ON MULTIPLE TRY SCHEMES AND THE PARTICLE METROPOLIS-HASTINGS ALGORITHM

Luca Martino[⋆]*, Jukka Corander*[⋆]

[⋆] Dep. of Mathematics and Statistics, University of Helsinki, 00014 Helsinki (Finland).

ABSTRACT

Markov Chain Monte Carlo (MCMC) methods are well-known Monte Carlo methodologies, widely used in different fields for statistical inference and stochastic optimization. The Multiple Try Metropolis (MTM) algorithm is an extension of the standard Metropolis-Hastings (MH) algorithm in which the next state of the chain is chosen among a set of candidates, according to certain weights. The Particle MH (PMH) algorithm is other advanced MCMC technique specifically designed for scenarios where the multidimensional target density can be easily factorized as multiplication of (lower - dimensional) conditional densities. Both are widely studied and applied in literature. In this note, we investigate similarities and differences among the MTM schemes and the PMH method.

Keywords: Markov chain Monte Carlo; Multiple Try Metropolis; Particle MCMC.

1. INTRODUCTION

Monte Carlo statistical methods are powerful tools for numerical inference and stochastic optimization [15]. 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 proposed pdf, usually to approximate an otherwise-incalculable (analytically) integral [7, 6]. MCMC algorithms produce a Markov chain with a stationary distribution that coincides with the target pdf.

The Metropolis-Hastings (MH) algorithm [12, 4] is the most famous MCMC technique. It can be applied to almost any target distribution but, in practice, the performance dependence crucially on the choice of the proposal pdf. In some cases, the Markov chain generated by the MH algorithm can remain trapped almost indefinitely in a local mode meaning that, in practice, convergence can be very slow.

The *Multiple-Try Metropolis* (MTM) method of [8], [7, Chapter 5] 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 is can explore a larger portion of the sample space in fewer iterations. A famous special case of MTM, well-known in molecular simulation field, is the *orientational bias Monte Carlo* technique [3].

Several generalizations of the basic MTM scheme [8] can be found in literature: with correlated candidates [10, 14], more general form of the weights and different frameworks [5, 11, 13, 18], with adaptive and interacting proposal pdfs [2, 9]. Interesting and related considerations about the use of multiple auxiliary variables for building acceptance probabilities within a MH approach can be found in [17].

Independently from the MTM schemes, *Particle MCMC methods* have been proposed [1] in literature. They are specially designed to solve inference problem in state space model applications. Here, we focus on the so-called *Particle (Independent) Metropolis-Hastings* (PMH) algorithm.

This work has been supported by the AoF grant 251170, and by the European Union 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861.

Table 1. **Batch Importance Sampling.**

1. Draw N samples, $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}$, from a (normalized) proposal pdf $q(\mathbf{x})$.
2. Compute the importance weights
$$w(\mathbf{x}^{(i)}) = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \quad i = 1, ..., N.$$
3. Normalize them
$$\bar{w}(\mathbf{x}^{(i)}) = \frac{w(\mathbf{x}^{(i)})}{\sum_{j=1}^{N} w(\mathbf{x}^{(j)})}, \quad i = 1, ..., N.$$
4. **Output:** A *weighted distribution*
$$\widehat{\pi}(d\mathbf{x}) = \sum_{i=1}^{N} \bar{w}(\mathbf{x}^{(i)}) \delta(d\mathbf{x} - \mathbf{x}^{(i)}), \tag{3}$$
is provided, where $\delta(d\mathbf{x} - \mathbf{x}^{(i)})$ denotes the Dirac mass at point $\mathbf{x}^{(i)}$. Under mild assumptions, $\widehat{\pi}$ is a consistent approximation of π, so that we can write
$$\widehat{I} = \sum_{i=1}^{N} \bar{w}(\mathbf{x}^{(i)}) f(\mathbf{x}^{(i)}) \approx I, \tag{4}$$
converges to I for $N \to +\infty$. Moreover, the normalizing constant Z can be also estimated as
$$\widehat{Z} = \frac{1}{N} \sum_{i=1}^{N} w(\mathbf{x}^{(i)}) \approx Z. \tag{5}$$

The authors in [1] also provide a short description of the relationships with other existing techniques. They mention and describe precisely the relationship with the so-called *configurational bias Monte Carlo* method [16],[7, Chapter 5], technique that is also strictly connected to the MTM scheme. They also allude quickly to the MTM method [8].

