Distributed Markov chain Monte Carlo

Lawrence Murray
CSIRO Mathematics, Informatics and Statistics
Perth, Western Australia
lawrence.murray@csiro.au

Abstract

We consider the design of Markov chain Monte Carlo (MCMC) methods for large-scale, distributed, heterogeneous compute facilities, with a focus on synthesising sample sets across multiple runs performed in parallel. While theory suggests that many independent Markov chains may be run and their samples pooled, the well-known practical problem of *quasi-ergodicity*, or poor *mixing*, frustrates this otherwise simple approach. Furthermore, without some mechanism for hastening the convergence of individual chains, overall speedup from parallelism is limited by the portion of each chain to be discarded as burn-in. Existing multiple-chain methods, such as parallel tempering and population MCMC, use a synchronous exchange of samples to expedite convergence. This work instead proposes mixing in an additional independent proposal, representing some hitherto best estimate or summary of the posterior, and cooperatively adapting this across chains. Such adaptation can be asynchronous, increases the ensemble's robustness to quasi-ergodic behaviour in constituent chains, and may improve overall tolerance to fault.

1 Introduction

We are interested in sampling some target distribution $p(X)$. Unable to draw samples from it directly, we instead develop a Markov chain $X_0, X_1, \ldots$ that, through some clever design of transition density $u(X_{t+1} \mid X_t)$, converges to $p(X)$ at equilibrium. Initialising the chain at $X_0 = \mathbf{x}_0$ by drawing from some *starting distribution* $p(X_0)$, it is simulated via repeated draws of $u(\cdot)$ until it has converged to stationarity after, say, T iterations. At this point, $X_T = \mathbf{x}_T$ is taken as an independent sample of $p(X)$, and the process repeated to draw additional samples.

This is the basic, "many short runs" form of the Markov chain Monte Carlo (MCMC) method. The period of simulation to equilibrium, commonly referred to as *burn-in*, is handled somewhat inefficiently in this approach, being performed for each sample. Realising that $\mathbf{x}_T$ is itself a sample of the equilibrium distribution, the second chain may be initialised there, and simulated for a much briefer period to diminish autocorrelations before the next sample is drawn. This leads to the "one long run" approach, and the popular adoption of a relatively lengthy burn-in followed by sampling at much shorter, regularly spaced intervals to deliver approximately independent samples [9].

There is no immediately obvious challenge to parallelising either approach: each of C chains, indexed by i, can be initialised and burned-in independently for T_i steps before one or more samples are drawn at intervals from each. As $C \to \infty$ and all $T_i \to \infty$, the ensemble is ergodic to $p(X)$ [9].

Unfortunately, rarely are chains of theoretical merit so ideally ergodic in practice. Within a finite, computationally-tractable number of steps, chains may be confined to isolated modes or mix poorly along strong correlations between variables (the latter a well known failure case of the Gibbs [8] sampler). This is referred to as poor *mixing*, or *quasi-ergodicity*, and implies that pooling samples from multiple runs will not necessarily give a fair representation of $p(X)$ any more than one run may. If several chains are quasi-ergodic to disparate modes, for instance, there is no guarantee that the weight attributed to each will be at all proportional to their relative likelihood mass, indeed, it will be more strongly influenced by the number of chains they confine. The initialisation of each chain potentially introduces additional bias, predisposing each to a subset of modes. As a result, care must be taken in the design of the starting

