Statistics and Probability Letters 82 (2012) 1445–1453

Contents lists available at SciVerse ScienceDirect

Statistics and Probability Letters

journal homepage: www.elsevier.com/locate/stapro

A multi-point Metropolis scheme with generic weight functions

Luca Martino [*], Victor Pascual Del Olmo, Jesse Read

Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911, Leganes, Madrid, Spain

ARTICLE INFO

Article history:
Received 30 November 2011
Received in revised form 4 April 2012
Accepted 5 April 2012
Available online 13 April 2012

Keywords:
Multiple Try Metropolis algorithm
Multi-point Metropolis algorithm
MCMC methods

ABSTRACT

The multi-point Metropolis algorithm is an advanced MCMC technique based on drawing several correlated samples at each step and choosing one of them according to some normalized weights. We propose a variation of this technique where the weight functions are not specified, i.e., the analytic form can be chosen arbitrarily. This has the advantage of greater flexibility in the design of high-performance MCMC samplers. We prove that our method fulfills the balance condition, and provide a numerical simulation. We also give new insight into the functionality of different MCMC algorithms, and the connections between them.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Monte Carlo statistical methods are powerful tools for numerical inference and stochastic optimization (see Robert and Casella (2004), for instance). Markov chain Monte Carlo (MCMC) methods are classical Monte Carlo techniques that generate samples from a target probability density function (pdf) by drawing from a simpler proposal pdf, usually to approximate an otherwise-incalculable (analytically) integral (Liu, 2004; Liang et al., 2010). MCMC algorithms produce a Markov chain with a stationary distribution that coincides with the target pdf.

The Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) is the most famous MCMC technique. It can be applied to almost any target distribution. In practice, however, finding a "good" proposal pdf can be difficult. In some applications, the Markov chain generated by the MH algorithm can remain trapped almost indefinitely in a local mode meaning that, in practice, convergence may not be reached.

The *Multiple-Try Metropolis* (MTM) method of Liu et al. (2000) is an extension of the MH algorithm in which the next state of the chain is selected among a *set* of independent and identically distributed (i.i.d.) samples. This enables the MCMC sampler to make large step-size jumps without a lowering the acceptance rate; and thus MTM can explore a larger portion of the sample space in fewer iterations.

An interesting special case of the MTM, well-known in molecular simulation field, is the *orientational bias Monte Carlo*, as described in Chapter 13 of Frenkel and Smit (1996) and Chapter 5 of Liu (2004), where i.i.d. candidates are drawn from a *symmetric* proposal pdf, and one of these is chosen according to some weights directly proportional to the target pdf. Here, however, the analytic form of the weight functions is fixed and unalterable.

Casarin et al. (in press) introduced a MTM scheme using different proposal pdfs. In this case the samples produced are independent but not identically distributed. In Qin and Liu (2001), another generalization of the MTM (called the *multi-point Metropolis* method) is proposed using correlated candidates at each step. Clearly, the proposal pdfs are also different in this case.

Moreover, in Pandolfi et al. (2010) an extension of the classical MTM technique is introduced where the analytic form of the weights is not specified. In Pandolfi et al. (2010), the same proposal pdf is used to draw samples, so that the candidates

[*] Corresponding author. Tel.: +34 695276294.
E-mail addresses: luca@tsc.uc3m.es (L. Martino), vpolmo@tsc.uc3m.es (V.P. Del Olmo), jesse@tsc.uc3m.es (J. Read).

0167-7152/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.spl.2012.04.008

generated in each step of the algorithm are i.i.d. Further interesting and related considerations about the use of auxiliary variables for building acceptance probabilities within a MH approach can be found in Storvik (2011).

In this paper, we draw from the two approaches (Qin and Liu, 2001; Pandolfi et al., 2010) to create a novel algorithm that selects a new state of the chain among correlated samples using generic weight functions, i.e., the analytic form of the weights can be chosen arbitrarily. Furthermore, we formulate the algorithm and the acceptance rule in order to fulfill the detailed balance condition.

Our method allows more flexibility in the design of efficient MCMC samplers with a larger coverage and faster exploration of the sample space. In fact, we can choose any bounded and positive weight functions to either improve performance or reduce computational complexity, independently of the chosen proposal pdf. Moreover, since in our approach the proposal pdfs are different, adaptive or interacting techniques can be applied, such as those introduced by Andrieu and Moulines (2006) and Casarin et al. (in press). An important advantage of our procedure is that, since in our procedure a new candidate is drawn from a conditional pdf which depends on the samples generated earlier during the same time step, it constructs an improved proposal by automatically building on the information obtained from the generated samples.

The rest of the paper is organized as follows. In Section 2 we recall the standard multi-point Metropolis algorithm. In Section 3 we introduce our novel scheme with generic weight functions and correlated samples. Section 4 provides a rigorous proof that the novel scheme satisfies the detailed balance condition. A numerical simulation is provided in Section 5 and finally, in Section 6, we discuss the advantages of our proposed technique and provide an insight into the relationships among different MTM schemes in literature.

