Journal of Classification 35 (2018)
DOI: 10.1007/s00357-018-9248-z

Two-Stage Metropolis-Hastings for Tall Data

Richard D. Payne

Texas A&M University, USA

Bani K. Mallick

Texas A&M University, USA

Abstract: This paper discusses the challenges presented by tall data problems associated with Bayesian classification (specifically binary classification) and the existing methods to handle them. Current methods include parallelizing the likelihood, subsampling, and consensus Monte Carlo. A new method based on the two-stage Metropolis-Hastings algorithm is also proposed. The purpose of this algorithm is to reduce the exact likelihood computational cost in the tall data situation. In the first stage, a new proposal is tested by the approximate likelihood based model. The full likelihood based posterior computation will be conducted only if the proposal passes the first stage screening. Furthermore, this method can be adopted into the consensus Monte Carlo framework. The two-stage method is applied to logistic regression, hierarchical logistic regression, and Bayesian multivariate adaptive regression splines.

Keywords: Bayesian inference; Logistic model; Bayesian multivariate adaptive regression splines; Markov chain monte carlo; Metropolis-hastings algorithm; Tall data.

1. Introduction

In the past twenty-five years, Bayesian statistics have become increasingly popular as they are capable of analyzing data with complex structures. Consequently, Bayesian methods have been proven to be effective in a wide range of applications. The rise in popularity is largely attributed

The authors would like to thank the reviewers for their comments which greatly improved the presentation of the material in this article. B.K. Mallick's research was supported by National Cancer Institute of the National Institute of Health under award number R01CA194391.

Corresponding Author's Address: R. Payne, Department of Statistics, 3143 Texas A&M University, College Station, TX 77843, e-mail: richard@stat.tamu.edu.

Published online

to simulation based algorithms which can approximate the complex posterior distributions of non-conjugate models, such as Markov Chain Monte Carlo (MCMC) methods including the Metropolis-Hastings (MH) algorithm (Robert and Casella, 2013).

The term “tall data” generally describes data in which $n >> p$, that is, when the number of observations is much larger than the number of predictors. For MCMC methods, as n increases, so does the computational demand of the algorithm. Specifically, for MH, the increased computational demand is driven by the complete scan of the data through likelihood evaluations on each iteration of the algorithm. If n is large enough, MCMC methods (including MH) are computationally infeasible.

There are several general methods to overcome this issue. The simplest method involves parallelizing the likelihood to speed up computation. Another method divides the data across multiple machines and performs independent parallel MCMC on each machine to sample from the posterior distribution (consensus Monte Carlo). The results are then aggregated using weighting (Scott et al., 2013). A third approach is to use subsampling methods to provide a faster estimation of the likelihood (Quiroz, Villani, and Kohn, 2014; Korattikara, Chen, and Welling, 2014; Bardenet, Doucet, and Holmes, 2014). See Bardenet, Doucet, and Holmes (2015) for a review of MCMC approaches for tall data.

We propose a method based on a two-stage Metropolis algorithm which uses a cheap estimate of the likelihood to determine if a full estimation of the likelihood is necessary. Furthermore, we compare the strengths and weaknesses of the general tall data MH methods of consensus, subsampling, and two-stage Metropolis, as well as briefly introduce the use of a combination of the consensus and two-stage methods. For definiteness, in the following, the focus of this paper is on the classification problem. However, the developed methodology can be extended to any model which is suitable to analyze tall data. The methods are applied to three datasets: marketing data from a Portuguese bank, loan data from Freddie Mac, and a simulated dataset. Logistic regression is applied to both the Portuguese bank and Freddie Mac datasets and an additional logistic hierarchical model is fit to the Freddie Mac dataset. Modifications to the techniques described in the papers above have been made to accommodate the features of these datasets and are explained further. Lastly, the two-stage method is applied to a simulated binary classification problem using Bayesian multivariate adaptive regression splines (BMARS).

In Section 2, we describe the existing methods for handling tall data and present the two-stage methodology. Section 3 applies the methods to the marketing, Freddie Mac, and simulated datasets. Section 4 provides a brief discussion and Section 5 concludes.

2. Methodology

We begin by briefly describing existing techniques to speed up MCMC computation for tall data applications.

2.1 Likelihood Parallelization

Perhaps the simplest way to adapt the Metropolis-Hastings algorithm for tall data is to compute the likelihood in parallel. In this method, the data are partitioned into p partitions and each is assigned to a separate process/core. On each iteration of the MH algorithm, the master process draws parameters from the proposal distribution and sends the proposed values to the other processes. Each process then computes the likelihood for its partition and passes this information (i.e. the sum of the log-likelihood) to the master process which sums the log-likelihood contributions from each partition and determines whether or not to accept the proposal. As long as there is no significant communication overhead between the processes, the MH algorithm's speed will be increased while still sampling from the true posterior distribution.

2.2 Consensus Monte Carlo

In the consensus Monte Carlo method, the data are randomly partitioned into p partitions. Subsequently, allow each partition to run a full MCMC simulation from a posterior distribution given its own data. Lastly, combine the posterior simulations from each partition to produce a set of global draws to reproduce the unified posterior distribution.

Suppose $\mathbf{y} = (y_1, \dots, y_n)$ denotes the full data and $\mathbf{y}_j$ is the data at the jth partition. We then represent the posterior distribution of $\boldsymbol{\beta}$ as

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto \prod_{j=1}^{p} p(\mathbf{y}_j|\boldsymbol{\beta})p(\boldsymbol{\beta})^{1/p},$$

where the prior distribution has been expressed as the product of the p components.

For each partition, a Metropolis sampler with a chain of length m is computed in parallel with the prior weight adjusted to p^{-1} its original weight. Once posterior samples are obtained from each of the partitions, the results are combined using a weighted average. The weight, W_i, for the ith partition is equal to the inverse of the posterior covariance matrix obtained from the Metropolis sampler. Let $\boldsymbol{\beta}_i$ be the posterior sample matrix from the ith partition. Thus, the final posterior sample, $\boldsymbol{\beta}$ is obtained using the following weighted average:

$$\boldsymbol{\beta} = \left(\sum_{i=1}^{p} \boldsymbol{\beta}_i W_i \right) \left(\sum_{j=1}^{p} W_j \right)^{-1} .$$

For details see Scott et al. (2013).

2.3 Subsampling Based Methods

In subsampling methods, a small subset of the data is used to estimate the likelihood function which is then used to evaluate the acceptance probabilities of the MH algorithm. In principle, subsampling reduces the data size and therefore a faster MCMC algorithm can be developed. Using an unbiased likelihood estimate in the MCMC chain still provides the correct stationary distribution (Andrieu and Roberts, 2009), however the efficiency of the MCMC chain depends on the variance of the estimator. Usually complete random sampling does not work well in this situation (i.e. the chain gets stuck for many iterations), but some general guidelines for estimating the full likelihood from a subsample in a Bayesian setting have been developed. Quiroz et al. (2014) suggest using a portion of the data prior to MCMC and fitting Gaussian processes or splines to approximate the log-likelihood. On each iteration of the MCMC chain, the log-likelihood is estimated for each observation. The data are then sampled with probability proportional to its estimated log-likelihood value, which reduces the variance of the estimator and improves the efficiency of the chain.

2.4 Two-Stage MH

Consider the usual Bayesian model setup where the posterior distribution of the parameter $\boldsymbol{\beta}$ given data $\mathbf{y}$ is given by

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}) , \tag{1}$$

where $p(\mathbf{y}|\boldsymbol{\beta})$ is the likelihood function and $p(\boldsymbol{\beta})$ is the prior distribution for the parameter vector $\boldsymbol{\beta}$. If a non-conjugate prior is selected, the posterior distribution $p(\boldsymbol{\beta}|\mathbf{y})$ often cannot be expressed in an explicit form and consequently MCMC methods must be used to simulate samples from this posterior distribution. More specifically, we use the Metropolis-Hastings (MH) algorithm to generate samples of $\boldsymbol{\beta}$s from $p(\boldsymbol{\beta}|\mathbf{y})$. The MH algorithm is described as follows.

