Stochastic Simulations

Autumn Semester 2019

Prof. Fabio Nobile *Assistant: Juan Pablo Madrigal Cianci*

Mini-project

submission deadline: 12 January 2020

The parallel tempering MCMC algorithm

1 Introduction and background

Perhaps the most commonly-used Markov chain Monte Carlo (MCMC) method is the random walk Metropolis (RWM) algorithm. However, despite its large applicability and widespread popularity, there are certain *undesirable* aspects of it. In particular, RWM tends to move around the equilibrium distribution in relatively small steps, with no tendency for the steps to proceed in the same direction, which means that, unfortunately, it can take a long time for the walker to explore all of the space. This can, in turn, become particularly damming when dealing with *difficult to explore* target distributions, such as those which are multi-modal or whose density concentrates around a non-linear lower-dimensional manifold; on the former case, if the modes of the distribution are far apart, the RWM would need to take large step sizes to be able to jump from mode to mode, which might cause a lot of rejections, ultimately producing samples that are largely correlated. On the latter case, the RWM algorithm will need to take very small step sizes to follow the curvature of the manifold, which will again produce samples with a large correlation. Ultimately, both cases result in the same conclusion: the chain will need to be run for a long time, to explore properly the whole distribution.

In recent years there has been an active development of computational techniques and algorithms to overcome these issues using a *tempering strategy*. Of particular importance to this mini-project is the parallel tempering (PT) algorithm [1, 2, 3] (also known as replica exchange), which finds its origins in the physics and molecular dynamics community. The general idea behind such methods is to simultaneously run K independent chains, each chain being invariant with respect to a *smoothed* (referred to as *tempered*) version of the posterior of interest μ, proposing to swap states between any two chains every so often. Such swap is then accepted using the standard Metropolis-Hastings (MH) acceptance-rejection rule. Intuitively, chains with a larger smoothing parameter (referred to as *temperature*) will typically be able to better explore the parameter space. Thus, by proposing to exchange states between chains targeting posteriors at different temperatures, it is possible for the chain of interest (i.e., the one targeting μ) to mix faster, and to avoid the undesirable behavior of some MCMC samplers, of getting "stuck" in a mode. In this mini project we will implement different variants of the parallel tempering algorithm, investigate their efficiency, and compare them to that of a simple RWM.

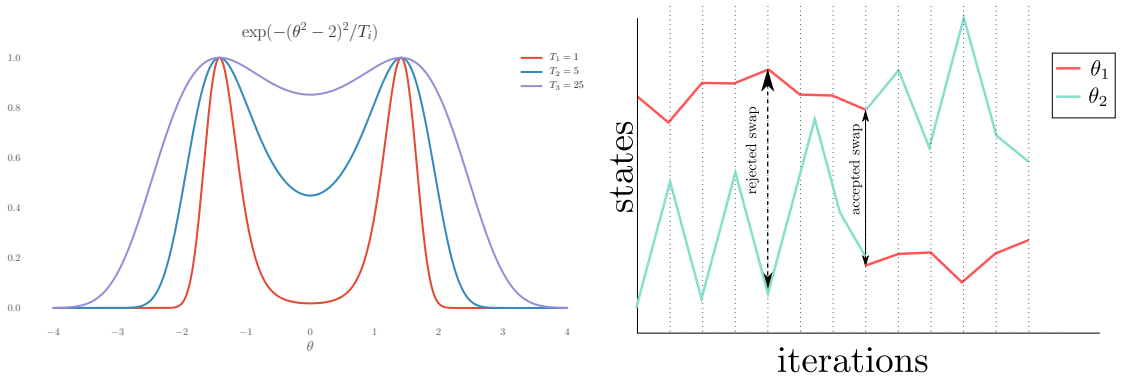

Figure 1: (**Left**) Tempered densities. As we can see, as $T \to \infty$ the density becomes *easier* to explore using RWM. (**Right**) Schematic of the parallel tempering algorithm.