2. Multi-point Metropolis algorithm

In the classical MH algorithm, a new possible state is drawn from the proposal pdf and the movement is accepted with a suitable decision rule. In the multi-point approach, several correlated samples are generated and, from these, a "good" one is chosen.

Specifically, consider a target pdf $p_o(x)$ known up to a constant (hence, we can evaluate $p(x) \propto p_o(x)$). Given a current state $x \in \mathbb{R}$ (we assume scalar values only for simplicity in the treatment), we draw N correlated samples each step from a sequence of different proposal pdfs $\{\pi_j\}_{j=1}^N$, i.e.,

$$y_1 \sim \pi_1(\cdot|x), y_2 \sim \pi_2(\cdot|x, y_1), y_3 \sim \pi_3(\cdot|x, y_1, y_2), \dots, y_N \sim \pi_N(\cdot|x, y_1, \dots, y_{N-1}). \tag{1}$$

Therefore, we can write the joint distribution of the generated samples as

$$q_N(y_1, \dots, y_N|x) = q_N(y_{1:N}|x) = \pi_1(y_1|x)\pi_2(y_2|x, y_1) \cdots \pi_N(y_N|x, y_{1:N-1}), \tag{2}$$

i.e.,

$$q_N(y_{1:N}|x) = \pi_1(y_1|x) \prod_{j=2}^{N} \pi_j(y_j|x, y_{1:j-1}) \tag{3}$$

where, for brevity, we use the notation $y_{1:j} \triangleq [y_1, \dots, y_j]$ and $y_{j:1} \triangleq [y_j, \dots, y_1]$ denotes the vector with the reverse order.

A "good" candidate among the generated samples is chosen according to weight functions

$$\omega_j(z_1, \dots, z_{j+1}) \in \mathbb{R}^{j+1} \to \mathbb{R}^+$$

where $z_1, \dots, z_{j+1}$, are generic variables and $j = 1, \dots, N$. The specific analytic form of the weights needed in this technique is

$$\omega_j(z_1, \dots, z_{j+1}) \triangleq p(z_1)\pi_1(z_2|z_1) \cdots \pi_N(z_j|z_{1:j-1})\lambda_j(z_1, \dots, z_{j+1}), \tag{4}$$

where $p(x) \propto p_o(x)$ is the target pdf, λ_j can be any bounded, positive, and *sequentially symmetric* function, i.e.,

$$\lambda_j(z_1, z_{2:j+1}) = \lambda_j(z_{j+1:2}, z_1). \tag{5}$$

Note that, since $q_j(z_{2:j+1}|z_1) = \pi_1(z_2|z_1) \cdots \pi_N(z_j|z_{1:j-1})$ (see Eq. (2)), we can rewrite the weight functions as

$$\omega_j(z_1, \dots, z_{j+1}) = p(z_1)q_j(z_{2:j+1}|z_1)\lambda_j(z_1, z_{2:j+1}). \tag{6}$$

2.1. Algorithm

Given a current state $x = x_t$, the multi-point Metropolis algorithm consists of the following steps:

1. Draw N samples $y_{1:N} = [y_1, y_2, \dots, y_N]$ from the joint pdf

$$q_N(y_{1:N}|x) = \pi_1(y_1|x) \prod_{j=2}^{N} \pi_j(y_j|x, y_{1:j-1})$$

namely, draw y_j from $\pi_j(\cdot|x, y_{1:j-1})$, with $j = 1, \dots, N$.

2. Calculate the weights $\omega_j(y_{j:1}, x)$ as in Eq. (4), and normalize them to obtain $\bar{\omega}_j, j = 1, \ldots, N$.
3. Draw a $y = y_k \in \{y_1, \ldots, y_N\}$ according to their weights $\bar{\omega}_1, \ldots, \bar{\omega}_N$.
4. Set

$$x_1^* = y_{k-1}, \qquad x_2^* = y_{k-2}, \ldots, x_{k-1}^* = y_1, \tag{7}$$

and finally $x_k^* = x$. Then, draw other "reference" samples

$$x_j^* \sim \pi_i(\cdot|y, x_{1:j-1}^*), \tag{8}$$

for $j = k+1, \ldots, N$. Note that for $j = k+1$ we have

$$\pi_j(\cdot|y, x_{1:j-1}^*) = \pi_j(\cdot|y, x_1^* = y_{k-1}, \ldots, x_{k-1}^* = y_1, x_k^* = x),$$

and, for $j = k+2, \ldots, N$, we have

$$\pi_j(\cdot|y, x_{1:j-1}^*) = \pi_j(\cdot|y, x_1^* = y_{k-1}, \ldots, x_{k-1}^* = y_1, x_k^* = x, x_{k+1:j-1}^*).$$

5. Compute $\omega_j(x_{j:1}^*, y)$ as in Eq. (4).
6. Let $x_{t+1} = y_k$ with probability

$$\alpha = \min\left[1, \frac{\sum_{j=1}^{N} \omega_j(y_{j:1}, x)}{\sum_{j=1}^{N} \omega_j(x_{j:1}^*, y)}\right], \tag{9}$$

otherwise set $x_{t+1} = x$ with probability $1 - \alpha$.
7. Set $t = t + 1$ and repeat from step 1.

