Parallelization for Markov chains Monte Carlo with heterogeneous runtimes

Stanislas du Ché[1] – Charles C. Margossian[2]

[1] Ecole Normale Supérieure SACLAY France and Intern at ISERP Columbia University USA (stanislasduche@gmail.com)
[2] Center for Computational Mathematics, Flatiron Institute, USA; formally Department of Statistics, Columbia University, USA

Contributions

- **Even with parallelization, adding more chains can reduce the efficiency of MCMC, because we always wait for the slowest chain to finish.**
- **We propose instead to only wait for the fastest chains.**

Waiting for the slowest chains

Case study: ODE-based models

- Models in pharmacometrics integrate a scientific model inside a statistical model.
- The likelihood is parametrized by the solution to an ODE, motivated by a biological model [1].
- In this poster, we will focus on the Michaelis-Menten Model [2].
- **Depending on where we are in the parameter space (Figure 1.) Solving ODEs can be numerically challenging and take an excessive amount of time.**
- **This results in chains with wildly heterogeneous runtimes (Figure 2.), and a diminished return in efficiency when running many chains and waiting for the slowest chain to finish (Figure 3.).**

Details of the experiment

- We fit the model using Stan's Hamiltonian Monte Carlo, with 500 warmup and 500 sampling iterations. Convergence of the chains is evaluated using $\hat{R}$.
- The ODE is solved using a Runge-Kutta $4^{th}/5^{th}$ order integrator (RK 45). Using the more conservative backward differentiation (BDF) integrator leads to a slower average runtime but more stable behavior. RK45 is usually more efficient to solve non-stiff ODEs whereas BDF is faster for stiff ones. A stiff ODE is unstable numerically when a small perturbation is applied to its parameters.
- In this model we use **weakly informative priors** and **over-dispersed initializations.** Stronger priors (if available) and better initialization can stabilize the chains.

Observation

The probability of having a « lagging » chain increases with the total number of chains.

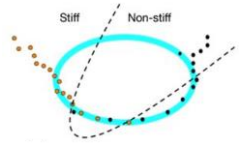

Figure 1. *[2] Evolution of 2 chains (orange and black dots) across the parameter space. The ODE behavior varies across the space, and it can be stiff or non-stiff depending on its location. The blue ellipsoid represents the sampling region, where the posterior probability mass concentrates.*

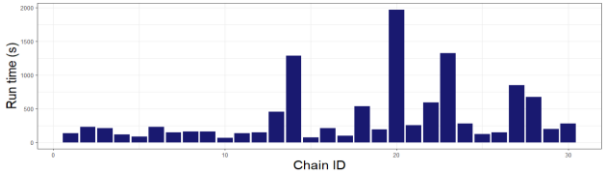

Figure 2. *Runtime of 30 Markov chains when fitting the Michaelis-Menten model in Stan [3]. A Runge-Kutta 4th/5th order integrator is used to solve the model's ODE and evaluate the likelihood. While some chains finish in **seconds**, others require **more than an hour**. Additional analysis finds that all chains are converging to the same distribution (Figure 5.).*

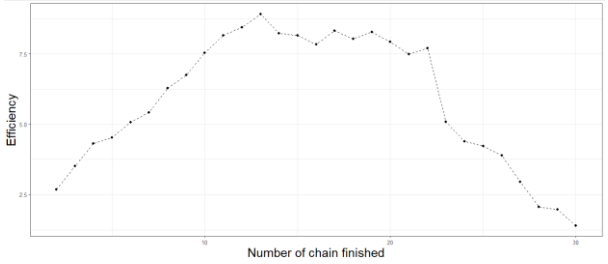

Figure 3. *Efficiency of MCMC as measured by ESS/s. Adding more chains improves the ESS. However, **waiting for the slowest chain to finish can reduce the overall ESS/s resulting in a lost in efficiency.** The maximum Efficiency is obtained after the 13 fastest chains finished, out of 30.*

Waiting for the fastest chains

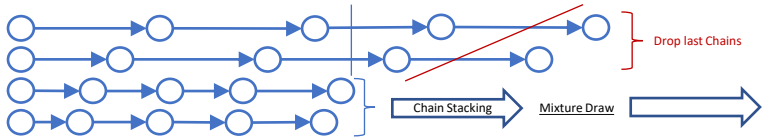

Figure 4. Racing-Chains MCMC. *We sample chains in parallel. Each time a chain finishes, we realize a mixture sampling draw via chain stacking [4]. We assess the ESS of this draw and check its reliability through a Pareto-k diagnostic [5]. Finally, if the draw is satisfying, we keep it and drop the other chains.*