The relationship between MTM and PMH deserves a more careful look. This the aim of this work. We introduce a slight variant of a MTM technique with an independent proposal density. The structure of this algorithm coincides exactly with the PMH method although the mechanism of generation of the candidates is different, as discussed in the sequel.

2. BASIC BACKGROUND

In this section, we recall quickly basic material. Let $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ be the target density and $q(\mathbf{x})$ the proposal pdf, with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$. Without loss of generality, for sake of simplicity, we assume the proposal $q(\mathbf{x})$ be normalized. Consider we are interested in computing efficiently the integral
$$I = \int_{\mathcal{X}} f(\mathbf{x}) \bar{\pi}(\mathbf{x}) d\mathbf{x} = \frac{1}{Z} \int_{\mathcal{X}} f(\mathbf{x}) \pi(\mathbf{x}) d\mathbf{x}, \tag{1}$$
where
$$Z = \int_{\mathcal{X}} \pi(\mathbf{x}) d\mathbf{x}. \tag{2}$$
A classical Monte Carlo strategy is to apply the IS method, directly on the space $\mathcal{X} \subseteq \mathbb{R}^D$. See Table 1.

2.1. Sequential Importance Sampling (SIS) and Sequential Monte Carlo (SMC)

It is nontrivial to find a "good" proposal pdf for doing importance sampling in an high dimensional space $\mathbf{x} \in \mathbb{R}^D$. A efficient and one of the most useful strategy is to draw sequentially the proposed samples. Given a variable of

interest

$$\mathbf{x} = [x_1, ..., x_D]^\top = x_{1:D},$$

the corresponding target pdf $\bar{\pi}(\mathbf{x})$ can be always written as (using the chain rule)

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) = \pi_1(x_1)\pi_2(x_2|x_1)\cdots\pi_{D-1}(x_{D-1}|x_{1:D-2})\pi_D(x_D|x_{1:D-1}).$$

Note that, in some applications, the target is already decomposed, however in general one needs to marginalize several times the target $\bar{\pi}(\mathbf{x})$. We can also consider a proposal density decomposed in the same way

$$q(\mathbf{x}) = q_1(x_1)q_2(x_2|x_1)\cdots q_{D-1}(x_{D-1}|x_{1:D-2})q_D(x_D|x_{1:D-1}).$$

By definition, the importance weight is given by

$$w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})} = \frac{\pi_1(x_1)\pi_2(x_2|x_1)\cdots\pi_D(x_D|x_{1:D-1})}{q_1(x_1)q_2(x_2|x_1)\cdots q_D(x_D|x_{1:D-1})}.$$

The previous expression suggests a recursive procedure of computing the importance weights:

$$\begin{cases} w(x_1) = u_1 = \dfrac{\pi(x_1)}{q(x_1)}, \\ w(x_{1:d}) = w(x_{1:d-1})u_d = w(x_{1:d-1})\dfrac{\pi(x_d|x_{1:d-1})}{q(x_d|x_{1:d-1})}, \quad d = 2, \ldots, D, \end{cases} \tag{9}$$

where

$$u_d = \frac{\pi(x_d|x_{1:d-1})}{q(x_d|x_{1:d-1})}. \tag{10}$$

Thus, we can also write $w(x_{1:d}) = \prod_{j=1}^{d} u_j$. Then, we can easily calculate the sequence of weights $w(x_1), w(x_{1:2}), \ldots, w(x_{1:D})$, and recalling that $\mathbf{x} = x_{1:D}$, at the end of the recursion we have

$$w(\mathbf{x}) = \prod_{d=1}^{D} u_d.$$

Table 2 summarises the complete SIS algorithm.