The kernel of this technique satisfies the detailed balance condition as shown in Qin and Liu (2001). However, to fulfill this condition, the algorithm requires the weights to be defined *exactly* with the form in Eq. (4).

3. Extension with generic weight functions

Now, we consider generic weight functions $\omega_j(z_1, \ldots, z_{j+1}) \in \mathbb{R}^{j+1} \to \mathbb{R}^+$, that have to be (a) bounded and (b) positive. In this case, the algorithm can be described as follows.

1. Draw N samples $y_{1:N} = [y_1, y_2, \ldots, y_N]$ from the joint pdf

$$q_N(y_{1:N}|x) = \pi_1(y_1|x) \prod_{j=2}^{N} \pi_j(y_j|x, y_{1:j-1})$$

namely, draw y_j from $\pi_j(\cdot|x, y_{1:j-1})$, with $j = 1, \ldots, N$.
2. Choose some suitable (bounded and positive) weight functions. Then, calculate each weight $\omega_j(y_{j:1}, x)$, and normalize them to obtain $\bar{\omega}_j, j = 1, \ldots, N$.
3. Draw a $y = y_k \in \{y_1, \ldots, y_N\}$ according to $\bar{\omega}_1, \ldots, \bar{\omega}_N$, and set $\bar{W}_y = \bar{\omega}_k$, i.e.,

$$\bar{W}_y \triangleq \frac{\omega_k(y_{k:1}, x)}{\sum_{j=1}^{N} \omega_j(y_{j:1}, x)}. \tag{10}$$

4. Set

$$x_1^* = y_{k-1}, \qquad x_2^* = y_{k-2}, \ldots, x_{k-1}^* = y_1, \tag{11}$$

and finally $x_k^* = x$. Then, draw the remaining "reference" samples

$$x_j^* \sim \pi_j(\cdot|y, x_{1:j-1}^*), \tag{12}$$

for $j = k+1, \ldots, N$.
5. Compute the general weights $\omega_j(x_{j:1}^*, y)$ and calculate the normalized weight

$$\bar{W}_x \triangleq \frac{\omega_k(x_{k:1}^*, y)}{\sum_{j=1}^{N} \omega_j(x_{j:1}^*, y)}. \tag{13}$$

6. Set $x_{t+1} = y_k$ with probability

$$\alpha = \min\left[1, \frac{p(y)\pi_1(x_1^*|y)\pi_2(x_2^*|y, x_1^*)\cdots\pi_k(x_k^*|y, x_1^*, \ldots, x_{k-1}^*)}{p(x)\pi_1(y_1|x)\pi_2(y_2|x, y_1)\cdots\pi_k(y_k|x, y_1, \ldots, y_{k-1})}\frac{\bar{W}_x}{\bar{W}_y}\right]. \tag{14}$$

We can rewrite it in a more compact form as

$$\alpha = \min\left[1, \frac{p(y)q_k(x_{1:k}^*|y)}{p(x)q_k(y_{1:k}|x)}\frac{\bar{W}_x}{\bar{W}_y}\right], \tag{15}$$

where we recall

$$q_k(y_{1:k}|x) = \pi_1(y_1|x)\prod_{j=2}^{k}\pi_j(y_j|x, y_{1:j-1}) \tag{16}$$

where k is the index of the chosen sample y_k.

Otherwise, set $x_{t+1} = x$ with probability $1 - \alpha$.

7. Set $t = t + 1$ and repeat from step 1.

We emphasize that in the algorithm above we have not specifically defined the weight functions.

3.1. *Examples of weight functions*

The weight functions must to be bounded and positive. The choice can depend on some criteria such as improving performance or reducing computational complexity. If the target density is bounded, two possibilities are

$$\omega_j(z_1, \ldots, z_{j+1}) = p(z_1), \tag{17}$$

or

$$\omega_j(z_1, \ldots, z_{j+1}) = p(z_1)p(z_2)\cdots p(z_{j+1}), \tag{18}$$

with $j = 1, \ldots, N$. Other possible choices are the following

$$\omega_j(z_1, \ldots, z_{j+1}) = \left[\frac{p(z_1)}{q_j(z_{1:j}|z_{j+1})}\right]^{\theta}, \tag{19}$$

where $\theta > 0$ is a positive constant, or

$$\omega_j(z_1, \ldots, z_{j+1}) = \frac{p(z_j)}{q_1(z_j|z_{j+1})}\frac{p(z_{j-1})}{q_2(z_{j-1:j}|z_{j+1})}\cdots\frac{p(z_1)}{q_j(z_{1:j}|z_{j+1})}, \tag{20}$$

and a third possible choice

$$\omega_j(z_1, \ldots, z_{j+1}) = \frac{p(z_1)}{\pi_{j+1}(z_1|z_{j+1:2})}, \tag{21}$$

where $\pi_{j+1}(z_1|z_{j+1:2})$ is the $j+1$-th proposal pdf used in step 1 of the algorithm. It is important to remark that the z-variables are ordered such that z_1 is the most recently generated sample, z_j is the first drawn sample, and z_{j+1} represents the *previous step* of the chain.

