Multiple-try Stochastic Search for Bayesian Variable Selection

by

Xu Chen

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Surya T. Tokdar, Advisor

_____
Merlise A. Clyde

_____
Galen Reeves

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Statistical Science
in the Graduate School of Duke University
2017

Abstract

Multiple-try Stochastic Search for Bayesian Variable Selection

by

Xu Chen

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Surya T. Tokdar, Advisor

_____
Merlise A. Clyde

_____
Galen Reeves

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Statistical Science in the Graduate School of Duke University
2017

Abstract

Variable selection is a key issue when analyzing high-dimensional data. The explosion of data with large sample size and dimensionality brings new challenges to this problem in both inference efficiency and computational complexity. To alleviate these problems, a scalable Markov chain Monte Carlo (MCMC) sampling algorithm is proposed by generalizing multiple-try Metropolis to discrete model space and further incorporating neighborhood-based stochastic search. In this thesis, we study the behaviors of this MCMC sampler in the "large p small n" scenario where the number of predictors p is much greater than the number of observations n. Extensive numerical experiments including simulated and real data examples are provided to illustrate its performance. Choices of tunning parameters are discussed.

To my parents

Contents

Abstract **iv**

List of Tables **viii**

List of Figures **ix**

Acknowledgements **x**

1 Introduction **1**

2 Background **4**

2.1 Bayesian variable selection 4

2.2 Neighborhood-based stochastic search MCMC samplers 6

2.3 Multiple-try Metropolis 7

3 A paired-move multiple-try stochastic search sampler **9**

3.1 Paired-move neighborhood search 10

3.2 A paired-move multiple-try stochastic search MCMC algorithm . . . 11

3.2.1 A mixed discrete multiple-try sampler 12

3.2.2 A paired-move multiple-try stochastic search sampler 13

3.3 Adaptive predictor importance 15

4 Numerical studies **17**

4.1 Effectiveness of pMTM 18

4.2 Simulated examples 20

4.3 Real data examples 27

4.3.1 Near-infrared (NIR) Spectroscopy Analysis of Biscuit Doughs 27
4.3.2 Riboflavin Production with Bacillus Subtilis 27
4.4 Computational efficiency 29
5 Discussion **32**
A Proof of Theorem 3.2.2 **34**
Bibliography **36**

List of Tables

4.1 Independent design with $(n, p_0) = (100, 8)$ 23

4.2 Compound symmetry with $(n, p, p_0) = (100, 1000, 5)$ 24

4.3 Autoregressive correlation with $(n, p, p_0) = (100, 1000, 12)$ 25

4.4 Group structure with $(n, p, p_0) = (100, 1000, 8)$ 26

4.5 The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with fat as response. . . . 28

4.6 The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with sucrose as response. 29

4.7 The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with dry flour as response. 29

4.8 The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with water as response. . 30

4.9 The fitting and prediction results of different methods (averaged over 10 replicates) for riboflavin dataset. 30

List of Figures

4.1 Logarithm of the median number of marginal likelihood evaluations needed to find the true model 19

4.2 Logarithm of the median number of marginal likelihood evaluations needed to find the true model for *ada-pMTM* with different expected number of trails (M). 19

4.3 Heat maps of correlation matrices of the first 100 predictors. Left: NIR spectroscopy example with all 700 predictors highly correlated. Right: Riboflavin gene expression example with a grouped correlation structure. 28

4.4 Logarithm of the mean number of marginal likelihood evaluations within 10 seconds 31

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. Surya Tokdar, for his invaluable advice and perpetual encouragement during this project. He listens to my ideas and provides me insightful feedbacks. I would not have finished this thesis without his help. Except for research projects, he also inspires me on my personal development with his experience and knowledge.

I would like to thank Dr. Merlise Clyde for serving as my committee member and her wonderful lectures on Bayesian linear models. I thank my committee member Dr. Galen Reeves for his time and support. I would also like to thank Dr. Robert Wolpert and Dr. Li Ma, who taught me core statistics courses with enthusiasm and insightful ideas. Further thanks go to my first year advisor Dr. Scott Schmidler for giving me helpful guidance when I was a fresh graduate student.

I would also like to thank my English teacher and friend Diane Bryson for her generous and continued support. Thanks also go to all my friends at Duke and in China.