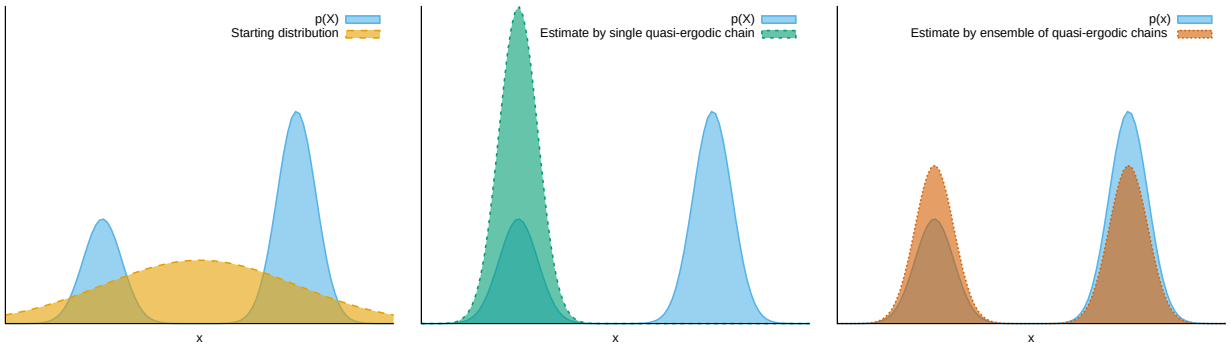

Figure 1: Two examples of quasi-ergodicity. $p(X)$ is the target distribution, consisting of two isolated modes; **(left)** the starting distribution, from which Markov chains are initialised; **(centre)** a typical example of initialising and running a single Markov chain that is quasi-ergodic, discovering only one of the modes; **(right)** the expected result of pooling samples from a collection of such chains: while both modes are detected, their weighting is determined by their equal basins of attraction [19] under the starting distribution rather than their likelihood mass.

distribution as well as the transition density [7]. Figure 1 gives a simple example of these problems of quasi-ergodicity for a univariate Gaussian mixture. For a more elaborate example using a double-well system, see Frantz et al. [6].

Whatever the strategy selected, Amdahl's law [1] bites: if some portion ρ of steps, $0 < \rho \leq 1$ and typically up to .5, must be removed as burn-in from each chain, the maximum clock-time speedup through parallelisation is limited to $1/\rho$. The performance gains of multiple chains might then be disappointing without some additional strategy to reduce ρ. Adaptation [11, 12, 2] and tempering [6, 9, 17, 10, 19] are both established means of achieving this for single chains[1], but note that the independent adaptation of each chain can deliver at most a constant factor improvement to overall speedup in the number a chains. A more successful strategy is one that reduces ρ *relative to the number of chains*, and indeed, the rate at which this occurs would be a critical assessor of such a method. Population MCMC [15] attempts this via an evolutionary [3] selection, crossing and mutation of multiple chains. Craiu et al. [5] more explicitly target the posterior with an ensemble of chains, using the covariance of samples across all chains to adapt the proposal covariance for a set of Metropolis-Hastings [18, 13] chains.

The approach adopted here attempts to construct a global best estimate of the posterior at any given step, and mixes this in as a *remote* component with whatever *local* proposal the chain has adopted. This does not preclude adaptation or tempering of that local proposal, admits strategies such as Gibbs, and indeed permits a heterogeneous blend of strategies across chains if so desired. Rather than trying to ensure that each chain mixes well, we attempt, via the remote proposal, to ensure that the ensemble to mixes well, in spite of any quasi-ergodic behaviour that might be apparent in individual chains due to the local proposals chosen for them.

The interaction of chains via proposals rather than by a direct exchange of samples seems a useful way forward to facilitate asynchronous communication. This applies to the work of Craiu et al. [5] also. Furthermore, as the failure of any chain might merely deprive the others of a timely proposal update, fault tolerance around this framework seems attainable. In contrast, failure of a chain in a parallel tempering [10] setup, for example, would require removal of the chain, potentially leaving too great a gap in temperature between newfound neighbouring chains for samples to cross between them; while certainly not insurmountable, a more elaborate strategy might be necessary.

In looking then toward a successful MCMC strategy in a large-scale, distributed computing context, we desire:

- a multiple-chain MCMC method robust to the quasi-ergodicity of individual chains,
- scaling of the burn-in proportion, ρ, relative to the number of chains,
- asynchronous communication between chains, and
- fault tolerance.

[1] In the case of tempering, this may be via the addition of chains ergodic to higher-temperature distributions, but note that typically only one remains ergodic to the target of interest.

