Global consensus Monte Carlo

Lewis Rendell [1], **Adam M. Johansen** [1], **Anthony Lee** [2], **Nick Whiteley** [2]
L.Rendell@warwick.ac.uk, A.M.Johansen@warwick.ac.uk, Anthony.Lee@bristol.ac.uk, Nick.Whiteley@bristol.ac.uk
[1] Department of Statistics, University of Warwick, UK [2] School of Mathematics, University of Bristol, UK

THEORY ◀

Introduction

For problems involving large data sets, it may be convenient or necessary to partition the data across multiple cores or machines.

Consider a target density given by

$$\pi(z) \propto \mu(z) \prod_{j=1}^{b} f_j(z)$$

where f_j is computable on the jth machine, and depends on y_j, the jth subset of the data. We wish to generate samples distributed according to π.

Existing approaches include:

- Scott et al. (2016), who propose running one MCMC chain on each computing node, with target densities proportional to $\mu(z)^{1/b} f_j(z)$. The samples are combined in a way that implicitly assumes approximate Gaussianity.
- Xu et al. (2014), who approximate each f_j by a density from an exponential family.

We propose a novel procedure, motivated by the global variable consensus optimisation algorithm of Boyd et al. (2011), itself based upon ideas of Bertsekas and Tsitsiklis (1989).

The instrumental model

We introduce an instrumental hierarchical model by associating a local 'proxy' variable x_j with each subset of the data, and introducing a top-level parameter λ *(see DAG, above right)*.

Specifically, on an extended state space we define

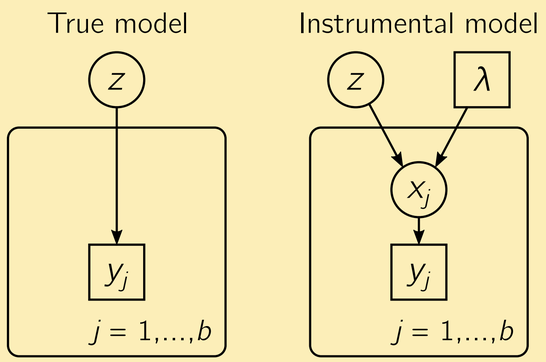
$$\pi_\lambda(z, x_{1:b}) \propto \mu(z) \prod_{j=1}^{b} K_\lambda(z, x_j) f_j(x_j),$$

where $\{K_\lambda : \lambda > 0\}$ is a family of Markov transition densities.

Assume that f_j is bounded, and that $\int K_\lambda(z, x) f_j(x) \mathrm{d}x \to f_j(z)$ pointwise as $\lambda \to 0$. Then the z-marginal of π_λ converges in total variation to π, so that for bounded functions φ,

$$\int \varphi(z) \pi_\lambda(z) \mathrm{d}z \to \int \varphi(z) \pi(z) \mathrm{d}z.$$

We look to generate samples distributed according to the artificial joint distribution π_λ; for λ sufficiently close to 0, the z-marginal forms an approximation to the true target distribution.



Gibbs sampler

For given λ, a π_λ-reversible Markov chain is obtained by considering the full conditional densities:

$$\pi_\lambda(z \mid x_{1:b}) \propto \mu(z) \prod_{j=1}^{b} K_\lambda(z, x_j),$$

$$\pi_\lambda(x_j \mid z) \propto K_\lambda(z, x_j) f_j(x_j).$$

A two-variable Gibbs sampler may be constructed, where the two variables are z and $x_{1:b}$. In distributed settings, the drawing of a new $x'_j \sim \pi_\lambda(\cdot \mid z)$ may occur on the jth computing node; the new values $x'_{1:b}$ may then be sent to a central node, in order to draw a new $z' \sim \pi_\lambda(\cdot \mid x'_{1:b})$.

SMC sampler

The parameter λ may be chosen to balance computational tractability with fidelity to the true model, in a form of bias–variance tradeoff. When λ is close to 0, MCMC chains targeting $\pi_\lambda(z, x_{1:b})$ may mix poorly.

To this end, one may approximate a sequence of distributions $\pi_{\lambda_0}, \pi_{\lambda_1}, \ldots$ using an SMC sampler. By choosing a decreasing sequence of values λ_p, such an approach is observed to produce low-variance estimators for small values of λ.

Work is ongoing on a procedure to specify λ in an automated manner, by adaptively specifying this sequence, and using SMC variance estimators to assess the performance of each λ value.

▶ EXAMPLES

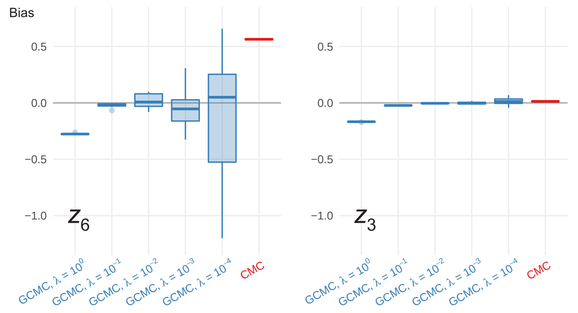
Binary regression

Binary logistic regression models are commonly used with A/B testing data – in web design for example, to determine which content choices lead to maximised user interaction (such as the user clicking on a link to a product for sale).