Finally, I am immensely grateful to my parents, for their eternal support and encouragement.

1

Introduction

The explosion of data with large sample size and dimensionality brings new challenges to classical statistical methods. Under the "large p small n" scenario, such as gene expression and network data, sparse models are essential to improve prediction accuracy, reduce computational complexity, and enhance model interpretability. In the context of linear regression, variable selection is crucial when the underlying true model is known to have a sparse representation or only a subset of predictors has predictive power.

Consider the canonical Gaussian linear regression model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \qquad \varepsilon \sim \mathsf{N}(0, \mathbf{I}_n/\phi)$$

where $\mathbf{Y} \in \mathbb{R}^n$ is a response vector and $\mathbf{X} \in \mathbb{R}^{n\times p}$ is a design matrix for n samples and p predictors. Assume $\mathbf{Y}$ and $\mathbf{X}$ are standardized and hence an intercept is not included. $\boldsymbol{\beta} \in \mathbb{R}^p$ is an unknown regression coefficient vector. Accurately recovering the support of and estimating $\boldsymbol{\beta}$ is the central task of variable selection. A wide variety of methods have been proposed to approach this problem. Many popular

selection procedures are based on solving the penalized likelihood

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \text{Pen}\,(\boldsymbol{\beta}) \tag{1.1}$$

by optimization methods. Akaike's information criterion (AIC), Bayesian information criterion (BIC), and Mellows' C_p are of this form when using ℓ_0 penalty, that is, $\text{Pen}\,(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_0$, where $\|\boldsymbol{\beta}\|_0$ is the number of nonzero coefficients in $\boldsymbol{\beta}$. Such methods achieve the best trade-off between goodness-of-fit and the number of predictors in the model. However, a comparison between all 2^p possible models is computationally intractable when $p > 30$. It turns out that for ℓ_q penalty, $q = 1$ is the largest possible value inducing a sparse solution and also the smallest value inducing a convex optimization problem. *lasso* [Tibshirani (1996)] employs the standard ℓ_1 penalty. Yuan and Lin (2006), Zou (2006), Bogdan et al. (2015), and Tibshirani et al. (2005) generalized this idea by considering groups, weights and differences of predictors. Other widely-used penalties include *SCAD* [Fan and Li (2001)], *elastic net* [Zou and Hastie (2005)], and *MC+* [Zhang et al. (2010)]. Many of these methods enjoy asymptotic guarantees.

There is also a large amount of literature in Bayesian variable selection. A typical recipe is to impose shrinkage priors over the model space and the regression coefficients. Among these methods, Park and Casella (2008), Hans (2009), and Polson et al. (2014) provide Bayesian interpretations of penalized likelihood discussed above. Other well known shrinkage priors include Zellner's g-prior [Zellner (1986)], "spike-and-slab" prior [Mitchell and Beauchamp (1988), George and McCulloch (1993)] and horseshoe [Polson et al. (2012)]. In addition to shrinkage models, a key ingredient in Bayesian variable selection is the sampling algorithm. Stochastic search variable selection (SSVS) [George and McCulloch (1993)] is a systematic-scan Gibbs sampler which explore the model space by successively scanning over all predictors. This sequential scheme may suffer from slow convergence when the number of predictors is

large or the predictors are correlated. To mitigate this problem, Ročková and George (2014) adopts EM algorithm to deterministically move towards the posterior modes. Another type of algorithms adopt the idea of neighborhood-based stochastic search. Madigan et al. (1995) and Raftery et al. (1997) introduce a Metropolis-Hastings algorithm called MC^3 which moves to a one-step away neighborhood in each iteration. Hans et al. (2007) further include "swap" moves, replacing one current predictor by a predictor excluded in the current model, to efficiently identify models with large posterior probability. Bottolo et al. (2010) extends the idea of evolutionary Monte Carlo to allow for jumping between local modes in a model space. Clyde et al. (2011) develops a Bayesian adaptive sampling algorithm which use sampling without replacement to sequentially learn the marginal inclusion probabilities of predictors.

