Markov Chain Monte Carlo

After six decades of continual development, MCMC has proven to be a very powerful and typically unique computational tool for analyzing data of complex structures. However, it has encountered intrinsic difficulties in dealing with large-scale problems:

- Big sample size (N): Repeated scans of the full dataset
- Large dimension (p): Curse of dimensionality
- Big N + Large p.

Stochastic Gradient MCMC (SGMCMC)

- Stochastic gradient Langevin dynamics (SGLD)
- stochastic gradient Hamiltonian Monte Carlo
- stochastic gradient thermostats
- stochastic gradient Fisher scoring
- preconditioned SGLD

Refer to Ma et al. (2015) for a complete framework of these algorithms.

Attractive Features of SGMCMC

- Their transitional kernel is defined by a stochastic differential equation with the moving direction guided by the gradient of the log-target density function.
- Only an inaccurate gradient of the log-target density function, which can be evaluated on a mini-batch of data, is required for each transition.

These two distinctive features make SGMCMC algorithms highly scalable, exhibiting similar costs as many stochastic optimization algorithms.

Difficulties with SGMCMC

In their current setup, the SGMCMC algorithms are only applicable to a small class of problems for which

(i) the parameter space has a fixed dimension; and

(ii) the log-posterior density is differentiable with respect to the parameters.

Other Mini-batch Methods

- Split-and-merge: split the big dataset into a number of smaller subsets, conduct the Bayesian analysis for each subset separately (running in parallel), and then aggregate the posterior samples generated for each subset to make correct inference for the full data posterior. Existing methods include consensus Monte Carlo (Scott et al., 2016), WASP Monte Carlo (Srivastava et al., 2018), and double parallel Monte Carlo (Xue and Liang 2018).
- mini-batch Metropolis-Hastings sampler: Korattikara et al. (2014), Bootstrap MH (Liang et al., 2016), Bardenet et al. (2017). These samplers are often biased due to an inaccurate assessment for uncertainty of the acceptance/rejection probability caused by subsampling.

Stochastic gradient Langevin dynamics (SGLD)

- Let $\boldsymbol{X}_N = (X_1, X_2, \ldots, X_N)$ denote a set of N independent and identically distributed (iid) observations.
- Let $L(\boldsymbol{\theta}|\boldsymbol{X}_N) = \sum_{i=1}^{N} \log f(x_i|\boldsymbol{\theta}) + \log \pi(\boldsymbol{\theta})$ denote the log-posterior density of a statistical model parameterized by $\boldsymbol{\theta}$, where $f(x_i|\boldsymbol{\theta})$ is the density function of X_i, and $\pi(\boldsymbol{\theta})$ is the prior density of $\boldsymbol{\theta}$.

The SGLD algorithm iterates via the equation

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \frac{\epsilon_{t+1}}{2} \partial_{\boldsymbol{\theta}} \widehat{L}(\boldsymbol{\theta}_t|\boldsymbol{X}_N) + \sqrt{\epsilon_{t+1}\tau}\eta_{t+1}, \quad \eta_{t+1} \sim N(0, I_d), \tag{1}$$

where d is the dimension of $\boldsymbol{\theta}$, I_d is a $d \times d$-identity matrix, ϵ_{t+1} is the step size (also known as the learning rate), τ is the temperature, and $\hat{L}(\boldsymbol{\theta}|\boldsymbol{X}_N)$ denotes an estimate of $L(\boldsymbol{\theta}|\boldsymbol{X}_N)$.

Stochastic gradient Langevin dynamics (SGLD)

Under the assumption that the estimation error

$$\xi_t = \hat{L}(\boldsymbol{\theta}_t|\boldsymbol{X}_N) - L(\boldsymbol{\theta}_t|\boldsymbol{X}_N)$$

is a white noise or martingale difference noise, Sato and Nakagawa (2014) proved that $\boldsymbol{\theta}_t$ converges weakly to the posterior

$$\pi(\boldsymbol{\theta}|\boldsymbol{X}_N) \propto \exp\{L(\boldsymbol{\theta}|\boldsymbol{X}_N)/\tau\}, \quad \text{as } \epsilon_t \to 0.$$

Stochastic gradient Langevin dynamics (SGLD)

Welling and Teh (2011) proposed to estimate $L(\boldsymbol{\theta}|\boldsymbol{X}_N)$ using a mini-batch sample by setting

$$\widehat{L}(\boldsymbol{\theta}|\boldsymbol{X}_N) = \frac{N}{n}\sum_{i=1}^{n} \log f(x_i^*|\boldsymbol{\theta}) + \log \pi(\boldsymbol{\theta}),$$

where $\{x_1^*, x_2^*, \ldots, x_n^*\}$ denotes a subsample of size n, and set the step size to a non-increasing sequence satisfying

$$\sum_{t=0}^{\infty} \epsilon_{t+1} = \infty, \quad \sum_{t=0}^{\infty} \epsilon_{t+1}^2 < \infty,$$

as in a stochastic approximation algorithm.

