Marginal Markov Chain Monte Carlo (MCMC)

Mohamed Tarek[1,2], Jose Storopoli[1], Chris Elrod[1,3], Joga Gobburu[1,4], Vijay Ivaturi[1,4]

[1]Pumas-AI Inc., USA

[2]Business School, University of Sydney, Australia

[3]JuliaHub Inc., USA

[4]University of Maryland Baltimore, USA

Motivation

The number of parameters in hierarchical models increases with the number of sub-groups. In pharmacometrics, each sub-group is a subject/patient with individual parameters η_i for each subject i. There are also other population-level parameters θ shared between all the subjects. In most cases, the number of parameters per subject is < 10 and the number of population parameters is < 100. However, the number of subjects can be $\gg 100$. Running Markov Chain Monte Carlo (MCMC) on such models can often be extremely time-consuming where the No-U-Turn Sampler (NUTS) [1] algorithm with a diagonal mass matrix often finds extremely small step sizes in the adaptation phase. This can lead to extremely slow sampling with an estimated completion time of hours or even days.

The heavy correlation which exists between θ and the η_is in the posterior often means that the $P(\eta_i \mid \theta, \text{data})$ can be significantly different in each MCMC iteration. One alternative is to marginalize η_i for all i using the Laplace method [2]. The Laplace method uses a Gaussian approximation of $P(\eta_i \mid \theta, \text{data})$, common in pharmacometrics [3] when maximizing the marginal likelihood. This is similar in spirit to the integrated nested Laplace approximation (INLA) algorithm [4] which uses a similar Laplace-based approach.

Marginalizing η_i comes with the additional cost of computing the marginal log likelihood of the population parameters as well as its gradient. However, this additional computational cost can be shown to be over-compensated with computational gains made using the NUTS algorithm on the resulting small dimensional problem without significant loss in sampling accuracy. There are two main contributions in this work:

1. We generalize the so-called first order conditional estimation (FOCE) [3] method to approximate the Hessian required in the Laplace method and use it to marginalize non-Gaussian random variables.
2. We derive the gradient of the marginal likelihood in the general non-Gaussian case using the implicit function theorem.

Marginal MCMC Algorithm

We adopt a "Marginalize-Then-Sample" approach by integrating the subject-specific parameters ηs out and then sampling from the marginal posterior only. The marginal posterior probability is proportional to:

$$\begin{aligned} P(\theta \mid \text{data}) &\propto P(\theta, \text{data}) \\ &= \int_{\eta_K} \cdots \int_{\eta_1} P(\theta, \ldots, d\eta_K, \text{data}) d\eta_1 \ldots d\eta_K \end{aligned}$$

where K is the number of subject-specific parameters. MCMC can sample from $P(\theta \mid \text{data})$ given $P(\theta, \text{data})$.

For hierarchical models, we can further break down the joint probability into the product of independent conditionals and $P(\theta)$:

$$P(\theta, \eta_1, \ldots, \eta_K, \text{data}) = P(\theta) \cdot \prod_{i=1}^{K} P(\eta_i, \text{data}_i \mid \theta)$$

where data_i is the data of subject i, which simplifies to:

$$\begin{aligned} P(\theta, \text{data}) &= \int_{\eta_K} \cdots \int_{\eta_1} P(\theta) \cdot \prod_{i=1}^{K} P(\eta_i, \text{data}_i \mid \theta) d\eta_1 \ldots d\eta_K \\ &= P(\theta) \cdot \prod_{i=1}^{K} \int_{\eta_i} P(\eta_i, \text{data}_i \mid \theta) d\eta_i \end{aligned}$$

Each of the above smaller integrals has a small dimension.

Approximate Marginalization

The Marginal MCMC algorithm in Pumas [6] uses approximate integration methods, e.g. the Laplace method or the FOCE approximation of the Laplace method to compute the subject-specific integrals. This approximation is often accurate if $P(\eta_i|\theta, \text{data}_i)$ is approximately Gaussian, which is true if the model is conditionally identifiable with respect to η_i after fixing θ and there is enough data per subject to identify the true parameters η_i.

Efficiency and Cost

There is a tradeoff between the cost per HMC step (marginal MCMC is more expensive) and the number of steps per proposal (marginal MCMC requires less steps). However, the computational cost of marginal MCMC can be more effectively parallelized by parallelizing the subject integral computations. The computational effort per subject per HMC step in marginal MCMC is higher than the joint MCMC method. This makes up for the parallelism overhead of having to manage and communicate with multiple threads/processes.

Table 1: Marginal versus Joint MCMC.

