Parallelizing MCMC via Random Forest

Changye WU; Christian ROBERT
wu@ceremade.dauphine.fr ; xian@ceremade.dauphine.fr
CEREMADE, Université Paris-Dauphine

Abstract

For Bayesian analyses of so-called big data models, the divide-and-conquer MCMC approach splits the whole data set into smaller batches, runs MCMC algorithm over each batch to produce parameter samples, and combines these towards producing an approximation of the posterior distribution. In this spirit, we introduce random forests into this method and use each subposterior or partial posterior as a proposal distribution to implement importance sampling. Unlike the existing divide-and-conquer MCMC solutions, our method is based on scaled subposteriors, whose scale factor is not necessarily restricted to 1 or to the number of batches. Through several experiments, we show that our methods performs satisfactorily aginst the existing solutions in low-dimensional setting, for both Gaussian and severely non-Gaussian cases, and under model misspecification.

Problem

Denote by $\mathcal{X} = \{x_1, x_2, \cdots, x_N\}$ the whole set of observations, where $x_i \sim p_\theta(\cdot)$ i.i.d. and $\theta \in \Theta \subset \mathbb{R}^d$. Let $\pi_0(\theta)$ denote the prior distribution on Θ. In Bayesian analysis, the target of interest is the posterior distribution:

$$\pi(\theta|\mathcal{X}) \propto \gamma(\theta) = \pi_0(\theta)\prod_{i=1}^{N} p(x_i|\theta)$$

As a standard approach to sample from $\pi(\theta|\mathcal{X})$, MCMC algorithms with Metropolis-Hastings step require at each iteration to sweep over the whole data set. When the size of the data set, N, is too large, evaluating the acceptance ratio

$$1 \wedge \frac{\gamma(\theta^*)q(\theta|\theta^*)}{\gamma(\theta)q(\theta^*|\theta)}$$

is too costly an operation and rules out the applicability of MCMC algorithms in some Bayesian inferences.

Methodology

Splitting the data set $\mathcal{X}$ into subsets $\mathcal{X}_1, \cdots, \mathcal{X}_K$, each with the same size $m = \frac{N}{K}$. For each subset $\mathcal{X}_k$, $k = 1, \cdots, K$, we define the associated λ_k-subposterior as

$$\pi_k^{\lambda_k}(\theta|\mathcal{X}_k) = \frac{(\gamma_k(\theta|\mathcal{X}_k))^{\lambda_k}}{Z_{k,\lambda_k}}, \quad \gamma_k(\theta|\mathcal{X}_k) = \pi_0(\theta)^{\frac{1}{K}} \prod_{x\in\mathcal{X}_k} p(x|\theta),$$

where Z_{k,λ_k} is the normalizing constant of $(\gamma_k(\theta|\mathcal{X}_k))^{\lambda_k}$.

Firstly, we run some MCMC algorithms on$\pi_k^{\lambda_k}(\theta|\mathcal{X}_k)$ to obtain MCMC samples $\{\theta_1^k, \theta_2^k, \cdots, \theta_T^k\}$. In the Metropolis–Hastings acceptance ratio, we therefore evaluate $\log \gamma_k(\theta|\mathcal{X}_k)$ for each proposal. As a byproduct, we thus obtain evaluations of $\log \gamma_k(\theta|\mathcal{X}_k)$ at some parameter values $\{\vartheta_1^k, \vartheta_1^k, \cdots, \vartheta_{T_k}^k\}$, which are the quantities proposed by the MCMC algorithms.

Secondly, calling in a random forests on the learning set

$$\left\{\left(\vartheta_1^k, \log \gamma_k(\vartheta_1^k|\mathcal{X}_k)\right), \left(\vartheta_2^k, \log \gamma_k(\vartheta_2^k|\mathcal{X}_k)\right), \cdots, \left(\vartheta_{T_k}^k, \log \gamma_k(\vartheta_{T_k}^k|\mathcal{X}_k)\right)\right\}$$

provides an estimator, f_k, of the value of the unnormalised partial log-likelihoods, $\log\{\gamma_k(\theta|\mathcal{X}_k)\}$. These estimators f_k, $k = 1, \cdots, K$, provide an approximation of $\pi(\theta|\mathcal{X})$ by

$$f(\theta) = \exp\left\{\sum_{k=1}^{K} f_k(\theta)\right\}$$

up to a multiple constant.

Thirdly, for the k-th MCMC sample set, weighting each sample θ_t^k with weight w_t^k,

$$w_t^k \propto \exp\left\{\sum_{k=1}^{K} f_k(\theta_t^k) - \lambda_k \log \gamma_k(\theta_t^k|\mathcal{X}_k)\right\}, t = 1, \cdots, T,$$

by importance sampling, provides k-th approximation of the posterior

$$\hat{\pi}_k = \sum_{t=1}^{T} w_t^k \delta_{\theta_t^k}$$

Finally, averaging these K discrete measures,

$$\tilde{\pi} = \frac{1}{K}\sum_{k=1}^{K} \hat{\pi}_k$$

gives an approximation of the posterior distribution.