Stochastic gradient Langevin dynamics (SGLD)

For the constant stepsize SGLD algorithm, Sato and Nakagawa (2014) established the weak convergence for the path average, which implies that for any continuously differentiable and polynomial growth function $\rho(\boldsymbol{\theta})$,

$$\frac{1}{T}\sum_{t=1}^{T}\rho(\boldsymbol{\theta}_t) \xrightarrow{p} E_\pi \rho(\boldsymbol{\theta}), \tag{2}$$

where $\xrightarrow{p}$ denotes convergence in probability.

Stochastic gradient Langevin dynamics (SGLD)

About the choice of ϵ and the subsample size n:

- According to Dalalyan and Karagulyan (2017), they should be chosen such that $\epsilon Np/n = o(1)$ under the assumptions that n and p will increase with N, while ϵ will decrease with N, where p denotes the dimension of $\boldsymbol{\theta}$.
- Based on practical experience, Nagapetyan et al. (2017) suggested to choose $\epsilon < 1/N$ and $n \approx N^2\epsilon/2$.

Variable selection for linear regression

For illustration, we consider the problem of variable selection for linear regression:

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\beta} + \sigma\boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{y}$ is an N-dimensional response vector, $\boldsymbol{Z}$ is an $N \times p$ design matrix, $\boldsymbol{\beta}$ is a p-dimensional vector of regression coefficients, σ is the standard deviation, and $\boldsymbol{\varepsilon} \sim N(0, I_N)$.

This problem has been well studied in Bayesian statistics, for which a variety of methods has been developed including SSVS (George and McCullloch, 1993), Bayesian Lasso (Park and Casella, 2008), split-and-merge (Song and Liang, 2015), among others.

When N is large, repeated scans of the full dataset during iterations can make the Bayesian methods far less than practical.

Extended SGLD Algorithm

- Let $\boldsymbol{\theta}$ denote a latent vector which links the model S and the associated parameter vector $\boldsymbol{\beta}_S$ via the relation

$$\boldsymbol{\beta}_S = \boldsymbol{\theta} * \boldsymbol{\gamma}_S, \tag{4}$$

  where $\boldsymbol{\gamma}_S$ is a binary vector indicating the variables included in the model S, and $*$ denotes elementwise multiplication.
- Let $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_{[-S]}$ denote the elements of $\boldsymbol{\theta}$ with the corresponding variables belonging to S and not belonging to S, respectively.
- Let $\pi(\boldsymbol{\gamma}_S)$ denote the prior probability of model S.
- Let $\pi(\boldsymbol{\theta}_S)$ and $\pi(\boldsymbol{\theta}_{[-S]})$ denote the prior density function of $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_{[-S]}$, respectively. Assume that $\pi(\boldsymbol{\theta}|\boldsymbol{\gamma}_S) = \pi(\boldsymbol{\theta}_S)\pi(\boldsymbol{\theta}_{[-S]})$.

Extended SGLD Algorithm

Lemma 1. Assume $\pi(\boldsymbol{\theta}|\gamma_S) = \pi(\boldsymbol{\theta}_S)\pi(\boldsymbol{\theta}_{[-S]})$. If (4) holds, then

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \boldsymbol{X}_N) = \sum_{\gamma_S} \pi(\gamma_S \mid \boldsymbol{\theta}, \boldsymbol{X}_N) \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \gamma_S, \boldsymbol{X}_N), \quad (5)$$

where $\pi(\gamma_S \mid \boldsymbol{\theta}, \boldsymbol{X}_N)$ and $\pi(\boldsymbol{\theta} \mid \gamma_S, \boldsymbol{X}_N)$ denote the conditional posterior of γ_S and $\boldsymbol{\theta}$, respectively.

