Parallelizing MCMC with Random Partition Trees

Xiangyu Wang*, Richard (Fangjian) Guo†, Katherine A. Heller‡*, David B. Dunson*

*Department of Statistical Science, †Department of Computer Science, Duke University

http://bit.ly/1RdhuBW

Introduction

The modern scale of data has brought new challenges to Bayesian inference. In particular, conventional MCMC algorithms are computationally very expensive for large data sets. A promising approach is the embarrassingly parallel MCMC (EP-MCMC). Existing EP-MCMC algorithms are limited by approximation accuracy and difficulty in resampling. Here we propose a new EP-MCMC algorithm PART that solves these problems. The new algorithm applies random partition trees to combine the subset posterior draws, which is distribution-free, easy to resample from and can adapt to multiple scales.

Features of PART

- Fast sub-posterior estimation
- Efficient density aggregation
- Efficient resampling

Embarrassingly Parallel MCMC

- Partition data into m subsets
- Run MCMC independently in parallel on each subset with your favorite algorithm
- Aggregate the sub-chains (sub-posteriors) by some algorithm

PART Algorithm

PART relies on the product density equation (PDE). Assuming X is the observed data and θ is the parameter of interest, when the observations are iid conditioned on θ, for any partition of $X = X^{(1)} \cup X^{(2)} \cup \cdots \cup X^{(m)}$, then

$$p(\theta | X) \propto \pi(\theta)p(X | \theta) \propto p(\theta | X^{(1)})p(\theta | X^{(2)}) \cdots p(\theta | X^{(m)}),$$

if the prior on the full data and subsets satisfy $\pi(\theta) = \prod_{i=1}^{m} \pi_i(\theta)$.

Step 1: Space Partitioning

Since the full data posterior is $p(\theta | X) \propto \prod_{i=1}^{m} p(X^{(i)} | \theta) \prod_{i=1}^{m} f_i^{(i)}(\theta)$. We use a random partition tree or multi-scale histogram to estimate each sub-posterior density $f_i^{(i)}(\theta)$. Let $F_k$ be the collection of all $R^d$-partitions formed by $K$ disjoint rectangular blocks, where a rectangular block takes the form of

$$A_k = \{ (l_{k,1}, r_{k,1}), (l_{k,2}, r_{k,2}), \cdots, (l_{k,p}, r_{k,p}) \} \subset \mathbb{R}^d$$

for some $l_{k,p} < r_{k,p}$. A $K$-block histogram is then defined as

$$\hat{f}^{(i)}(\theta) = \frac{1}{N_i} \sum_{A_k \in F_K} \mathbb{1}(\theta \in A_k),$$

where $\{ A_k : k = 1, 2, \cdots, K \} \in F_K$ are the blocks and $N_i$ is the total number of posterior samples on the $i$th subset and of those inside the block $A_k$, respectively (assuming the same $N$ across subsets). A random partition tree is built like a decision tree by recursively (1) randomly selecting a dimension and (2) choosing an optimal cutting point by $\hat{f}(\theta)$ to bisect that dimension, until the probability or area of a block drops below a threshold.

Step 2: Density Aggregation

By constraining all trees to take the same partitioning $\{ A_k \}$, the aggregated density is still a tree with the same structure. It can be computed by block-wise multiplication and a renormalization in $O(MK)$. Let $Z = \sum_{k=1}^{K} \prod_{i=1}^{m} \frac{\hat{f}^{(i)}(\hat{\theta})}{f_i^{(i)}} |A_k|^{-1}$ be the normalizing constant, $w_i$ is the updated weights, and $\theta_k(\theta) = \text{unif}(\theta | A_k)$ is the block-wise distribution.

Why not Kernel Density Estimate?

$O(n^m)$ mixture components — very slow mixing.

Variance Reduction & Smoothing

Random Tree Ensemble

Inspired by random forests, the full posterior is estimated by averaging $T$ independent trees, which are built in parallel. Resampling from the aggregated posterior consists of:

- choosing a tree uniformly at random
- sampling a block $k$ by weight $w_k$
- sampling $\theta \sim \theta_k(\theta)$

Local Gaussian Smoothing

The blockwise uniform distribution can be replaced by a Gaussian distribution $\theta_k = N(\theta_k|\mu_k, \Sigma_k)$, with mean and covariance estimated “locally” by samples within the block. A multiplied Gaussian approximation is used:

$$\Sigma_k = \sum_{i=1}^{m} \frac{\hat{f}^{(i)}(\hat{\theta})}{f_i^{(i)}} |A_k|^{-1} \Sigma_i^{-1} \hat{\Sigma}_k^{-1}, \quad \mu_k = \sum_{i=1}^{m} \frac{\hat{f}^{(i)}(\hat{\theta})}{f_i^{(i)}} |A_k|^{-1} \hat{\Sigma}_k^{-1} \hat{\theta}_k,$$

where $\hat{\Sigma}_k$ and $\hat{\mu}_k$ are estimated with the $i$th subset.

Features of PART

- Fast sub-posterior estimation
- Efficient density aggregation
- Efficient resampling

Related Works

- Simple averaging and Weighted averaging
  (Consensus Monte Carlo [1]): weights are optimally chosen for a Gaussian posterior.
- Weierstrass rejection sampler: subset posterior samples are passed through a rejection sampler based on the Weierstrass transform [2].
- Parametric density product: a product of Laplacian approximations to sub-posteriors [3].
- Nonparametric/ Semiparametric density product: a product of kernel density estimates (or its semiparametric variant) for subset posteriors [3], which is sampled with an independent Metropolis chain.

References