Remark. *Note that, in this case, there are two possible estimators for approximating Z. The estimator $\widehat{Z}$ in Eq.* (5) *coincides exactly with those in Eq.* (7)*. The alternative estimator $\widetilde{Z}$ in Eq.* (8) *is derived in this way Denoting as*

$$Z_d = \int_{\mathcal{X}_d} \pi_d(x_d|x_{1:d-1}) \cdot \pi_2(x_2|x_1)\pi_1(x_1)dx_{1:d},$$

then, for the basic arguments in Eq. (5)*, we can write*

$$\widehat{\frac{Z_d}{Z_{d-1}}} = \frac{1}{N}\sum_{i=1}^{N} u_d^{(i)}. \tag{11}$$

Therefore, finally we obtain (considering $\hat{Z}_0 = 1$)

$$\widetilde{Z} = \widehat{Z_D} = \prod_{d=1}^{D} \widehat{\frac{Z_d}{Z_{d-1}}} = \frac{\widehat{Z_1}}{Z_0} \cdots \widehat{\frac{Z_{D-1}}{Z_{D-2}}}\widehat{\frac{Z_D}{Z_{D-1}}} = \prod_{d=1}^{D}\left[\frac{1}{N}\sum_{i=1}^{N} u_d^{(i)}\right], \tag{12}$$

Table 2. Sequential Importance Sampling (SIS).

1. Set $w(x_1^{(i)}) = u_1 = \frac{\pi_1(x_1^{(i)})}{q_1(x_1^{(i)})}$ for $i = 1, ..., N$.
2. For $d = 2, \ldots, D$:
   (a) **Propagate:** draw $x_d^{(i)}$ from $q(x_d|x_{1:d-1}^{(i)})$, $i = 1, \ldots, N$.
   (b) **Weigh:** compute recursively the importance weights

$$w(x_{1:d}^{(i)}) = w(x_{1:d-1}^{(i)}) \frac{\pi_d(x_d|x_{1:d-1}^{(i)})}{q_d(x_d|x_{1:d-1}^{(i)})}, \quad u_d^{(i)} = \frac{\pi_d(x_d|x_{1:d-1}^{(i)})}{q_d(x_d|x_{1:d-1}^{(i)})}. \tag{6}$$

3. Finally, we have the complete importance weights

$$w(\mathbf{x}^{(i)}) = w(x_{1:D}^{(i)}) = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})} = \prod_{d=1}^{D} u_d^{(i)}, \quad i = 1, ..., N.$$

4. Normalize them

$$\bar{w}(\mathbf{x}^{(i)}) = \frac{w(\mathbf{x}^{(i)})}{\sum_{j=1}^{N} w(\mathbf{x}^{(j)})}, \quad i = 1, ..., N.$$

5. **Output:** A *weighted distribution*

$$\widehat{\pi}(d\mathbf{x}) = \sum_{i=1}^{N} \bar{w}(\mathbf{x}^{(i)}) \delta(d\mathbf{x} - \mathbf{x}^{(i)}),$$

the estimator

$$\widehat{I} = \sum_{i=1}^{N} \bar{w}(\mathbf{x}^{(i)}) f(\mathbf{x}^{(i)}) \approx I,$$

and *two possible estimators* of the normalizing constant

$$\widehat{Z} = \frac{1}{N} \sum_{i=1}^{N} w(\mathbf{x}^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \left[\prod_{d=1}^{D} u_d^{(i)} \right] \approx Z. \tag{7}$$

and

$$\widetilde{Z} = \prod_{d=1}^{D} \left[\frac{1}{N} \sum_{i=1}^{N} u_d^{(i)} \right] \approx Z. \tag{8}$$

that is exactly the estimator in Eq. (8).

Sequential Monte Carlo (SMC) techniques [7, 15] combine this sequential construction of the importance weights in a IS estimation scheme, with *resampling* steps. A well-known examples are the so-called *particle filters* often used in state space model applications. Hence, a generic SMC technique is a SIS method using resampling schemes in step 2 of SIS. We show an example of in Table 3. In general, the resampling could be applied only in certain steps following some criterion. If the resampling is done every iteration then the method is often called *bootstrap filter*.

