.001$ and $\psi = 3/2$ as suggested in Jasra et al. (2007).

For the MH algorithms we build chain i ergodic to π^{ξ_i} and construct different scales for the chains of the population as in Jasra et al. (2007). For the ith chain the proposal variance $\sigma_i = \sigma_1/(1 + \gamma_i)$ with $\sigma_1 = 0.5$.

We report the estimates for each mean μ_i in Table 2. The AE column shows the maximum bias (over the four means). One can see easily that, on average, the AIMTM yields the smallest errors within each tempering scheme. Note that the results are not directly comparable with the ones in Table 1 because in the experiments without tempering all the chains of the population have the same target and all samples are used to estimate the parameters of the mixture. In the experiments with the different tempering schemes we consider, for each algorithm, the output of the chain with $\xi_i = 1$,

Table 2 Estimates of $\mu_1, \dots, \mu_4$. For the MHs we have sampled $T = 100{,}000$ draws using $N = 100$ chains. The AIMTM results are based on $T = 10{,}000$ samples obtained using $N = 100$ auxiliary chains and $M = 10$ proposals within the chain ergodic to π. The maximum absolute bias (AE) is reported for each algorithm and tempering scheme

	1	2	3	4	AE
	M1				
MH	1.81	0.73	1.02	1.79	0.97
MH1	0.64	1.62	0.91	1.59	0.86
MH2	0.81	1.75	1.12	1.99	0.69
AIMTM-IS	1.83	1.43	1.98	1.37	0.48
AIMTM-TA	0.89	1.15	1.92	1.81	0.61
	M2				
MH	0.84	0.72	1.41	0.93	0.78
MH1	1.67	1.57	1.06	1.84	0.54
MH2	1.71	1.32	1.52	1.01	0.49
AIMTM-IS	1.44	1.91	1.37	1.26	0.41
AIMTM-TA	1.86	1.19	1.51	1.49	0.36
	M3				
MH	0.82	1.25	0.83	0.97	0.68
MH1	1.79	1.42	1.33	0.98	0.52
MH2	0.99	1.27	1.63	1.69	0.51
AIMTM-IS	1.19	1.97	1.16	1.12	0.47
AIMTM-TA	1.37	1.04	1.86	1.77	0.46

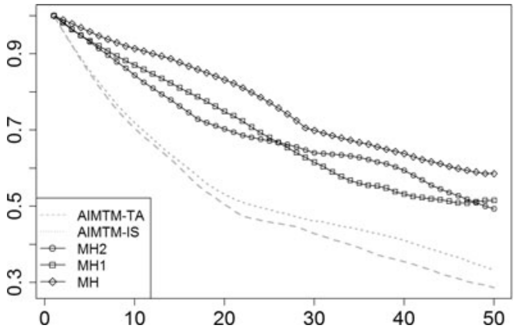

Fig. 4 Autocorrelation functions obtained by averaging over 10 independent runs of each algorithm for the population of MH and AMTM chains

which has the target π. The results in Table 2 show that the AIMTM algorithms outperform the population of MH, MH1 and MH2 chains for the three different tempering schemes. The logarithm and power decay schemes seem to give the best result when combined with the AIMTM.

The gain in efficiency with respect to the populations of MH-type algorithms is evident also from the ACF functions presented in Fig. 4. The ACF have been obtained by averaging over 10 independent runs of the algorithms considered in the comparison.

4.1.2 *Multivariate normal mixture*

We compare, for a high-dimensional target distribution, the population Monte Carlo MH with cross-over algorithm and the IMTM-TA. The target considered is the multivariate mixture of two normals with a sparse variance-covariance structure

$$\frac{1}{3}\mathcal{N}_{20}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{2}{3}\mathcal{N}_{20}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{6}$$

with $\boldsymbol{\mu}_1 = (3, \dots, 3)'$, $\boldsymbol{\mu}_2 = (10, \dots, 10)'$ and $\boldsymbol{\Sigma}_j$, with $j = 1, 2$, generated independently from a Wishart distribution $\boldsymbol{\Sigma}_j \sim \mathcal{W}_{20}(\nu, \mathbf{I}_{20})$ where $\nu = 21$ is the degrees of freedom parameter.

The comparison is based on $T = 100{,}000$ samples obtained from $N = 20$ parallel MH chains with cross-over interactions and on $T = 10{,}000$ samples obtained from $N = 20$ IMTM-TA chains, each using $M = 10$ proposal distributions. The jth proposal distribution for the ith IMTM chain, $T_j^{(i)}(\mathbf{y}|\mathbf{x}_n^{(i)})$, is Gaussian with variance-covariance matrix $\Lambda_i = (0.1 + 5i)\mathbf{I}_{20}$ for all $j = 1, \dots, M$. For MH, MH1 and MH2 populations of chains, we consider Gaussian random walk proposals with scales in the same range as the IMTM proposals.

The autocorrelation functions given in Fig. 5 are averaged over the 20 dimensions of the target, the different chains of the population and over 10 replicates of the experiment. One can see that the population of MTM chains outperforms, in terms of estimation efficiency, the populations of MHs with cross-over.