A.1 MH Algorithm

1. At the tth iteration generate $\boldsymbol{\beta}$ from the proposal distribution $q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)$ where $\boldsymbol{\beta}_t$ is the current state.

2. Accept $\boldsymbol{\beta}$ as a posterior sample with probability

$$h(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = \min\left\{1, \frac{q(\boldsymbol{\beta}_t|\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathbf{y})}{q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)p(\boldsymbol{\beta}_t|\mathbf{y})}\right\} . \tag{2}$$

3. $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}$ with probability $h(\boldsymbol{\beta}_t, \boldsymbol{\beta})$ and $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t$ with probability $1 - h(\boldsymbol{\beta}_t, \boldsymbol{\beta})$.

At each iteration, the probability of moving from the state $\boldsymbol{\beta}_t$ to next state $\boldsymbol{\beta}$ is $q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)h(\boldsymbol{\beta}_t, \boldsymbol{\beta})$, hence the transition kernel for the Markov Chain $\boldsymbol{\beta}_t$ is

$$T(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)h(\boldsymbol{\beta}_t, \boldsymbol{\beta}) + \left\{1 - \int q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)h(\boldsymbol{\beta}_t, \boldsymbol{\beta})d\boldsymbol{\beta}\right\} I(\boldsymbol{\beta} = \boldsymbol{\beta}_t) ,$$

where $I()$ is the indicator function. Due to the iterative nature of the algorithm, the likelihood function $p(\mathbf{y}|\boldsymbol{\beta})$ needs to be evaluated repeatedly which is expensive when n is large. Hence, we need to modify the MH algorithm to adapt it for tall data problems.

In the MH algorithm described in A.1, the evaluation of the likelihood is expensive in the tall data situation. Generally the MCMC chain requires thousands of iterations to converge. Furthermore, we need to generate a large number of samples to quantify the uncertainty in the parameters. We use the two-stage MH algorithm where the proposal distribution $q()$ is adapted to the target distribution using an approximate likelihood based model. These algorithms have been used previously (Christen and Fox, 2005; Higdon, Lee and Bi, 2002; Mondal et al., 2014), usually for solving expensive inverse problems. For our purposes, instead of testing each proposal by the exact likelihood based model directly, initially the algorithm tests the proposal by the approximate likelihood based model which is much cheaper to compute. If the proposal is accepted by the initial test, then an exact likelihood based computation will be conducted and the proposal will be further tested as in the MH algorithm method described in A.1. Otherwise, the proposal will be rejected by the approximate model and a new proposal will be generated from $q()$. The approximate likelihood based model filters the unacceptable proposals and avoids the expensive full likelihood computations.

A.2 Two-Stage MH Algorithm

Let $\hat{p}(\mathbf{y}|\boldsymbol{\beta})$ be an approximation of the full likelihood, and let the approximate posterior distribution be represented as $p^*(\boldsymbol{\beta}|\mathbf{y}) \propto \hat{p}(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$. Then the Two-Stage MH Algorithm proceeds as follows:

1. At the tth iteration generate $\boldsymbol{\beta}^{'}$ from the proposal distribution $q(\boldsymbol{\beta}^{'}|\boldsymbol{\beta}_t)$.
2. Take a real proposal as

$$\boldsymbol{\beta} = \begin{cases} \boldsymbol{\beta}^{'} & \text{with probability } \delta(\boldsymbol{\beta}_t, \boldsymbol{\beta}^{'}) \\ \boldsymbol{\beta}_t & \text{with probability } 1 - \delta(\boldsymbol{\beta}_t, \boldsymbol{\beta}^{'}) \, , \end{cases}$$

where

$$\delta(\boldsymbol{\beta}_t, \boldsymbol{\beta}^{'}) = \min\left\{1, \frac{q(\boldsymbol{\beta}_t|\boldsymbol{\beta}^{'})p^*(\boldsymbol{\beta}^{'}|\mathbf{y})}{q(\boldsymbol{\beta}^{'}|\boldsymbol{\beta}_t)p^*(\boldsymbol{\beta}_t|\mathbf{y})}\right\} \, .$$

3. Accept $\boldsymbol{\beta}$ as a posterior sample with probability

$$\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = \min\left\{1, \frac{Q(\boldsymbol{\beta}_t|\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathbf{y})}{Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)p(\boldsymbol{\beta}_t|\mathbf{y})}\right\} \, , \tag{3}$$

where $Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t) = \delta(\boldsymbol{\beta}_t, \boldsymbol{\beta})q(\boldsymbol{\beta}|\boldsymbol{\beta}_t) + \{1 - \int \delta(\boldsymbol{\beta}_t, \boldsymbol{\beta})q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)d\boldsymbol{\beta}\}I(\boldsymbol{\beta} = \boldsymbol{\beta}_t)$.

4. Hence take $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}$ with probability $\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})$ and $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t$ with probability $1 - \rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})$.

At each iteration, the probability of moving from the state $\boldsymbol{\beta}_t$ to next state $\boldsymbol{\beta}$ is $q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})$, hence the transition kernel for the Markov Chain $\boldsymbol{\beta}_t$ is

$$T(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta}) + \left\{1 - \int q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})d\boldsymbol{\beta}\right\} I(\boldsymbol{\beta} = \boldsymbol{\beta}_t).$$

In the above algorithm, if the trial proposal $\boldsymbol{\beta}^{'}$ is rejected by the approximate posterior then no further computation is needed. Thus, the expensive exact posterior computation can be avoided for those proposals which are unlikely to be accepted. This is just an adaption of the proposal using the approximate posterior where the transition kernel can be written as $K(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = \rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)$ for $\boldsymbol{\beta} \neq \boldsymbol{\beta}_t$ and $K(\boldsymbol{\beta}_t, \{\boldsymbol{\beta}_t\}) = 1 - \int_{\boldsymbol{\beta} \neq \boldsymbol{\beta}_t} \rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})Q(\boldsymbol{\beta}_t|\boldsymbol{\beta})d\boldsymbol{\beta}$ for $\boldsymbol{\beta} = \boldsymbol{\beta}_t$. It is simple to show that the detailed balance condition $p(\boldsymbol{\beta}_t|\mathbf{y})K(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathbf{y})K(\boldsymbol{\beta}, \boldsymbol{\beta}_t)$ is always satisfied under some minor regularity conditions like the regular MH algorithm.

Result 1: *The detailed balance condition is satisfied under the regularity conditions of the MH algorithm. That is,* $p(\boldsymbol{\beta}_t|\mathbf{y})K(\boldsymbol{\beta}_t, \boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathbf{y})$ $K(\boldsymbol{\beta}, \boldsymbol{\beta}_t)$.