3. MULTIPLE TRY METROPOLIS (MTM) SCHEMES

The Multiple Try Metropolis (MTM) algorithm [8] is an MCMC technique, where N candidates are generated each iterations. According to certain weights, one candidate is chosen and accepted as new state with a certain probability α. The MTM steps with a generic proposal $q(\mathbf{x}|\mathbf{x}_{k-1})$, depending on the previous state, are summarized in Table 4. For $N = 1$, the MTM algorithm becomes the standard Metropolis-Hastings (MH) method. We have conisidered importance weights for favouring the comparison with the other techniques in this work. However, different kind of

Table 3. Step 2 of SIS for a Sequential Monte Carlo (SMC) method.

2. For $d = 2, \ldots, D$:
   (a) **Propagate:** draw $x_d^{(i)}$ from $q(x_d|x_{1:d-1}^{(i)})$, $i = 1, \ldots, N$.
   (b) **Weigh:** compute recursively the importance weights

$$w(x_{1:d}^{(i)}) = w(x_{1:d-1}^{(i)}) \frac{\pi_d(x_d|x_{1:d-1}^{(i)})}{q_d(x_d|x_{1:d-1}^{(i)})}, \quad u_d^{(i)} = \frac{\pi_d(x_d|x_{1:d-1}^{(i)})}{q_d(x_d|x_{1:d-1}^{(i)})}. \tag{13}$$

   (c) **Resample:** draw N samples from

$$\widehat{\pi}_d(dx_{1:d}) = \frac{1}{\sum_{i=1}^N w(x_{1:d}^{(i)})} \sum_{i=1}^N w(x_{1:d}^{(i)}) \delta(dx_{1:d} - x_{1:d}^{(i)}),$$

   Note that this is equivalent to sample N times a multinomial pmf with probabilities $\frac{w(x_{1:d}^{(i)})}{\sum_{i=1}^N w(x_{1:d}^{(i)})}$.

Table 4. General MTM algorithm.

1. Set $k = 1$, choose a initial state $\mathbf{x}_0$ and a value K.
2. Draw N samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x}|\mathbf{x}_{k-1})$, $i = 1, \ldots, N$.
3. Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights

$$w(\mathbf{x}^{(i)}) = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)}|\mathbf{x}_{k-1})}, \quad i = 1, \ldots, N.$$

4. Draw $N-1$ "reference" samples $\mathbf{z}^{(j)} \sim q(\mathbf{x}|\mathbf{x}^*)$, $j = 1, \ldots, N-1$, and set $z^{(N)} = \mathbf{x}_{k-1}$.
5. Compute the importance weights also for the reference points,

$$w(\mathbf{z}^{(i)}) = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)}|\mathbf{x}^*)}, \quad i = 1, \ldots, N.$$

6. Set $\mathbf{x}_k = \mathbf{x}^*$ with probability

$$\alpha = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w(\mathbf{x}^{(i)})}{\sum_{i=1}^N w(\mathbf{z}^{(i)})},$$

   otherwise, with probability $1 - \alpha$, set $\mathbf{x}_k = \mathbf{x}_{k-1}$.
7. If $k < K$, then set $k = k + 1$ and repeat from step 2.

weights could be applied [8, 11].We have denoted $a \wedge b = \min[a, b]$. The MTM method generates a reversible chain that converges to $\bar{\pi}(\mathbf{x})$ [8, 11]. See Appendix A for the proof.

If the proposal pdf is independent from the previous state of the chain, i.e., $q(\mathbf{x})$, the algorithm can be simplified. indeed, the steps 4 and 5 can be removed in the MTM scheme. Namely, one does not need to generate the reference samples at step 4. Indeed, in this case, we could directly set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \ldots, N-1$. The simplified algorithm (I-MTM) is given in Table 4.

3.1. Alternative version of the I-MTM method (I-MTM2)

In this work, we notice that the I-MTM method can be designed in an alternative way. With a proposal pdf independent from the previous state, we have seen that we can set $\mathbf{z}^{(j)} = \mathbf{x}^{(j)}$, $j = 1, \ldots, N-1$, because each

Table 5. MTM with independent proposal (I-MTM).