We use this example to report on the trade-off between computing time and root mean square error (RMSE) improvement for different dimensions of the target's support.

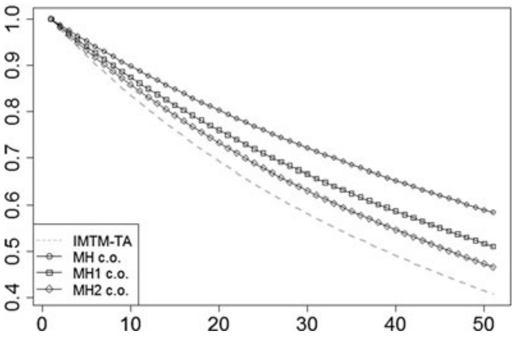

Fig. 5 Autocorrelation function (ACF) for the MH with cross-over and the IMTM algorithm. The ACF is obtained by averaging over the 20 components of the multivariate chain, the different chains of the population and over 10 replicates

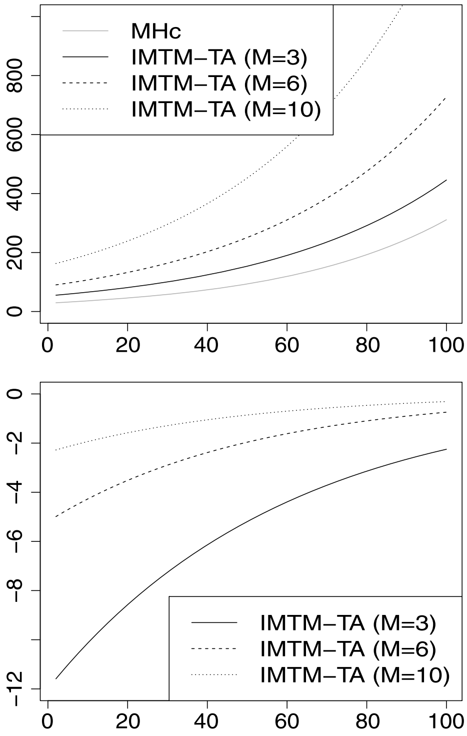

Fig. 6 Comparison of efficiency for the MH with cross-over and the IMTM-TA for different dimensions of the target distribution. Results are based on $T = 10{,}000$ samples obtained using the IMTM-TA with $N = 20$ chains and $M = 3$ (*solid line*), $M = 6$ (*dashed line*) and $M = 10$ (*dotted line*) proposals and $T = 30{,}000$ iterations of $N = 20$ parallel MH chains with cross-over (MHc). *Top panel*: Computing time in minutes. *Bottom panel*: Percentage of relative RMSE reduction per minute of added running time as defined in (7)

The comparison is based on the IMTM-TA and parallel MH with cross-over. More specifically we run the IMTM-TA for $T = 10{,}000$ iterations with $N = 20$ chains and with a different number of proposals, $M \in \{3, 6, 10\}$ and we run the MH with cross-over for $T = 30{,}000$ iterations with $N = 20$ chains. The algorithms are implemented in R and run on a machine with a Xeon X3430 2.40 GHz CPU and a Linux system. In the top panel of Fig. 6 we show the CPU times (in minutes) of each algorithm for target dimension ranging from 2 to 100. In each dimension $d = 2, \dots, 100$, $\boldsymbol{\mu}_1 = 3 \times \mathbf{1}_d$, $\boldsymbol{\mu}_2 = 10 \times \mathbf{1}_d$ ($\mathbf{1}_d$ is the d-dimensional vector with all components equal to 1) and $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ are independently generated from the Wishart distribution with $d + 1$ degrees of freedom and scale matrix $\mathbf{I}_d$. In the bottom panel of Fig. 6 we also provide an estimate of the relative RMSE reduction (in percentage) per additional minute of computation brought in by IMTM-TA over the MH with cross-over, i.e. we report

$$RF = 100 \times \frac{\Delta_{RMSE}/\text{RMSE}_{MH}}{\text{Time}_{IMTM\text{-}TA} - \text{Time}_{MH}}, \tag{7}$$

where $\Delta_{RMSE} = \text{RMSE}_{IMTM\text{-}TA} - \text{RMSE}_{MH}$. One may be tempted to conclude, based on the bottom part of Fig. 6, that it is always advantageous to use a small number of proposals. However, the astute reader may have noticed that the number of chains is fixed throughout this study at $N = 20$ and it would be difficult to generalize these results to other values of N or other target distributions. The interplay between the dimension of the target, the number of chains and the number of proposals is more subtle and based on our experiments with IMTM we recommend using a large number of chains, i.e. $N \geq 100$ and $M/N \in [5\%, 20\%]$.