Clearly, owing to the great flexibility in the construction of the weight functions, it can be difficult to assert which is the best choice in terms of the performances of the algorithm. However, evidently, in general, including more statistical information in the weights can improve performance yet, at the same time, increases the computational cost of the designed technique.

More specific theoretical or empirical studies are needed to clear up this issue. Indeed, observe that the point of the best selection of the weights is even unclear in the classical MTM by Liu et al. (2000), as for the method in Pandolfi et al. (2010), for instance.

3.2. *Relationship with the independent multiple tries scheme*

In Pandolfi et al. (2010) i.i.d. candidates are proposed at each time step. The acceptance probability α in Eqs. (14)–(15) may appear similar to the acceptance probability in Pandolfi et al. (2010). However, note that the expression of α in Eq. (15) is different to the acceptance probability in Pandolfi et al. (2010) for two main reasons:

(a) the first factor $\frac{p(y)q_k(x_{1:k}^*|y)}{p(x)q_k(y_{1:k}|x)}$ is distinct (see Eqs. (16)), and

(b) the definition and computation of $\bar{W}_x$ and $\bar{W}_y$ (see Eqs. (10) and (13)) are also different since here the weight functions take into account the previous generated samples (in the same time step).

If here we set $\pi_j(y_j|x, y_{1:j-1}) = \pi(y_j|x)$ for all $j = 1, \ldots, N$, then the steps of our algorithm coincide exactly with those of the technique in Pandolfi et al. (2010) *except for* step 4. Indeed, the procedures of choosing the "reference" points are different in the two methods (in our case, some of them are fixed while in Pandolfi et al. (2010) all the reference points are chosen randomly). We can find a specular difference between the methods in Liu et al. (2000) and Qin and Liu (2001).

3.3. Multi-point Metropolis as specific case

In the case when the weight functions are chosen as in Eq. (6), i.e., $\omega_k(z_1, \ldots, z_{j+1}) = p(z_1)q_j(z_{2:j+1}|z_1)\lambda_j(z_1, \ldots, z_{j+1})$, where

$$\lambda_j(z_1, z_{2:j+1}) = \lambda_j(z_{j+1:2}, z_1), \tag{22}$$

is sequentially symmetric, then our scheme coincides exactly with the standard multi-point Metropolis method in Qin and Liu (2001). Indeed, first of all we can rewrite the expression (15) as

$$\alpha = \min\left[1, \frac{p(y)q_k(x^*_{1:k}|y)}{p(x)q_k(y_{1:k}|x)}\frac{\omega_k(x^*_{k:1}, y)}{\omega_k(y_{k:1}, x)}\frac{\sum_{j=1}^{N}\omega_j(y_{j:1}, x)}{\sum_{j=1}^{N}\omega_j(x^*_{j:1}, y)}\right]. \tag{23}$$

Then, recalling the Eq. (11), i.e., $x^*_1 = y_{k-1}, x^*_2 = y_{k-2}, \ldots, x^*_{k-1} = y_1, x^*_k = x$ and $y = y_k$, the two weights $\omega_k(x^*_{k:1}, y)$ and $\omega_k(y_{k:1}, x)$ can be expressed exactly as

$$\begin{aligned}\omega_k(x^*_{k:1}, y) &= \omega_k(x^*_k = x, x^*_{k-1} = y_1, \ldots, x^*_1 = y_{k-1}, y = y_k)\\ &= p(x)q_k(y_{1:k}|x)\lambda_k(x, y_{1:k}),\end{aligned}$$

and

$$\begin{aligned}\omega_k(y_{k:1}, x) &= \omega_k(y_k = y, y_{k-1} = x^*_1, \ldots, y_1 = x^*_{k-1}, x = x^*_k)\\ &= p(y)q_k(x^*_{1:k}|y)\lambda_k(y, x^*_{1:k}),\end{aligned}$$

respectively. Therefore replacing the weights $\omega_k(x^*_{k:1}, y)$ and $\omega_k(y_{k:1}, x)$ in Eq. (23), we obtain

$$\alpha = \min\left[1, \frac{\lambda_k(x, y_{1:k})}{\lambda_k(y, x^*_{1:k})}\frac{\sum_{j=1}^{N}\omega_j(y_{j:1}, x)}{\sum_{j=1}^{N}\omega_j(x^*_{j:1}, y)}\right] = \min\left[1, \frac{\sum_{j=1}^{N}\omega_j(y_{j:1}, x)}{\sum_{j=1}^{N}\omega_j(x^*_{j:1}, y)}\right],$$

that coincides with acceptance probability in Eq. (9) of the standard multi-point Metropolis algorithm. Note that we have considered $\lambda_k(x, y_{1:k}) = \lambda_k(y, x^*_{1:k})$. Indeed, since $x^*_k = x$ we can write $\lambda_k(x, y_{1:k}) = \lambda_k(y, x^*_{1:k-1}, x)$, then because $x^*_{1:k-1} = y_{k-1:1}$, we obtain $\lambda_k(x, y_{1:k}) = \lambda_k(y, y_{k-1:1}, x)$, and as $y = y_k$, finally we have

$$\lambda_k(x, y_{1:k}) = \lambda_k(y_{k:1}, x),$$

that is exactly the condition assumed in Eq. (22). In the following, we show the proposed technique satisfies the detailed balance condition.

