Scaling up Bayesian Inference

David Dunson

Departments of Statistical Science, Mathematics & ECE, Duke University

May 1, 2017
Motivation & background

EP-MCMC

aMCMC

Discussion
Complex & high-dimensional data

Interest in developing new methods for analyzing & interpreting complex, high-dimensional data
Interest in developing new methods for analyzing & interpreting complex, high-dimensional data

Arise routinely in broad fields of sciences, engineering & even arts & humanities
Interest in developing new methods for analyzing & interpreting complex, high-dimensional data

Arise routinely in broad fields of sciences, engineering & even arts & humanities

Despite huge interest in big data, there are vast gaps that have fundamentally limited progress in many fields
‘Typical’ approaches to big data

There is an increasingly immense literature focused on big data
‘Typical’ approaches to big data

- There is an increasingly immense literature focused on big data.
- Most of the focus has been on optimization-style methods.
‘Typical’ approaches to big data

- There is an increasingly immense literature focused on big data
- Most of the focus has been on optimization-style methods
- Rapidly obtaining a point estimate even when sample size $n$ & overall ‘size’ of data is immense
‘Typical’ approaches to big data

- There is an increasingly immense literature focused on big data.
- Most of the focus has been on optimization-style methods.
- Rapidly obtaining a point estimate even when sample size $n$ & overall ‘size’ of data is immense.
- **Bandwagons**: many people work on quite similar problems, while critical open problems remain untouched.
My focus - probability models

"I wish we hadn't learned probability 'cause I don't think our odds are good."
My focus - probability models

General probabilistic inference algorithms for complex data

“I wish we hadn't learned probability 'cause I don't think our odds are good.”
My focus - probability models

General probabilistic inference algorithms for complex data

We would like to be able to handle arbitrarily complex probability models
My focus - probability models

- General probabilistic inference algorithms for complex data
- We would like to be able to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

"I wish we hadn't learned probability 'cause I don't think our odds are good."
My focus - probability models

- General probabilistic inference algorithms for complex data
- We would like to be able to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

“I wish we hadn’t learned probability ’cause I don’t think our odds are good.”
My focus - probability models

General probabilistic inference algorithms for complex data

We would like to be able to handle arbitrarily complex probability models

Algorithms scalable to huge data - potentially using many computers

Accurate uncertainty quantification (UQ) is a critical issue
My focus - probability models

General probabilistic inference algorithms for complex data

We would like to be able to handle arbitrarily complex probability models

Algorithms scalable to huge data - potentially using many computers

Accurate uncertainty quantification (UQ) is a critical issue

Robustness of inferences also crucial
My focus - probability models

General probabilistic inference algorithms for complex data

We would like to be able to handle arbitrarily complex probability models

Algorithms scalable to huge data - potentially using many computers

Accurate uncertainty quantification (UQ) is a critical issue

Robustness of inferences also crucial

Particular emphasis on scientific applications - limited labeled data
Bayesian methods offer an attractive general approach for modeling complex data.
Bayesian methods offer an attractive general approach for modeling complex data.

Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$
Bayes approaches

Bayesian methods offer an attractive general approach for modeling complex data.

Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$
\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta) d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.
$$

Often $\theta$ is moderate to high-dimensional & the integral in the denominator is intractable.
Bayesian methods offer an attractive general approach for modeling complex data. Choosing a prior \( \pi(\theta) \) & likelihood \( L(Y^{(n)}|\theta) \), the posterior is

\[
\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.
\]

Often \( \theta \) is moderate to high-dimensional & the integral in the denominator is intractable. Accurate analytic approximations to the posterior have proven elusive outside of narrow settings.
Bayes approaches

Bayesian methods offer an attractive general approach for modeling complex data.

Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta) d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$ 

Often $\theta$ is moderate to high-dimensional & the integral in the denominator is intractable.

Accurate analytic approximations to the posterior have proven elusive outside of narrow settings.

Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms remain the standard.
Bayes approaches

Bayesian methods offer an attractive general approach for modeling complex data.

Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$
\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta) d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.
$$

Often $\theta$ is moderate to high-dimensional & the integral in the denominator is intractable.

Accurate analytic approximations to the posterior have proven elusive outside of narrow settings.

Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms remain the standard.

