Introduction

For problems involving large data sets, it may be practical or necessary to distribute the data across multiple processors. We consider a target probability density function given by

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

where $f_j$ is computable on processor $j$, requiring consideration of $y_j$, the $j$th subset of the full data set. We wish to generate samples distributed according to the corresponding distribution.

Existing approaches to this problem include:

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

The instrumental model

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

We introduce an instrumental hierarchical model by associating an instrumental variable $x_j$ with each subset of the data, and a introducing a top-level parameter $\lambda$:

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

where $\{K_j : \lambda \in \mathbb{R}_+\}$ is a family of Markov transition densities. The $z$-marginal of $\pi_\lambda$ is

$$\pi_\lambda(z) \propto \mu(z) \prod_{j=1}^{b} K_j(z, x_j) dx.$$ 

We assume that $f_j$ is bounded, and assume that this family satisfies $\int K_j(z, x_j) f_j(x) dx \rightarrow f_j(z)$ pointwise as $\lambda \rightarrow 0$. This implies convergence in total variation of $\pi_\lambda$ to $\pi$, so that for bounded functions $\varphi$:

$$\int \varphi(z) \pi(z) dz \rightarrow \int \varphi(z) \pi(z) dz.$$

Examples

We compare our algorithm (GCMC) with the consensus Monte Carlo algorithm (CMC) proposed by Scott et al. (2016). In both examples, we aim to estimate $\int \varphi(z) dz$.

Lognormal toy example

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

- Let $\mu(x) = \mathcal{LN}(x; \mu_0, \sigma_0^2)$
- Let $f_j(x) = \mathcal{LN}(x; x_j, \sigma_j^2)$

For GCMC, use lognormal transition kernels:$K_j(x, x_j) = \mathcal{LN}(x; \log(x), \lambda)$. The treatment of the prior in the CMC algorithm results in large biases when an asymmetric prior is used (although a reparameterisation would solve this issue in the toy example). GCMC avoids this, but requires a careful choice of $\lambda$, as demonstrated here with $b = 10$. In each case, $10^6$ samples were used, following burn-in.

### Examples (continued)

Binary regression

Binary logistic regression models are commonly used in A/B testing settings — 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 $\eta_i \in \{0, 1\}$ and vectors $\xi_i \in \{0, 1\}^d$ of binary covariates.
- The data are split into $b$ subsets;
- The relevant to distributed settings is that draws a new $x_j$ according to the density $\pi_\lambda(x_j | z)$ may then be sent to a central node that draws a new $z'$ according to $\pi_\lambda(z' | x_1:b)$. By adaptively specifying the sequence of values $\lambda_1, \lambda_2, \ldots$, we may then be sent to a central node that draws a new $z'$ according to $\pi_\lambda(z' | x_1:b)$.

SMC sampler

The parameter $\lambda$ may be chosen to balance computational tractability with fidelity to the true model, in a form of bias–variance tradeoff. To this end, we use $\pi_\lambda$, $\pi_\lambda$, $\ldots$, $\pi_\lambda$ to form an SMC sampler. If $\pi_\lambda$-invariant MCMC kernels are used (such as those formed by the Gibbs procedure), then the potential functions depend only on the transition kernels $K_j$.

By adaptively specifying the sequence of values $\lambda$, and using appropriate variance estimators, this approach could be used to specify $\lambda$ in an automated manner.

References