4. Proof of the detailed balance condition

To guarantee that a Markov chain generated by an MCMC method converges to the target distribution $p(x) \propto p_o(x)$, the kernel $A(y|x)$ of the corresponding algorithm fulfills the following detailed balance condition[1]

$$p(x)A(y|x) = p(y)A(x|y).$$

First of all, we have to find the kernel $A(y|x)$ of the algorithm, i.e., the conditional probability to move from x to y. For simplicity, we consider the case $x \neq y$ (case $x = y$ is trivial). The kernel can be expressed as

$$A(y = y_k|x) = \sum_{j=1}^{N} h(y = y_k|x, k = j), \tag{24}$$

[1] Note that the detailed balance condition is sufficient but not necessary condition. Namely, the detailed balance ensures invariance. The converse is not true. Markov chains that satisfy the detailed balance condition are called *reversible*.

where $h(y = y_k|x, k = j)$ is the probability of accepting $x_{t+1} = y_k$ given $x_t = x$ when the chosen sample y_k is the j-th candidate, i.e., when $y_k = y_j$.

In the sequel, we study just one $h(y = y_k|x, k)$ for a generic $k \in \{1, \dots, N\}$. Indeed, if $h(y = y_k|x, k)$ fulfills the detailed balance condition (it is symmetric w.r.t. x and y), then $A(y|x)$ also satisfies the detailed balance because it is a sum of symmetric functions. Therefore, we want to show that

$$p(x)h(y|x, k) = p(y)h(x|y, k),$$

for a generic $k \in \{1, \dots, N\}$. Following the steps above of the algorithm, we can write

$$p(x)h(y|x,k) = p(x) \int \cdots \int \left[\prod_{j=1}^{N} \pi_j(y_j|x, y_{1:j-1}) \right] \frac{\omega_k(y_{k:1}, x)}{\sum_{j=1}^{N} \omega_j(y_{j:1}, x)} \left[\prod_{i=k+1}^{N} \pi_i(x_i^*|y, x_{1:i-1}^*) \right]$$
$$\cdot \min\left[1, \frac{p(y)q_k(x_{1:k}^*|y)}{p(x)q_k(y_{1:k}|x)} \frac{\bar{W}_x}{\bar{W}_y}\right] dy_{1:k-1} dy_{k+1:N} dx_{k+1:N}^*.$$

Note that each factor inside the integral corresponds to a step of the method described in the previous section. The integral is over all auxiliary variables. Recalling the definition of the joint probability $q_k(y_{1:k}|x)$ and $\bar{W}_y$, the expression can be simplified to

$$p(x)h(y|x,k) = p(x) \int \cdots \int q_k(y_{1:k}|x) \cdot \left[\prod_{j=k+1}^{N} \pi_j(y_j|x, y_{1:j-1}) \right] \cdot \bar{W}_y \cdot \left[\prod_{i=k+1}^{N} \pi_i(x_i^*|y, x_{1:i-1}^*) \right]$$
$$\cdot \min\left[1, \frac{p(y)q_k(x_{1:k}^*|y)}{p(x)q_k(y_{1:k}|x)} \frac{\bar{W}_x}{\bar{W}_y}\right] dy_{1:k-1} dy_{k+1:N} dx_{k+1:N}^*,$$

and we only arrange it, obtaining

$$p(x)h(y|x,k) = \int \cdots \int p(x) q_k(y_{1:k}|x) \bar{W}_y \left[\prod_{j=k+1}^{N} \pi_j(y_j|x, y_{1:j-1}) \right] \left[\prod_{i=k+1}^{N} \pi_i(x_i^*|y, x_{1:i-1}^*) \right]$$
$$\cdot \min\left[1, \frac{p(y)q_k(x_{1:k}^*|y)}{p(x)q_k(y_{1:k}|x)} \frac{\bar{W}_x}{\bar{W}_y}\right] dy_{1:k-1} dy_{k+1:N} dx_{k+1:N}^*.$$

Now, we multiply the two members of the function $\min[\cdot, \cdot]$ by the factor $p(x)q_k(y_{1:k}|x)\bar{W}_y$ so that

$$p(x)h(y|x,k) = \int \cdots \int \left[\prod_{j=k+1}^{N} \pi_j(y_j|x, y_{1:j-1}) \right] \left[\prod_{i=k+1}^{N} \pi_i(x_i^*|y, x_{1:i-1}^*) \right]$$
$$\cdot \min\left[p(x)q_k(y_{1:k}|x)\bar{W}_y, p(y)q_k(x_{1:k}^*|y)\bar{W}_x\right] dy_{1:k-1} dy_{k+1:N} dx_{k+1:N}^*.$$