Proof: When $\boldsymbol{\beta} = \boldsymbol{\beta}_t$, the result is trivial. When $\boldsymbol{\beta} \neq \boldsymbol{\beta}_t$ we have

$$\begin{aligned} p(\boldsymbol{\beta}_t|\mathbf{y})K(\boldsymbol{\beta}_t,\boldsymbol{\beta}) &= p(\boldsymbol{\beta}_t|\mathbf{y})\rho(\boldsymbol{\beta}_t,\boldsymbol{\beta})Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t) \\ &= p(\boldsymbol{\beta}_t|\mathbf{y})\min\left\{1, \frac{Q(\boldsymbol{\beta}_t|\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathbf{y})}{Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)p(\boldsymbol{\beta}_t|\mathbf{y})}\right\}Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t) \\ &= \min\left\{p(\boldsymbol{\beta}_t|\mathbf{y})Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t), Q(\boldsymbol{\beta}_t|\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathbf{y})\right\} \\ &= \min\left\{\frac{p(\boldsymbol{\beta}_t|\mathbf{y})Q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)}{p(\boldsymbol{\beta}|\mathbf{y})Q(\boldsymbol{\beta}_t|\boldsymbol{\beta})}, 1\right\}p(\boldsymbol{\beta}|\mathbf{y})Q(\boldsymbol{\beta}_t|\boldsymbol{\beta}) \\ &= \rho(\boldsymbol{\beta},\boldsymbol{\beta}_t)p(\boldsymbol{\beta}|\mathbf{y})Q(\boldsymbol{\beta}_t|\boldsymbol{\beta}) \\ &= p(\boldsymbol{\beta}|\mathbf{y})K(\boldsymbol{\beta},\boldsymbol{\beta}_t)\,. \end{aligned}$$

■

Result 2: *The acceptance probability can be expressed as*

$$\rho(\boldsymbol{\beta}_t,\boldsymbol{\beta}) = \min\left\{1, \frac{p^*(\boldsymbol{\beta}_t|\mathbf{y})p(\boldsymbol{\beta}|\mathbf{y})}{p^*(\boldsymbol{\beta}|\mathbf{y})p(\boldsymbol{\beta}_t|\mathbf{y})}\right\}.$$

Proof: If $\boldsymbol{\beta} = \boldsymbol{\beta}_t$ then the result is trivial since $\rho(\boldsymbol{\beta}_t,\boldsymbol{\beta}) = 1$. For $\boldsymbol{\beta} \neq \boldsymbol{\beta}_t$

$$\begin{aligned} Q(\boldsymbol{\beta}_t|\boldsymbol{\beta}) &= \delta(\boldsymbol{\beta},\boldsymbol{\beta}_t)q(\boldsymbol{\beta}_t|\boldsymbol{\beta}) \\ &= \frac{1}{p^*(\boldsymbol{\beta}|\mathbf{y})}\min\{q(\boldsymbol{\beta}_t|\boldsymbol{\beta})p^*(\boldsymbol{\beta}|\mathbf{y}), q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)p^*(\boldsymbol{\beta}_t|\mathbf{y})\} \\ &= \frac{q(\boldsymbol{\beta}|\boldsymbol{\beta}_t)p^*(\boldsymbol{\beta}_t|\mathbf{y})}{p^*(\boldsymbol{\beta}|\mathbf{y})}\delta(\boldsymbol{\beta}_t,\boldsymbol{\beta}) \\ &= \frac{p^*(\boldsymbol{\beta}_t|\mathbf{y})}{p^*(\boldsymbol{\beta}|\mathbf{y})}Q(\boldsymbol{\beta}|\beta_t). \end{aligned}$$

Substituting this in the expression of $\rho(\boldsymbol{\beta}_t,\boldsymbol{\beta})$ we obtain the required expression.

■

It is important to note that the methodology above is general enough to be applied to any computationally expensive MH sampler. However, for definiteness in following, the method is applied to a few specific classification models. The success of the two-stage method in any given model depends on the construction of a computationally cheap and accurate estimate of the likelihood. The accuracy and speed of the likelihood estimator governs the efficiency of the MCMC chain. For instance, if the likelihood estimator $\hat{p}(\mathbf{y}|\boldsymbol{\beta}')$ severely underestimates $p(\mathbf{y}|\boldsymbol{\beta}')$, then $\delta(\boldsymbol{\beta}_t,\boldsymbol{\beta}')$ will be small and the proposal will be rejected (even if it might be a reasonable candidate). On the other hand, if $\hat{p}(\mathbf{y}|\boldsymbol{\beta}')$ severely overestimates $p(\mathbf{y}|\boldsymbol{\beta}')$, then it will likely pass the first stage and get rejected in the second stage since $\rho(\boldsymbol{\beta}_t, \boldsymbol{\beta})$ decreases as a function of $p^*(\boldsymbol{\beta}|\mathbf{y}) = \hat{p}(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$; thus the algorithm will compute the full likelihood for an unfavorable candidate. Consequently, it is important to select an accurate approximation to the likelihood. Specific likelihood approximations will be discussed for the examples in Section 3.

2.5 Combining Consensus with Two-Stage MH

For larger data sets which may not fit in RAM, we propose a combination of the consensus and the two-stage Metropolis methods. This is identical to the consensus method with the exception that each partition uses the two-stage Metropolis sampler rather than the usual Metropolis sampler. Since two-stage MH will draw from the same distribution as MH on each partition, the results of the consensus method will remain the same.

3. Applications

The methods introduced above were implemented on three datasets. For initial testing, the methods were implemented on a relatively small dataset with just over 40,000 observations from a phone marketing campaign conducted by a Portuguese bank. A larger dataset of approximately 2.3 million observations consisting of individual household loan data from Freddie Mac was used to test how the methods scale. In both, logistic regression was used to classify observations, the latter also employs a hierarchical model. Lastly, the two-stage method was implemented on a BMARS model with a large (10^6 observations) simulated dataset.

3.1 Logistic Regression Model

We are considering a binary classification problem where the response $\mathbf{y}$ takes the value 0 or 1 where $\mathbf{y} = (y_1, \ldots, y_n)$ and we have a vector of covariates $\mathbf{x}$. We use a logit link function to link the ith response with the covariates as

$$\begin{aligned} y_i \mid \boldsymbol{\beta}, \mathbf{x}_i &\sim \text{Bernoulli}\{\pi(\mathbf{x}_i)\} \\ \pi(\mathbf{x}_i) &= \{1 + \exp(-\mathbf{x}_i\boldsymbol{\beta})\}^{-1} \\ \boldsymbol{\beta} &\sim \text{Multivariate-Normal}(\mathbf{0}, \Sigma_0)\,, \end{aligned}$$

where $\boldsymbol{\beta}$ is the l dimensional vector of classification parameters, $\mathbf{x}_i$ is the ith row of the design matrix ($i = 1, \ldots, n$), and a Gaussian prior is placed on $\boldsymbol{\beta}$. The model's posterior distribution can be expressed as $p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}) \propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta})p(\boldsymbol{\beta})$.

In the logistic regression models for both the Portuguese bank and Freddie Mac datasets, we estimate the log-likelihood using a variant of the case-control approximate likelihood (Raftery et al., 2012). To understand the approximation, it is important to realize the log-likelihood for a logistic regression model can be written as two sums:

$$\log\{p(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\} = \sum_{i:y_i=1} \{\theta_i - \log(1+e^{\theta_i})\} + \sum_{i:y_i=0} -\log(1+e^{\theta_i}), \quad (4)$$

where $\theta_i = \log\{\pi(\mathbf{x}_i)/[1-\pi(\mathbf{x}_i)]\} = \mathbf{x}_i\boldsymbol{\beta}$.

If the data are sparse, then the computation of the first sum will be relatively cheap, and only the second summation needs to be estimated. We use a subsampling method where a random sample of a observations is taken from the failed outcomes (i.e. $y_i = 0$). The second sum in (4) is estimated by multiplying the average log-likelihood of the a sampled observations by $n_0 = \sum_{i=1}^{n} I(y_i = 0)$, the number of failures in the dataset. Let A be the index values of the subsample of size a. Thus, the original log-likelihood is estimated as

$$\widehat{\log}\{p(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\} = \sum_{i:y_i=1} \{\theta_i - \log(1+e^{\theta_i})\} + \frac{n_0}{a}\sum_{i\in A} -\log(1+e^{\theta_i}). \quad (5)$$

We note $\widehat{\log}\{p(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\}$ is an unbiased estimate of the log-likelihood as $E[\widehat{\log}\{p(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\}] = \log\{p(\mathbf{y}|\boldsymbol{\beta},\mathbf{X})\}$. We could obtain an unbiased estimate of the likelihood by making a bias correction (Quiroz et al., 2014) but that is not necessary for our method as we are doing further filtering of the proposal using the exact likelihood method.