Lemma 1 provides us a Monte Carlo estimate for $\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \boldsymbol{X}_N)$ by averaging over the samples drawn from the posterior distribution $\pi(\gamma_S|\boldsymbol{\theta}, \boldsymbol{X}_N)$.

eSGLD for variable selection: Plain version(Algorithm 1)

(i) (Simulating models) Simulate models $\gamma_{S_1}^{(t)}, \ldots, \gamma_{S_m}^{(t)}$ according to the conditional posterior $\pi(\gamma_S|\boldsymbol{\theta}^{(t)}, \boldsymbol{X}_N)$, where $\boldsymbol{\theta}^{(t)}$ denotes the sample of $\boldsymbol{\theta}$ drawn at iteration t.

(ii) (Updating $\boldsymbol{\theta}$) Update $\boldsymbol{\theta}^{(t)}$ by setting

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\epsilon_{t+1}}{2m} \sum_{k=1}^{m} \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}^{(t)}|\gamma_{S_k}^{(t)}, \boldsymbol{X}_N) + \sqrt{\epsilon_{t+1}\tau}\eta_{t+1},$$

where $\eta_{t+1} \sim N(0, I_d)$, and ϵ_{t+1} can be set to a small constant or decrease as in the SGLD algorithm.

eSGLD with subsampling

Without loss of generality, we assume that N is a multiple of n, i.e., N/n is an integer. Let $\boldsymbol{X}_{n,N} = \{X_n, \ldots, X_n\}$ denote a duplicated dataset from the subsample, whose total sample size is also N. Therefore

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \boldsymbol{X}_{n,N}) &= \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{X}_{n,N} \mid \boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}) \\ &= \sum_{\gamma_S} \pi(\gamma_S \mid \boldsymbol{\theta}, \boldsymbol{X}_{n,N}) \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \gamma_S, \boldsymbol{X}_{n,N}), \end{aligned} \tag{6}$$

where the first equality implies $\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \boldsymbol{X}_{n,N})$ is an unbiased estimator of $\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} \mid \boldsymbol{X}_N)$.

eSGLD with subsampling (Algorithm 2)

(i) (Subsampling) Draw a subsample of size n from the full dataset $\boldsymbol{X}_N$ at random, and denote the subsample by $\boldsymbol{X}_n^{(t)}$, where t indexes the iteration.

(ii) (Simulating models) Simulate models $\gamma_{S_1,n}^{(t)}, \ldots, \gamma_{S_m,n}^{(t)}$ from the conditional posterior $\pi(\gamma_S|\boldsymbol{\theta}^{(t)}, \boldsymbol{X}_{n,N}^{(t)})$, where $\boldsymbol{\theta}^{(t)}$ denotes the sample of $\boldsymbol{\theta}$ drawn at iteration t.

(iii) (Updating $\boldsymbol{\theta}$) Update $\boldsymbol{\theta}^{(t)}$ by setting

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\epsilon_{t+1}}{2m} \sum_{k=1}^{m} \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}^{(t)}|\gamma_{S_k}^{(t)}, \boldsymbol{X}_{n,N}^{(t)}) + \sqrt{\epsilon_{t+1}\tau}\eta_{t+1},$$

where $\eta_{t+1} \sim N(0, I_d)$, and ϵ_{t+1} can be set to a small constant or decrease as in the SGLD algorithm.

Validity of eSGLD for variable selection

Let $R_n^{(t)}$ denote the index of the subsamples drawn at iteration t. The joint distribution of the eSGLD samples is given by

$$\pi(\boldsymbol{\theta}, R_n, \boldsymbol{\gamma}_S | \boldsymbol{X}_N) = \pi(\boldsymbol{\gamma}_S | \boldsymbol{\theta}, \boldsymbol{X}_{n,N}) P(R_n) \pi(\boldsymbol{\theta} | \boldsymbol{X}_N).$$

Suppose that we are interested in estimating the probability $\pi(\boldsymbol{\gamma}_S | \boldsymbol{X}_N)$. Based on the MCMC output, it can be estimated by

$$\widehat{\pi}(\boldsymbol{\gamma}_S | \boldsymbol{X}_N) = \frac{1}{T} \sum_{t=1}^{T} I(\boldsymbol{\gamma}_S^{(t)} = \boldsymbol{\gamma}_S). \tag{7}$$