4.2 Stochastic volatility

The estimation of the stochastic volatility (SV) model due to Taylor (1994) still presents challenging issues in both offline (Celeux et al. 2006) and sequential (Casarin and Marin 2009) inference contexts. First, the nonlinear structure of the model makes parameter estimation difficult. Second, the high dimension of the sampling space hinders the use of the data-augmentation and prevents the reliable joint estimation of the parameters and the latent variables. As highlighted in Casarin et al. (2009) using multiple chains with a chain interaction mechanism could lead to a substantial improvement in the MCMC method for this kind of model. We consider the SV model given in Celeux et al. (2006)

$$y_t | h_t \sim \mathcal{N}\left(0, e^{h_t}\right),$$

$$h_t | h_{t-1}, \boldsymbol{\theta} \sim \mathcal{N}\left(\alpha + \phi h_{t-1}, \sigma^2\right),$$

$$h_0 | \boldsymbol{\theta} \sim \mathcal{N}\left(0, \sigma^2/(1 - \phi^2)\right),$$

with $t = 1, \dots, T$ and $\boldsymbol{\theta} = (\alpha, \phi, \sigma^2)$. For the parameters we assume the noninformative prior (see Celeux et al. 2006)

$$\pi(\boldsymbol{\theta}) \propto 1/(\sigma\beta)\mathbb{I}_{(-1,1)}(\phi),$$

where $\beta^2 = \exp(\alpha)$. In order to simulate from the posterior we consider the full conditional distributions and apply a Gibbs algorithm. If we define $\mathbf{y} = (y_1, \dots, y_T)$ and

$\mathbf{h} = (h_0, \ldots, h_T)$ then the full conditionals for β and ϕ are the inverse gamma distributions

$$\beta^2|\mathbf{h}, \mathbf{y} \sim \mathcal{IG}\left((T-1)/2, \sum_{t=1}^{T} y_t^2 \exp(-h_t)/2\right),$$

$$\sigma^2|\phi, \mathbf{h}, \mathbf{y} \sim \mathcal{IG}\left((T-1)/2, \sum_{t=2}^{T} (h_t - \phi h_{t-1})^2/2 + h_1^2(1-\phi^2)\right)$$

and ϕ and the latent variables have non-standard full conditionals

$$\pi(\phi|\sigma^2, \mathbf{h}, \mathbf{y}) \propto (1-\phi^2)^{1/2} \exp\left(-\frac{\phi^2}{2\sigma^2} \sum_{t=2}^{T-1} h_t^2 - \frac{\phi}{\sigma^2} \sum_{t=2}^{T} h_t h_{t-1}\right) \times \mathbb{I}_{(-1,+1)}(\phi),$$

$$\pi(h_t|\alpha, \phi, \sigma^2, \mathbf{h}, \mathbf{y}) \propto \exp\left\{-\frac{1}{2\sigma^2}[(h_t - \alpha - \phi h_{t-1})^2 - (h_{t+1} - \alpha - \phi h_t)^2] - \frac{1}{2}[h_t + y_t^2 \exp(-h_t)]\right\}.$$

In order to sample from the posterior we use the IMTM-IS within Gibbs algorithm. Particularly, in the IMTM step for ϕ, we follow Celeux et al. (2006), and use as proposal, a truncated normal distribution on $(-1, 1)$ with mean and variance

$$\sum_{t=2}^{T} h_t h_{t-1} \Big/ \sum_{t=2}^{T-1} h_t^2 \quad \text{and} \quad \sigma^2 \Big/ \sum_{t=1}^{T-1} y_t^2.$$

One of the most difficult issues is related to the choice of the proposal distribution for h_t. In this paper we follow a standard approach based on the second-order Taylor approximation of the term $\exp\{h_t\}$, in the full conditional of h_t, around the mean μ_t of the distribution of $h_t|h_{t-1}, \phi, \sigma^2$. The approach has been introduced by Shephard and Pitt (1997) and has been adapted to the context of iterated importance sampling by Celeux et al. (2006). The proposal distribution for h_t, $1 \le t \le T$ is $\mathcal{N}(A_t, B_t)$ where

$$A_1 = \frac{\phi h_2 \sigma^{-2} + 0.5 \exp(-\phi h_2) y_1^2 (1 + \phi h_2) \beta^{-2} - 0.5}{\sigma^{-2} + 0.5 \exp(-\phi h_2) y_1^2 \beta^{-2}},$$

$$A_t = \frac{(1+\phi^2)\mu_t \sigma^{-2} + 0.5 \exp(-\mu_t) y_t^2 (1+\mu_t)\beta^{-2} - 0.5}{(1+\phi^2)\sigma^{-2} + 0.5 \exp(-\mu_t) y_t^2 \beta^{-2}}, \quad \forall t = 2, \ldots, T-1,$$

$$A_T = \frac{\phi h_{T-1} \sigma^{-2} + 0.5 \exp(-\phi h_{T-1}) y_T^2 (1 + \phi h_{T-1}) \beta^{-2} - 0.5}{\sigma^{-2} + 0.5 \exp(-\phi h_{T-1}) y_T^2 \beta^{-2}}$$

and

$$B_1 = (\sigma^{-2} + 0.5 \exp(-\phi h_2) y_1^2 \beta^{-2})^{-1},$$

$$B_t = \frac{(1+\phi^2)\mu_t \sigma^{-2} + 0.5 \exp(-\mu_t) y_t^2 (1+\mu_t)\beta^{-2} - 0.5}{(1+\phi^2)\sigma^{-2} + 0.5 \exp(-\mu_t) y_t^2 \beta^{-2}}, \quad \forall t = 2, \ldots, T-1,$$

$$B_T = \left[\sigma^{-2} + 0.5 \exp(-\phi h_{T-1}) y_T^2 \beta^{-2}\right]^{-1}.$$

The IMTM implementation uses M different independent proposals which are obtained from the population of chains according to the design outlined in Algorithm 3.

