Scalable Bayes via Parallelization and Posterior Aggregation

Sarah Shy

December 3, 2019

Motivation

Recall MCMC

- Goal: Sample from a posterior distribution / approximate the posterior
- Idea: Construct a Markov Chain whose stationary distribution is the target distribution

Motivation

Recall MCMC

- Goal: Sample from a posterior distribution / approximate the posterior
- Idea: Construct a Markov Chain whose stationary distribution is the target distribution

Computing Problem

- The MC converges to the target distribution after **infinite** iterations
- We run the algorithm for a long time

Motivation

Recall MCMC

- Goal: Sample from a posterior distribution / approximate the posterior
- Idea: Construct a Markov Chain whose stationary distribution is the target distribution

Computing Problem

- The MC converges to the target distribution after **infinite** iterations
- We run the algorithm for a long time

(One) Solution: Parallelization

Goal: Speed up MCMC for approximating posterior distribution

Idea: Exploit multiple processors to run several independent chains and combine them.

Split data → MCMC Sampler → Aggregate all posteriors

Motivation

Recall MCMC

- Goal: Sample from a posterior distribution / approximate the posterior
- Idea: Construct a Markov Chain whose stationary distribution is the target distribution

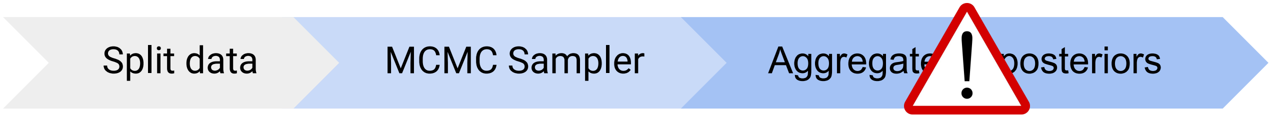
Computing Problem

- The MC converges to the target distribution after **infinite** iterations
- We run the algorithm for a long time

(One) Solution: Parallelization

Goal: Speed up MCMC for approximating posterior distribution

Idea: Exploit multiple processors to run several independent chains and combine them.



Easy!

You’re good at this

Aggregating Posteriors

We have: several subset posterior measures (noisy approximations to the posterior based on the full data)

We want to: aggregate them to obtain a single posterior measure that is an approximation to the true posterior

Aggregating Posteriors

We have: several subset posterior measures (noisy approximations to the posterior based on the full data)

We want to: aggregate them to obtain a single posterior measure that is an approximation to the true posterior

WASserstein **P**osterior (Srivastava et al., 2015)

Why it dominates:

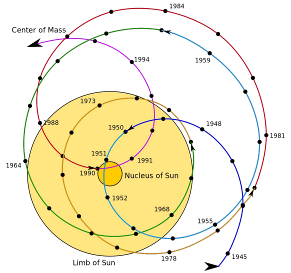
- ✅ No need for a kernel or tuning parameters (unlike other methods)
- ✅ WASP can be estimated “efficiently” via a linear program
- ✅ Resulting posterior is a “good” approximation

WASP calculates the **Wasserstein barycenter (WB)** of the subset posterior measures

First, some background

Wasserstein Barycenter?

Barycenter (Wikipedia)
the center of mass of two or more bodies that orbit one another and is the point about which the bodies orbit



Wasserstein metric
a distance function defined between probability measures on a given metric space

Wasserstein barycenter
The mean of a set of probability measures (the measure that minimizes the sum of its Wasserstein distances to each element in that set)

p^{th} Wasserstein distance

(Ω, d)	Metric space, metric
$P(\Omega)$	The set of Borel probability measures on Ω
$\Pi(\mu, \nu)$	The set of all probability measures on Ω^2 that have marginals μ and ν (The set of all "couplings" of μ and ν)

Let $\mu, \nu \in P(\Omega)$

p^{th} Wasserstein distance

(Ω, d)	Metric space, metric
$P(\Omega)$	The set of Borel probability measures on Ω
$\Pi(\mu, \nu)$	The set of all probability measures on Ω^2 that have marginals μ and ν (The set of all "couplings" of μ and ν)