2 Method

For each of C chains, indexed by i, and currently in state $\boldsymbol{\theta}_i$, let the proposed move of each chain, $\boldsymbol{\theta}'_i$, be drawn from:

$$q_i(\boldsymbol{\theta}'_i) := (1-\alpha) l_i(\boldsymbol{\theta}'_i|\boldsymbol{\theta}_i) + \alpha R_i(\boldsymbol{\theta}'_i)\,, \tag{1}$$

where α is a mixing portion ($0 \leq alpha < 1$), $l_i(\boldsymbol{\theta}'_i|\boldsymbol{\theta}_i)$ a *local* proposal and $R_i(\boldsymbol{\theta}'_i)$ a *remote* proposal formed, and adapted, according to the progress of other chains. The local proposal may be a Gibbs update, random walk or otherwise, and indeed need not be the same for all chains. We propose that the remote proposal is an aggregation over C components $r_j(\boldsymbol{\theta}_i)$, where the jth component is contributed by the jth chain, and typically expected to be that chain's best estimate of the posterior to date in some closed form that is readily communicable to other chains (e.g. a Gaussian mixture). The following aggregation to form $R_i(\cdot)$ is then suggested:

$$R_i(\boldsymbol{\theta}'_i) \propto \max_{j=1}^{C} r_j(\boldsymbol{\theta}'_i)\,. \tag{2}$$

The motivation here is to ensure that $R_i(\cdot)$ in any region is not influenced by the number of chains operating in the vicinity. Loosely speaking, if the regions explored by all chains are disjoint, this approximates an equally weighted mixture. In intersecting regions, only one chain contributes to the computation.

$R_i(\boldsymbol{\theta}'_i)$ is known only up to its normalising constant, but this is sufficient for acceptance ratio calculations with a Metropolis-Hastings criterion. Samples are readily drawn via rejection sampling with the mixture $(1/C)\sum_{j=1}^{C} r_j(\boldsymbol{\theta}'_i)$, noting that this is always greater than or equal to $(1/C)R_i(\boldsymbol{\theta}'_i)$.

Because the $R_i(\cdot)$ are dependent on past states of the C chains, the individual chains, and indeed the full ensemble, are no longer Markovian. Note, however, that as long as the adaptation of $R_i(\cdot)$ vanishes as the number of steps increases, suitable properties for convergence to the target distribution are conveyed [2].

An appropriate communication scheme must be designed to facilitate each chain updating its remote components with other chains. The implementation here uses asynchronous all-pairs point-to-point communication, with each chain i updating and sending its component $r_i(\cdot)$ at each step with some probability given by a parameter β. This will not scale well, but is sufficient for preliminary investigations here.

3 Experiments

The method is applied to parameter estimation in a nonlinear marine biogeochemical state-space model, consisting of 16 state variables and 16 parameters. The model is a stochastic Lotka-Volterra [16, 20] type differential model of phytoplankton and zooplankton interactions, with conserved nitrogen currency and climatic forcing. A simulated data set is used, with five of the 16 state variables observed at daily intervals over a period of one year.

MCMC is applied to estimation of the parameters, using a random-walk Metropolis-Hastings [18, 13] with orthogonal Gaussian proposal set to a scaled version of the prior covariance. At each step the likelihood of the parameters is computed using an unscented Kalman filter [14, 21] to marginalise out the state. Four methods are compared:

1. the base random walk,
2. the random walk adapted according to the method of Haario et al. [12],
3. the random walk mixed with $R_i(\cdot) \equiv r_i(\cdot)$, and
4. the random walk mixed with $R_i(\cdot) \equiv \max_{j=1}^{C} r_j(\cdot)$.

Note that no interaction between chains occurs in the first three cases. The third case isolates gains of the mixture proposal within single chains from cooperative adaptation of multiple chains. In all cases adaptation starts after 500 steps, and 25000 steps are taken in total. For mixture runs, $\alpha = 0.2$, $\beta = 0.1$, and $r_i(\cdot)$ at step t is simply the maximum likelihood Gaussian over the preceding $\lfloor t/2 \rfloor$ samples.