Validity of eSGLD for variable selection

Conditioned on $(\boldsymbol{\theta}^{(t)}, R_n^{(t)})$, $t = 1, 2, \ldots, T$, i.e., taking the conditional expectation for each indicator $I(S_t = S)$, we get an alternative estimator

$$\begin{aligned} \widetilde{\pi}(\gamma_S|\boldsymbol{X}_N) &= \frac{1}{T}\sum_{t=1}^{T} E(I(\gamma_S^{(t)} = \gamma_S)|\boldsymbol{\theta}^{(t)}, \boldsymbol{X}_{n,N}^{(t)}) \\ &= \frac{1}{T}\sum_{t=1}^{T} \pi(\gamma_S|\boldsymbol{\theta}^{(t)}, \boldsymbol{X}_{n,N}^{(t)}). \end{aligned} \tag{8}$$

Validity of eSGLD for variable selection

(A.1) The posterior $\pi_* = \pi(\boldsymbol{\theta}|\boldsymbol{X}_N)$ is strongly log-concave:

$$f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}') - \nabla f(\boldsymbol{\theta}')^T(\boldsymbol{\theta} - \boldsymbol{\theta}') \geq \frac{q_N}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta, \tag{9}$$

where $f(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta}|\boldsymbol{X}_N)$, and q_N is a positive number depending on N and satisfying the limit $\lim_{N\to\infty} q_N = \infty$. In general, we have $q_N = O(N)$.

(A.2) The posterior $\pi_* = \pi(\boldsymbol{\theta}|\boldsymbol{X}_N)$ is gradient-Lipschitz, i.e.,

$$\|\nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta}')\|_2 \leq Q_N\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta, \tag{10}$$

where $Q_N \leq c_0 N$ is a positive constant for some constant c_0. Note that we must have $q_N \leq N_N$.

Validity of eSGLD for variable selection

(A.3) $E\{\max_{S\in\mathcal{S}} E_{\boldsymbol{X}_N}[\|\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\gamma_S, \boldsymbol{X}_N)\|^2|\boldsymbol{\theta}]\} = O(N^2 p)$, where $E_{\boldsymbol{X}_N}$ denotes expectation with respect to the distribution of $\boldsymbol{X}_N$.

(A.4) Let $L_N(\gamma_S, \boldsymbol{\theta}) = \log p(X_N|\gamma_S, \boldsymbol{\theta})/N$ and let $\{L_N^{(i)}(\boldsymbol{\theta}) : i = 1, 2, \ldots, |\mathcal{S}\|\}$ be the descending order statistics of $L_N(\gamma_S, \boldsymbol{\theta})$'s, where $\mathcal{S}$ denotes the set of all possible models with the cordiality $|\mathcal{S}| = 2^p$. Assume that there exists a constant $\delta > 0$ such that $\inf_{\theta\in\Theta}(L_N^{(1)}(\theta) - L_N^{(2)}(\theta)) \geq \delta$.

Validity of eSGLD for variable selection

Theorem 1. [Asymptotic consistency of subsampling estimators] Assume the conditions (A.1)-(A.3), which are given in the Appendix.

(i) If m, p, n are increasing with N such that $N/n \geq m \succ \sqrt{p}$, and a constant learning rate $\epsilon \prec \frac{mn}{Np}$ is used, then $W_2(\pi_t, \pi_*) \to 0$ as $t \to \infty$, where π_t denotes the distribution of $\boldsymbol{\theta}^{(t)}$, $\pi_*(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\boldsymbol{X}_N)$, and $W_2(\cdot, \cdot)$ denotes the second order Wasserstein distance between two distributions.

(ii) If $\rho(\boldsymbol{\theta})$ is α-Lipschitz for some constant $\alpha > 0$, then $\sum_{t=1}^{T} \rho(\boldsymbol{\theta}^{(t)})/T \xrightarrow{p} \pi_*(\rho)$ as $T \to \infty$, where $\xrightarrow{p}$ denotes convergence in probability and $\pi_*(\rho) = \int_\Theta \rho(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{X}_N)d\boldsymbol{\theta}$.

(iii) If condition (A.4) and $n \succ p$ are further satisfied, then $\frac{1}{mT}\sum_{t=1}^{T}\sum_{i=1}^{m} I(\gamma_{S_i,n}^{(t)} = \gamma_S) - \pi(\gamma_S|\boldsymbol{X}_N) \xrightarrow{p} 0$ as $T \to \infty$ and $N \to \infty$.

Validity of eSGLD for variable selection

Theorem 2. Assume the conditions of Theorem **??** hold. If, instead of a constant learning rate, a decreasing learning rate $\epsilon_t = O(1/t^{\kappa})$ is used for some $0 < \kappa < 1$, then the unweighted estimators given in parts (ii) and (iii) of Theorem **??** are still valid.