Scaling MCMC to big & complex settings challenging.
MCMC & Computational bottlenecks

MCMC constructs Markov chain with stationary distribution \( \pi_n(\theta | Y^{(n)}) \)
MCMC & Computational bottlenecks

- MCMC constructs Markov chain with stationary distribution
  \[ \pi_n(\theta | Y^{(n)}) \]
- A *transition kernel* is carefully chosen & iterative sampling proceeds
MCMC & Computational bottlenecks

- MCMC constructs Markov chain with stationary distribution \( \pi_n(\theta | Y^{(n)}) \)
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Time per iteration increases with # of parameters/unknowns
MCMC & Computational bottlenecks

- MCMC constructs Markov chain with stationary distribution \( \pi_n(\theta | Y^{(n)}) \)
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Time per iteration increases with # of parameters/unknowns
- Mixing worse as dimension of data increases

Motivation & background 6
MCMC constructs Markov chain with stationary distribution 
\[ \pi_n(\theta | Y^{(n)}) \]
A transition kernel is carefully chosen & iterative sampling 
proceeds
Time per iteration increases with # of parameters/unknowns
Mixing worse as dimension of data increases
Storing & basic processing on big data sets is problematic
MCMC & Computational bottlenecks

- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta | Y^{(n)})$
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Time per iteration increases with # of parameters/unknowns
- Mixing worse as dimension of data increases
- Storing & basic processing on big data sets is problematic
- Usually multiple likelihood and/or gradient evaluations at each iteration
Solutions

- Embarrassingly parallel (EP) MCMC: run MCMC in parallel for different subsets of data & combine.
Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.

- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.
Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.
- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.
- **Hybrid algorithms**: run MCMC for a subset of the parameters & use a fast estimate for the others.
Embarrassingly parallel (EP) MCMC: run MCMC in parallel for different subsets of data & combine.

Approximate MCMC: Approximate expensive to evaluate transition kernels.

Hybrid algorithms: run MCMC for a subset of the parameters & use a fast estimate for the others.

Designer MCMC: define clever kernels that solve mixing problems in high dimensions
Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.
- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.
- **Hybrid algorithms**: run MCMC for a subset of the parameters & use a fast estimate for the others.
- **Designer MCMC**: define clever kernels that solve mixing problems in high dimensions
- I’ll focus on EP-MCMC & aMCMC in remainder
Outline

Motivation & background

EP-MCMC

aMCMC

Discussion
Embarrassingly parallel MCMC

Divide large sample size $n$ data set into many smaller data sets stored on different machines.

- Draw posterior samples for each subset posterior in parallel.
- 'Magically' combine the results quickly & simply.
Embarrassingly parallel MCMC

Divide large sample size $n$ data set into many smaller data sets stored on different machines
- Draw posterior samples for each subset posterior in parallel
Embarrassingly parallel MCMC

- Divide large sample size $n$ data set into many smaller data sets stored on different machines
- Draw posterior samples for each subset posterior in parallel
- ‘Magically’ combine the results quickly & simply
Toy Example: Logistic Regression

\[ \text{pr}(y_i = 1|x_{i1}, \ldots, x_{ip}, \theta) = \frac{\exp \left( \sum_{j=1}^{p} x_{ij} \beta_j \right)}{1 + \exp \left( \sum_{j=1}^{p} x_{ij} \beta_j \right)}. \]

 Subset posteriors: ‘noisy’ approximations of full data posterior.
Toy Example: Logistic Regression

\[
\text{pr}(y_i = 1|x_{i1}, \ldots, x_{ip}, \theta) = \frac{\exp\left(\sum_{j=1}^{p} x_{ij} \beta_j\right)}{1 + \exp\left(\sum_{j=1}^{p} x_{ij} \beta_j\right)}.
\]

- Subset posteriors: 'noisy' approximations of full data posterior.
- 'Averaging' of subset posteriors reduces this 'noise' & leads to an accurate posterior approximation.
Stochastic Approximation

Full data posterior density of \textit{inid} data $Y^{(n)}$

$$
\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta)}{\int_{\Theta} \prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta) d\theta}.
$$

$\gamma = O(k)$ chosen to minimize approximation error
Stochastic Approximation

- Full data posterior density of \( \text{inid} \) data \( Y^{(n)} \)
  \[
  \pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta) d\theta}.
  \]

- Divide full data \( Y^{(n)} \) into \( k \) subsets of size \( m \):
  \( Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]}) \).