1. Set $k = 1$, choose a initial state $\mathbf{x}_0$ and a value K.
2. Draw N samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x})$, $i = 1, \ldots, N$.
3. Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights

$$w(\mathbf{x}^{(i)}) = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \quad i = 1, \ldots, N.$$

4. Set $\mathbf{x}_k = \mathbf{x}^*$ with probability

$$\alpha = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w(\mathbf{x}^{(i)})}{\sum_{i=1}^N w(\mathbf{x}^{(i)}) - w(\mathbf{x}^*) + w(\mathbf{x}_{k-1})} = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w(\mathbf{x}^{(i)})}{\sum_{i=1}^N w(\mathbf{z}^{(i)})},$$

where $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}\} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \setminus \{\mathbf{x}^*\} \cup \{\mathbf{x}_{k-1}\}$. Otherwise, with probability $1 - \alpha$, set $\mathbf{x}_k = \mathbf{x}_{k-1}$.

5. If $k < K$, then set $k = k + 1$ and repeat from step 2.

Table 6. Alternative I-MTM algorithm (I-MTM2).

1. Set $k = 1$, choose a initial vector $\mathbf{x}_0$, a value K and $\widehat{Z}_0 = \frac{1}{N}$.
2. Draw N samples from $\mathbf{x}^{(i)} \sim q(\mathbf{x})$, $i = 1, \ldots, N$.
3. Choose one sample $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the importance weights

$$w(\mathbf{x}^{(i)}) = \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}, \quad i = 1, \ldots, N.$$

4. Set $\mathbf{x}_k = \mathbf{x}^*$ and $\widehat{Z}_k = \frac{1}{N}\sum_{i=1}^N w(\mathbf{x}^{(i)})$ with probability

$$\alpha = 1 \quad \wedge \quad \frac{\frac{1}{N}\sum_{i=1}^N w(\mathbf{x}^{(i)})}{\widehat{Z}_{k-1}} = 1 \quad \wedge \quad \frac{\sum_{i=1}^N w(\mathbf{x}^{(i)})}{N\widehat{Z}_{k-1}},$$

otherwise, with probability $1 - \alpha$, set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widehat{Z}_k = \widehat{Z}_{k-1}$.

5. If $k < K$, then set $k = k + 1$ and repeat from step 2.

$\mathbf{x}^{(j)}$ is itself drawn from $q(\mathbf{x})$.

With the same arguments, we can also use the samples generated in the previous iteration of the algorithm as reference points, since all the samples are generated independently from the same proposal pdf. Namely, the alternative version of the I-MTM is summarized in Table 6.

4. PARTICLE METROPOLIS-HASTINGS (PMH) ALGORITHM

Consider again that the the target can be easily expressed in this form

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) = \pi_1(x_1)\pi_2(x_2|x_1)\cdots\pi_D(x_D|x_{1:D-1}).$$

The PMH method [1] is another MCMC technique proposed independently from the MTM algorithm, specifically designed for this framework. The complete description is provided in Table 7.

Table 7. Particle Metropolis-Hastings (PMH) algorithm.

1. Set $k = 1$, choose a initial state $\mathbf{x}_0$, a value K and $\widehat{Z}_0 = 1$.
2. Using a proposal pdf
$$q(\mathbf{x}) = q_1(x_1)q_2(x_2|x_1)\cdots q_D(x_D|x_{1:D-1}),$$
run a *sequential Monte Carlo* method (see Section **??**) with N particles. Thus, finally we obtain a collection of N weighted samples
$$\{x_{1:D}^{(i)}, w(x_{1:D}^{(i)})\}_{i=1}^N = \{\mathbf{x}^{(i)}, w(\mathbf{x}^{(i)})\}_{i=1}^N.$$
Namely, we have a weighted approximation of the target distribution
$$\widehat{\pi}(d\mathbf{x}) = \sum_{i=1}^N \bar{w}(\mathbf{x}^{(i)})\delta(d\mathbf{x} - \mathbf{x}^{(i)}).$$
and, moreover we also obtain
$$\widetilde{Z} = \prod_{d=1}^D \left[\frac{1}{N}\sum_{i=1}^N u_d^{(i)}\right] \approx Z$$
as in Eq. (12).
3. Draw $\mathbf{x}^* \sim \widehat{\pi}$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the weights $w_D^{(i)}$, $i = 1, ..., N$.
4. Set $\mathbf{x}_k = \mathbf{x}^*$ and $\widehat{Z}_k = \widehat{Z}^*$ with probability
$$\alpha = 1 \quad \wedge \quad \frac{\widetilde{Z}^*}{\widetilde{Z}_{k-1}},$$
otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$ and $\widetilde{Z}_k = \widetilde{Z}_{k-1}$.
5. If $k < K$, set $k = k + 1$ repeat from step 2.