In this thesis, we introduce and study the behaviors of a scalable MCMC sampler [Qamar and Tokdar (2014)] for efficiently sampling the model space by generalizing the multiple-try Metropolis [Liu et al. (2000)] and further incorporating neighborhood-based stochastic search [Hans et al. (2007)]. The rest of the thesis is organized as follows. In Chapter 2, we start by establishing notations and presenting the hierarchical formulation of Bayesian variable selection. Conjugate priors for regression coefficients and the predictor inclusion vector are introduced. Then we briefly review the neighborhood-based stochastic search and multiple-try Metropolis algorithms. We introduce a scalable MCMC sampler for predictor inclusion vector by generalizing the multiple-try Metropolis and combining with neighborhood-based stochastic search in Chapter 3. Extensive simulation studies are provided in Chapter 4 to examine the effectiveness of the proposed algorithm according to inference accuracy, prediction performances, and computational efficiency. We conclude in Chapter 5 with discussions on the future research directions.

2

Background

2.1 Bayesian variable selection

In Bayesian paradigm, variable selection is typically performed by introducing a p dimensional binary latent indicator vector $\boldsymbol{\gamma} \in \{0,1\}^p$. Denote the set of indices of predictors $\{1, 2, ..., p\}$ as $[p]$. For each $i \in [p]$, $\gamma_i = 1$ if $\mathbf{X}_i$ is included in the model. $\boldsymbol{\gamma}$ can also be viewed as the set of indices of active predictors (i.e., a subset of $[p]$) in the affiliated model $\mathcal{M}_{\boldsymbol{\gamma}}$ where $|\boldsymbol{\gamma}|$ and $\boldsymbol{\gamma}^c$ denote the cardinality and complement of $\boldsymbol{\gamma}$. Under $\mathcal{M}_{\boldsymbol{\gamma}}$, a conjugate hierarchical model is typically constructed as follows:

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}, \phi \sim \mathsf{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}/\phi) \tag{2.1}$$

An independent prior is obtained by specifying $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \mathbf{I}_{|\boldsymbol{\gamma}|}$. Another conventional choice is a g-prior where $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = g(\mathbf{X}_{\boldsymbol{\gamma}}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1}$ [Zellner (1986)]. This type of prior preserves correlation structures of design matrices and leads to simple closed-form marginal likelihoods. Models with small size are preferred when larger values of g are adopted. See Liang et al. (2008) for a detailed discussion of the effects of g.

$$\phi \sim \mathsf{Gamma}(a, b) \tag{2.2}$$

Generally, a and b are chosen to be small constants, resulting in a non-informative prior for ϕ. However, ϕ is expected to be larger when including more predictors in the model. Therefore, George and McCulloch (1993) and Dobra et al. (2004) consider relating a or b to $|\boldsymbol{\gamma}|$. When $a, b \to 0$, we get a popular improper prior $\pi(\phi) \propto 1/\phi$.

$$\pi(\boldsymbol{\gamma} \mid \tau) = \prod_{j=1}^{p} \tau^{\gamma_j}(1-\tau)^{1-\gamma_j} = \tau^{|\boldsymbol{\gamma}|}(1-\tau)^{p-|\boldsymbol{\gamma}|} \tag{2.3}$$

The prior for $\boldsymbol{\gamma}$ only depends on its size. Fixing $\tau = 1/2$ yields a uniform distribution for all 2^p models with expected model size of $p/2$. This prior fails to penalize large models. A more reasonable approach is to treat τ as a hyperparameter with a Beta prior. See Scott et al. (2010) for theoretical properties of this prior.

$$\tau \sim \mathsf{Beta}(u, v) \tag{2.4}$$

Let d^* be the expected model size. We may set $u = d^*$ and $v = p - d^*$ resulting in $\mathbf{E}[\tau] = d^*$ and $\mathbf{Var}[\tau] \approx d^*/p^2$ when $d^* = o(p)$. A marginal Beta-binomial distribution for $\boldsymbol{\gamma}$ is

$$\pi(\boldsymbol{\gamma}) = \frac{B(|\boldsymbol{\gamma}| + u, p - |\boldsymbol{\gamma}| + v)}{B(u, v)} \tag{2.5}$$

where $B(\cdot, \cdot)$ is the Beta function.