\( \gamma \) chosen to minimize approximation error
Stochastic Approximation

Full data posterior density of *inid* data $Y^{(n)}$

$$
\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^{n} p_i(y_i \mid \theta) \pi(\theta)}{\int_{\Theta} \prod_{i=1}^{n} p_i(y_i \mid \theta) \pi(\theta) \, d\theta}.
$$

Divide full data $Y^{(n)}$ into $k$ subsets of size $m$:

$Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]})$.

Subset posterior density for $j$th data subset

$$
\pi_m^{\gamma}(\theta \mid Y_{[j]}) = \frac{\prod_{i \in [j]} (p_i(y_i \mid \theta))^\gamma \pi(\theta)}{\int_{\Theta} \prod_{i \in [j]} (p_i(y_i \mid \theta))^\gamma \pi(\theta) \, d\theta}.
$$
Stochastic Approximation

Full data posterior density of $\text{inid}$ data $Y^{(n)}$

$$
\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta)}{\int_{\Theta} \prod_{i=1}^{n} p_i(y_i \mid \theta)\pi(\theta) d\theta}.
$$

Divide full data $Y^{(n)}$ into $k$ subsets of size $m$:

$Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]})$.

Subset posterior density for $j$th data subset

$$
\pi_{m}^{\gamma}(\theta \mid Y_{[j]}) = \frac{\prod_{i \in [j]} (p_i(y_i \mid \theta))^{\gamma} \pi(\theta)}{\int_{\Theta} \prod_{i \in [j]} (p_i(y_i \mid \theta))^{\gamma} \pi(\theta) d\theta}.
$$

$\gamma = O(k)$ - chosen to minimize approximation error
Barycenter in Metric Spaces

Space of probability measures $\mathcal{M}$
Barycenter in Metric Spaces

\[ \Pi_M = \arg\min_{\Pi \in \mathcal{M}} \sum_{i=1}^{n} \rho^2(\Pi, \Pi_i) \]

Space of probability measures \( \mathcal{M} \) with metric \( \rho \)
2-Wasserstein distance between $\mu, \nu \in \mathcal{P}_2(\Theta)$

$$W_2(\mu, \nu) = \inf \left\{ \left( \mathbb{E}[d^2(X, Y)] \right)^{\frac{1}{2}} : \text{law}(X) = \mu, \text{law}(Y) = \nu \right\}.$$
WAsserstein barycenter of Subset Posteriors (WASP)

Srivastava, Li & Dunson (2015)

2-Wasserstein distance between $\mu, \nu \in \mathcal{P}_2(\Theta)$

$$W_2(\mu, \nu) = \inf \left\{ \left( \mathbb{E}[d^2(X, Y)] \right)^{\frac{1}{2}} : \text{law}(X) = \mu, \text{law}(Y) = \nu \right\}.$$  

$\Pi^\gamma_m(\cdot | Y[j])$ for $j = 1, \ldots, k$ are combined through WASP

$$\Pi^\gamma_n(\cdot | Y^{(n)}) = \arg\min_{\Pi \in \mathcal{P}_2(\Theta)} \frac{1}{k} \sum_{j=1}^{k} W_2^2(\Pi, \Pi^\gamma_m(\cdot | Y[j])).$$  

[Agueh & Carlier (2011)]
**WAsserstein barycenter of Subset Posteriors (WASP)**

Srivastava, Li & Dunson (2015)

💡 2-Wasserstein distance between $\mu, \nu \in \mathcal{P}_2(\Theta)$

$$W_2(\mu, \nu) = \inf \left\{ \left( \mathbb{E}[d^2(X, Y)] \right)^{\frac{1}{2}} : \text{law}(X) = \mu, \text{law}(Y) = \nu \right\}.$$

💡 $\Pi^\gamma_m (\cdot \mid Y_{[j]})$ for $j = 1, \ldots, k$ are combined through WASP

$$\Pi^\gamma_n (\cdot \mid Y^{(n)}) = \arg\min_{\Pi \in \mathcal{P}_2(\Theta)} \frac{1}{k} \sum_{j=1}^{k} W_2^2 (\Pi, \Pi^\gamma_m (\cdot \mid Y_{[j]})).$$ [Agueh & Carlier (2011)]

💡 Plugging in $\hat{\Pi}^\gamma_m (\cdot \mid Y_{[j]})$ for $j = 1, \ldots, k$, a linear program (LP) can be used for fast estimation of an atomic approximation
Minimizing Wasserstein is solution to a discrete optimal transport problem

Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_1^j}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_2^l}$, and $M_{12} \in \mathbb{R}^{J_1 \times J_2}$ be the matrix of square differences in atoms $\{\theta_1^j\}, \{\theta_2^l\}$.