5. RELATIONSHIP BETWEEN MTM AND PMH

A simple look at the alternative version of the MTM technique with independent proposal (I-MTM2), introduced in Section 3.1, and the PMH method, shows that are strictly related. Indeed, the structure of the two algorithms coincides. The links and differences are listed below:

- The main difference lies that the candidates in PMH are generated sequentially, using a SMC procedure. If the resampling steps in the SMC are not applied them I-MTM2 and PMH are *exactly* the same algorithm, where the candidates are drawn in a *batch* setting or *sequential* way. Namely, I-MTM2 generates directly $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_D^{(i)}]$ from $q(\mathbf{x})$ whereas PMH draws sequentially each component $x_d^{(i)}$ from $q_d(x_d|x_{1:d-1}^{(i)})$.

- Therefore, the resampling steps is the real difference between the generation procedures of PMH and I-MTM2. Owing to the use of the resampling, the candidates $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ proposed by PMH are not independent, differently from the MTM schemes considered in this work. Without resampling, the generated samples $\mathbf{x}^{(i)} = x_{1:D}^{(i)}$ would be independent as in I-MTM2.

- I-MTM2 uses the estimator $\widehat{Z}$ in Eqs. (5)-(7) whereas PMH uses the estimator $\widetilde{Z}$ in Eqs. (8)-(12). However, they are both estimators of the same unknown parameter: the normalizing constant Z.

- The PMH approach is preferable in high dimension, when the target can be factorized, since the use of the resampling steps provides a better proposal generation procedure.

Thus a novel possible version of PMH, equivalent to I-MTM, is given in Table 8.

Table 8. Alternative PMH.

1. Set $k = 1$, choose a initial state $\mathbf{x}_0$, a value K and $\widehat{Z}_0 = 1$.
2. Using a proposal pdf
$$q(\mathbf{x}) = q_1(x_1)q_2(x_2|x_1)\cdots q_D(x_D|x_{1:D-1}),$$
run a *sequential Monte Carlo* method (see Section **??**) with N particles. Thus, finally we obtain a collection of N weighted samples
$$\{x_{1:D}^{(i)}, w(x_{1:D}^{(i)})\}_{i=1}^N = \{\mathbf{x}^{(i)}, w(\mathbf{x}^{(i)})\}_{i=1}^N.$$
Namely, we have a weighted approximation of the target distribution
$$\widehat{\pi}(d\mathbf{x}) = \sum_{i=1}^N \bar{w}(\mathbf{x}^{(i)})\delta(d\mathbf{x} - \mathbf{x}^{(i)}).$$
and, moreover we also obtain
$$\widehat{Z} = \frac{1}{N}\sum_{i=1}^N \bar{w}(\mathbf{x}^{(i)}) \approx Z.$$
as in Eq. (12).
3. Draw $\mathbf{x}^* \sim \widehat{\pi}$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability proportional to the weights $w_D^{(i)}$, $i = 1, ..., N$.
4. Set $\mathbf{x}_k = \mathbf{x}^*$ with probability
$$\alpha = 1 \quad \wedge \quad \frac{N\widehat{Z}}{N\widehat{Z} - w(\mathbf{x}^*) + w(\mathbf{x}_{k-1})},$$
otherwise set $\mathbf{x}_k = \mathbf{x}_{k-1}$.
5. If $k < K$, set $k = k + 1$ repeat from step 2.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

[2] R. Casarin, R. Craiu, and F. Leisen. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 23(2):185–200, 2013.