Therefore, it is straightforward that the expression above is symmetric in x and y. Indeed, we can exchange the notations of x and y, and x_i^* and y_j, respectively, and the expression does not vary. Then we can write

$$p(x)h(y|x, k) = p(y)h(x|y, k). \tag{25}$$

We can repeat the same development for each k obtaining

$$p(x)A(y|x) = p(y)A(x|y), \tag{26}$$

that is the detailed balance condition. Therefore, the generated Markov chain converges to our target pdf.

5. Toy example

Now we provide a simple numerical simulation to show an example of multi-point scheme with generic weight functions and compare it with the technique in Pandolfi et al. (2010). Let $X \in \mathbb{R}$ be a random variable[2] with bimodal pdf

$$p_o(x) \propto p(x) = \exp\left\{-(x^2 - 4)^2/4\right\}. \tag{27}$$

Our goal is to draw samples from $p_o(x)$ using our proposed multi-point technique.

[2] Note that we consider a scalar variable only to simplify the treatment. Clearly, all the considerations and algorithms are valid for multi-dimensional variables.

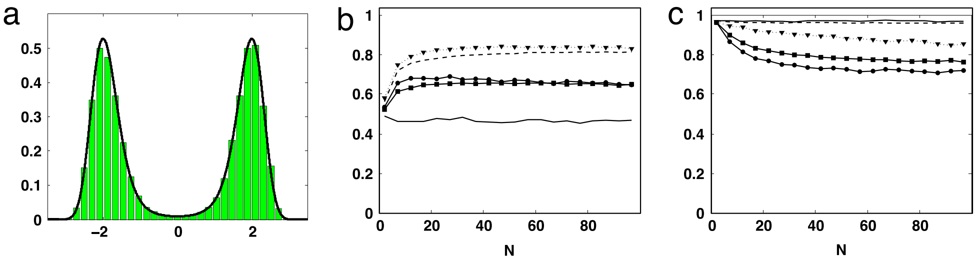

Fig. 1. (a) The target density $p_o(x)$ (solid line) and the normalized histogram of the samples generated using the proposed scheme and with $N = 10$. (b) The mean acceptance probability of jumping in a new state, depending on the number of tries N. We show the results of the technique in Pandolfi et al. (2010) (dashed line for weights $\omega_j^{(1)}$ and dotted line with triangles for $\omega_j^{(3)}$) and our method (squares for $\omega_j^{(1)}$, solid line with $\omega_j^{(2)}$, and with circles with $\omega_j^{(3)}$). (c) Estimated linear correlation coefficient depending on the number of tries N for the different techniques.

We consider Gaussian densities as proposal pdfs (a standard choice)

$$\pi_j(y_j|x_t, y_{1:j-1}) \propto \exp\left\{-\frac{(y_j-\mu_j)^2}{2\sigma^2}\right\} \tag{28}$$

where we use $\sigma^2 = 1$ and

$$\mu_j = \frac{\gamma_1}{i-1}(x_t + y_1 + \cdots + y_{i-2}) + \gamma_2 y_{i-1}, \tag{29}$$

i.e, μ is a weighted mean ($\gamma_1 + \gamma_2 = 1$) of the previous state x_t and the previous generated samples (at the same time step). Specifically, we set $\gamma_1 = 0.2$ and $\gamma_2 = 0.8$.

Moreover, we choose very simple weight functions depending only on first variable and on the target pdf

$$\omega_j^{(1)}(z_1, z_2, \ldots, z_{j+1}) = [p(z_1)]^{\theta}, \tag{30}$$

with $\theta = 1/2$. Note that $p(\cdot)$ is bounded and also positive (since it is a pdf). This kind of weight cannot be used in the multi-point scheme of Qin and Liu (2001), expect for $\theta = 1$ and using a specific sequence of the proposal pdfs. Moreover, for $\theta = 1$ this weight function can be also used in a standard MTM of Liu et al. (2000) if the chosen proposal density $\pi(y|x)$ is symmetric (i.e, $\pi(y|x) = \pi(x|y)$ and choosing $\lambda(x, y) = \frac{1}{\pi(x|y)}$).

We also compare the performances of the proposed algorithms with the weights as

$$\omega_j^{(2)}(z_1, \ldots, z_{j+1}) = p(z_1)p(z_2)\cdots p(z_{j+1}), \tag{31}$$

and

$$\omega_j^{(3)}(z_1, \ldots, z_{j+1}) = \frac{p(z_1)}{\pi_{j+1}(z_1|z_{j+1:2})}. \tag{32}$$

Then, we run the proposed multi-point algorithm with different numbers N of candidates and calculate the estimated acceptance rate (the averaged probability of accepting a movement) and linear correlation coefficient (between one state of the chain and the next). We also run the method in Pandolfi et al. (2010) with proposal pdf $\pi(y_j|x_t) \propto \exp\left\{-\frac{(y_j-x_t)^2}{2\sigma^2}\right\}$ and compare the performances, using weight functions as in Eq. (30) and the third type in Eq. (32). Because the samples are generated independently, we do not compare using weights in Eq. (31), as statistically this no longer makes sense.