- Data set formed of responses $\eta_i \in \{-1, 1\}$ and vectors ξ_i of binary covariates (originally belonging to $\{0, 1\}^d$, but centred in a pre-processing step). The data are split into b subsets; $f_j(z) = \prod_i \sigma(\eta_i z^{\mathrm{T}} \xi_i)$, where the product is taken over those indices i included in the jth data subset, and σ is the logistic function.
- We use a zero-mean normal prior μ.
- For our algorithm, we use normal transition kernels: $K_\lambda(z, x) = \mathcal{N}(x; z, \lambda)$.

We compare our algorithm (GCMC) with the consensus Monte Carlo algorithm (CMC) proposed by Scott et al. (2016). We aim to estimate the posterior mean $\int z \pi(z) \mathrm{d}z$.

We demonstrate on a simple data set with $d = 6$ covariates, split into $b = 8$ subsets, each comprising 512 data. We use various choices of λ for GCMC.



These boxplots show the biases of estimates of the posterior mean of two components, as obtained from 10 runs of each setup.

When estimating some parameters, CMC can introduce a far larger bias than GCMC, as the latter is more robust to deviations from Gaussianity, though must λ be chosen appropriately to control the variance. In the case above left, the corresponding covariate is rarely observed in some of the data subsets, so the f_j are skewed in this dimension, and are poorly approximated by Gaussians.

For parameters corresponding to more frequently-observed covariates, GCMC performs comparably to CMC if λ is chosen suitably *(above right)*.

Log-normal toy model

Let $\mathcal{LN}(x; \mu, \sigma^2)$ denote the density at x of a log-normal distribution with parameters (μ, σ^2).

- Let $\mu(z) = \mathcal{LN}(z; \mu_0, \sigma_0^2)$.
- Let $f_j(z) = \mathcal{LN}(y_j; \log(z), \sigma_j^2)$.
- For our algorithm, we use log-normal transition kernels: $K_\lambda(z, x) = \mathcal{LN}(x; \log(z), \lambda)$.

Although the CMC algorithm of Scott et al. is exact for Gaussian models, this does not apply to reparametrisations, as is the case here. In GCMC, for small enough λ we would still expect the z-marginal of π_λ to be a good approximation of the true target.

We demonstrate on an example with $b = 32$; in each case, 10^6 samples were used, following burn-in. Estimating the first moment $\int z \pi(z) \mathrm{d}z \approx 1.131$:

	Bias squared	Variance
GCMC, $\lambda = 1$	1.32×10^{-3}	4.66×10^{-5}
GCMC, $\lambda = 10^{-1}$	5.45×10^{-5}	2.32×10^{-4}
GCMC, $\lambda = 10^{-2}$	8.32×10^{-5}	6.03×10^{-4}
GCMC, $\lambda = 10^{-3}$	1.52×10^{-3}	1.26×10^{-2}
CMC	2.37×10^{-2}	1.41×10^{-3}

When estimating higher moments, GCMC results in far smaller biases than CMC, if a small enough λ is used. For example, estimating the fifth moment $\int z^5 \pi(z) \mathrm{d}z \approx 3.445$:

	Bias squared	Variance
GCMC, $\lambda = 1$	$1.53 \times 10^{+1}$	6.57×10^{-2}
GCMC, $\lambda = 10^{-1}$	1.15×10^{-1}	9.84×10^{-2}
GCMC, $\lambda = 10^{-2}$	7.34×10^{-2}	2.07×10^{-1}
GCMC, $\lambda = 10^{-3}$	1.49×10^{-1}	9.45×10^{0}
CMC	$3.49 \times 10^{+3}$	$3.49 \times 10^{+3}$

▼ REFERENCES

Bertsekas, D. P., and Tsitsiklis, J., N. (1989). 'Parallel and distributed computation: numerical methods'. Prentice-Hall, Engelwood Cliffs.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). 'Distributed optimization and statistical learning via the alternating direction method of multipliers'. Foundations and Trends in Machine Learning, 3(1):1–122.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). 'Bayes and big data: the consensus Monte Carlo algorithm'. International Journal of Management Science and Engineering Management, 11(2):78–88.

Xu, M., Teh, Y. W., Zhu, J., and Zhang, B. (2014). 'Distributed context-aware Bayesian posterior sampling via expectation propagation'. In: Advances in Neural Information Processing Systems, pages 3356–3364.

Lewis Rendell is funded by EPSRC grant number EP/M508184/1.