It has been recognized that the single-move Gibbs sampler updates sequentially the latent variables and is inefficient. A possible remedy (see Shephard and Pitt 1997) consists in simulating jointly groups of latent variables (this approach is henceforth referred to as *blocking*). The IMTM algorithms proposed here can be extended to accommodate the blocking procedure. Implementation of the classical MTM (Algorithm 1) for this example was proposed by So (2006) who discussed three types of MTM multi-move Gibbs samplers: the autoregressive MTM, the independent kernel MTM (IKMTM) and the posterior mode direction sampling for the Bayesian analysis of state space models. The algorithms extend in the MTM context the block sampling strategy for state space models introduced by Shephard and Pitt (1997). Since the three algorithms are based on a single chain and use the same distribution for generating the proposals of the MTM, at each iteration they are able to explore only one direction of the state space. A combination of our IMTM strategy with one of the algorithms in So (2006) can use at each iteration different proposal distributions with different directions. In our paper we consider the IKMTM algorithm that relies on an independent proposal distribution for the block of latent variables and generates multiple draws from the proposal distribution to explore the state space. We propose to combine the blocking strategy of the IKMTM with our interacting kernel strategy thus obtaining an IMTM-IS algorithm with blocking which is denoted IMTM-IS-b.

We consider in our simulations two parameter settings, $(\alpha, \phi, \sigma^2) = (0, 0.99, 0.01)$ and $(\alpha, \phi, \sigma^2) = (0, 0.9, 0.1)$, which correspond, in a financial stock market context, to daily and weekly frequency data, respectively. Note that, as reported in Casarin and Marin (2009), inference in the daily example is more difficult. We compare the IMTM-within-Gibbs algorithms with a population of MH-within-Gibbs in terms of Mean Square Error (MSE) for the parameters and of cumulative RMSE for the latent variables. We carry out the comparison based on the MSE and the SD by running

Table 3 Mean square error (MSE) and standard deviation (in parenthesis) for the parameter estimation with IMTM-IS, IMTM-IS-b and MH within Gibbs algorithms. Top panel: daily data. Bottom panel: weekly data

θ	Value	MSE		
		IMTM-IS	IMTM-IS-b	MH
	Daily data			
α	0	0.00541	0.001289	0.01520
		(0.00121)	(0.000946)	(0.000782)
ϕ	0.99	0.05979	0.008735	0.083013
		(0.00266)	(0.001283)	(0.004432)
σ^2	0.01	0.00069	0.000315	0.005318
		(0.00029)	(0.000195)	(0.000389)
	Weekly data			
α	0	0.000608	0.000243	0.000839
		(0.000249)	(0.000929)	(0.000722)
ϕ	0.9	0.00724	0.006273	0.04096
		(0.00038)	(0.000291)	(0.00423)
σ^2	0.1	0.00077	0.000717	0.00716
		(0.00033)	(0.000989)	(0.00106)

the algorithms on 20 independent simulated datasets of 1000 observations. In our comparison we account for the computational cost by sampling $T = 50{,}000$ draws from the posterior using the population of MH with $N = 20$ chains, and only $T = 10{,}000$ samples using the IMTM-IS and IMTM-IS-b within Gibbs, each with $N = 20$ interacting chains and $M = 5$ proposals. For IMTM-IS-b we use blocks of size 5 (see Shephard and Pitt 1997, for a discussion on the choice of the block size). We left the study of the optimal block size for further research as the main goal of our simulation study is to demonstrate the IMTM algorithm's ability to break down the dependence in the single-move sampler and thus to improve the efficiency of the Monte Carlo sample. We also expect that, due to the degeneracy of the selection weights, the efficiency of our algorithm will deteriorate as the size of the block increases.

The results for the parameter estimation when applying IMTM-IS and IMTM-IS with blocking (IMTM-IS-b) are presented in Table 3 and show an effective improvement in the estimates, both for weekly and daily data, when compared to the results of a MH algorithm with an equivalent computational load.

Figure 7 shows the estimated maximum ACF for the 1000 components associated to the latent process $\{h_t\}_{t=1,\dots,T}$ with $T = 1000$. The maximum ACF is evaluated over the chains in the Monte Carlo population and over 10 independent replicates for the population MH (dashed line), IMTM-IS (solid black line) and IMTM-IS-b (solid grey line) algorithms. Figure 7 shows the results for the daily data (top panel) and for the weekly data (bottom panel). In both setups IMTM-IS and IMTM-IS-b outperform the population MH in terms of estimation efficiency. We should notice that there is an efficiency improvement when using block sampling and that the improvement is larger for the daily dataset than for the weekly dataset.