Moreover, observe that in the scheme of Pandolfi et al. (2010) (where the candidates are drawn independently), the weight functions in Eq. (32) become $\omega^{(3)}(y_j, x_{t-1}) = \frac{p(y_j)}{\pi(y_j|x_{t-1})}$ where x_{t-1} is the previous step of the chain.[3]Note also that this particular choice of weights $\omega^{(3)}$ can be used in the standard MTM of Liu et al. (2000) (by choosing $\lambda(x, y) = \frac{1}{\pi(y|x)\pi(x|y)}$) and, in this case, the technique of Pandolfi et al. (2010) coincides with a standard MTM.

Fig. 1(a) depicts the target density $p_o(x)$ (solid line) and the normalized histogram of 100, 000 samples drawn using our proposed scheme and $N = 10$. Fig. 1(b)–(c) illustrate the mean acceptance probability and the estimated correlation coefficient (for different values of N and averaged using 5000 runs) of the two techniques and different choice of weights: our method is shown with squares using $\omega_j^{(1)}$, with solid line using $\omega_j^{(2)}$ and with circles using $\omega_j^{(3)}$. The performances of the

[3] Note that, in the expression of the weights $\omega^{(3)}(y_j, x_{t-1})$, we remove the subscript j because in Pandolfi et al. (2010) the analytic form of the weights is the same for each generated sample y_j, $j = 1, \ldots, N$.

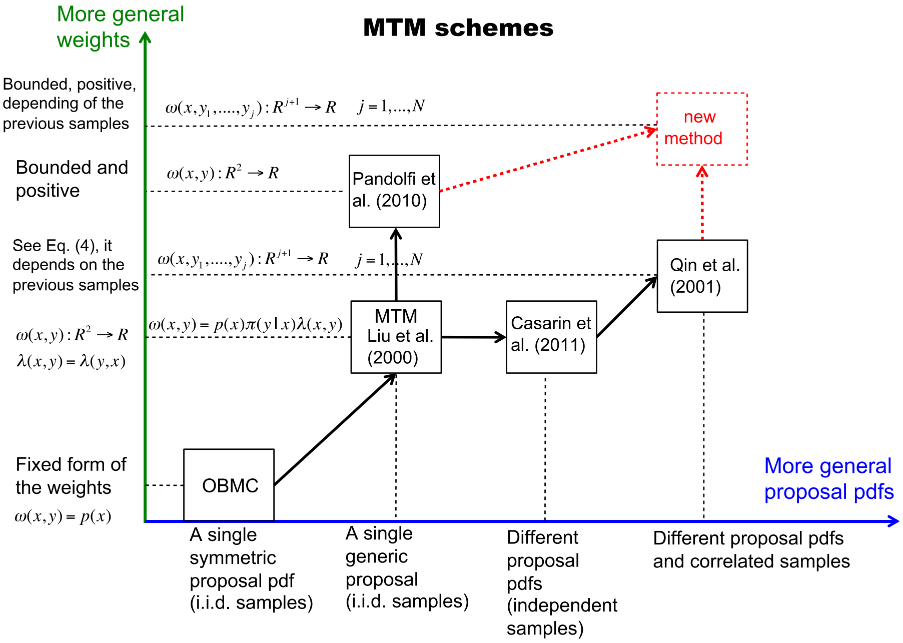
MTM schemes
More general weights
Bounded, positive, depending of the previous samples
$\omega(x,y_1,....,y_j): R^{j+1}\to R$ $j=1,...,N$
new method
Bounded and positive
$\omega(x,y): R^2\to R$
Pandolfi et al. (2010)
See Eq. (4), it depends on the previous samples
$\omega(x,y_1,....,y_j): R^{j+1}\to R$ $j=1,...,N$
Qin et al. (2001)
$\omega(x,y): R^2\to R$
$\lambda(x,y)=\lambda(y,x)$
$\omega(x,y)=p(x)\pi(y|x)\lambda(x,y)$
MTM Liu et al. (2000)
Casarin et al. (2011)
Fixed form of the weights
$\omega(x,y)=p(x)$
OBMC
More general proposal pdfs
A single symmetric proposal pdf (i.i.d. samples)
A single generic proposal (i.i.d. samples)
Different proposal pdfs (independent samples)
Different proposal pdfs and correlated samples

Fig. 2. Comparison of different MTM schemes in literature according to the flexibility in the choice of the proposal and weight functions. With the acronym OBMC we indicate the *orientational bias Monte Carlo* introduced by Frenkel and Smit (1996).

method in Pandolfi et al. (2010) are depicted with dashed line corresponding to the first choice $\omega_j^{(1)}$, and dotted line with triangles for $\omega_j^{(3)}$.