	Marginal MCMC	**Joint MCMC**
Number of parameters	Low	High
Accuracy	Approximate even for infinite samples unless the conditional posterior $P(\eta_i \mid \theta, \text{data}_i)$ is Gaussian	Exact with infinite samples
Cost per HMC step	High	Low
Mass matrix	Dense	Diagonal (by default)
Max tree depth	Often low	Often high
Parallelism	Efficient for all models	Efficient only for difficult models

Results

We ran experiments in Pumas [6] using a non-linear regression model, comparing treatment versus placebo arms, using synthetic data, in a `Intel Xeon Platinum 8375C CPU @ 2.90GHz 32GB RAM` Linux virtual machine with 8 virtual CPUs. Each arm was fitted using both Marginal and Joint MCMC using NUTS with 5 to 100 subjects in increments of 5 subjects per arm, each subject has 26 observations. The sampling was performed using 4 parallel MCMC chains with parallel computation of the log probability over the subjects. The number of total samples were 1'000, with 500 (half) adaptation (warmup) samples, the target acceptance ratio was 0.8. Marginal MCMC was sampled using a dense matrix, and Joint MCMC was sampled using a diagonal matrix. For Marginal MCMC we used the FOCE algorithm for the approximate marginalization. The model has 6 population parameters (θ) and 2 subject parameters (η), and one parameter represents the efficacy of the treatment arm.Figures 1 and 2 display the results using the effective sample size (ESS) per second, and standard deviation of the parameter of interest (both discarding warmup); respectively.

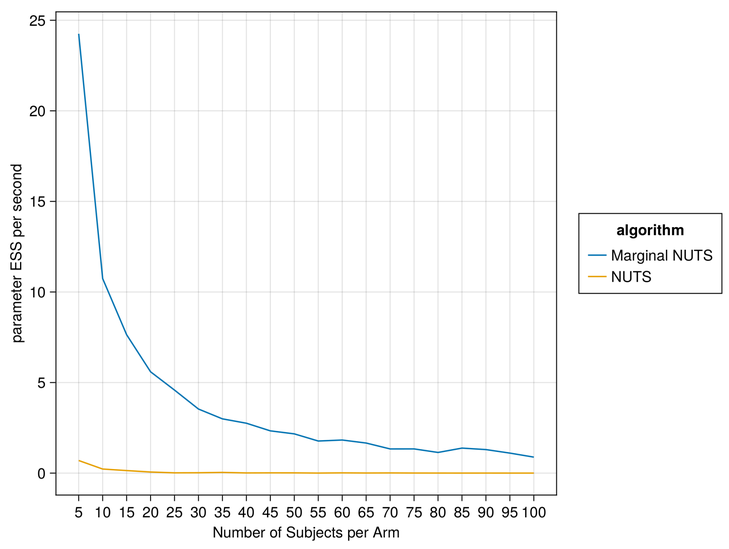

Figure 1: Parameter of interest ESS/sec (higher is better).

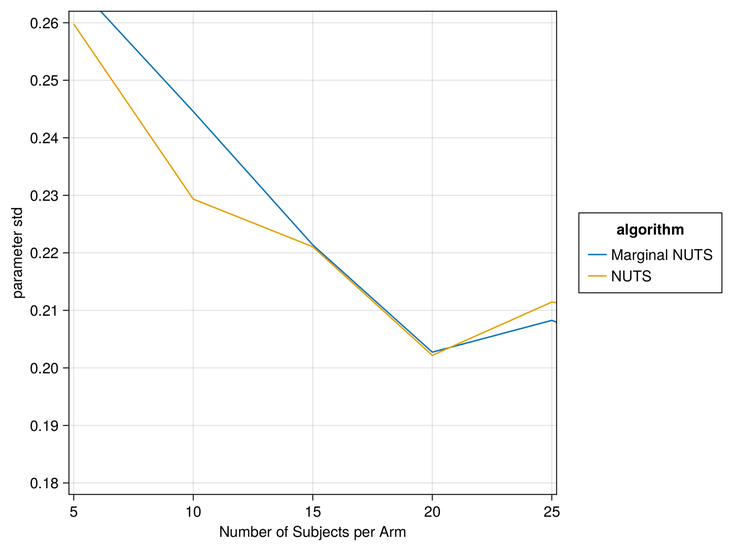

Figure 2: Parameter of interest standard deviation.

References

[1] M. Hoffman, A. Gelman, The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *Journal of Machine Learning Research*, 2014.

[2] C. Margossian, A. Vehtari, D. Simpson, R. Agrawal, Hamiltonian Monte Carlo using an adjoint-differentiated Laplace approximation: Bayesian inference for latent Gaussian models and beyond, *NeurIPS*, 2020.

[3] Y. Wang, Derivation of various NONMEM estimation methods, *J Pharmacokinet Pharmacodyn*, 2007.

[4] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B*, 2009.

[5] R. Neal, MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2011.

[6] C. Rackauckas, et al., Accelerated predictive healthcare analytics with Pumas, a high performance pharmaceutical modeling and simulation platform, 2020.