[3] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*. Academic Press, San Diego, 1996.

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

[5] G. Kobayashi and H. Kozumi. Generalized multiple-point metropolis algorithms for approximate bayesian computation. *Journal of Statistical Computation and Simulation (DOI:10.1080/00949655.2013.836652)*, 2013.

[6] F. Liang, C. Liu, and R. Caroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics, England, 2010.

[7] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[8] J. S. Liu, F. Liang, and W. H. Wong. The Multiple-Try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, March 2000.

[9] L. Martino, R. Casarin, F. Leisen, and D. Luengo. Adaptive sticky generalized Metropolis. *arXiv:1308.3779*, 2013.

[10] L. Martino, V. P. Del Olmo, and J. Read. A multi-point Metropolis scheme with generic weight functions. *Statistics & Probability Letters*, 82(7):1445–1453, 2012.

[11] L. Martino and J. Read. On the flexibility of the design of multiple try metropolis schemes. *Computational Statistics (DOI 10.1007/s00180-013-0429-2)*, 2013.

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

[13] S. Pandolfi, F. Bartolucci, and N. Friel. A generalized multiple-try version of the reversible jump algorithm. *Computational Statistics & Data Analysis (available online)*, 2013.

[14] Z. S. Qin and J. S. Liu. Multi-Point Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 172:827–840, 2001.

[15] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[16] J. I. Siepmann and D. Frenkel. Configurational bias Monte Carlo: a new sampling scheme for flexible chains. *Molecular Physics*, 75(1):59–70, 1992.

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

[18] Y. Zhang and W. Zhang. Improved generic acceptance function for multi-point Metropolis algorithms. *2nd International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT-2012)*, 2012.

A. BALANCE CONDITION FOR MTM

The detailed balance condition [7, 15] is a sufficient condition to guarantee that a Markov chain has as invariant distribution exactly the pdf $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$. We have to prove that the kernel $A(\mathbf{y}|\mathbf{x})$ of the MTM method corresponding algorithm (probability of accepting a generated sample $\mathbf{y}$ given the previous state value $\mathbf{x}$) fulfills the following equation

$$\pi(\mathbf{x})A(\mathbf{y}|\mathbf{x}) = \pi(\mathbf{y})A(\mathbf{x}|\mathbf{y}).$$

We will consider $x \neq y$, since the case $x = y$ is trivial. The kernel (for $x \neq y$) can be expressed as

$$A(\mathbf{y} = \mathbf{y}^{(c)}|\mathbf{x}) = \sum_{i=1}^{N} h(\mathbf{y}^{(c)}|\mathbf{x}, c = i),$$

where $h(\mathbf{y}^{(c)}|\mathbf{x}, c = i)$ is the probability of accepting the new state $\mathbf{x}_{k+1} = \mathbf{y}^{(c)}$ given the previous one $\mathbf{x}_c = \mathbf{x}$, when the chosen sample $\mathbf{y} = \mathbf{y}^{(c)}$ is the c-th candidate. However, since the $\mathbf{y}^{(i)}$ are exchangeable (and also independent), for symmetry we have $h(\mathbf{y}^{(i)}|\mathbf{x}, i) = h(\mathbf{y}^{(j)}|x, j)\ \forall i, j \in \{1, ..., N\}$. Hence, we can also write

$$A(\mathbf{y}^{(c)}|\mathbf{x}) = N \cdot h(\mathbf{y}^{(c)}|\mathbf{x}, c),$$

where $c \in \{1, ..., N\}$ and we recall N is the total number of proposed candidates $\mathbf{y}^{(i)}$. Then, we need to show that

$$\pi(\mathbf{x})h(\mathbf{y}|\mathbf{x}, c) = \pi(\mathbf{y})h(\mathbf{x}|\mathbf{y}, c),$$

for a generic $c \in \{1, ..., N\}$. Following each step of the algorithm above, we can write