1.1 The parallel tempering algorithm

Suppose that we are interested in sampling from a probability density[1] $\mu_1 : R^d \supset X \mapsto \mathbb{R}_+$ given by

$$\mu_1(\theta) = \frac{e^{-\Phi(\theta)}}{Z_1}, \quad Z_1 = \int_X e^{-\Phi(\theta)} \mathrm{d}\theta, \quad \theta \in X \tag{1}$$

for some positive function $\Phi : X \mapsto \mathbb{R}_+$ that we will call *potential*. In addition, let $\{T_i\}_{i=1}^K$ be a set of K *temperatures* such that $1 = T_1 < T_2 < \cdots < T_K \leq \infty$ and consider the following *tempered* target probability densities $\mu_i : X \mapsto \mathbb{R}_+$

$$\mu_i(\theta) := \frac{e^{-\Phi(\theta)/T_i}}{Z_i}, \quad Z_i := \int_X e^{-\Phi(\theta)/T_i} \mathrm{d}\theta \quad \theta \in X. \tag{2}$$

In the case where $T_K = \infty$, we set μ_K as the uniform density on X. Notice that μ_1 corresponds to the target posterior distribution that we are interested in.

We say that for $i = 2, \ldots, K$, each distribution μ_i is a *tempered* version of μ_1. In general, the $1/T_i$ term in (2) serves as a "smoothing" factor, which in turn makes μ_i easier to explore as $T_i \to \infty$, as seen on Figure 1 (left).

The idea behind tempering methods is to sample from all posterior distributions μ_i simultaneously using a μ_i-reversible Markov transition kernel p_i on each chain (where this kernel may correspond to, e.g., a RWM kernel), proposing to exchange states between chains targeting $\mu_i, \mu_j,\ j \neq i,\ i, j \in \{1, \ldots, K\}$ every certain number of steps N_s of the MH algorithm, and accepting such swap with the usual MH accept-reject rule. A depiction of such process is shown in Figure 1 (right). Carefully choosing how to do this swapping procedure can in turn result in a much more efficient sampling than the single-temperature counterpart. A pseudo-code for a simple implementation of the parallel tempering algorithm is presented in Algorithm 1. We remark that it is not obvious how to choose the temperatures, however, a *rule of thumb* is to choose $T_i = aT_{i-1}$ for some $a > 1$.

[1] On a slight abuse of notation, we are denoting density and distribution by the same symbol.

Algorithm 1 Simple parallel tempering.

function SIMPLE PARALLEL TEMPERING($N, \{p_i\}_{i=1}^N, \{\mu_i\}_{i=1}^N, \mu^0, N_s$)
  Sample $\theta_i^{(1)} \sim \mu^0,\ i = 1, \dots, K$.
  for $n = 1, 2, \dots, N-1$ **do**
    # Do one step of MH on each chain
    for $k = 1, \dots, K$ **do**
      sample $\theta_k^{(n+1)} \sim p_k(\theta_k^{(n)}, \cdot)$
    end for
    # Swapping step
    if $\mod(n, N_s) = 0$ **then**
      Sample $i \sim \{1, 2, \dots, K-1\}$.
      Swap states $\theta_i^{(n+1)}$ and $\theta_{i+1}^{(n+1)}$ with probability

$$\alpha_{\text{swap}}^{(n)} = \min\left\{1, \frac{\mu_{i+1}(\theta_i^{(n+1)})}{\mu_i(\theta_i^{(n+1)})} \frac{\mu_i(\theta_{i+1}^{n+1})}{\mu_{i+1}(\theta_{i+1}^{(n)})}\right\}$$

    end if
  end for
  Output $\{\theta_1^{(n)}\}_{n=1}^N$.
end function

2 Goals of the project

1. Show that the PT algorithm produces samples $\{\theta_1^{(i)}\}_{i=1}^N$ that are (asymptotically) distributed as μ_1.

2. We begin with a simple test case. Consider the potential function

$$\Phi(\theta) := \gamma(\theta^2 - 1)^2, \tag{3}$$

   where $\gamma \in \mathbb{R}_+$ is some positive parameter. Implement both a random walk Metropolis and a parallel tempering algorithm with $N_s = 1$ and different 4 temperatures to sample from $\mu_1(\theta) = \exp(-\Phi(\theta))/Z_1$, for $\gamma = 1, 2, 4, 8, 16$. Compare the results obtained using the standard MCMC diagnostic tools (auto-correlation plots, trace-plots, histograms of the samples, effective sample size, etc). Investigate the results obtained for different choices of temperature.