Let $\mu, \nu \in P(\Omega)$

p^{th} Wasserstein distance: $W_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \mathbf{E}[d(X, Y)^p] \right)^{1/p}$

p^{th} Wasserstein distance

(Ω, d)	Metric space, metric
$P(\Omega)$	The set of Borel probability measures on Ω
$\Pi(\mu, \nu)$	The set of all probability measures on Ω^2 that have marginals μ and ν (The set of all "couplings" of μ and ν)

Let $\mu, \nu \in P(\Omega)$

p^{th} Wasserstein distance: $W_p(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \mathbf{E}[d(X,Y)^p] \right)^{1/p}$

Wasserstein Barycenter of N measures $\{\nu_1, \cdots, \nu_N\} \in P(\Omega)$: $\arg\min_{\tau} \frac{1}{N} \sum_{i=1}^{N} W_p^p(\tau, \nu_i)$

… in the the context of parallel MCMC

- Approximate subset posterior probability measures ν_i with empirical measures:

$$\hat{\nu}_i = \hat{\Pi}_i(\cdot) = \sum_{j=1}^{S} \frac{1}{S} \delta_{\theta ij}(\cdot), \qquad i = 1, \ldots, N$$

⇒ 2nd order Wasserstein distance

… in the the context of parallel MCMC

- Approximate subset posterior probability measures ν_i with empirical measures:

$$\hat{\nu}_i = \hat{\Pi}_i(\cdot) = \sum_{j=1}^{S} \frac{1}{S} \delta_{\theta ij}(\cdot), \qquad i = 1, \ldots, N$$

$\Rightarrow$ 2nd order Wasserstein distance

- WASP **intractable** in most cases. Uh-oh.

… in the the context of parallel MCMC

- Approximate subset posterior probability measures ν_i with empirical measures:

$$\hat{\nu_i} = \hat{\Pi}_i(\cdot) = \sum_{j=1}^{S} \frac{1}{S} \delta_{\theta ij}(\cdot), \qquad i = 1, \dots, N$$

⇒ 2nd order Wasserstein distance

- WASP **intractable** in most cases. Uh-oh.

Approximate: $\hat{\overline{\Pi}}_i(\cdot) = \sum_{i=1}^{N}\sum_{j=1}^{S} a_{ij}\delta_{\theta_{ij}}(\cdot)$ constrained to $0 \leq a_{ij} \leq 1, \quad \sum_{i=1}^{N}\sum_{j=1}^{S} a_{ij} = 1$

… in the the context of parallel MCMC

- Approximate subset posterior probability measures ν_i with empirical measures:

$$\hat{\nu_i} = \hat{\Pi}_i(\cdot) = \sum_{j=1}^{S} \frac{1}{S} \delta_{\theta ij}(\cdot), \qquad i = 1, \ldots, N$$

⇒ 2nd order Wasserstein distance

- WASP **intractable** in most cases. Uh-oh.

Approximate: $\hat{\overline{\Pi}}_i(\cdot) = \sum_{i=1}^{N} \sum_{j=1}^{S} a_{ij} \delta_{\theta_{ij}}(\cdot)$ constrained to $0 \leq a_{ij} \leq 1, \quad \sum_{i=1}^{N} \sum_{j=1}^{S} a_{ij} = 1$

A linear program! Yay!

Several existing algorithms to solve this linear program. See Cuturi (2014), Carlier et al. (2015), Srivastava et al. (2015)

My Contribution

- Total compute time of WASP, compared with other algorithms
- Application to a real-world problem: modeling radial velocity of a star in a binary system

My Contribution

- Total compute time of WASP, compared with other algorithms
- Application to a real-world problem: modeling radial velocity of a star in a binary system

Toy example (16+): Gaussian Mixture

- 100,000 samples from mixture of two 2-dim Gaussians: $f_{\text{mix}}(\mathbf{y}|\theta) = \sum_{i=1}^{2} \pi_i N_2(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i)$

My Contribution

- Total compute time of WASP, compared with other algorithms
- Application to a real-world problem: modeling radial velocity of a star in a binary system