Optimal transport polytope: $T(a, b) = \text{set of doubly stochastic matrices with row sums } a \text{ and column sums } b$.

The objective is to find $T \in T(a, b)$ minimizing $\text{tr}(T M_{12})$.

For WASP, generalize to multimargin optimal transport problem - entropy smoothing has been used previously.

We can avoid such smoothing & use sparse LP solvers - negligible computation cost compared to sampling.
LP Estimation of WASP

Minimizing Wasserstein is solution to a discrete optimal transport problem

Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $M_{12} \in \mathcal{R}^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$. 

Entropy smoothing has been used previously - we can avoid such smoothing & use sparse LP solvers - negligible computation cost compared to sampling.
Minimizing Wasserstein is solution to a discrete optimal transport problem

Let \( \mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}} \), \( \nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}} \) & \( \mathbf{M}_{12} \in \mathbb{R}^{J_1 \times J_2} \) = matrix of square differences in atoms \( \{\theta_{1j}\}, \{\theta_{2l}\} \).

Optimal transport polytope: \( \mathcal{T}(\mathbf{a}, \mathbf{b}) = \) set of doubly stochastic matrices w/ row sums \( \mathbf{a} \) & column sums \( \mathbf{b} \).
LP Estimation of WASP

Minimizing Wasserstein is solution to a discrete optimal transport problem

Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $M_{12} \in \mathbb{R}^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.

Optimal transport polytope: $\mathcal{T}(a, b) = \text{set of doubly stochastic matrices w/ row sums } a \text{ & column sums } b$

Objective is to find $T \in \mathcal{T}(a, b)$ minimizing $\text{tr}(T^T M_{12})$
Minimizing Wasserstein is solution to a discrete optimal transport problem

Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $M_{12} \in \mathbb{R}^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.

Optimal transport polytope: $\mathcal{T}(a, b)$ = set of doubly stochastic matrices w/ row sums $a$ & column sums $b$

Objective is to find $T \in \mathcal{T}(a, b)$ minimizing $\text{tr}(T^T M_{12})$

For WASP, generalize to multimargin optimal transport problem - entropy smoothing has been used previously
LP Estimation of WASP

Minimizing Wasserstein is solution to a discrete optimal transport problem.

Let \( \mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_1j}, \ \nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_2l} \) & \( \mathbf{M}_{12} \in R^{J_1 \times J_2} \) = matrix of square differences in atoms \( \{\theta_1j\}, \{\theta_2l\} \).

Optimal transport polytope: \( \mathcal{T}(a, b) \) = set of doubly stochastic matrices w/ row sums \( a \) & column sums \( b \).

Objective is to find \( T \in \mathcal{T}(a, b) \) minimizing \( \text{tr}(T^T \mathbf{M}_{12}) \).

For WASP, generalize to multimargin optimal transport problem - entropy smoothing has been used previously.

We can avoid such smoothing & use sparse LP solvers - negligible computation cost compared to sampling.
Theorem (Subset Posteriors)

Under “usual” regularity conditions, there exists a constant $C_1$ independent of subset posteriors, such that for large $m$,

$$\mathbb{E}_{P[j]} W_2^2 \{ \prod_{\gamma}^m (\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot) \} \leq C_1 \left( \frac{\log^2 m}{m} \right)^{\frac{1}{\alpha}} j = 1, \ldots, k,$$
WASP: Theorems

Theorem (Subset Posteriors)

Under “usual” regularity conditions, there exists a constant $C_1$ independent of subset posteriors, such that for large $m$,

$$\mathbb{E}_{P_{\theta_0}^{[j]}} W_2^2 \{ \Pi_m^\gamma (\cdot | Y_{[j]}), \delta_{\theta_0}(\cdot) \} \leq C_1 \left( \frac{\log^2 m}{m} \right)^{\frac{1}{\alpha}} \quad j = 1, \ldots, k,$$

Theorem (WASP)

Under “usual” regularity conditions and for large $m$,

$$W_2 \left\{ \Pi_n^\gamma (\cdot | Y^{(n)}), \delta_{\theta_0}(\cdot) \right\} = O_{P_{\theta_0}^{(n)}} \left( \sqrt{\frac{\log^{2/\alpha} m}{km^{1/\alpha}}} \right).$$
Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2015)

