Lookahead for Parallelizing Exactly Convergent MCMC

Gregor Robinson

13 July 2016

1 Introduction

Exactly convergent Markov Chain Monte Carlo (MCMC) methods are attractive for their theoretical foundation and easy intution for those practiced in Markov processes or statistical physics. MCMC typically requires generating a great deal of samples from the distribution of interest. Because exactly convergent MCMC inherently draws samples in serial, the walltime required can preclude its application to processes that require great computational effort to generate proposals. Many practitioners therefore resort to approximate methods that lack distributional convergence guarantees, either to expedite the serial process or to parallelize the effort (e.g. with multiple interacting chains).

These notes present a way to parallelize MCMC, while maintaining the strong convergence guarantees of algorithms such as Metropolis-Hastings and Gibbs samplers. We will call this method 'MCMC lookahead'. Discussion will use the concepts and language of traditional Metropolis-Hastings, but can easily be extended to other sampling mechanisms that use a proposal/rejection scheme.

Lookahead is suitable for applications where the acceptance rate is low, the computational effort of generating a proposal is high, and the cost of communicating a proposal state to another computer is sufficiently low. These criteria must be considered in relation to each other. For example, the communication cost in terms of walltime must be weighed against the walltime required to generate a proposal.

2 Language, notation, and review of M-H algorithm

It is helpful to introduce language and notation by reviewing the Metropolis-Hastings algorithm. Given an initial state X_i, a proposal $X_{i+1}^{(j)}$ is drawn from a proposal distribution with density $g(X_{i+1}^{(j)}|X_i)$. The superscript emphasizes that the proposals occur in a specified order, which will be useful below. This proposal is accepted with probability density $A(X_{i+1}^{(j)}|X_i)$. The transition probability density defined in this way is thus given by $p(X_{i+1}^{(j)}|X_i) = g(X_{i+1}^{(j)}|X_i)A(X_{i+1}^{(j)}|X_i)$. To ensure detailed balance as a sufficient but

not necessary way to make the Markov process ergodic, one commonly imposes the condition

$$A(X_{i+1}^{(j)}|X_i) = \min\left(1, \frac{p(X_{i+1}^{(j)})}{p(X_i)} \frac{g(X_i|X_{i+1}^{(j)})}{g(X_{i+1}^{(j)}|X_i)}\right).$$

The ratio of probabilities $p(X_{i+1}^{(j)})/p(X_i)$ can be computed as a ratio of likelihoods, so a great deal of discussion here will refer to the likelihood function. Evaluating a likelihood for a model, particularly for models of dynamical systems, often involves running a computationally expensive process model based on given parameters. For that reason, the text here will often refer to the process model and its computational considerations.

3 Lookahead

Lookahead is the technique of exploring possible future states of the Markov chain before completely evaluating the likelihood of the present state. Consider the tree of possible proposals: the root is the present state of the Markov chain, its direct descendents are a sequence of proposals from that state, second degree descendents are each sequences of proposals that would be offered if their direct ancestor was accepted by the MCMC sampler, and so on.

To introduce the process of generating and walking this tree, consider just the direct descendents of the present state X_i. These descendents, $X_{i+1}^{(j)}, \quad j \in \mathbb{N}$, are independent and identically distributed draws from the proposal distribution around the present state. The chain should generally consider each proposal in a predefined sequence to preserve ergodicity, but that does not preclude one from starting to evaluate the likelihoods (and running a process model) in parallel for a number of possible proposals.

Just as it is valid to evaluate the likelihood of a number of future proposals that originate from the present state, it is also valid to initate the same lookahead process on children of possible proposals even before they are accepted as part of the Markov chain.

For example, suppose that we simultaneously initiate the process model for two proposals $X_{i+1}^{(1)}$ and $X_{i+1}^{(2)}$. We can also initiate the process model for two possible proposals $X_{i+1}^{(2,1)}$ and $X_{i+1}^{(2,2)}$ drawn from the proposal distribution around $X_{i+1}^{(2)}$ even before the process model and likelihood evaluation are complete for the proposal $X_{i+1}^{(1)}$ that is immediately under consideration for acceptance by the Markov chain.

4 Pruning

For certain problems, it may be useful to focus computational effort by pruning the lookahead tree. This is particularly of interest for problems that either (1) permit approximation of the likelihood function, or (2) exhibit substantially heterogeneous walltime in computing the likelihood function.

4.1 Interpolating the likelihood function

One possible choice in approximating the likelihood function is to interpolate it from known values. The lookahead tree can then be selectively traversed according to branches with a high probability of being accepted by the MCMC sampler, so that one spends computational resources running the full process model only on lookahead paths that are likely to matter.

A particularly attractive choice for interpolating a relatively high-dimensional likelihood function is based on weighted superposition of radial basis functions (RBF). Radial basis interpolation has a major advantage of being relatively insensitive to the location of nodes, permitting the interpolant to reuse the effort of running a process model and computing the likelihood at those points already investigated by the MCMC sampler.

The RBF interpolation scheme takes the form

$$L(\mathbf{x};\theta) \approx \sum_{k=1}^{N_{nodes}} w_k \phi\left(\|\theta_{\mathbf{i}} - \theta\|\right),$$

where each $\theta_{\mathbf{i}}$ are nodes on which the values $L(\mathbf{x};\theta)$ are know, and weights w_i are chosen so that the interpolant is idential to the actual function values at those nodes.

At a minimum, interpolation of the likelihood function in this manner requires a number of nodes comparable to the effective dimensionality of the target distribution. If the target distribution is rugged, many more nodes may be required. Therefore pruning the lookahead tree is only useful where (1) the number of MCMC proposals required for convergence is much larger than the effective dimension of the target distribution, and (2) it is substantially faster to solve a dense symmetric linear system of dimension suitable to attain fair approximation of the likelihood function.

In cases where interpolation of the likelihood function does not provide accurate results, it may still be possible to approximate a distribution of likelihoods faster than the full process model can be evaluated. This is a possible topic for future research.

4.2 Heterogeneous walltime

If the walltime required to run a process model substantially varies, it may be worth traversing the lookahead tree with parallel conditional Markov chains. Each conditional Markov chain would assume that its originating state, a proposal from the actual present state, will be accepted into the primary chain. Upon completing process models for the proposals from the actual present state, they are evaluated for acceptance as usual. Once a proposal is accepted, it is then possble to immediately "fast forward" to wherever the corresponding conditional chain has arrived (discarding the other conditional chains).