The above approximation to the likelihood yields the following result:

Result 3: *When the proposal β is promoted from the first stage, then for large a it can be shown that $\rho(\cdot,\cdot)$ goes towards 1. Thus, the two-stage MH algorithm only calculates the original full data likelihood when there is a high probability of acceptance of the proposal.*

3.1.1 Portuguese Bank Data

The Portuguese bank dataset was obtained from the University of California Irvine Machine Learning Archive and were analyzed in a recent paper by Moro, Cortez, and Rita (2014). The binary dependent variable of interest was whether or not a client subscribed to a term deposit after contact through a telephone marketing campaign. The predictor variables of interest were the client's previous promotion outcome (non-existent, failure, success), age (years), and type of contact (telephone, cellular), education level (8 categories), and marital status (married, divorced, single, unknown). The categorical variables were treated as nominal values and the continuous variable age was logged and centered. A vague prior was placed on $\boldsymbol{\beta}$ resulting in the following model:

$$\begin{aligned} y_i \mid \boldsymbol{\beta}, \mathbf{x}_i &\sim \text{Bernoulli}\{\pi(\mathbf{x}_i)\} \\ \pi(\mathbf{x}_i) &= \{1 + \exp(-\mathbf{x}_i\boldsymbol{\beta})\}^{-1} \\ \boldsymbol{\beta} &\sim \text{Multivariate-Normal}(\mathbf{0}, \Sigma_0 = I * 10^2)\,, \end{aligned}$$

where $\boldsymbol{\beta}$ is the vector of coefficients, Σ_0 is the prior covariance matrix for $\boldsymbol{\beta}$, I is the identity matrix and $\mathbf{x}_i$ is the ith row of the design matrix, $i = 1, \ldots, n$.

The two-stage, consensus, and standard MH algorithms were coded in Fortran and run for 100,000 iterations with a burn in of 5,000 values. The subsampling method was considerably slower and was consequently run for only 10,000 iterations with the same burn in of 5,000. In the consensus method, the data were randomly split into 14 partitions. In the two-stage method, a single random subsample of 1,400 observations was taken prior to MCMC. This subsample was used to approximate the log-likelihood during the first stage on each iteration of the two-stage MH algorithm.

In the subsampling method, a thin-plate spline surface was fit to a subsample of the data (1,000 observations) prior to MCMC. For simplicity, in this smaller dataset, the thin-plate spline surface treated the categorical predictors as continuous. Although this resulted in a somewhat crude approximation to the log-likelihood surface, using a subsample of around 7,000 observations on each iteration allowed the MCMC chain to mix satisfactorily. To get a better idea of the speed of the subsampling method if a better spline surface was fit, it was also run subsampling 100 observations rather than 7,000. The number of likelihood evaluations per second for the MH and subsampling method (100 and 7,000 observations) were 355, 23.9, and 1.8 respectively.

Figures 1-3 compare the posterior densities of the two-stage, subsampling and consensus methods to the standard Metropolis sampler results. Figure 1 shows that the two-stage method matches the results obtained by the unmodified MH algorithm which is expected based on the theoretical results above. Figure 2 indicates that the subsampling method was effective in capturing the true posterior distribution since the subsampling method can be arbitrarily close to the true posterior based on the subsample size (Quiroz et al., 2014). Figure 3 shows that the consensus method matches the true posterior very well, with the exception of β_{10} and β_{15} which have a larger spreads and are slightly biased. Interestingly, β_{10} and β_{15} are the coefficients for education ('illiterate'), and marital status ('unknown') which have only 18 and 80 cases in the dataset respectively. Paradoxically, as Scott

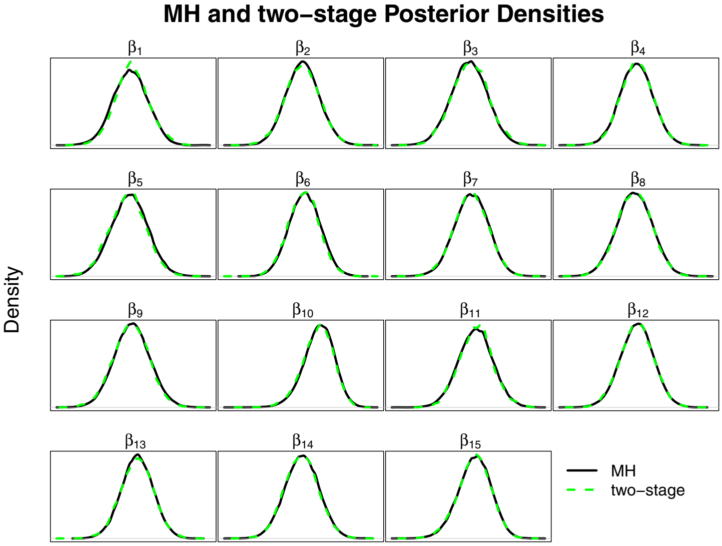

Figure 1. Posterior densities from the Portuguese bank data, two-stage Metropolis vs. Metropolis-Hastings. The MH and two-stage MH methods are represented by the solid and dashed lines, respectively. (See online version for color.) The posterior densities are nearly indistinguishable between the methods.

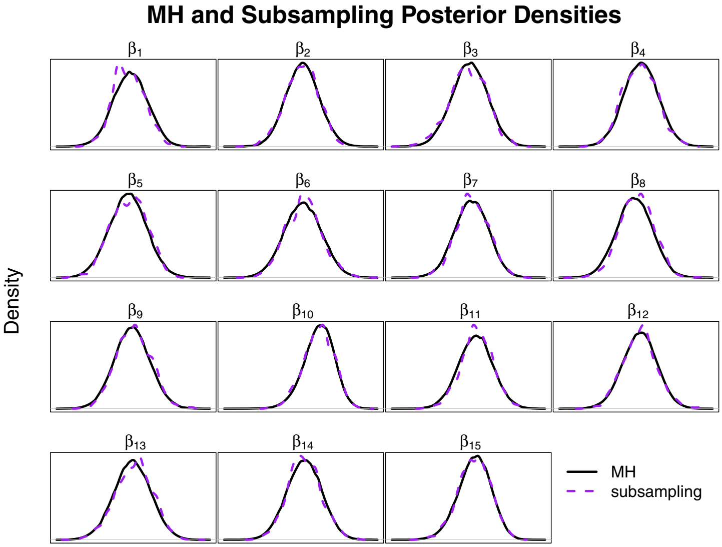

Figure 2. Posterior densities from the Portuguese bank data, subsampling vs. MH. The MH and subsampling methods are represented by the solid and dashed lines, respectively. (See online version for color.) The posterior densities are nearly indistinguishable between the methods.