 נראה(point & interval estimates for different 1-d functionals - multidimensional posterior difficult to interpret)
**Simple & Fast Posterior Interval Estimation (PIE)**

Li, Srivastava & Dunson (2015)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2015)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2015)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*
Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2015)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*
- Strong theory showing accuracy of the resulting approximation
Simple & Fast Posterior Interval Estimation (PIE)

*Li, Srivastava & Dunson (2015)*

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*
- Strong theory showing accuracy of the resulting approximation
- Can be implemented in *STAN*, which allows powered likelihoods
We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$.
We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$

As subset sample size $m$ increases, $W_2$ distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$
Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$.
- As subset sample size $m$ increases, $W_2$ distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$.
- Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0, 1)$, so $m$ can increase very slowly relative to $k$ (recall $n = mk$).
We show 1-d WASP $\Pi_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$

As subset sample size $m$ increases, $W_2$ distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$

Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0,1)$, so $m$ can increase very slowly relative to $k$ (recall $n = mk$)

Their biases, variances, quantiles only differ in high orders of the total sample size
We show 1-d WASP $\Pi_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$.

As subset sample size $m$ increases, $W_2$ distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$.

Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0,1)$, so $m$ can increase very slowly relative to $k$ (recall $n = mk$).

Their biases, variances, quantiles only differ in high orders of the total sample size.

**Conditions:** standard, mild conditions on likelihood + prior finite 2nd moment & uniform integrability of subset posteriors.
Results

We have implemented for rich variety of data & models...
We have implemented for rich variety of data & models
Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
Results

We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB
Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB
- WASP/PIE is much faster than MCMC & highly accurate
Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB
- WASP/PIE is much faster than MCMC & highly accurate
- Carefully designed VB implementations often do very well
Outline

Motivation & background

EP-MCMC

aMCMC

Discussion
Different way to speed up MCMC - replace expensive transition kernels with approximations
Different way to speed up MCMC - replace expensive transition kernels with approximations. For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data.
Different way to speed up MCMC - replace expensive transition kernels with approximations

For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data

Can potentially vastly speed up MCMC sampling in high-dimensional settings
Different way to speed up MCMC - replace expensive transition kernels with approximations

For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data

Can potentially vastly speed up MCMC sampling in high-dimensional settings

Original MCMC sampler converges to a stationary distribution corresponding to the exact posterior
Different way to speed up MCMC - replace expensive transition kernels with approximations

For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data

Can potentially vastly speed up MCMC sampling in high-dimensional settings

Original MCMC sampler converges to a stationary distribution corresponding to the exact posterior

Not clear what happens when we start substituting in approximations - may diverge etc
aMCMC Overview

* aMCMC is used routinely in an essentially *ad hoc* manner
aMCMC Overview

- aMCMC is used routinely in an essentially *ad hoc* manner
- Our goal: obtain theory guarantees & use these to target design of algorithms

Outline:
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing
- ‘Exact’ chain converges to stationary distribution corresponding to exact posterior
- Approximate kernel in exact chain with more computationally tractable alternative
aMCMC Overview

- aMCMC is used routinely in an essentially *ad hoc* manner
- Our goal: obtain theory guarantees & use these to target design of algorithms
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing
aMCMC Overview

- aMCMC is used routinely in an essentially *ad hoc* manner
- Our goal: obtain theory guarantees & use these to target design of algorithms
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing
- ‘exact’ chain converges to stationary distribution corresponding to exact posterior
aMCMC Overview

- aMCMC is used routinely in an essentially *ad hoc* manner
- Our goal: obtain theory guarantees & use these to target design of algorithms
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing
- ‘exact’ chain converges to stationary distribution corresponding to exact posterior
- Approximate kernel in exact chain with more computationally tractable alternative
Sketch of theory

Define $s_\epsilon = \tau_1(\mathcal{P}) / \tau_1(\mathcal{P}_\epsilon) = \text{computational speed-up}$, $\tau_1(\mathcal{P}) =$ time for one step with transition kernel $\mathcal{P}$
Define $s_\epsilon = \tau_1(\mathcal{P}) / \tau_1(\mathcal{P}_\epsilon) = \textit{computational speed-up}$, $\tau_1(\mathcal{P}) = \text{time for one step with transition kernel } \mathcal{P}$

**Interest**: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_{\Theta} f(\theta) \Pi(d\theta|x)$
Sketch of theory

Define $s_\epsilon = \tau_1(\mathcal{P}) / \tau_1(\mathcal{P}_\epsilon) = \text{computational speed-up}$, $\tau_1(\mathcal{P}) =$ time for one step with transition kernel $\mathcal{P}$

Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_\Theta f(\theta) \Pi(d\theta|x)$

We provide tight, finite sample bounds on $L_2$ error.
Define $s_c = \tau_1(\mathcal{P}) / \tau_1(\mathcal{P}_c) = \text{computational speed-up}$, $\tau_1(\mathcal{P}) =$ time for one step with transition kernel $\mathcal{P}$

Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_{\Theta} f(\theta) \Pi(d\theta|x)$

We provide tight, finite sample bounds on $L_2$ error

aMCMC estimators win for low computational budgets but have asymptotic bias
Sketch of theory

Define \( s_\varepsilon = \tau_1(\mathcal{P}) / \tau_1(\mathcal{P}_\varepsilon) = \text{computational speed-up} \), \( \tau_1(\mathcal{P}) = \text{time for one step with transition kernel } \mathcal{P} \)

Interest: optimizing computational time-accuracy tradeoff for estimators of

\[
\Pi f = \int_{\Theta} f(\theta) \Pi(d\theta|x)
\]

We provide tight, finite sample bounds on \( L_2 \) error

aMCMC estimators win for low computational budgets but have asymptotic bias

Often larger approximation error \( \rightarrow \) larger \( s_\varepsilon \) & rougher approximations are better when speed super important
Replace the full data likelihood with

$$L_\epsilon(x | \theta) = \left( \prod_{i \in V} L(x_i | \theta) \right)^{N/|V|},$$

for randomly chosen subset $V \subset \{1, \ldots, n\}$. 
Ex 1: Approximations using subsets

Replace the full data likelihood with

$$L_{\epsilon}(x \mid \theta) = \left( \prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|},$$

for randomly chosen subset $V \subset \{1, \ldots, n\}$.

Applied to Pólya-Gamma data augmentation for logistic regression.
Ex 1: Approximations using subsets

Replace the full data likelihood with

\[ L_\epsilon(x \mid \theta) = \left( \prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|} \]

for randomly chosen subset \( V \subset \{1, \ldots, n\} \).

Applied to Pólya-Gamma data augmentation for logistic regression

Different \( V \) at each iteration – trivial modification to Gibbs
Ex 1: Approximations using subsets

Replace the full data likelihood with

\[ L_\epsilon(x \mid \theta) = \left( \prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|}, \]

for randomly chosen subset \( V \subset \{1, \ldots, n\} \).

Applied to Pólya-Gamma data augmentation for logistic regression

Different \( V \) at each iteration – trivial modification to Gibbs

Assumptions hold with high probability for subsets > minimal size (wrt distribution of subsets, data & kernel).
Application to SUSY dataset

$n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
Application to SUSY dataset

- \( n = 5,000,000 \) (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from \( |V| = 1,000 \) to \( 4,500,000 \)

Rate at which loss \( \to 0 \) with \( \epsilon \) heavily dependent on loss

For small computational budget & focus on posterior mean estimation, small subsets preferred

As budget increases & loss focused more on tails (e.g., for interval estimation), optimal \( |V| \) increases
Application to SUSY dataset

- \( n = 5,000,000 \) (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from \( |V| = 1,000 \) to 4,500,000
- Considered different losses as function of \( |V| \)
Application to SUSY dataset

$n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates

Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$

Considered different losses as function of $|V|$

Rate at which loss $\rightarrow 0$ with $\epsilon$ heavily dependent on loss
Application to SUSY dataset

$n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates

Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$

Considered different losses as function of $|V|$

Rate at which loss $\rightarrow 0$ with $\epsilon$ heavily dependent on loss

For small computational budget & focus on posterior mean estimation, small subsets preferred
Application to SUSY dataset

- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$
- Considered different losses as function of $|V|$
- Rate at which loss $\to 0$ with $\epsilon$ heavily dependent on loss
- For small computational budget & focus on posterior mean estimation, small subsets preferred
- As budget increases & loss focused more on tails (e.g., for interval estimation), optimal $|V|$ increases
Application 2: Mixture models & tensor factorizations

We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

a very useful model for multivariate categorical data.
We also considered a nonparametric Bayes model:

\[
\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hcj}^{(j)},
\]

a very useful model for multivariate categorical data

Dunson & Xing (2009) - a data augmentation Gibbs sampler
We also considered a nonparametric Bayes model:

\[
\Pr(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi^{(j)}_{hc_j},
\]

a very useful model for multivariate categorical data

Dunson & Xing (2009) - a data augmentation Gibbs sampler

Sampling latent classes computationally prohibitive for huge \( n \)
We also considered a nonparametric Bayes model:

\[
pr(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},
\]

a very useful model for multivariate categorical data.

