Interacting Multiple Try Algorithms

Roberto Casarin, Radu Craiu, Fabrizio Leisen

April 23, 2012

1 Introduction

In the universe of Markov chain Monte Carlo (MCMC) algorithms, one of the most widely used class of algorithms is defined by the Metropolis-Hastings (MH). An important generalization of the standard MH formulation is represented by the multiple-try Metropolis (MTM) [7]. The MTM setup have been extended in many directions. [2] propose to use antithetic and quasi-Monte Carlo samples to generate the proposals and to improve the efficiency of the algorithm. [1] propose to use recent advances in population Monte Carlo and adaptive MCMC.

The class of population Monte Carlo procedures has been designed to address the inefficiency of classical MCMC samplers in complex applications involving multimodal and high dimensional target distributions. 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. This connects naturally with the class of adaptive MCMC samplers in which a chain's own path is used for tuning the transition kernel "on the go" (see [4]).

In the next sections we present the interacting MCMC sampling design for the MTM proposed in [1] and link the use of stochastic overrelaxation, random-ray Monte Carlo method (see [7]) and simulated annealing to IMTM.

2 Interacting Monte Carlo Chains for MTM

Suppose that of interest is sampling from a distribution π that has support in $\mathcal{Y} \subset \mathbb{R}$ and is known up to a normalizing constant. For defining the interacting MTM algorithm (IMTM) [1] consider a population $X^{(i)} = \{x_n^{(i)}\}_{n\in\mathbb{N}}$,

$i = 1, \dots, N$, of N chains. We assume that the ith chain has MTM transition kernel with M_i different proposals $T_j^{(i)}$, $j = 1, \dots, M_i$.

The interacting mechanism allows each proposal distribution to possibly depend on the values of the chains at the previous step. Formally, let $\Xi_n = \{x_n^{(i)}\}_{i=1}^N$ be the vector of values taken at iteration $n \in \mathbb{N}$ by the population of chains. Essentially, we allow each proposal distribution used in updating the population at iteration $n+1$ to depend on Ξ_n (see Algorithm 3.1). One expects that the chains in the population are spread throughout the sample space and thus the proposals generated are a good representation of the sample space $\mathcal{Y}$.

The transition kernel of the population of chains and the validity of the algorithm are given in [1]. For a discussion on the choice of the number of chains N, the number of proposals M, and of the function $\lambda_j^{(i)}(x, y)$ see [1].

3 Some generalizations

3.1 Stochastic Overrelaxation

Stochastic overrelaxation (SOR) is a MCMC technique developed for normal densities and subsequently extended for non-normal targets. The idea behind this approach is to induce negative correlation between consecutive draws of a single MCMC process.

Interacting Multiple Try Algorithm (IMTM)

- For $i = 1, \dots, N$
  1. Let $x = x_n^{(i)}$, for $j = 1, \dots, M_i$ draw $y_j \sim T_j^{(i)}(\cdot | x_n^{(1:i-1)}, x, x_n^{(i+1:N)})$ independently and compute
  $$w_j^{(i)}(y_j, x) = \pi(y_j) T_j^{(i)}(y_j | x_n^{(1:i-1)}, x, x_n^{(i+1:N)}) \lambda_j^{(i)}(y_j, x).$$
  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 $x_j^* \sim T_j^{(i)}(\cdot | x_n^{(1:i-1)}, y, x_n^{(i+1:N)})$, let $x_J^* = x_n^{(i)}$ and compute
  $$w_j^{(i)}(x_j^*, y) = \pi(x_j^*) T_j^{(i)}(x_j^* | x_n^{(1:i-1)}, y, x_n^{(i+1:N)}) \lambda_j^{(i)}(x_j^*, y).$$

4. Set $x_{n+1}^{(i)} = y$ with probability

$$\rho_i = \min\left\{1, \frac{w_1^{(i)}(y_1, x) + \ldots + w_{M_i}^{(i)}(y_{M_i}, x)}{w_1^{(i)}(x_1^*, y) + \ldots + w_{M_i}^{(i)}(x_{M_i}^*, y)}\right\}$$

and $x_{n+1}^{(i)} = x_n^{(i)}$ with probability $1 - \rho_i$.

Within the MTM algorithm we induce negative correlation between the proposals and the current state of the chain, x, by assuming that $(y_1, \ldots, y_{M-1}, x)^T \sim N_{d\times M}(\mathbf{0}, V)$ where V has a structure imposed by the desired negative dependence between the proposals $y_1, \ldots, y_n$'s and x, specifically

$$V = \begin{pmatrix} \Sigma_1 & \Psi_{12} & \ldots & \Psi_{1M} \\ \Psi_{12} & \Sigma_2 & \ldots & \Psi_{2M} \\ \ldots & \ldots & \ldots & \ldots \\ \Psi_{1M} & \Psi_{2M} & \ldots & \Sigma_M \end{pmatrix}.$$

One possible choice is to set $\Psi_{ij} = 0$ whenever $i, j \neq M$ and $\Psi_{iM} = \Sigma_i^{1/2} R_{iM} \Sigma_M^{1/2}$ where R_{iM} is a correlation matrix which corresponds to extreme negative correlation [**?**, see]for a discussion of extreme dependence]crameng2 between any two components (of same index) from y_i and x, for any $1 \leq i \leq M - 1$.

This falls within the context of dependent proposals as discussed by [2]. However, here we consider the case where the proposals and the current state are negatively correlated. This essentially ensures that no proposals are exceedingly close to the current location of the chain.

3.2 Multiple Random-ray Monte Carlo

The use of different proposals for the MTM algorithm allows also to extend the random-ray Monte Carlo method given in [7]. In particular the proposed algorithm allows to deal with multiple search directions at each iteration of the chains. At the n-th iteration of the chain, in order to update the set of chains Ξ_n, the algorithm performs for each chain $x_n^{(r)} \in \Xi_n$, with $r = 1, \ldots, N$, the following steps:

1. Evaluate the gradient $\log \pi(x)$ at $x_n^{(r)}$ and find the mode a_n along $x_n^{(r)} + r u_n$ where $u_n = x_n^{(r)} - x_{n-1}^{(r)}$.

2. Sample $I_1, \dots, I_M$ from the uniform $\mathcal{U}_{\{1,\dots,r-1,r+1,\dots,N\}}$.

3. Let $e_{n,j} = (a_n - x_n^{(I_j)})/||a_n - x_n^{(I_j)}||$ and sample r_j from $\mathcal{N}(0, \sigma^2)$.

and then use the set of proposals T_j which depends on $e_{n,j}$ to perform a MTM transition with different proposals as in the IMTM algorithm.

References

[1] Casarin, R., Craiu, R. and Leisen, F. (2012). Interacting multiple try algorithms with different proposal distributions. *Statist. Comput.*, forthcoming.

[2] Craiu, R. V. and Lemieux, C. (2007). Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling. *Statistics and Computing* **17** 109–120.

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

[4] Craiu, R. V., Rosenthal, J. S. and Yang, C. (2009). Learn from thy neighbor: Parallel-chain adaptive and regional MCMC. *Journal of the American Statistical Association* **104** 1454–1466.

[5] Haario, H., Saksman, E. and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli* **7** 223–242.

[7] Liu, J., Liang, F. and Wong, W. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association* **95** 121–134.