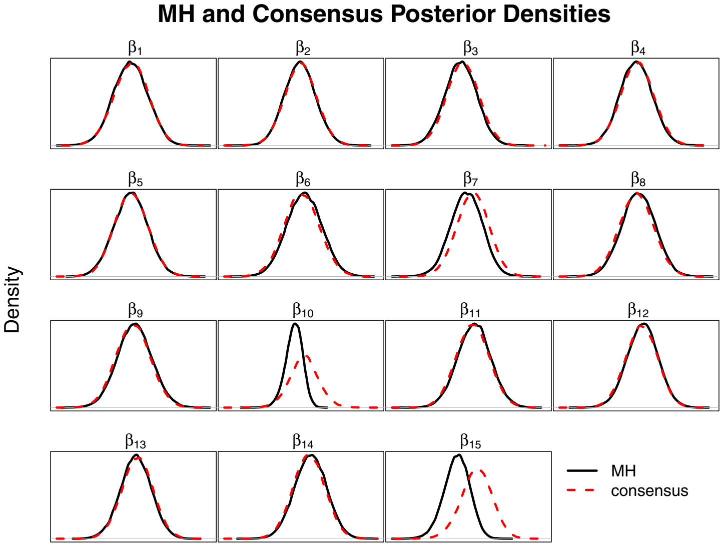

Figure 3. Posterior densities from the Portuguese bank data, consensus Monte Carlo vs. MH. The MH and consensus Monte Carlo methods are represented by the solid and dashed lines, respectively. (See online version for color.) The posterior densities are nearly indistinguishable between the methods.

et al. (2013) points out, we are suffering from a case of small sample bias in a large dataset, which is a potential issue in consensus Monte Carlo applications.

Since the Portuguese bank dataset is relatively small (approximately 40,000 observations), we refer the reader to the next section to better understand how these methods might scale to larger datasets, as well as a more detailed comparison of the speed and efficiency of the methods.

3.1.2 Freddie Mac Data, Logistic Regression

The loan data from Freddie Mac was obtained in September 2015 from Freddie Mac's website. The data consists of approximately 2.3 million loans which Freddie Mac acquired during 2009-2010 and contains monthly performance data on each loan. The binary dependent variable of interest is whether or not a loan was foreclosed by the end of September 2014.

To understand and quantify the effects of various covariates on foreclosure, a logistic model was used. Covariates of interest include the date of the first mortgage payment, FICO score, debt to income ratio, original principal balance of the loan, and first-time home-buyer status (yes, no, unknown). Each variable was transformed, centered, and scaled as appropriate. A vague prior was placed on β yielding the following logistic model:

$$\begin{aligned} y_i \mid \boldsymbol{\beta}, \mathbf{x}_i &\sim \text{Bernoulli}\{\pi(\mathbf{x}_i)\} \\ \pi(\mathbf{x}_i) &= \{1+\exp(-\mathbf{x}_i\boldsymbol{\beta})\}^{-1} \\ \boldsymbol{\beta} &\sim \text{Multivariate-Normal}(\mathbf{0}, \Sigma_0 = I * 10^2)\,, \end{aligned} \qquad (6)$$

where $\boldsymbol{\beta}$ is the vector of coefficients, Σ_0 is the covariance matrix for $\boldsymbol{\beta}$, I is the identity matrix of appropriate dimension and $\mathbf{x}_i$ is the ith row of the design matrix, $i = 1, \ldots, n$.

The coefficients of the model were estimated using the usual MH, consensus MH, and two-stage MH algorithm, all of which were coded in Fortran. To provide a fair comparison with the consensus Monte Carlo algorithm, both the MH and two-stage MH algorithm were parallelized with $p = 14$ partitions as described in Section 2.1 using Open MPI for Fortran. In the two-stage MH algorithm, the s observations used to approximate the log-likelihood in the first stage were selected by randomly selecting s/p observations from each partition prior to the start of the MH algorithm.

The subsampling MH method was not employed on the Freddie Mac dataset since it was not likely to computationally competitive in this setting. Since the data are extremely sparse, the likelihood can be easily calculated for the cases when $y_i = 1$, so subsampling would only need to be employed when $y_i = 0$. For full implementation, three separate spline surfaces would be required to be fit for each category of first-time home-buyer status (no, yes, unknown). Even if a relatively small sample was used for each spline surface approximation (e.g. several thousand observations), the corresponding matrices to calculate the spline fits would be in total far larger than the design matrix itself, and that computation is only the first step. Furthermore, the subsampling method is not likely to see the gains of parallelization that the MH and two-stage MH algorithms receive since more data will need to pass between processes and/or each process will have to calculate the same information in parallel (which defeats the purpose of parallelization).

The usual MH, consensus, and two-stage MH algorithms were run for 100,000 iterations with a burn-in period of 5,000. Parameters were updated sequentially and proposal variances were chosen such that the acceptance rate for each parameter was near 50%. Figure 4 plots the densities of the posterior distribution of $\beta_1, \ldots, \beta_7$. The densities are essentially indistinguishable between the MH, two-stage, and consensus methods. The execution times were 118, 115, and 81 minutes for the parallelized MH, consensus, and two-stage methods, respectively. In this particular application, the autocorrelation in the two-stage method was slightly more persistent than the regular MH algorithm. Consequently, it is of interest to evaluate the efficiency of the three MCMC chains, accounting for autocorrelation. This can be done by measuring the effective draws per minute (EDPM), which is a measure of the equivalent number of independent posterior draws per minute

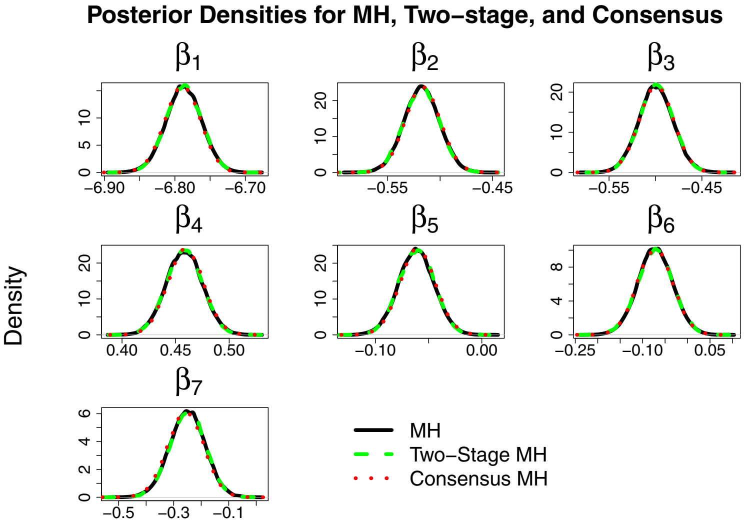

Figure 4. Posterior densities from Freddie Mac data. MH, two-stage MH, and consensus MH. (See online version for color.) The posterior densities are nearly indistinguishable between the methods.

the MCMC chain represents. The EDPM diagnostic incorporates both the execution time and autocorrelation of the chain to measure its efficiency:

$$\text{EDPM} = t^{-1}\left(\frac{n}{1+2\sum_{k=1}^{\infty}\rho_k}\right),$$

where n is the number of MCMC iterations, t = execution time of the MCMC chain in minutes, and ρ_k is the autocorrelation at the kth lag of the chain. EDPM can be calculated by estimating ρ_k with $\hat{\rho}_k$, the sample autocorrelation of the MCMC chain. To compare the efficiency of two MCMC chains, we can compute the relative effective draws per minute (REDPM) as

$$\text{REDPM} = \frac{\text{EDPM}_{\text{Algorithm 1}}}{\text{EDPM}_{\text{Algorithm 2}}}.$$

Figure 5 plots the REDPM of the two-stage and consensus methods relative to the MH method for each coefficient, $\beta_1, \ldots, \beta_7$. Also plotted are the REDPM values for the MCMC chain thinned by keeping every 10th and 20th values of the chain. We note that the two-stage method had REDPM values which were always above 1, and with the exception of one parameter, was always above the REDPM values of the consensus method. The median REDPMs for the two-stage method were 1.27, 1.44, and 1.47 as contrasted with 1.03, 1.07, and 1.02 for the consensus method (no thinning, keeping