Dunson & Xing (2009) - a data augmentation Gibbs sampler.

Sampling latent classes computationally prohibitive for huge \( n \).

Use adaptive Gaussian approximation - avoid sampling individual latent classes.
We also considered a nonparametric Bayes model:

\[ pr(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)}, \]

a very useful model for multivariate categorical data

Dunson & Xing (2009) - a data augmentation Gibbs sampler

Sampling latent classes computationally prohibitive for huge \( n \)

Use adaptive Gaussian approximation - avoid sampling individual latent classes

We have shown Assumptions 1-2, Assumption 2 result more general than this setting
We also considered a nonparametric Bayes model:

\[
\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},
\]

a very useful model for multivariate categorical data

- Dunson & Xing (2009) - a data augmentation Gibbs sampler
- Sampling latent classes computationally prohibitive for huge \( n \)
- Use adaptive Gaussian approximation - avoid sampling individual latent classes
- We have shown Assumptions 1-2, Assumption 2 result more general than this setting
- Improved computation performance for large \( n \)
Application 3: Low rank approximation to GP

Gaussian process regression, $y_i = f(x_i) + \eta_i, \eta_i \sim N(0, \sigma^2)$
Application 3: Low rank approximation to GP

Gaussian process regression, $y_i = f(x_i) + \eta_i, \eta_i \sim N(0, \sigma^2)$

$f \sim GP$ prior with covariance $\tau^2 \exp(-\phi||x_1 - x_2||^2)$
Application 3: Low rank approximation to GP

- Gaussian process regression, \( y_i = f(x_i) + \eta_i, \eta_i \sim N(0, \sigma^2) \)
- \( f \sim GP \) prior with covariance \( \tau^2 \exp(-\phi \|x_1 - x_2\|^2) \)
- Discrete-uniform on \( \phi \) & gamma priors on \( \tau^{-2}, \sigma^{-2} \)
Application 3: Low rank approximation to GP

- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi ||x_1 - x_2||^2)$
- Discrete-uniform on $\phi$ & gamma priors on $\tau^{-2}, \sigma^{-2}$
- Marginal MCMC sampler updates $\phi, \tau^{-2}, \sigma^{-2}$
Application 3: Low rank approximation to GP

- Gaussian process regression, \( y_i = f(x_i) + \eta_i, \eta_i \sim N(0, \sigma^2) \)
- \( f \sim GP \) prior with covariance \( \tau^2 \exp(-\phi||x_1 - x_2||^2) \)
- Discrete-uniform on \( \phi \) & gamma priors on \( \tau^{-2}, \sigma^{-2} \)
- Marginal MCMC sampler updates \( \phi, \tau^{-2}, \sigma^{-2} \)
- We show Assumption 1 holds under mild regularity conditions on “truth”, Assumption 2 holds for partial rank-\( r \) eigen approximation to \( \Sigma \)
Application 3: Low rank approximation to GP

- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi ||x_1 - x_2||^2)$
- Discrete-uniform on $\phi$ & gamma priors on $\tau^{-2}, \sigma^{-2}$
- Marginal MCMC sampler updates $\phi, \tau^{-2}, \sigma^{-2}$
- We show Assumption 1 holds under mild regularity conditions on “truth”, Assumption 2 holds for partial rank-$r$ eigen approximation to $\Sigma$
- Less accurate approximations clearly superior in practice for small computational budget
Achieving uniform control of approximation error $\epsilon$ requires approximations adaptive to current state of chain.
Applications: General Conclusions

Achieving uniform control of approximation error $\epsilon$ requires approximations \textbf{adaptive} to current state of chain.

More accurate approximations needed farther from high probability region of posterior; good as chain rarely there.
Applications: General Conclusions

- Achieving uniform control of approximation error $\epsilon$ requires approximations **adaptive** to current state of chain.
- More accurate approximations needed farther from high probability region of posterior; good as chain rarely there.
- Approximations to conditionals of vector parameters are highly sensitive to 2nd moment.
Achieving uniform control of approximation error $\epsilon$ requires approximations adaptive to current state of chain.