	(a) Standard MCMC	(b) Standard MCMC	(c) Racing-chains	(d) Racing-chains + stacking
Chains	30	9	30 (9 kept)	9
Running time (s)	4918	775	**145**	**145**
Effective Sample Size	**2728**	1172	966	1107
Efficiency (ESS/s)	0.55	1.51	6.66	**7.63**

Table 1. *Comparison of sampling methods: (a) 30 chains in parallel. (b) Another draw launching 9 chains in parallel. (c) Keep only 9 fastest chains out of 30. (d) Keep only the fastest chains for which their stacking verifies ESS > ESS_{target} = 1000. Keeping only the fastest chains reduces computational time and increase the method's efficiency.*

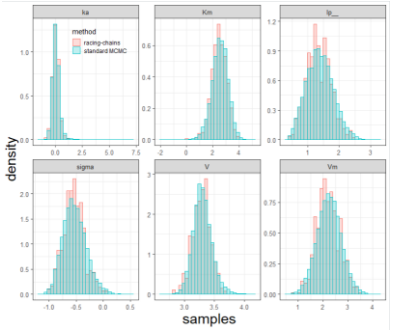

Figure 5. *Posterior distribution of the standard MCMC run for 30 chains in parallel and the racing-chains MCMC coming from the chain stacking of the fastest chains. The parameters drawn are those of the Michaelis-Menten, and "lp__" describes the unnormalized log posterior of the observations.*

Chain stacking [4] - principle

- The idea of chain stacking is to use parallel chains to explore as many modes as possible and then construct a stacked inference. The weights between the chains are found by maximizing the leave-one-out log predictive density (lpd_{loo}).
- From the sampling of the chains, we can build a mixture draw according to their respective weights and estimate its Effective Sample Size. Its convergence to posterior distribution is monitored by the stability of lpd_{loo} seen as a function of the number of chains in parallel.
- In our case, stacking is necessary as some chains may finish quickly but have low likelihood. A uniform draw will thus induce a bias. This is especially the case for very wide priors, and the sampler can not draw a Markov chain.

Discussion

- An open question is to understand what bias does dropping chains introduce? See the discussion by Rosenthal (2000) [6] on failing chains.
- In our example, slow chains get stuck in pathological regions of the parameter space with no probability mass and dropping these chains doesn't incur a bias (Figure 5); for another example, see [7]. Still, we can't expect this to always be true.
- Developing reliable convergence diagnostics and using unbiased Monte Carlo methods [8, 9, 10] may provide a solution.

References

[1] Flexible and efficient Bayesian pharmacometrics modeling using Stan and Torsten, Part I. C. Margossian, Y. Zhang and W. Gillespie. *CPT: Pharmacometrics & Systems Pharmacology*, 2022.
[2] Solving ODEs in a Bayesian context: challenges and opportunities. C. Margossian, L. Zhang, S. Weber and A. Gelman. Population Approach Group in Europe, 2021.
[3] Stan Development Team. YEAR. Stan Modeling Language Users Guide and Reference Manual, VERSION. https://mc-stan.org
[4] Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors. Yuling Yao, Aki Vehtari, and Andrew Gelman, 2020.
[5] Pareto Smoothed Importance Sampling. Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, Jonah Gabry,2022.
[6] Parallel computing and Monte Carlo algorithms. J. S. Rosenthal. Far East Journal of Theoretical Statistics, 4:207-236, 2000
[7] Bayesian model of planetary motion: exploring ideas for a modeling workflow when dealing with ordinary differential equations and multimodality. C. Margossian and A. Gelman. Stan Case Studies 7. 2020
[8] Annealed importance sampling. R. M. Neal. Statistics and Computing, 11:125 - 139, 2001
[9] Sequential Monte Carlo samplers. P. Del Moral, A. Doucet and A. Jasra. Journal of the Royal Statistical Society, Series B, 68:411-436, 2006
[10] Unbiased Markov chain Monte Carlo methods with couplings. P. E. Jacob, J. O'Leary and Y. F. Atchadé. Journal of the Royal Statistical Society, Series B, 82:543-600, 2020.