Remarks on eSGLD

- The total computational complexity of eSGLD is $O(N^{1+\varepsilon}p^{1-\varepsilon'})$ if measured in float computation, where $\varepsilon = \kappa/2 + \delta$ and the factor $0 < 1 - \varepsilon' < 1$ counts for the possible sparsity of the model. This is quite comparable with the computational complexity $O(Np^{1-\varepsilon'})$ achieved in general by the stochastic gradient descent (SGD) algorithm.
- For any integrable function $\rho(\gamma_S)$ such that $E_\pi|\rho(\gamma_S)|^{1+\varepsilon} < \infty$ for any $\varepsilon > 0$, $\frac{1}{T}\sum_{t=1}^{T}\rho(\gamma_S^{(t)})$ forms a consistent estimator of $E_\pi\rho(\gamma_S)$, where E_π denotes the expectation with respect to the posterior distribution.

An Illustrative Example

We considered the regression model:

$$\boldsymbol{y} = \boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{z}_3 + \boldsymbol{z}_4 + \boldsymbol{z}_5 - \boldsymbol{z}_6 - \boldsymbol{z}_7 - \boldsymbol{z}_8 + 0 \cdot \boldsymbol{z}_9 + \cdots + 0 \cdot \boldsymbol{z}_{100} + \varepsilon, \quad (11)$$

where $\varepsilon \sim N(0, I_N)$, and I_N denotes an $N \times N$ identity matrix. The explanatory variables $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{100}$ were generated such that they have a mutual correlation coefficient of 0.5. We considered three different values of N: 250, 500 and 1000. For each of them, we generated 10 independent datasets from the model (11).

An Illustrative Example

Table: Marginal inclusion probabilities estimates produced by Algorithm 2 for different sizes of datasets. CPU time was recorded for a run with one dataset on a Linux machine with Intel(R) Core(TM) i7-7700 CPU@3.60GHz.

N	True variables[a]	False variables[b]	CPU[c] (seconds)
250	0.9489 (0.0644)	0.0202 (0.0066)	2.4
500	1.0000 (0.0001)	0.0214 (0.0065)	6.0
1000	1.0000 (0.0000)	0.0249 (0.0063)	15.3

Large-Scale Linear Regression

To show that Algorithm 2 is suitable for large-scale Bayesian computing, we simulated 10 large datasets from an extended model of (11), where we set $p = 2000$ and $N = 50,000$ while keeping the true model unchanged.

The performance of the algorithm in variable selection is measured by the false selection rate (FSR) and negative selection rate (NSR):

$$FSR = \frac{\sum_{i=1}^{10} \left| \hat{S}_i \setminus S \right|}{\sum_{i=1}^{10} \left| \hat{S}_i \right|}, \quad NSR = \frac{\sum_{i=1}^{10} \left| S \setminus \hat{S}_i \right|}{\sum_{i=1}^{10} |S|}, \tag{12}$$

where S denotes the set of true variables, $\hat{S}_i$ denotes the set of selected variables from dataset i, and $|\hat{S}_i|$ denotes the size of $\hat{S}_i$.

Large-Scale Logistic Regression

We simulated 10 large datasets from the model

$$\text{logit}P(Y_i = 1) = \sum_{i=1}^{k} \beta_j z_{ij}, \quad i = 1, 2, \ldots, N, \tag{13}$$

where $N = 50,000$, $k = 2000$, $\beta_1 = \cdots = \beta_5 = 1$, $\beta_6 = \beta_7 = \beta_8 = -1$, and $\beta_9 = \cdots = \beta_p = 0$. The explanatory variables $\boldsymbol{z}_i$'s, where $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ip})^T$, were generated such that they have a mutual correlation coefficient of 0.5. The response variables were generated such that half of them have a value of 1 and the other half have a value of 0.

Large-Scale GLM

Table: Summary of numerical results of eSGLD for large datasets with $N = 50,000$ and $p = 2000$, where the CPU time was recorded for each dataset on a Linux machine with Intel(R) Core(TM) i7-7700 CPU@3.60GHz.