More accurate approximations needed farther from high probability region of posterior; good as chain rarely there.

Approximations to conditionals of vector parameters are highly sensitive to 2nd moment.

Smaller condition numbers for the covariance matrix of vector parameters mean less accurate approximations can be used.
Outline

Motivation & background

EP-MCMC

aMCMC

Discussion
Discussion

Proposed very general classes of scalable Bayes algorithms
Proposed very general classes of scalable Bayes algorithms

EP-MCMC & aMCMC - fast & scalable with guarantees
Discussion

- Proposed very general classes of scalable Bayes algorithms
- EP-MCMC & aMCMC - fast & scalable with guarantees
- Interest in improving theory - avoid reliance on asymptotics in EP-MCMC & weaken assumptions in aMCMC
Proposed very general classes of scalable Bayes algorithms

EP-MCMC & aMCMC - fast & scalable with guarantees

Interest in improving theory - avoid reliance on asymptotics in EP-MCMC & weaken assumptions in aMCMC

Useful to combine algorithms - e.g., run aMCMC for each subset
Discussion

- Proposed very general classes of scalable Bayes algorithms
- EP-MCMC & aMCMC - fast & scalable with guarantees
- Interest in improving theory - avoid reliance on asymptotics in EP-MCMC & weaken assumptions in aMCMC
- Useful to combine algorithms - e.g., run aMCMC for each subset
- By looking at algorithms through our theory lens, suggests new & improved algorithms
Discussion

- Proposed very general classes of scalable Bayes algorithms
- EP-MCMC & aMCMC - fast & scalable with guarantees
- Interest in improving theory - avoid reliance on asymptotics in EP-MCMC & weaken assumptions in aMCMC
- Useful to combine algorithms - e.g., run aMCMC for each subset
- By looking at algorithms through our theory lens, suggests new & improved algorithms
- Also, very interested in hybrid frequentist-Bayes algorithms
Hybrid high-dimensional density estimation

\[ y_i = (y_{i1}, \ldots, y_{ip})^T \sim f \text{ with } p \text{ large & } f \text{ an unknown density} \]

Ye, Canale & Dunson (2016, AISTATS)
Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

\[ y_i = (y_{i1}, \ldots, y_{ip})^T \sim f \text{ with } p \text{ large & } f \text{ an unknown density} \]

- Potentially use Dirichlet process mixtures of factor models

Discussion
Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

\[ y_i = (y_{i1}, \ldots, y_{ip})^T \sim f \text{ with } p \text{ large & } f \text{ an unknown density} \]

- Potentially use Dirichlet process mixtures of factor models
- Approach doesn’t scale well at all with \( p \)
Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- \( y_i = (y_{i1}, \ldots, y_{ip})^T \sim f \) with \( p \) large & \( f \) an unknown density
- Potentially use Dirichlet process mixtures of factor models
- Approach doesn’t scale well at all with \( p \)
- Instead use hybrid of Gibbs sampling & fast multiscale SVD
Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

\[ y_i = (y_{i1}, \ldots, y_{ip})^T \sim f \text{ with } p \text{ large & } f \text{ an unknown density} \]

- Potentially use Dirichlet process mixtures of factor models
- Approach doesn’t scale well at all with \( p \)
- Instead use hybrid of Gibbs sampling & fast multiscale SVD
- Scalable, excellent mixing & empirical/predictive performance
In the above we have put aside the mixing issues that can arise in big samples
In the above we have put aside the mixing issues that can arise in big samples.

Slow mixing → we need many more MCMC samples for the sample MC error.
What about mixing?

In the above we have put aside the mixing issues that can arise in big samples.

- Slow mixing → we need many more MCMC samples for the sample MC error.
- Common data augmentation algorithms for discrete data fail badly for large imbalanced data (Johndrow et al. 2016)
In the above we have put aside the mixing issues that can arise in big samples.

Slow mixing → we need many more MCMC samples for the sample MC error.

Common data augmentation algorithms for discrete data fail badly for large imbalanced data (Johndrow et al. 2016).

But such problems can be fixed via calibration (Duan et al. 2016).
In the above we have put aside the mixing issues that can arise in big samples

Slow mixing $\rightarrow$ we need many more MCMC samples for the sample MC error

Common data augmentation algorithms for discrete data fail badly for large imbalanced data \((\text{Johndrow et al. 2016})\)

But such problems can be fixed via calibration \((\text{Duan et al. 2016})\)

Interesting area for further research
Primary References