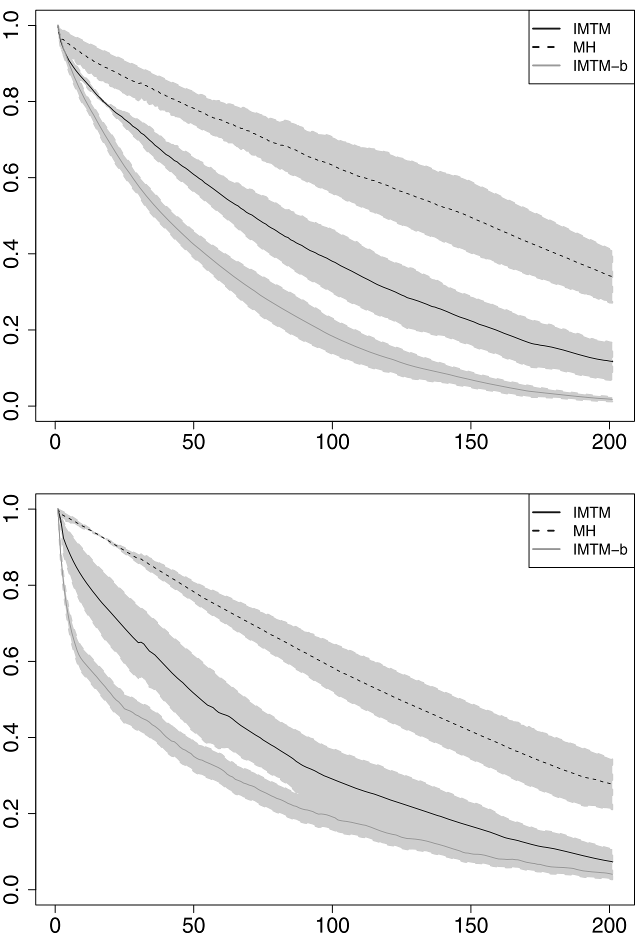

Fig. 7 Maximum value of the ACF associated with the 1000 latent process $\{h_t\}_{t=1,\dots,T}$, for daily (*top*) and weekly (*bottom*) datasets. The maximum value of the ACFs for the IMTM-IS (*solid black line*), MH (*dashed line*) and IMTM-IS-b (*solid gray line*) and the 90%HPD regions (*gray areas*) are estimated from different chains of the population and independent replicates

These results are similar to the results obtained for SV models in Celeux et al. (2006), Casarin and Marin (2009) and Casarin et al. (2009) for population Monte Carlo algorithms. We can conclude that while the IMTM shares some of its properties with other population Monte Carlo algorithms it has the advantage that the convergence of the algorithm relies upon the detail balance condition and no further theoretical results are needed.

Figure 8 show the HPD region at the 90% (grey areas) and the mean (black lines) of the cumulative RMSE of each algorithm for the weekly (top panel) and daily data (bottom panel). The statistics have been estimated from 10 independent experiments. The average RMSE shows that in

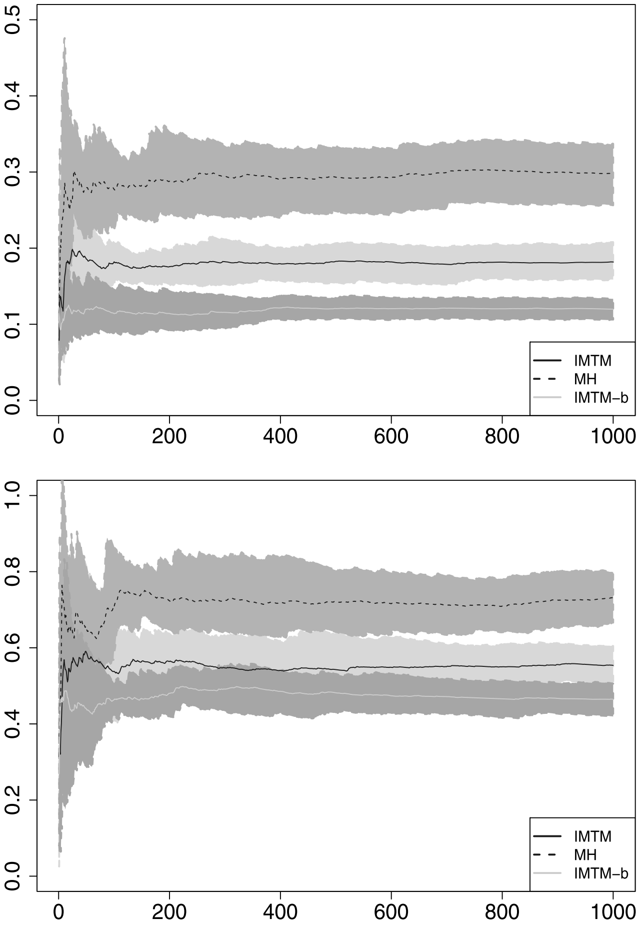

Fig. 8 Cumulative RMSE for the IMTM-IS (*solid black line*), MH (*dashed line*) and ITMT-IS-b (*solid gray line*) and the 90% HPD regions for the thre algorithms (*gray areas*) estimated on 20 independent experiments for both the daily (*top*) and weekly (*bottom*) datasets

both settings considered here, IMTM-IS (solid black line) and IMTMT-IS-b (solid grey line) are more efficient than the standard MH algorithm (dashed line).