Model	Algorithm	FSR	NSR	$\widehat{MSE}_1$	$\widehat{MSE}_0$	CPU(minutes)
Linear	eSGLD	0	0	$2.91 \times 10^{-3}(1.90 \times 10^{-3})$	$1.26 \times 10^{-7}(1.18 \times 10^{-8})$	2.5
	RJMH	—	—	—	—	7983.3
Logistic	eSGLD	0	0	$2.37 \times 10^{-2}(2.88 \times 10^{-3})$	$2.70 \times 10^{-4}(1.40 \times 10^{-4})$	2.9

A Pascal Challenge Dataset

The dataset *epsilon* has been used for Pascal large scale learning challenge in 2008. The raw dataset consists of 500,000 observations and 2000 features, which was split into two parts: 400,000 observations for training and 100,000 observations for testing. The training part is feature-wisely normalized to mean zero and variance one and then observation-wisely scaled to unit length. Using the scaling factors of the training part, the testing part is processed in a similar way.

A Pascal Challenge Dataset

Table: Numerical results for the dataset *epsilon*, where the results of eSGLD were averaged over 10 independent runs with the standard deviation reported in the parentheses, and the CPU time was recorded for a run on a Linux machine with Intel(R) Core(TM) i7-7700 CPU@3.60GHz.

Algorithm	Setting	Model size	Prediction Error(ν)	CPU (minutes)
eSGLD	$C_0 = 2.5$	62.7(2.6)	0.1422 (1.80×10^{-3})	17.4
Lasso	$\lambda = 0.0239$	63	0.1644	21.2
SIS-MCP/SCAD	—	—	—	> 1440

MovieLens Data

The dataset contains 10,000,054 ratings of 10,681 movies by 71,567 users. The ratings vary from 0.5 to 5 in increments of 0.5. The whole dataset contains information of the rating, time of the rating, and the genre of the movie (with 19 possible categories).

MovieLens Data

This dataset has been modeled by mixed effect models. Let N be the total number of users, and let s_i denote the number of ratings of user i. Set

$$\boldsymbol{y}_i = \boldsymbol{W}_i\beta + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{e}_i, \quad \boldsymbol{b}_i \sim N_q(0, \Sigma),\ \Sigma = \sigma^2\boldsymbol{D},\ e_i \sim N_{r_i}(0, \sigma^2\boldsymbol{I}_{r_i}),$$

where $\boldsymbol{y}_i \in \mathbb{R}^{s_i}$, $\boldsymbol{W}_i \in \mathbb{R}^{s_i \times p}$, $\beta \in \mathbb{R}^p$, $\boldsymbol{Z}_i \in \mathbb{R}^{s_i \times q}$, $\boldsymbol{b}_i \in \mathbb{R}^q$ for $i = 1, 2, \ldots, N$. Let r_{ij} be the rating of user i for movie j, the response is defined as $\boldsymbol{y}_i = (r_{ij_1}, r_{ij_2}, \ldots, r_{ij_{s_i}})^T$. Here it is assumed that $(\boldsymbol{y}_i,\ \boldsymbol{W}_i,\ \boldsymbol{Z}_i),\ i = 1, 2, \ldots, N$ are iid random samples with varying dimensions. We let $\boldsymbol{W}_i$ denote the collection of the intercept, genre predictor, popularity predictor and previous predictor, and let $\boldsymbol{Z}_i$ be the same as $\boldsymbol{W}_i$.

eSGLD for missing data problems

(i) (Subsampling) Draw a subsample of size n from the full dataset $\boldsymbol{X}_N$ at random, and denote the subsample by $\boldsymbol{X}_n^{(t)}$.

(ii) (Imputation) Simulate models (viewed as missing data) $\gamma_{S_1,n}^{(t)}, \ldots, \gamma_{S_m,n}^{(t)}$ from the conditional posterior $\pi(\gamma_S|\boldsymbol{\theta}^{(t)}, \boldsymbol{X}_n^{(t)})$.

(iii) (Updating $\boldsymbol{\theta}$) Update $\boldsymbol{\theta}^{(t)}$ by setting

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\epsilon_{t+1}}{2}\left\{\frac{N}{mn}\sum_{k=1}^{m}\nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{\theta}^{(t)}|\boldsymbol{X}_n^{(t)}, \gamma_{S_k,n}^{(t)}) - \frac{N-n}{n}\nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{\theta}^{(t)})\right\} + \sqrt{\epsilon_{t+1}\tau}\eta_{t+1},$$

where $\eta_{t+1} \sim N(0, I_d)$, and ϵ_{t+1} can be set to a small constant or decrease as in the SGLD algorithm.