$$\begin{aligned}
&\pi(\mathbf{x})h(\mathbf{y}^{(c)}|\mathbf{x}, c) = \\
&\pi(\mathbf{x}) \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \left[\prod_{j=1}^{N} q(\mathbf{y}^{(j)}|\mathbf{x})\right] \frac{\omega(\mathbf{y}^{(c)})}{\sum_{i=1}^{N} \omega(\mathbf{y}^{(i)})} \left[\prod_{j=1}^{N-1} q(\mathbf{z}^{(j)}|\mathbf{y}^{(c)})\right] \cdot \\
&\min\left[1, \frac{\sum_{i=1}^{N} \omega(\mathbf{y}^{(i)})}{\sum_{i=1}^{N} \omega(\mathbf{z}^{(i)})}\right] \, dy_{1:c-1} dy_{c+1:N} dz_{1:N-1},
\end{aligned}$$

where $\omega(\mathbf{y}^{(i)}) = \frac{\pi(\mathbf{y}^{(i)})}{q(\mathbf{y}^{(i)})}$, $\omega(\mathbf{z}^{(i)}) = \frac{\pi(\mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)})}$ and recall that $\mathbf{z}^{(N)} = \mathbf{x}$. 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. We can rearrange easily the previous expression as

$$\begin{aligned}
&\pi(\mathbf{x})h(\mathbf{y}^{(c)}|\mathbf{x}, c) = \\
&\pi(\mathbf{x})q(\mathbf{y}^{(c)}|\mathbf{x})\omega(\mathbf{y}^{(c)}) \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \left[\prod_{j=1; j\neq c}^{N} q(\mathbf{y}^{(j)}|\mathbf{x})\right] \left[\prod_{j=1}^{N-1} q(\mathbf{z}^{(j)}|\mathbf{y}^{(c)})\right] \cdot \\
&\min\left[\frac{1}{\sum_{i=1}^{N} \omega(\mathbf{y}^{(i)})}, \frac{1}{\sum_{i=1}^{N} \omega(\mathbf{z}^{(i)})}\right] \, dy_{1:c-1} dy_{c+1:N} dz_{1:N-1},
\end{aligned}$$

and recalling that $\omega(\mathbf{y}^{(c)}) = \frac{\pi(\mathbf{y}^{(c)})}{q(\mathbf{y}^{(c)})}$ and replacing, we have

$$\begin{aligned}
&\pi(\mathbf{x})h(\mathbf{y}^{(c)}|\mathbf{x}, c) = \\
&\pi(\mathbf{x})\pi(\mathbf{y}^{(c)}) \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \left[\prod_{j=1; j\neq c}^{N} q(\mathbf{y}^{(j)}|\mathbf{x})\right] \left[\prod_{j=1}^{N-1} q(\mathbf{z}^{(j)}|\mathbf{y}^{(c)})\right] \cdot \\
&\min\left[\frac{1}{\sum_{i=1}^{N} \omega(\mathbf{y}^{(i)})}, \frac{1}{\sum_{i=1}^{N} \omega(\mathbf{z}^{(i)})}\right] \, dy_{1:c-1} dy_{c+1:N} dz_{1:N-1}.
\end{aligned} \tag{14}$$

Since $\mathbf{z}^{(N)} = \mathbf{x}$, it is straightforward to observe that the expression in Eq.(14) is symmetric in $\mathbf{x}$ and $\mathbf{y}^{(c)}$. Indeed, if we exchange $\mathbf{x}$ with $\mathbf{y}^{(c)}$, and all the $\mathbf{z}^{(i)}$ with $\mathbf{y}^{(i)}$, respectively, the expression does not vary. Thus we can write

$$\pi(\mathbf{x})h(\mathbf{y}^{(c)}|\mathbf{x}, c) = \pi(\mathbf{y}^{(c)})h(\mathbf{x}|\mathbf{y}^{(c)}, c).$$

Since we have assumed a generic c, finally we have $\pi(\mathbf{x})A(\mathbf{y}^{(c)}|\mathbf{x}) = \pi(\mathbf{y}^{(c)})A(\mathbf{x}|\mathbf{y}^{(c)})$, that is the balance condition.