Toy example (16+): Gaussian Mixture

- 100,000 samples from mixture of two 2-dim Gaussians: $f_{\text{mix}}(\mathbf{y}|\theta) = \sum_{i=1}^{2} \pi_i N_2(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i)$

  where $\boldsymbol{\mu}_1 = (1,2)$ $\boldsymbol{\mu}_2 = (7,8)$ $\Sigma_i = \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$ $\boldsymbol{\pi} = (0.3, 0.7)$

- Goal: approximate $p(\boldsymbol{\mu}_1|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$ and $p(\boldsymbol{\mu}_2|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$

My Contribution

- Total compute time of WASP, compared with other algorithms
- Application to a real-world problem: modeling radial velocity of a star in a binary system

Toy example (16+): Gaussian Mixture

- 100,000 samples from mixture of two 2-dim Gaussians: $f_{\text{mix}}(\mathbf{y}|\theta) = \sum_{i=1}^{2} \pi_i N_2(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i)$

  where $\boldsymbol{\mu}_1 = (1,2)$ $\boldsymbol{\mu}_2 = (7,8)$ $\Sigma_i = \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$ $\boldsymbol{\pi} = (0.3, 0.7)$

- Goal: approximate $p(\boldsymbol{\mu}_1|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$ and $p(\boldsymbol{\mu}_2|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$
- Priors: $\boldsymbol{\mu}_i|\Sigma_i \sim N_2(\mathbf{0}, 100\Sigma_i)$ $\boldsymbol{\pi} \sim \text{Dirichlet}\left(\frac{1}{2}, \frac{1}{2}\right)$ $\Sigma_i \sim \text{Inverse-Wishart}(2, 4I_2)$

My Contribution

- Total compute time of WASP, compared with other algorithms
- Application to a real-world problem: modeling radial velocity of a star in a binary system

Toy example (16+): Gaussian Mixture

- 100,000 samples from mixture of two 2-dim Gaussians: $f_{\text{mix}}(\mathbf{y}|\theta) = \sum_{i=1}^{2} \pi_i N_2(\mathbf{y}|\boldsymbol{\mu}_i, \Sigma_i)$

  where $\boldsymbol{\mu}_1 = (1,2)$ $\boldsymbol{\mu}_2 = (7,8)$ $\Sigma_i = \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$ $\boldsymbol{\pi} = (0.3, 0.7)$

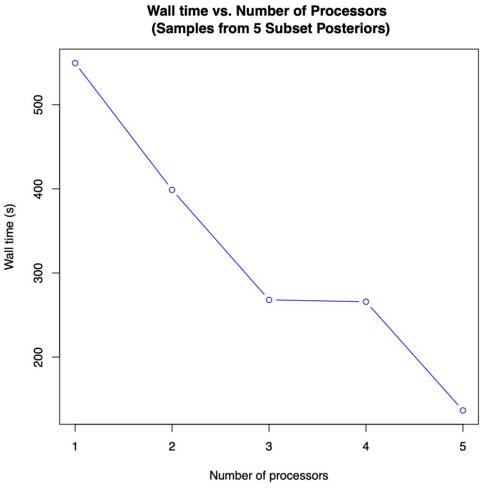
- Goal: approximate $p(\boldsymbol{\mu}_1|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$ and $p(\boldsymbol{\mu}_2|\mathbf{y}, \boldsymbol{\pi}, \Sigma_1, \Sigma_2)$

- Priors: $\boldsymbol{\mu}_i|\Sigma_i \sim N_2(\mathbf{0}, 100\Sigma_i)$ $\boldsymbol{\pi} \sim \text{Dirichlet}\left(\frac{1}{2}, \frac{1}{2}\right)$ $\Sigma_i \sim \text{Inverse-Wishart}(2, 4I_2)$