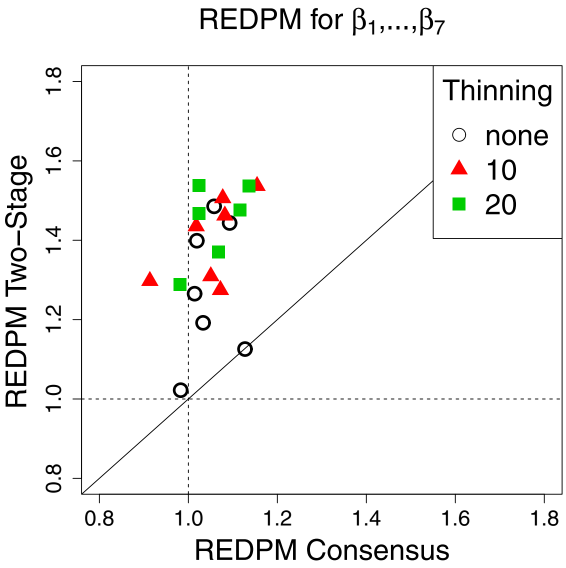

Figure 5. REDPM with respect to the MH algorithm for two-stage and consensus MH. REDPM is plotted for the original MCMC chain and the chain thinned every 10 and 20 values. (See online version for color.)

every 10th and 20th observations respectively). Thus in this application, the two-stage method appears to perform best in terms of speed, efficiency, and accuracy.

3.1.3 Freddie Mac, Hierarchical Logistic Regression

Bayesian statistics provide a simple way to fit hierarchical models, and with the help of MCMC, estimation of the parameters is generally straightforward. In addition to the covariates in the previous logistic regression model, the Freddie Mac dataset specifies which bank originally serviced the loan. It is of particular interest to understand how delinquency rates vary between banks during this time period. To accomplish this, we specify the following hierarchical model:

$$\begin{array}{rcl} y_{ij} \mid \theta_j, \boldsymbol{\beta}, \mathbf{x} & \sim & \text{Bernoulli}\{\pi(\mathbf{x}_{ij})\} \\ \pi(\mathbf{x}_{ij}) & = & [1+\exp\{-(\theta_j + \mathbf{x}_{ij}\boldsymbol{\beta})\}]^{-1} \\ \boldsymbol{\beta} & \sim & \text{Multivariate-Normal}(\mathbf{0}, \Sigma_0 = I * 10^2) \\ \theta_j & \overset{iid}{\sim} & \text{Normal}(0, \tau^2) \\ p(\tau) & \propto & \tau^{-2}, \end{array}$$

where $\boldsymbol{\beta}$ is the vector of coefficients (the same covariates as in (6) with an intercept), Σ_0 is the covariance matrix for the vague prior on $\boldsymbol{\beta}$, I is the identity matrix of appropriate dimension, $\mathbf{x}_{ij}$ is the row of the design matrix corresponding to observation y_{ij} and θ_j represents a random intercept term for the jth bank who serviced the loan, $j = 1, \ldots, k = 16$, $i = 1, \ldots, n_j$. Lastly, Jeffrey's prior was placed on τ.

In this model, interest lies primarily in the posterior distribution of τ, which provides us with an understanding of the variability of loan foreclosure rates between banks after controlling for the other covariates. For datasets which have relatively small n, the MH algorithm is straightforward to implement on this simple hierarchical model. The two-stage method is also easily extended to this hierarchical model, however, the consensus and the subsampling methods are not as easily implemented.

The two-stage and the usual MH algorithms were successfully implemented. As before, the data was partitioned into 14 partitions and the likelihoods were computed in parallel on each iteration. Both chains were run for 100,000 iterations with a burn-in period of 5,000. Parameters were updated sequentially and proposal distributions were chosen such that the acceptance rates for each parameter was near 50%. The two-stage method was implemented twice with sample sizes of 224,000 and 22,400 observations which were sampled prior to running the algorithm (1000 and 100 data values from each bank on each partition, roughly 10% and 5% of data). The total run times for the parallelized MH and the two-stage MH (10% and 5% subsample) were 1106, 849, and 639 minutes, respectively. We note that the variance of the proposal distribution for the two-stage MH with 5% subsampling was reduced (compared to the MH and two-stage MH with a 10% sample) in order to obtain the desired acceptance rate of the MCMC chain.

As in the other two applications, the posterior densities for the MH and two-stage MH for the 24 parameters were within Monte Carlo error (since both the MH and two-stage MH algorithm attain the correct stationary distribution). For brevity, we plot only the posterior distribution of τ since it is the primary parameter of interest (Figure 6).

Figure 7 plots the REDPM values of the 24 parameters for the hierarchical model. We note that the two-stage algorithm using 10% of the data consistently had REDPM values greater than 1, whereas using 5% of the data yielded more variability in the REDPM values, including 4 values lower than 1 on the un-thinned MCMC chain. In the 5% sampling case, all the values of REDPM < 1 were elements of $\boldsymbol{\beta}$, not $\boldsymbol{\theta}$. This may be an artifact of the sampling design since sampling was stratified by bank, and therefore some of the covariates may not have had adequate coverage in the smaller sample size. Thinning improves REDPM most dramatically for low REDPM values in the 5% sample, but otherwise doesn't seem to cause any

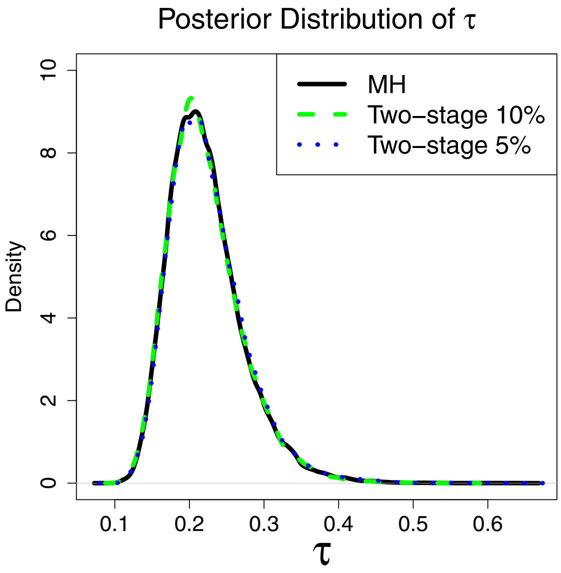

Figure 6. Posterior Distribution of τ for the MH and two-stage method subsampling 5% and 10% of the data. (See online version for color.) The posterior densities are nearly indistinguishable between the methods.

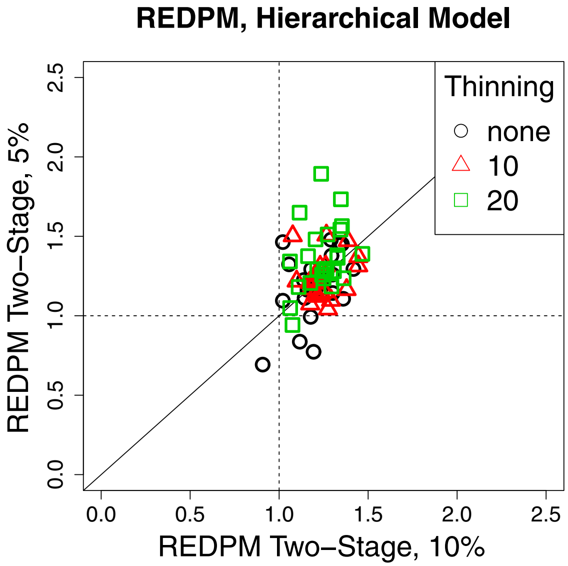

Figure 7. REDPM for the two-stage method with respect to the MH algorithm for subsample sizes of 5% and 10%. (See online version for color.)

major shifts in REDPM. Overall, the two-stage method showed increases in efficiency for the majority of the parameters.