MovieLens Data

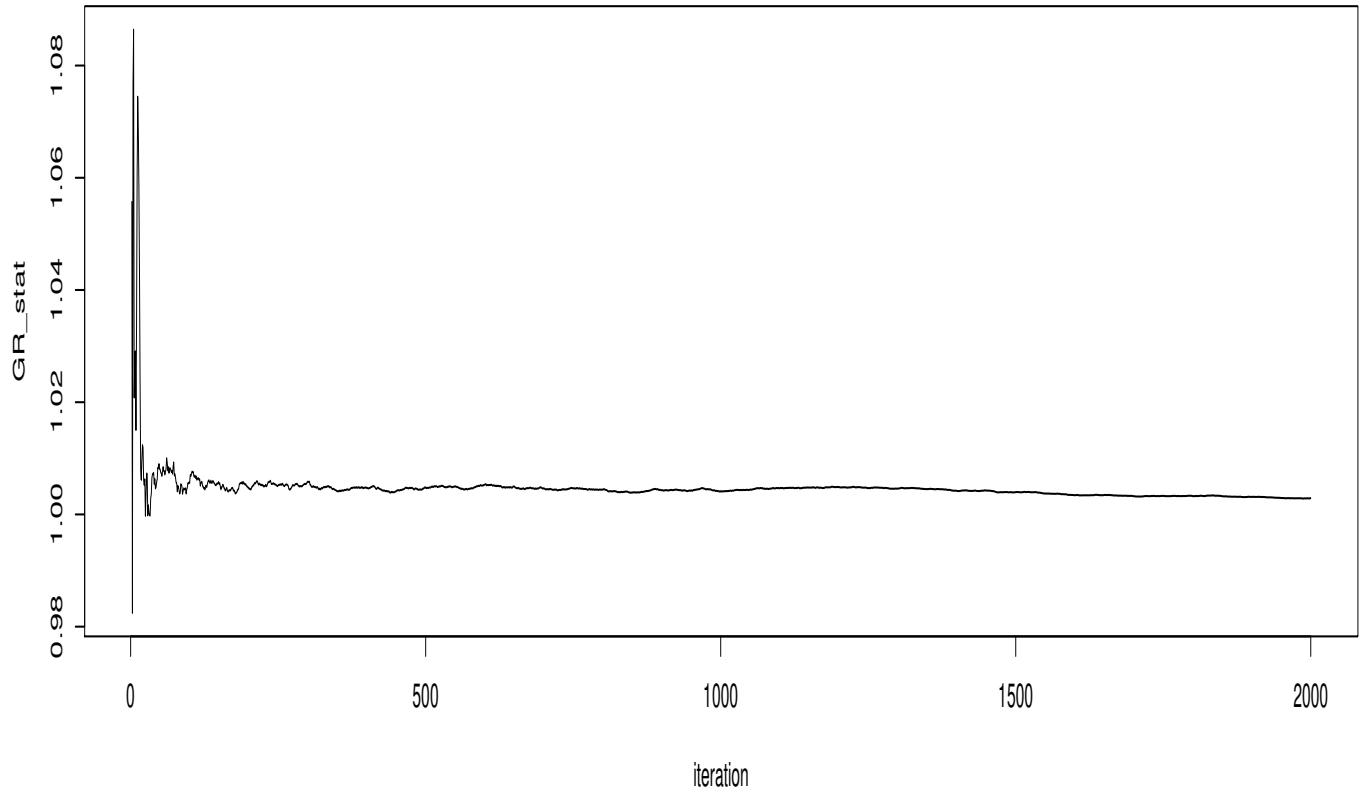

Figure: Convergene diagnostic for the eSGLD algorithm on MovieLen data with the Gelman-Rubin statistic, which was calculated based on 10 independent runs.