4.3 Loss of heterozygosity application

We consider here the problem of the genetic instability of esophageal cancers. During a neoplastic progression the cancer cells undergo a number of genetic changes and possibly lose entire chromosome sections. The loss of a chromosome section containing one allele by abnormal cells is called *Loss of Heterozygosity* (LOH). The LOH can be detected using laboratory assays on patients with two different alleles for a particular gene. Chromosome regions containing genes which regulate cell behavior, are hypothesized to have a high rates of LOH. Consequently the loss of these chromosome sections disables important cellular controls.

Chromosome regions with high rates of LOH are hypothesized to contain *Tumor Suppressor Genes* (TSGs), whose deactivation contributes to the development of esophageal cancer. Moreover the neoplastic progression is thought to produce a high level of background LOH in all chromosome regions.

In order to discriminate between “background” and TSGs LOH, the Seattle Barrett’s Esophagus research project (Barrett et al. 1996) has collected LOH rates from esophageal cancers for 40 regions, each on a distinct chromosome arm. The labeling of the two groups is unknown so Desai (2000) suggest to consider a mixture model for the frequency of LOH in both the “background” and TSG groups.

We consider the hierarchical Beta-Binomial mixture model proposed in Warnes (2001)

$$\begin{aligned} & f(x,n|\eta,\pi_1,\pi_2,\gamma) \\ &\quad = \eta\binom{n}{x}\pi_1^x(1-\pi_1)^{n-x} \\ &\qquad + (1-\eta)\binom{n}{x}\frac{\Gamma(1/\omega_2)}{\Gamma(\pi_2/\omega_2)\Gamma((1-\pi_2)/\omega_2)} \\ &\qquad \times \frac{\Gamma(x+\pi_2/\omega_2)\Gamma(n-x+(1-\pi_2)/\omega_2)}{\Gamma(n+1/\omega_2)} \end{aligned} \tag{8}$$

with x number of LOH sections, n the number of examined sections, $\omega_2 = \exp\{\gamma\}/(2(1+\exp\{\gamma\}))$. Let $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{n} = (n_1, \ldots, n_m)$ be a set of observations from $f(x,n|\eta,\pi_1,\pi_2,\gamma)$ and let us assume the following priors

$$\begin{aligned} &\eta \sim \mathcal{U}[0,1], \qquad \pi_1 \sim \mathcal{U}[0,1], \\ &\pi_2 \sim \mathcal{U}[0,1] \quad \text{and} \quad \gamma \sim \mathcal{U}[-30,30] \end{aligned} \tag{9}$$

with $\mathcal{U}$ the uniform distribution on $[a,b]$. Then the posterior distribution is

$$\pi(\eta,\pi_1,\pi_2,\gamma|\mathbf{x},\mathbf{n}) \propto \prod_{j=1}^{m} f(x_j,n_j|\eta,\pi_1,\pi_2,\gamma). \tag{10}$$

The parametric space is of dimension four: $(\eta,\pi_1,\pi_2,\gamma) \in [0,1]^3 \times [-30,30]$ and the posterior distribution has two well-separated modes making it difficult to sample using generic methods.

We apply the IMTM-IS algorithm $M = 4$ proposal functions selected between a population of $N = 100$ chains. The values of the population of chains (dots) at the last iteration on the subspace (π_1,π_2) is given in Fig. 9. The IMTM-IS is able to visit both regions of the parameter space and confirms the analysis of Craiu et al. (2009) and Warnes (2001).

5 Conclusions

In this paper we propose a new class of interacting multiple-try Metropolis algorithms that extends the existing literature in two directions. First , the multiple try transition kernel has

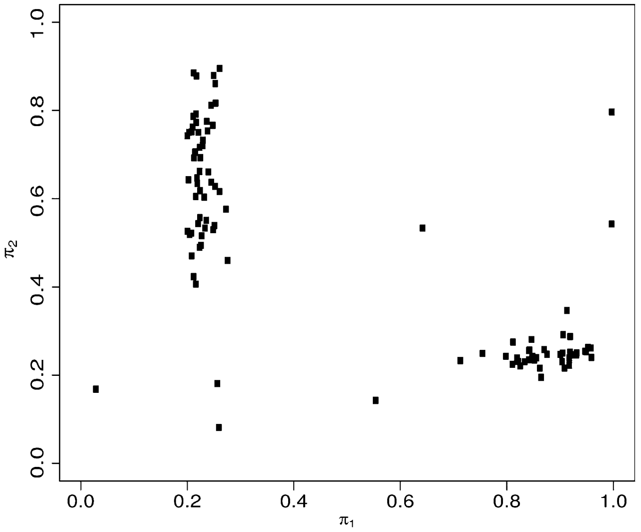

Fig. 9 Values of the population of chains (*dots*) at the last iteration on the subspace (π_1, π_2). The interaction is given by $M = 4$ proposal functions randomly selected between the population of $N = 100$ chains

been extended to allow to use of different proposal distributions and second, we propose a new interacting Monte Carlo algorithm for increasing the efficiency of MTM. We give a proof of validity of the algorithm and show on real and simulated examples the effective efficiency improvement. We have compared our IMTM with population MH and left for future research a comparison with importance sampling based methods such as the Population Monte Carlo methods or Sequential Monte Carlo methods described in Jasra et al. (2007). We note here that the use of antithetic and stratified sampling discussed by Craiu and Lemieux (2007) can be extended to the current setting. When implementing the IMTM sampler in practice one has to tune a number of simulation parameters. We are confident that some, if not all, these parameters can be changed "on the fly" based on principles developed within the class of adaptive MCMC. Future work will focus on building stronger ties between IMTM and the emerging area of adaptive MCMC.

