25th International Congress on Modelling and Simulation, Darwin, NT, Australia, 9 to 14 July 2023
mssanz.org.au/modsim2023

Adaptive MCMC parallelisation in Stan

T.N. Stenborg [a,b]

[a] ARC Training Centre in Data Analytics for Resources and Environments; [b] The University of Sydney
Email: travis.stenborg@sydney.edu.au

Abstract: Stan is a probabilistic programming language that uses Markov chain Monte Carlo (MCMC) sampling for Bayesian inference (Carpenter at el.). Stan sampling can be parallelised by running Markov chains m on separate processing cores n, i.e. ≥ 1 chain/core, for Amdahlian speedup (Annis et al.). An extension, introduced here, is adaptive parallelisation. First, prior to planned sampling, performance benchmarking was dynamically performed with $m = 4\ldots M$ chains distributed over $n = 1\ldots m$ cores (where M is a system's number of available cores, and using at least four chains is recommended (Vehtari et el.)). The best performing configuration (m, n) was then automatically adopted (github.com/tstenborg/Stan-Adaptive-Parallelisation).

To be relevant, benchmarking should proceed with the same data and compiled Stan model as the planned sampling. For efficiency, benchmarking was performed with fewer chain iterations than for inference proper, though using the same ratio of warmup to post-warmup iterations/chain ($1 : 1/m$, yielding an equal number of total draws per configuration). For further efficiency, comparison of only one evaluation of each configuration was made. One evaluation was deemed sufficient after measuring speedup variability, for an example problem and configuration near the middle of a test system's (Intel Core i7-10750H) non-hyperthreaded (m, n) configuration range. The simplifying assumption was made that results for the configuration were representative of the entire hyperthreaded and non-hyperthreaded range. Finally, for meaningful interconfiguration comparisons, a fixed seed was passed to the Stan random number generator.

Warmup iterations had a significant effect on optimum (m, n). Too few warmup iterations, though speeding up benchmarking, can leave Stan without enough adaptation time to determine efficient sampling parameters (Hecht et al.). That in turn, sometimes caused misoptimsation. Empirical testing suggested 80 warmup iterations as a minimum for adaptive parallelisation.

Figure 1 shows speedup as a function of (m, n) for the stan_model example built into RStan (top), and a test mixture model (23,315 members, bottom). Note the contrasting speedup for a small data, simple model (top) vs a large data, complex model (bottom). Stan MCMC with a naïve maximum (m, n) configuration wasn't always quickest. The fastest (m, n) was a function of model, data volume and warmup iterations.

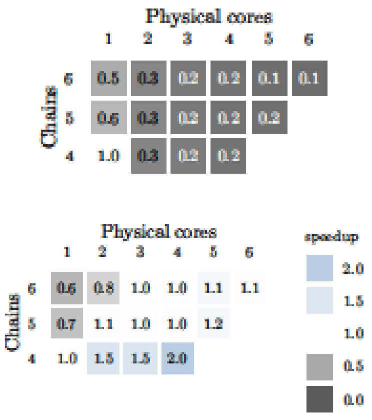

Figure 1. MCMC parallelisation and Stan model inference speedup. Simple model (top) vs complex model (bottom)

ACKNOWLEDGEMENTS

Co-funded by the ARC Training Centre in Data Analytics for Resources and Environments (project ICI9010031) and Australian National Health and Medical Research Council Ideas Grant GNT1186572.

REFERENCES

Annis J, Miller BJ, Palmeri TJ. 2017. Bayesian inference with Stan: A tutorial on adding custom distributions. Behav Res Methods. 49:863–86.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, et al. 2017. Stan: A Probabilistic Programming Language. J Stat Softw. 76(1):1–29.

Hecht M, Weirich S, Zitzmann S. 2021. Comparing the MCMC Efficiency of JAGS and Stan for the Multi-Level Intercept-Only Model in the Covariance- and Mean-Based and Classic Parametrization. Psych. 3(4):751–79.

Vehtari A, Gelman A, Simpson D, Carpenter B, Bürkner PC. 2021. Rank-Normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion). Bayesian Anal. 16(2):667–718.

Keywords: *Bayesian inference, Markov chain Monte Carlo sampling, parallel computing, Stan*