3. We now consider a more complicated target distribution. Let $X = [-2, 13] \times [-1, 3]$, and consider the potential given by

$$\Phi(\theta) = \log\left(\sum_{i=1}^{7} \exp\left(-\phi_i(\theta)\right)\right) \mathbb{I}_{[-2,13]\times[-1,3]}, \tag{4}$$

with

$$\phi_1(\theta) = -\frac{1}{2\sigma^2}\left((\theta_1 - 1.7)^2 + (\theta_2 - 1)^2 - 1.0)\right)^2 \mathbb{I}_{\{\{\theta_1 \leq 2.5\}},$$
$$\phi_2(\theta) = -\frac{1}{2\sigma^2}(\theta_2)^2 \mathbb{I}_{\{9 \leq \theta_1 < 11\}},$$
$$\phi_3(\theta) = -\frac{1}{2\sigma^2}(\theta_2 - 2)^2 \mathbb{I}_{\{[9 \leq \theta_1 < 11] \times [0 \leq \theta_2 < 2]\}},$$
$$\phi_4(\theta) = -\frac{1}{2\sigma^2}\left((\theta_1 - 7)^2 + (\theta_2 - 1)^2 - 1.0\right)^2,$$
$$\phi_5(\theta) = -\frac{1}{2\sigma^2}(\theta_2 + \theta_1 - 8)^2 \mathbb{I}_{\{7.7 < \theta_1 < 8.4\}},$$
$$\phi_6(\theta) = -\frac{1}{2\sigma^2}\left(\theta_2 - (m\theta_1 + 3.25)\right)^2 \mathbb{I}_{\{3.142 < \theta_1 < 4.86\}},$$
$$\phi_7(\theta) = -\frac{1}{2\sigma^2}\left((\theta_1 - 4)^2 + (\theta_2 - 1)^2 - 1\right)^2 \mathbb{I}_{\{[\theta_2 < 0.5] \cup [\theta_2 > 1.5]\}},$$

where $\sigma = 0.05$ and $m = -0.57$. We aim at sampling from $\mu_1(\theta) = \exp(-\Phi(\theta))/Z_1$. We remark that the difficulty in sampling from μ_1 arises from that fact that it is multi-modal, and its modes concentrate around a non-linear manifold. Repeat exercise 2 for this distribution and plot the density of the obtained samples. *Hint:* The target distribution has 4 well-separated modes.

4. We now consider two alternative formulations of PT. The first one, called *full parallel tempering*, considers sequential swaps between all chains. In this case, the swapping step in Algorithm 1 is replaced by Algorithm 2.

   The second one, called temperature-adaptive parallel tempering , adapts the temperatures at every iteration using one step of the Robbins-Monro algorithm as shown in Algorithm 3 to achieve a desired acceptance rate α^*. Here γ_n is called *the learning rate* and is chosen such that (i) $\sum_{n=0}^{\infty} \gamma_n = \infty$ and (ii) $\sum_{n=0}^{\infty} \gamma_n^2 < \infty$, and $L(T_i) := \log(T_{i+1} - T_i)$, for $i = 1, \ldots, K-1$. Notice that, clearly, $T_{k+1} = T_k + \exp(L(T_k))$.

   Implement these two variants of PT and repeat exercises 2 and 3 using 4 temperatures. For the adaptive PT case, consider $\gamma_n := \frac{0.6}{n}$ and take as initial log-temperatures $L(T_i) = 1,\ i = 1, 2, 3$ and $\alpha^* = 0.234$.

Algorithm 2 Full PT swapping step.

# Swapping step
for $i = 1 \ldots K-1$ **do**
Swap states $\theta_i^{(n+1)}$ and $\theta_{i+1}^{(n+1)}$ with probability

$$\alpha_{\text{swap}}^{(n)} = \min\left\{1, \frac{\mu_{i+1}(\theta_i^{(n+1)})}{\mu_i(\theta_i^{(n+1)})} \frac{\mu_i(\theta_{i+1}^{n+1})}{\mu_{i+1}(\theta_{i+1}^{(n)})}\right\}$$

end for

Algorithm 3 Adaptive PT.

Swapping and adaptive step

Sample $i \sim 1, 2, \ldots, K-1$.

Swap states $\theta_i^{(n+1)}$ and $\theta_{i+1}^{(n+1)}$ with probability

$$\alpha_{\text{swap}}^{(n)} = \min\left\{1, \frac{\mu_{i+1}(\theta_i^{(n+1)})}{\mu_i(\theta_i^{(n+1)})} \frac{\mu_i(\theta_{i+1}^{n+1})}{\mu_{i+1}(\theta_{i+1}^{(n)})}\right\}$$

Robbins Monro update of temperatures: Set

$(L(T_i))_n \leftarrow (L(T_i))_n + \gamma_n(\alpha_{\text{swap}}^{(n)} - \alpha^*)$.

References

[1] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[2] David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[3] Błażej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013.