Acknowledgements We would like to thank the Editor, Professor Gilles Celeux, an Associate Editor and two referees for constructive suggestions that have greatly improved the paper. The work of RVC has been supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

Appendix: Proof

Without loss of generality, we can set $M_i = N$, $\forall i$ and $x_n^{(i)} = x$. Fixed the ith chain, the conditional detailed balance is proved. This ensures the ergodicity of the chain.

Following the notations in Algorithm 3, let us define the following quantities

$$\bar{w}^{(i)}(y_{1:N}|x) = \sum_{j=1}^{N} w_j^{(i)}(y_j, x),$$

$$\bar{w}_{-k}^{(i)}(y_{1:N}|x) = \sum_{j \neq k}^{N} w_j^{(i)}(y_j, x)$$

and

$$S_N(J) = \frac{1}{\bar{w}^{(i)}(y_{1:N}|x)} \sum_{j=1}^{N} \delta_j(J) w_j^{(i)}(y_j, x)$$

with $J \in \mathcal{J} = \{1, \dots, N\}$ the empirical measure generated by different proposals and by the normalized selection weights.

Let $T^{(i)}(dy_{1:N} \mid x) = \bigotimes_{j=1}^{N} T_j^{(i)}(dy_j \mid \tilde{f}_n^{(i)}(x))$ the joint proposal for the multiple try and define $T_{-k}^{(i)}(dy_{1:N} \mid x) = \bigotimes_{j \neq k}^{N} T_j^{(i)}(dy_j \mid \tilde{f}_n^{(i)}(x))$. Let $A(x, y)$ be the actual transition probability for moving from x to y in the IMTM (Algorithm 3). Suppose that $x \neq y$, then the transition is a results two steps. The first step is a selection step which can be written as $y = y_J$ and $x_J^* = x$ with the random index J sampled from the empirical measure $S_N(J)$. The second step is a accept/reject step based on the generalized MH ratio which involves the generation of the auxiliary values x_j^* for $j \neq J$. Then

$$\begin{aligned}
&\pi(x)A(x, y) \\
&\quad = \pi(x) \int_{\mathcal{Y}^N} T^{(i)}(dy_{1:N} \mid x) \int_{\mathcal{J}} S_N(dJ) \\
&\qquad \times \int_{\mathcal{Y}^{N-1} \times \mathcal{Y}^2} T_{-J}^{(i)}(dx_{1:N}^* \mid y) \\
&\qquad \times \delta_x(dx_J^*)\delta_{y_J}(dy) \min\left\{1, \frac{\bar{w}^{(i)}(y_{1:N}|x)}{\bar{w}^{(i)}(x_{1:N}^*|y)}\right\} \\
&\quad = \pi(x) \sum_{j=1}^{N} \int_{\mathcal{Y}^{N-1}} T_{-j}^{(i)}(dy_{1:N} \mid x) T_j^{(i)}(y \mid \tilde{f}_n^{(i)}(x)) \\
&\qquad \times \int_{\mathcal{Y}^{N-1}} T_{-j}^{(i)}(dx_{1:N}^* \mid y) \\
&\qquad \times \frac{w_j^{(i)}(y, x)}{w_j^{(i)}(y, x) + \bar{w}_{-j}^{(i)}(y_{1:N}|x)} \\
&\qquad \times \min\left\{1, \frac{w_j^{(i)}(y, x) + \bar{w}_{-j}^{(i)}(y_{1:N}|x)}{w_j^{(i)}(x, y) + \bar{w}_{-j}^{(i)}(x_{1:N}^*|y)}\right\} \\
&\quad = \sum_{j=1}^{N} \frac{w_j^{(i)}(x, y) w_j^{(i)}(y, x)}{\lambda_j^{(i)}(y, x)} \int_{\mathcal{Y}^{2(N-1)}} T_{-j}^{(i)}(dy_{1:N} \mid x)
\end{aligned}$$

$$\times T_{-j}^{(i)}(dx_{1:N}^* \mid y) \min\left\{ \frac{1}{w_j^{(i)}(y,x)+\bar{w}_{-j}^{(i)}(y_{1:N}|x)}, \right.$$
$$\left. \frac{1}{w_j^{(i)}(x,y)+\bar{w}_{-j}^{(i)}(x_{1:N}^*|y)} \right\}$$

which is symmetric in x and y.