The consensus method can be applied to this model as long as all the loans originating from a particular bank are in the same partition. The method proposed by Scott et al. (2013) requires running independent MCMC chains in parallel and then combining the draws of τ in the usual manner, and then discarding the values of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Once the draws of τ are combined, these values are sent to each partition which independently draw new values of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ from $p(\boldsymbol{\beta}|\tau, \mathbf{X})$ and $p(\boldsymbol{\theta}|\tau, \mathbf{X})$. In our case, however, these distributions are not in standard form and are not easily sampled from. In implementation, the first consensus MCMC chain to obtain draws of τ was faster than the traditional MH algorithm with a parallelized likelihood but performed more slowly than the two-stage method (1106, 910, 849, 639 minutes for MH, consensus, and two-stage MH (10% and 5% subsampling) respectively). Since the speed of the first run of the consensus method was slower than the two-stage method and the two-stage MH was more efficient than the consensus method in the previous model, drawing values from $p(\boldsymbol{\beta}|\tau, \mathbf{X})$ and $p(\boldsymbol{\theta}|\tau, \mathbf{X})$ was not implemented.

The subsampling method can also in theory be applied to this model. However, this requires fitting 48 spline surfaces prior to running the MCMC (16 banks, 3 levels of first-time home-buyer status). These spline surfaces were fit using the methodology provided by Ma, Racine, and Yang (2015) using a subsample of $s = 16,000$ observations (1,000 observations per group). However, on each iteration, approximating the log-likelihood surface for the entire dataset requires 48 matrix multiplications of dimension $z_i \times s, \sum_{i=1}^{48} z_i = n - s - n_1 = 2,297,813 - 3,711 - 16,000 = 2,278,102$ where $n_1 = \sum I(y_i = 1)$. These matrices were too large to fit into RAM, thus we were unable to implement the subsampling MH. Even if the data did fit into RAM, the computational cost of estimating the likelihood contribution with splines would likely be greater than evaluating the likelihood directly. Furthermore, implementing the subsampling method in parallel is not likely to produce significant gains in computation time since it will require either calculating the same quantities on each process (which defeats the purpose parallelizing) or passing vectors of information (rather than scalars) between processes.

3.2 Bayesian MARS

The two-stage method also has applications in more complicated classification settings, including Bayesian multivariate adaptive regression splines (BMARS) (Friedman, 1991; Holmes and Denison, 2003). BMARS is a non-linear classification method which is extremely flexible for classification problems where the relationship between the response and covariates is complex, unknown, or otherwise difficult for the analyst to specify. It uses the data to adaptively choose splines and knots to flexibly model classification problems. Since the splines and knot locations are not known a priori, BMARS requires the use of reversible jump MCMC (Green, 1995) to explore a parameter space with varying dimension.

Even in this more complicated setting, implementing the two-stage MH requires only a few extra lines of code, but can still produce a faster MCMC chain. To quantify the effectiveness of the two-stage method using BMARS, one million observations were simulated from the following model:

$$y \sim \text{Bernoulli}\{[1+\exp(-\pi(\mu))]^{-1}\}$$
$$\mu = x_1 + x_2 - x_3 - x_4 + x_1x_2 - .5x_1x_3 - x_2x_3 + .2x_1x_2x_3;$$
$$x_1 \sim U(0,1),\ x_2 \sim N(0,1),\ x_3 \sim U(0,2),\ x_4 \sim N(0,2^2)\,,$$

where $U(a,b)$ denotes a uniform distribution on the interval (a,b). The BMARS method was implemented using the prior distributions outlined by Holmes and Denison (2003), to which we refer the reader to their paper for details. The two-stage method was implemented by choosing a random subsample prior to MCMC which was used in each iteration to approximate the likelihood. The log-likelihood approximation in the first stage was calculated as $\hat{l} = (n/a)l_{sub}$ where n and a are the number of observations in the whole dataset and subsample, respectively, and l_{sub} is the log-likelihood contribution of the subsample.

The BMARS algorithm was run a total of 10 times on this simulated dataset. The usual BMARS algorithm was run 5 times and the average is summarized in the first row of Table 1. The two-stage method was implemented on the remaining five runs with various subsampling percentages. Once the priors are in place, there are only two parameters which need to be specified in the BMARS method: the maximum number of interactions allowed for the basis functions (which was chosen to be 3), and a tuning parameter (the proposal standard deviation of the spline coefficients).

From Table 1, it is clear that the two-stage method is faster than the usual BMARS MCMC, with all two-stage runs producing a 30%-40% reduction in time. As with the previous examples, as the subsampling percentage decreased, the acceptance rate of the two-stage MCMC chain decreased and the speed increased. When sampling only one percent of the data, the acceptance rate was very low, so it was re-run with a smaller proposal standard deviation. This led to an increase in the acceptance rate with a slight reduction in speed.

Due to the varying dimension of the parameter space during MCMC, comparing efficiency of the MCMC chain is not straightforward. Conse-

Table 1. The results of the BMARS MCMC with the two-stage BMARS MCMC. The first row corresponds to the average of 5 runs of the usual BMARS algorithm to which the two-stage runs are compared. The last column is the ratio of the two-stage time to the regular MCMC time in the first row. Note that as the subsampling percentage decreased, the speed of the two-stage algorithm increased while the acceptance rate decreased for a fixed proposal standard deviation (SD).

Subsample Percentage	SD	Acceptance Rate	Time (sec)	Time Ratio
-	.0005	.31	41,594	-
15	.0005	.18	28,134	.68
10	.0005	.15	27,422	.66
5	.0005	.12	25,384	.61
1	.0005	.06	25,144	.60
1	.0001	.21	28,261	.68

quently, to determine the effectiveness of the two-stage method, one thousand observations were used as a test set, and the predictions based on both MCMC chains were compared and are shown in Figure 8. The top-left panel of Figure 8 compares the predicted probabilities of two BMARS runs (neither implementing the two-stage method) to provide a visual of Monte Carlo error. The two-stage method with 15%, 10%, and 5% subsampling produced predictions which appear to be within Monte Carlo error of the usual BMARS algorithm. The two-stage 1% subsampling (sd = .0005) showed slightly more variability and 1% subsampling (sd = .0001) shows a fair amount of variability in the predictions. Interestingly, although the predictions between the BMARS and two-stage methods become more variable as the subsampling percentage decreased, the root-mean-square error (RMSE) of the predictions were essentially the same (RMSE for the 5 BMARS runs: .0806, .0805, .0806, .0806, .0807; RMSE for the 5 two-stage runs: .0807, .0806, .0808, .0804, .0806 for 15%, 10%, 5%, 1% (sd = .0005), 1% (sd = .0001), respectively). This gives further evidence that the two-stage method can still be highly effective even when using a very small percentage of the data as a subsample.