We can see although the proposed technique always attains smaller acceptance rates, the resulting correlations are always smaller than the correlations obtained by the other method, except using weights $\omega_j^{(2)}$ in Eq. (31). Moreover, the best results are obtained with the proposed technique using the weights $\omega_j^{(3)}$ in Eq. (32). In this case, the correlation decreases when N increases, up to 0.72 with $N = 100$.

6. Discussion

In this work, we have introduced a Metropolis scheme with multiple correlated points where the weight functions are not defined specifically, i.e., the analytic form can be chosen arbitrarily. We proved that our novel scheme satisfies the detailed balance condition.

Our approach draws from two different approaches (Pandolfi et al., 2010; Qin and Liu, 2001) to form a novel efficient and flexible multi-point scheme.

The multi-point approach with correlated samples provides different advantages over the standard MTM. For instance, the multi-point procedure can iteratively improve the proposal pdfs in two different ways. Firstly, since the proposal pdfs can be distinct, as in Casarin et al. (in press), it is possible to tune the parameters of each proposal in every time step. Secondly, since the candidates are generated sequentially, successive proposal pdfs can be improved learning from the previously produced samples during the same time step.

Moreover, in our technique, the only constraints of the weight functions are that they must be bounded and positive, unlike in the existing multi-point Metropolis algorithm (Qin and Liu, 2001) which is based on a specific definition of the weight functions. Here the weights can be chosen with respect to some criteria such as improving performance or reducing computational complexity. Thus our method avoids any control or check the existence of a suitable function λ and, therefore, the selection of the weight functions is broader and easier.

It is interesting to observe that, in general, the function λ may depend on the proposal pdf for a specific choice of weights and, in some cases, may entail certain constraints on the proposal pdf (such as that it be symmetric, for instance). An important consequence of this, it is that the weights can be chosen independently of the specific proposal pdf used in the algorithm. Namely, the proposal distribution and the weight functions can be selected separately, to fit well to the specific problem and to improve the performance of the technique. However, further theoretical or empirical studies are needed to determine the best choice of weight functions given a certain proposal and target density.

Furthermore, unlike in Pandolfi et al. (2010), in our method the weights can depend on the previous candidates, and the dimension of the weight functions grows from $\mathbb{R}^2$ to $\mathbb{R}^N$, thus being more general and potentially more powerful. Fig. 2 illustrates the relationships among different MTM schemes according to the flexibility in the choice of the proposal and weight functions. Finally, we have also shown a numerical simulation and a simple multi point scheme that provides good performances reducing the correlation in the produced chain.

Acknowledgments

We would like to thank the Reviewer for his comments which have helped us to improve the first version of manuscript. Moreover, this work has been partially supported by Ministerio de Ciencia e Innovación of Spain (project MONIN, ref. TEC-2006-13514-C02- 01/TCM, Program Consolider-Ingenio 2010, ref. CSD2008- 00010 COMONSENS, and Distribuited Learning Communication and Information Processing (DEIPRO) ref. TEC2009-14504-C02-01) and Comunidad Autonoma de Madrid (project PROMULTIDIS-CM, ref. S-0505/TIC/0233).

References

Andrieu, C., Moulines, E., 2006. On the ergodicity properties of some adaptive MCMC algorithms. The Annals of Applied Probability 16 (3), 1462–1505.

Casarin, R., Craiu, R., Leisen, F., 2011. Interacting multiple try algorithms with different proposal distributions. Statistics and Computing. http://dx.doi.org/10.1007/s11222-011-9301-9 (in press).

Frenkel, D., Smit, B., 1996. Understanding Molecular Simulation: From Algorithms to Applications. Academic Press, San Diego.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57 (1), 97–109.

Liang, F., Liu, C., Caroll, R., 2010. Advanced Markov Chain Monte Carlo Methods: Learning From Past Samples. In: Wiley Series in Computational Statistics, England.

Liu, J.S., 2004. Monte Carlo Strategies in Scientific Computing. Springer.

Liu, J.S., Liang, F., Wong, W.H., 2000. The multiple-try method and local optimization in metropolis sampling. Journal of the American Statistical Association 95 (449), 121–134.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Equations of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1091.

Pandolfi, S., Bartolucci, F., Friel, N., 2010. A generalization of the Multiple-try Metropolis algorithm for Bayesian estimation and model selection. Journal of Machine Learning Research 9, 581–588. (Workshop and Conference Proceedings Volume 9: AISTATS 2010).

Qin, Z.S., Liu, J.S., 2001. Multi-point Metropolis method with application to hybrid Monte Carlo. Journal of Computational Physics 172, 827–840.

Robert, C.P., Casella, G., 2004. Monte Carlo Statistical Methods. Springer.

Storvik, G., 2011. On the flexibility of Metropolis–Hastings acceptance probabilities in auxiliary variable proposal generation. Scandinavian Journal of Statistics 38 (2), 342–358.