Figure 2 plots the evolution of the $\hat{R}^p$ statistic of Brooks and Gelman [4] across steps for each method. This is a measure of the convergence between multiple chains, approaching one as chains converge. Clearly the mixture strategy with sharing between chains delivers the fastest convergence rate. There is some apparent improvement in convergence in all methods as the number of chains increase, not just in the method with interactions between chains. This may be an artifact of increased sample size, or the $\hat{R}^p$ measure itself, and requires further investigation. It precludes, at this stage, an estimate of how well the strategy improves convergence as the number of chains increases.

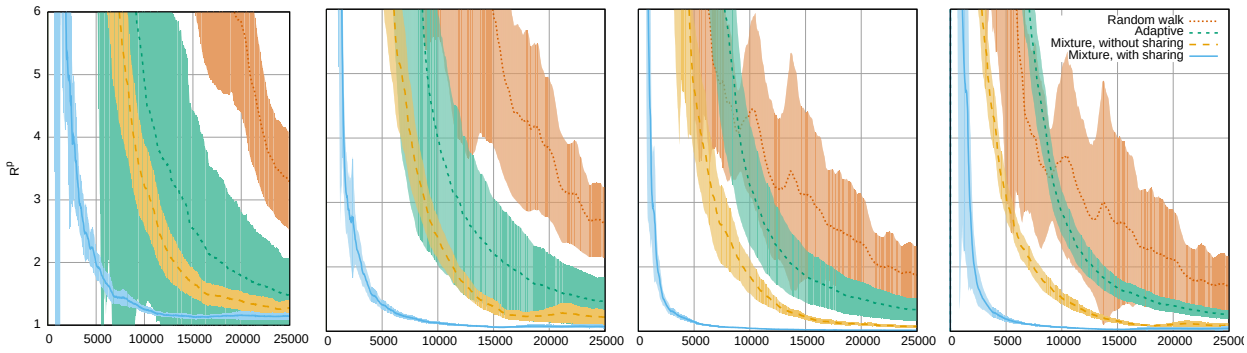

Figure 2: Evolution of the $\hat{R}^p$ statistic of Brooks and Gelman [4] across steps for each method, with **(left to right)** 2, 4, 8 and 16 chains. Lines indicate mean across 20 runs, and shaded areas a half standard deviation either side.

References

[1] G. Amdahl. Validity of the single processor approach to achieving large-scale computing capabilities. *AFIPS Conference Proceedings*, 30:483–485, 1967.

[2] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistical Computing*, 18:343–373, 2008.

[3] T. Back. *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[4] S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7:434–455, 1998.

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

[6] D. Frantz, D. Freeman, and J. Doll. Reducing quasi-ergodic behavior in Monte Carlo simulations by J-walking: Applications to atomic clusters. *Journal of Chemical Physics*, 93:2769–2784, 1990.

[7] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple simulations. *Statistical Science*, 7:457–511, 1992.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[9] C. Geyer. Markov chain Monte Carlo maximum likelihood. In E. Keramidas, editor, *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163, 1991.

[10] C. J. Geyer and E. A. Thompson. Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

[11] W. R. Gilks, G. O. Roberts, and S. K. Sahu. Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association*, 93:1045–1054, 1998.

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

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

[14] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*, 1997.

[15] K. B. Laskey and J. W. Myers. Population Markov chain Monte Carlo. *Machine Learning*, 50:175–196, 2003.

[16] A. Lotka. *Elements of physical biology*. Williams & Wilkins, 1925.

[17] E. Marinari and G. Parisi. Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, 19:451–458, 1992.

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[19] R. M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366, 1996.

[20] V. Volterra. *Animal Ecology*, chapter Variations and fluctuations of the number of individuals in animal species living together. McGraw-Hill, 1931. Translated from 1928 edition by R.N. Chapman.

[21] E. A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of IEEE Symposium on Adaptive Systems for Signal Processing Communications and Control*, pages 153–158, 2000.