4. Discussion

Perhaps the most pressing question regarding two-stage MH is how to select the subsample size. From experience the authors note that for a fixed proposal distribution variance, decreasing the subsampling percentage will at some point decrease acceptance rates of the MCMC chain. This is due to the fact that the estimate of the likelihood is either overestimating or underestimating the likelihood ratio which causes proposed parameter values to be discarded by either the first stage (if the estimate of the likelihood ratio is too small) or the second stage (if the estimate of the likelihood ratio is too large). If too small of a subsample is used, the variance of the proposal

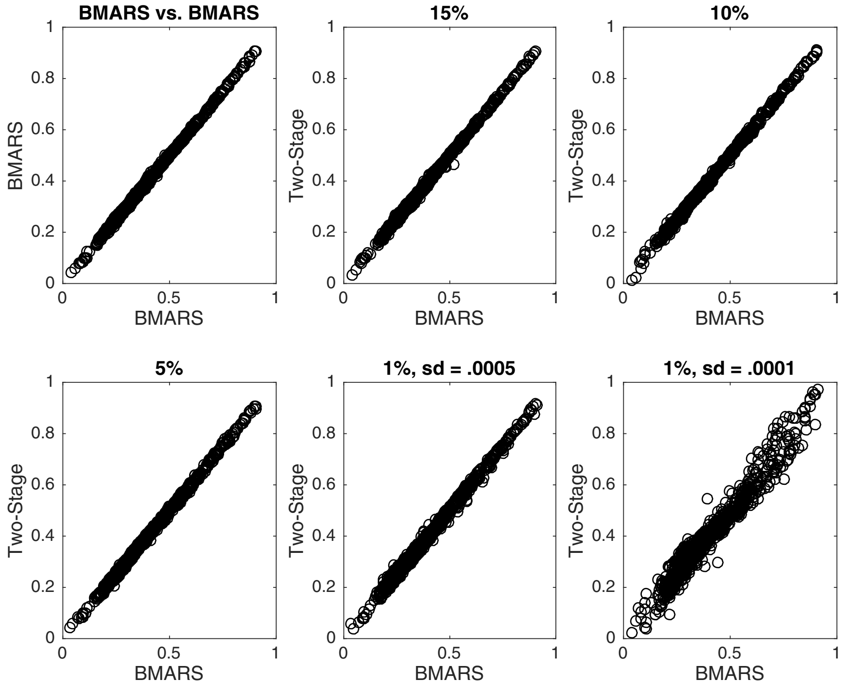

Figure 8. Comparison of the test-set predictions between the BMARS MCMC and the two-stage BMARS MCMC for various subsampling percentages. The top-left panel compares the predicted probabilities between two BMARS runs for a visual of Monte Carlo error.

distribution will need to be reduced to obtain the desired acceptance rate of the MCMC chain. A smaller subsample will increase the speed of the chain, but will likely increase the autocorrelation of the chain since the variance of the proposal distribution will need to be reduced. Even so, the hierarchical model for the Freddie Mac data still performed well with sampling only 5%-10% of the data, and the BMARS application performed well even with 1%-15% subsampling. This indicates that the speed and efficiency of the two-stage method may be somewhat robust to the subsample size used to approximate the log-likelihood.

One of the main advantages of the two-stage method is its simplicity and ease of implementation. It requires taking only one subsample prior to the MCMC algorithm and then adding a few lines of code to implement the first screening stage. Furthermore, it can be applied to any model in which a computationally cheap estimate of the likelihood can be obtained. Even using naive likelihood approximations, the two-stage method has performed well. If more precise likelihood estimates can be acquired for a particular model, the two-stage method may be even more effective at screening out bad proposals (although the speed will still depend on a computationally cheap likelihood estimate).

The consensus method is also generally straightforward in simple models, but even in hierarchical models it places restrictions on how the data can be partitioned and may require sampling from distributions which cannot be sampled from directly (which adds another potentially computationally demanding layer). The subsampling method requires the most effort to implement since it requires fitting spline surfaces to the data. Furthermore, these spline surfaces may require very large matrix multiplications to provide the approximation to the likelihood surface on each iteration of the MCMC.

The success of the two-stage method on the complex BMARS method indicates that it has potential in many other applications. Other potential non-linear classification methods include relevance vector machine (Tipping, 2001) and support vector machine models (Mallick, Ghosh, and Ghosh, 2005). This can also be extended in a multivariate responses framework (Holmes and Mallick, 2003). Perhaps most importantly, the two-stage method is not limited to classification problems. It can be applied to any model where a computationally cheap and accurate approximation of the likelihood can be constructed.

5. Conclusion

The results from this paper indicate there are a number of tall data Bayesian methods which are effective in obtaining/approximating the posterior distribution more quickly than traditional methods. Two-stage MH is simple to implement, fast, and overall more efficient than consensus, subsampling, or unmodified MH algorithms in our applications. Combining two-stage MH with the consensus method shows promise for even larger datasets in which the data cannot fit in RAM. Future extensions to this work include applying the method to handle more complicated likelihoods, and finding better likelihood approximations which are still computationally cheap to evaluate.

References

ANDRIEU, C., and ROBERTS, G.O. (2009), “The Pseudo-Marginal Approach for Efficient Monte Carlo Computations”, *The Annals of Statistics*, *37*(2), 697–725.

BARDENET, R., DOUCET, A., and HOLMES, C. (2014), “Towards Scaling Up Markov Chain Monte Carlo: An Adaptive Subsampling Approach”, in *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32, pp. 405–413.

BARDENET, R., DOUCET, A., and HOLMES, C. (2015), “On Markov Chain Monte Carlo Methods for Tall Data”, arXiv preprint arXiv:1505.02827.

CHRISTEN, J.A., and FOX, C. (2005), "Markov Chain Monte Carlo Using an Approximation", *Journal of Computational and Graphical Statistics*, *14*(4), 795–810.

FRIEDMAN, J.H. (1991), "Multivariate Adaptive Regression Splines", *The Annals of Statistics*, *19*(1), 1–67.

GREEN, P.J. (1995), "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination", *Biometrika*, *82(4)*, 711–732.

HIGDON, D., LEE, H., and BI, Z. (2002), "A Bayesian Approach to Characterizing Uncertainty in Inverse Problems Using Coarse and Fine-Scale Information", *Institute of Electrical and Electronics Engineers Transactions on Signal Processing*, *50(2)*, 389–399.

HOLMES, C., and DENISON, D. (2003), "Classification with Bayesian MARS", *Machine Learning*, *50(1)*, 159–173.

HOLMES, C., and MALLICK, B. (2003), "Generalized Nonlinear Modeling with Multivariate Free-Knot Regression Splines", *Journal of the American Statistical Association*, *98(462)*, 352–368.

KORATTIKARA, A., CHEN, Y., and WELLING, M. (2014), "Austerity in MCMC land: Cutting the Metropolis-Hastings Nudget", in *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*, Vol. 32, pp. 181–189.

MA, S., RACINE, J.S., and YANG, L. (2015), "Spline Regression in the Presence of Categorical Predictors", *Journal of Applied Econometrics*, *30(5)*, 705–717.

MALLICK, B.K., GHOSH, D., and GHOSH, M. (2005), "Bayesian Classification of Tumours by Using Gene Expression Data", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67(2)*, 219–234.

MONDAL, A., MALLICK, B., EFENDIEV, Y., and DATTA-GUPTA, A. (2014), "Bayesian Uncertainty Quantification for Subsurface Inversion Using a Multiscale Hierarchical Model", *Technometrics*, *56(3)*, 381–392.

MORO, S., CORTEZ, P., and RITA, P. (2014), "A Data-Driven Approach to Predict the Success of Bank Telemarketing", *Decision Support Systems*, *62*, 22–31.

QUIROZ, M., VILLANI, M., and KOHN, R. (2014), "Speeding Up MCMC by Efficient Data Subsampling", arXiv preprint arXiv:1404.4178.

RAFTERY, A.E., NIU, X., HOFF, P.D., and YEUNG, K.Y. (2012), "Fast Inference for the Latent Space Network Model Using a Case-Control Approximate Likelihood", *Journal of Computational and Graphical Statistics*, *21(4)*, 901–919.

ROBERT, C., and CASELLA, G. (2013), *Monte Carlo Statistical Methods*, New York: Springer Science & Business Media.

SCOTT, S.L., BLOCKER, A.W., BONASSI, F.V., CHIPMAN, H., GEORGE, E., and MCCULLOCH, R. (2013), "Bayes and Big Data: The Consensus Monte Carlo Algorithm", in *Economics, Finance and Business Bayes 250 Conference*, Vol. 16.

TIPPING, M.E. (2001), "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research*, *1(Jun)*], 211–244.