MovieLens Data

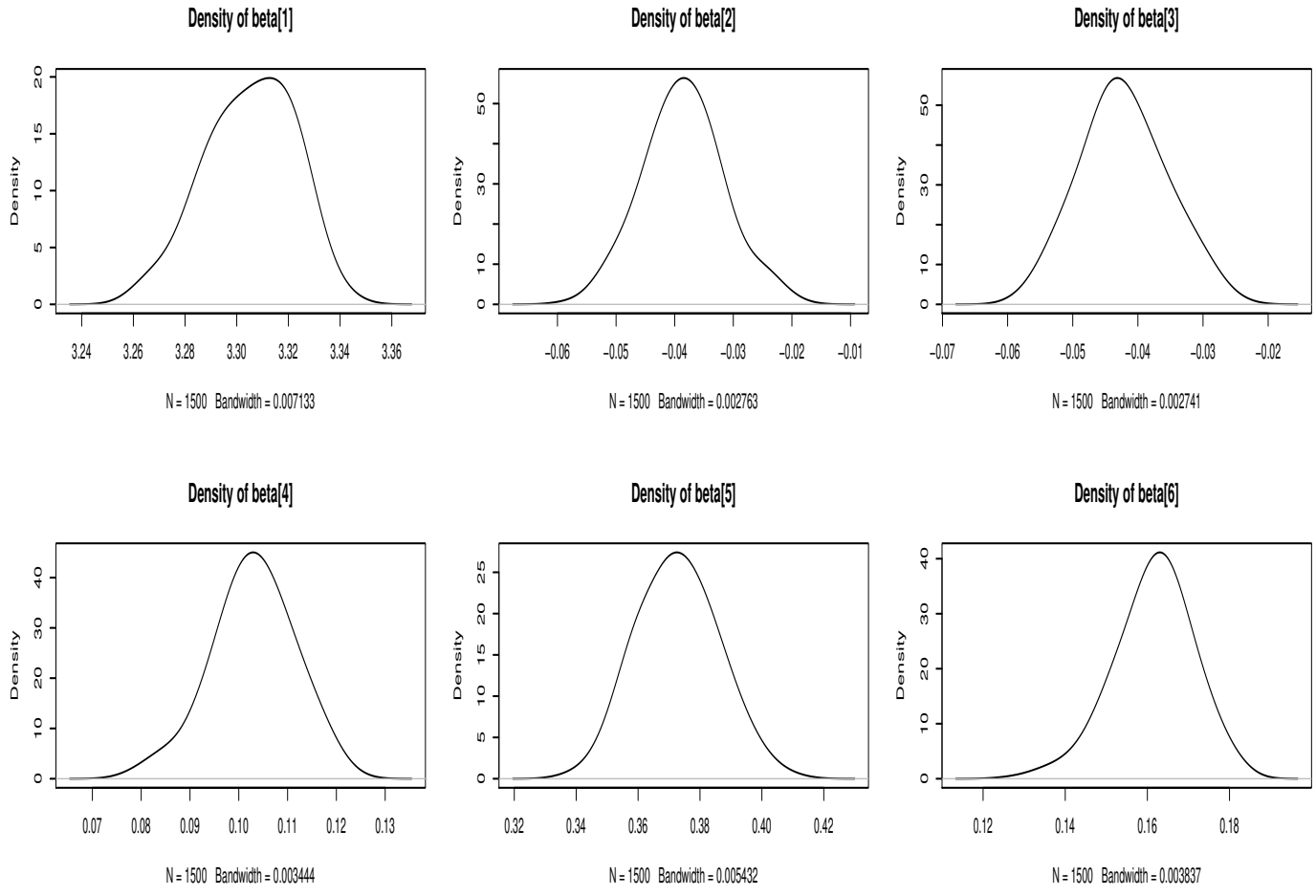

Figure: Posterior densities of $\boldsymbol{\beta}$ estimated from the posterior samples generated by the eSGLD algorithm in 1500 iterations.

Big-N-Large-P Problems

We suggest a combined use of the eSGLD algorithm and the split-and-merge strategy (Song and Liang, 2015):

(i) spit the full dataset into a number of subsets of lower dimensions;

(ii) apply eSGLD to each subset independently, which is to select the true variables included in the subset as well as the surrogate variables to the true variables missed in the subset;

(iii) collect all the variables selected from each subset as explanatory variables to form a new regression, and apply eSGLD to the new regression to select appropriate variables.

For linear regression, the consistency of this procedure has been established in Song and Liang (2015). For logistic regression, its consistency has been recently established by us and will be reported elsewhere.

Conclusion

- eSGLD is highly scalable with the use of mini-batch samples and can be much more efficient than traditional MCMC algorithms
- eSGLD has a comparable computational complexity with the SGD algorithm.
- Compared to frequentist methods, it can produce more accurate variable selection and prediction results, while exhibiting similar CPU costs when the dataset contains a large number of samples.

The eSGLD algorithm has much alleviated the pain of Bayesian methods in large-scale computing!

More Possible Applications of eSGLD

- Sparse deep learning (Liang et al., 2018)
- Hidden Markov model
- model-based clustering
- random coefficient models