Computation complexity

The computing budget of our approach is made of three components

- At the divide-and-conquer stage, the computating cost is $\mathcal{O}(T_k N/K)$ on each subsample and we generate a total of T samples points, where T may differ from T_k according to the techniques of burn-in and thinning of MCMC.
- At the regression training stage, the cost of each random forest is $\mathcal{O}(T_k \log T_k)$.
- At the combination stage, the cost of importance sampling is $\mathcal{O}(KT \log T_k)$ for weighting all samples over all subposteriors.

A Bimodal Posterior

$$X_n \sim \frac{1}{2}\mathcal{N}(\theta_1, 2) + \frac{1}{2}\mathcal{N}(\theta_1 + \theta_2, 2), n = 1, \cdots, N, (\theta_1^0, \theta_2^0) = (0, 1)$$

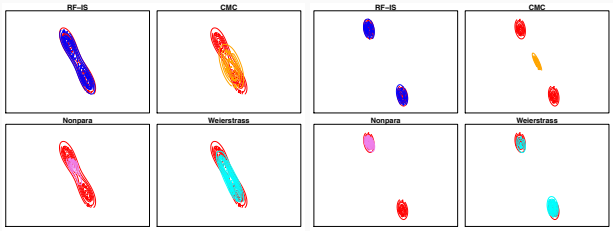

Fig. 1: Comparison of the contours of the true posterior (red), RF-IS (blue), consensus Monte Carlo (orange), KDE (violet) and Weierstrass sampler (cyan). (left) $K = 20, N = 200$, (right) $K = 50, N = 10000$.

A Moon-shaped Posterior

$$X_n \sim \mathcal{N}(\sqrt{\theta_1} + \sqrt{\theta_2}, 2), \theta_1 \geq 0, \theta_2 \geq 0.(\theta_1^0, \theta_2^0) = (\frac{1}{4}, \frac{1}{4})$$

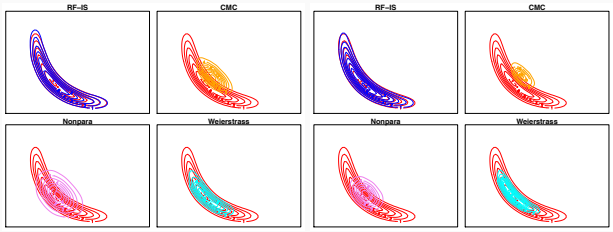

Fig. 2: Comparison of the contours of the true posterior (red), RF-IS (blue), consensus Monte Carlo (orange), KDE (violet) and Weierstrass sampler (cyan). (left) $K = 10, N = 1000$, (right) $K = 20, N = 1000$.

A Misspecification Example

$$X \sim \mathcal{N}(\mu, \sigma^2), \theta = (\mu, \sigma^2)$$

Dataset1: $X \sim \mathcal{N}(0, 1)$; Dataset2: $X \sim \mathcal{LN}(0, 1)$

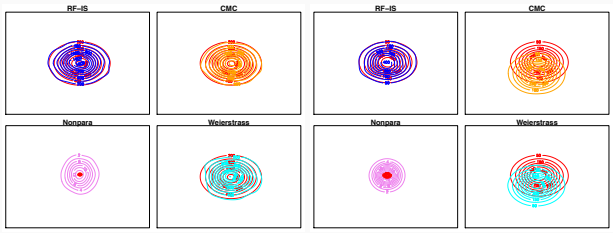

Fig. 3: Comparison of the contours of the true posterior (red), RF-IS (blue), consensus Monte Carlo (orange), KDE (violet) and Weierstrass sampler (cyan) with comfiguration $K = 10, N = 10000$. (left) Normal (right) log-Normal

Time Comparisons

Model	RF-IS	CMC	Nonpara	Weierstrass
Bimodal Posterior	1	0.94	8.11	0.97
Moon-shaped Posterior	1	0.96	6.93	0.93
Normal	1	0.92	2.33	0.93
Log-Normal	1	0.90	15.19	0.98

Conclusion

1. Advantages:
   (a) The scale factor is not necessarily restricted to be 1 or K.
   (b) Random Forest is easy to implement, has a strong learning ability of non-linear relations and is robust.
   (c) The prediction abilities of random forests are scalable, that is, given a training set of size T , the cost of predicting the output of a new input is of order $O(\log(T))$.
2. Limitations:
   (a) The curse of dimensionality.
   (b) The selection of scale factor.