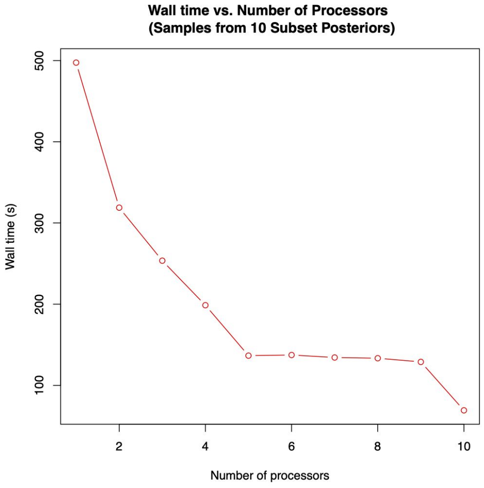
- Sampler: Gibbs
- Data split: 5 subsets, 5000 samples per chain

Preliminary results

5 subset posteriors

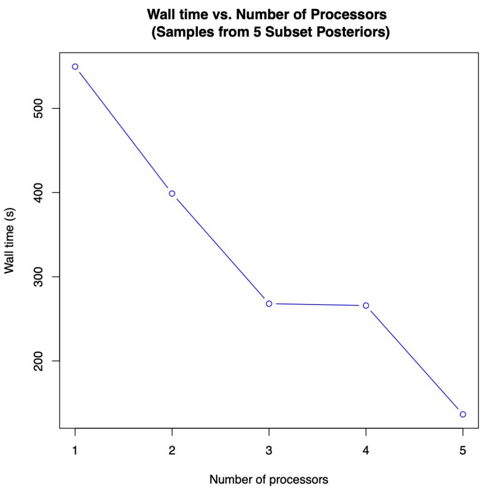


10 subset posteriors

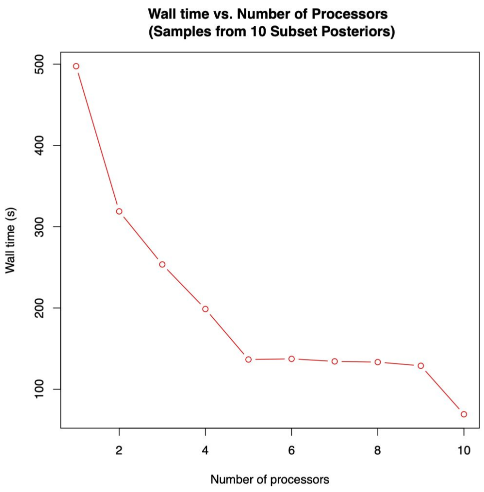


Preliminary results

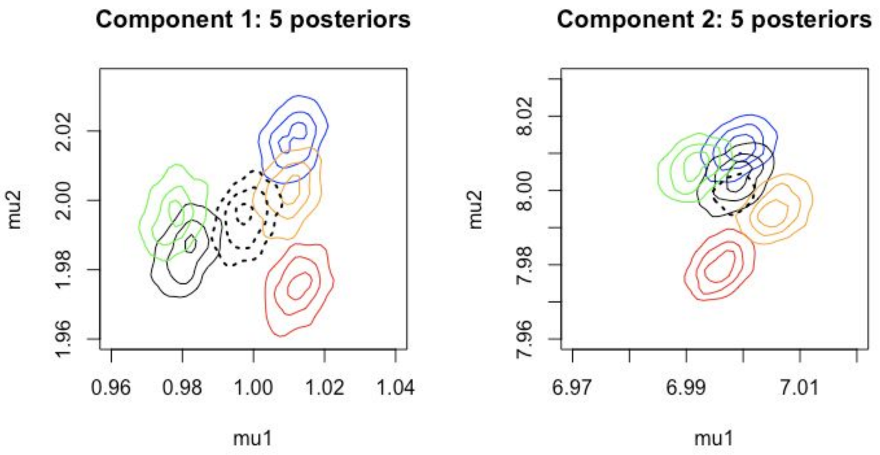
5 subset posteriors



10 subset posteriors

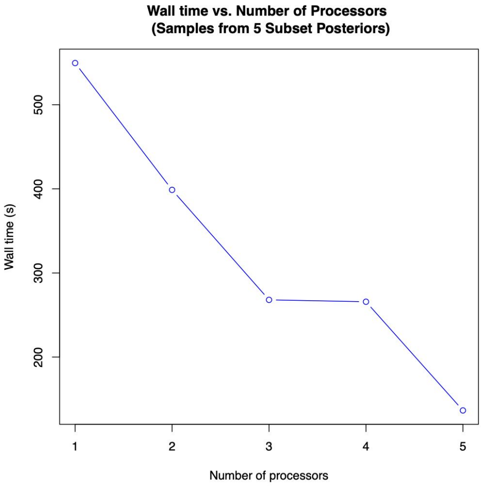


Subset Posterior Approximations

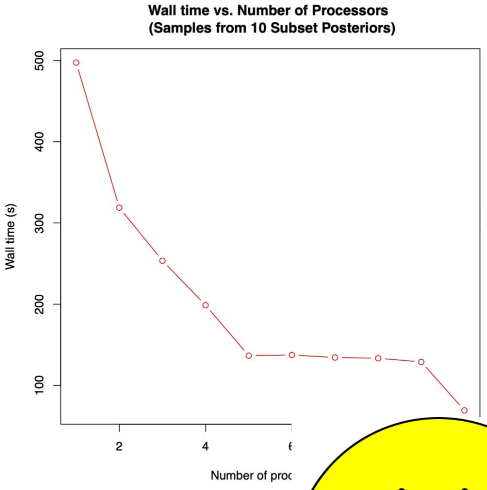


Preliminary results

5 subset posteriors

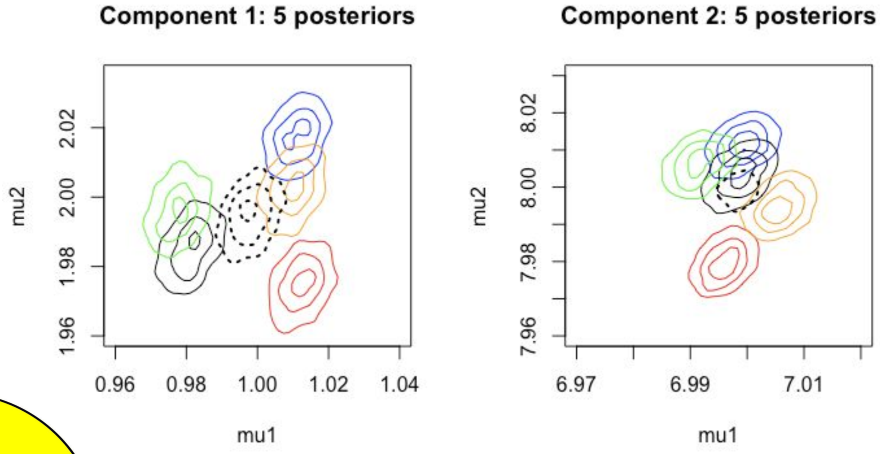


10 subset posteriors



Not bad!

Subset Posterior Approximations



Makes sense!

Alternative algorithms to explore

See paper for theoretical justification of WASP and comparison to other methods:

- **Consensus Monte Carlo** (CMC, Scott et al., 2016)

  weighted average of samples

- **Semiparametric density product** (SDP, Neiswanger et al., 2014)

  kernel smooth each subset posterior density, multiply together to approximate the posterior density

- **Parallel MCMC with M-Posterior** (Minsker et al., 2017)

  posterior aggregation method, robust to outliers, but less efficient

WASP: Scalable Bayes via barycenters of subset posteriors (Srivastava et al., 2015)

Alternative algorithms to explore

See paper for theoretical justification of WASP and comparison to other methods:

- **Consensus Monte Carlo** (CMC, Scott et al., 2016)

  weighted average of samples

- **Semiparametric density product** (SDP, Neiswanger et al., 2014)

  kernel smooth each subset posterior density, multiply together to approximate the posterior density

- **Parallel MCMC with M-Posterior** (Minsker et al., 2017)

  posterior aggregation method, robust to outliers, but less efficient

Note: Parallelizing can only take us so far. No substitute for good samplers.

WASP: Scalable Bayes via barycenters of subset posteriors (Srivastava et al., 2015)