References

Atchadé, Y., Roberts, G.O., Rosenthal, J.S.: Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. Stat. Comput. **21**, 555–568 (2011)

Barrett, M., Galipeau, P., Sanchez, C., Emond, M., Reid, B.: Determination of the frequency of loss of heterozygosity in esophageal adenocarcinoma nu cell sorting, whole genome amplification and microsatellite polymorphisms. Oncogene **12** (1996)

Bédard, M., Douc, R., Moulines, E.: Scaling analysis of multiple-try MCMC methods. Technical report, Université de Montréal (2010)

Campillo, F., Rakotozafy, R., Rossi, V.: Parallel and interacting Markov chain Monte Carlo algorithm. Math. Comput. Simul. **79**, 3424–3433 (2009)

Cappé, O., Gullin, A., Marin, J., Robert, C.P.: Population Monte Carlo. J. Comput. Graph. Stat. **13**, 907–927 (2004)

Casarin, R., Marin, J.-M.: Online data processing: Comparison of Bayesian regularized particle filters. Electron. J. Stat. **3**, 239–258 (2009)

Casarin, R., Marin, J.-M., Robert, C.: A discussion on: approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations by Rue, H., Martino, S. and Chopin, N. J. R. Stat. Soc. B **71**, 360–362 (2009)

Celeux, G., Marin, J.-M., Robert, C.: Iterated importance sampling in missing data problems. Comput. Stat. Data Anal. **50**, 3386–3404 (2006)

Chauveau, D., Vandekerkhove, P.: Improving convergence of the Hastings-Metropolis algorithm with an adaptive proposal. Scand. J. Stat. **29**, 13 (2002)

Craiu, R.V., Lemieux, C.: Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling. Stat. Comput. **17**, 109–120 (2007)

Craiu, R.V., Meng, X.L.: Multi-process parallel antithetic coupling for forward and backward MCMC. Ann. Stat. **33**, 661–697 (2005)

Craiu, R.V., Rosenthal, J.S., Yang, C.: Learn from thy neighbor: parallel-chain adaptive and regional MCMC. J. Am. Stat. Assoc. **104**, 1454–1466 (2009)

Del Moral, P.: Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications. Springer, Berlin (2004)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc. B **68**, 411–436 (2006)

Desai, M.: Mixture models for genetic changes in cancer cells. Ph.D. thesis, University of Washington (2000)

Früwirth-Schnatter, S.: Finite Mixture and Markov Switching Models. Springer, Berlin (2006)

Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences (with discussion). Stat. Sci. 457–511 (1992)

Geyer, C.J., Thompson, E.A.: Annealing Markov chain Monte Carlo with applications to ancestral inference. Tech. rep. 589, University of Minnesota (1994)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109 (1970)

Heard, N.A., Holmes, C., Stephens, D.: A quantitative study of gene regulation involved in the immune response od anophelinemosquitoes: an application of Bayesian hierarchical clustering of curves. J. Am. Stat. Assoc. **101**, 18–29 (2006)

Jasra, A., Stephens, D.A., Holmes, C.: Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modelling. Stat. Sci. **20**, 50–67 (2005)

Jasra, A., Stephens, D., Holmes, C.: On population-based simulation for static inference. Stat. Comput. **17**, 263–279 (2007)

Jennison, C.: Discussion of Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, by A.F.M. Smith and G.O. Roberts. J. R. Stat. Soc. B **55**, 54–56 (1993)

Liu, J., Liang, F., Wong, W.: The multiple-try method and local optimization in Metropolis sampling. J. Am. Stat. Assoc. **95**, 121–134 (2000)

Liang, F., Wong, W.: Real parameter evolutionary Monte Carlo with applications to Bayesian mixture models. J. Am. Stat. Assoc. **96**, 653–666 (2001)

Marinari, E., Parisi, G.: Simulated tempering: a new Monte Carlo scheme. Europhys. Lett. **19**, 451–458 (1992)

Mengersen, K., Robert, C.: The pinball sampler. In: Bernardo, J., Dawid, A., Berger, J., West, M. (eds.) Bayesian Statistics 7. Springer, Berlin (2003)

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equations of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)

Neal, R.M.: Sampling from multimodal distributions using tempered transitions. Tech. rep. 9421, University of Toronto (1994)

Pandolfi, S., Bartolucci, F., Friel, N.: A generalization of the multiple-try Metropolis algorithm for Bayesian estimation and model selection. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy, pp. 581–588 (2010a)

Pandolfi, S., Bartolucci, F., Friel, N.: A generalized Multiple-try Metropolis version of the Reversible Jump algorithm. Tech. rep. (2010b). http://arxiv.org/pdf/1006.0621

Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. Genetics **155**, 945–959 (2000)

Richardson, S., Green, P.J.: On Bayesian analysis of mixtures with an unknown number of components (with discussion). J. R. Stat. Soc. B **4**(59), 731–792 (1997)

Shephard, N., Pitt, M.: Likelihood analysis of non-Gaussian measurement time series. Biometrika **84**, 653–667 (1997)

So, M.K.P.: Bayesian analysis of nonlinear and non-Gaussian state space models via multiple-try sampling methods. Stat. Comput. **16**, 125–141 (2006)

Taylor, S.: Modelling stochastic volatility. Math. Finance **4**, 183–204 (1994)

Warnes, G.: The Normal kernel coupler: an adaptive Markov chain Monte Carlo method for efficiently sampling from multi-modal distributions. Technical report, George Washington University (2001)