All the simulations performed in the thesis adopt g-prior with $g = n$ (i.e., unit information prior [Kass and Wasserman (1995)]), $\pi(\phi) \propto 1/\phi$ and a Beta-binomial prior for $\boldsymbol{\gamma}$. Under these settings, the marginal likelihood is given by

$$\mathcal{L}_n(\mathbf{Y} \mid \boldsymbol{\gamma}) = \int \pi(\mathbf{Y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \phi)\pi(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \phi, \boldsymbol{\gamma})\pi(\phi)\mathrm{d}\boldsymbol{\beta}_{\boldsymbol{\gamma}}\mathrm{d}\phi \tag{2.6}$$

$$= \frac{\Gamma(n/2)(1+g)^{n/2}}{\pi^{n/2}\|\mathbf{Y}\|_2^n} \frac{(1+g)^{-|\boldsymbol{\gamma}|/2}}{[1+g(1-R^2_{\boldsymbol{\gamma}})]^{n/2}} \tag{2.7}$$

where $\Gamma(\cdot)$ is the Gamma function and $R^2_{\boldsymbol{\gamma}}$ is the ordinary coefficient of determination for the model $\mathcal{M}_{\boldsymbol{\gamma}}$

$$R^2_{\boldsymbol{\gamma}} = \frac{\mathbf{Y}^T \mathbf{P}_{\mathbf{X}_{\boldsymbol{\gamma}}} \mathbf{Y}}{\|\mathbf{Y}\|_2^2} \tag{2.8}$$

with $\mathbf{P}_{\mathbf{X}_\gamma} = \mathbf{X}_\gamma(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma^T$ the projection matrix onto the column space of $\mathbf{X}_\gamma$.

2.2 Neighborhood-based stochastic search MCMC samplers

Let $T(\boldsymbol{\gamma}, \cdot)$ be a proposal transition function over $N(\boldsymbol{\gamma})$, the neighborhood set of $\boldsymbol{\gamma}$. Here $T(\boldsymbol{\gamma}, \boldsymbol{\gamma}') > 0 \iff T(\boldsymbol{\gamma}', \boldsymbol{\gamma}) > 0$ is required to guarantee reversibility. Then a Metropolis-Hastings (MH) random walk neighborhood search algorithm is implemented iteratively as follows:

1. Randomly select a proposal state $\boldsymbol{\gamma}' \in N(\boldsymbol{\gamma})$ according to $T(\boldsymbol{\gamma}, \cdot)$.

2. Accept proposal $\boldsymbol{\gamma}'$ with probability α where

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y})T(\boldsymbol{\gamma}', \boldsymbol{\gamma})}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y})T(\boldsymbol{\gamma}, \boldsymbol{\gamma}')}\right\}$$

   otherwise stay at $\boldsymbol{\gamma}$.

This algorithm generates an irreducible, aperiodic, and positive recurrent Markov chain. Let 1_j be a $p \times 1$ vector with j^{th} element 1 and others 0. Then neighborhood set considered by Hans et al. (2007) consists of three types of moves:

1. Add an inactive predictor: $N_A(\boldsymbol{\gamma}) = \{\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}' = \boldsymbol{\gamma} + 1_j, j \in \boldsymbol{\gamma}^c\}$

2. Remove an active predictor: $N_R(\boldsymbol{\gamma}) = \{\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}' = \boldsymbol{\gamma} - 1_j, j \in \boldsymbol{\gamma}\}$

3. Swap an active predictor with an inactive predictor: $N_S(\boldsymbol{\gamma}) = \{\boldsymbol{\gamma}' \mid \boldsymbol{\gamma}' = \boldsymbol{\gamma} - 1_j + 1_k, (j,k) \in \boldsymbol{\gamma} \times \boldsymbol{\gamma}^c\}$

and $N(\boldsymbol{\gamma}) = N_A(\boldsymbol{\gamma}) \cup N_R(\boldsymbol{\gamma}) \cup N_S(\boldsymbol{\gamma})$. Swap move is actually the combination of adding and removing. Yang et al. (2016) further unifies 1 and 2 into one class based on Hamming distance.

A MCMC sampler built on "shotgun stochastic search" (SSS) algorithm provided in Hans et al. (2007) can be obtained immediately by defining

$$T(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \frac{S(\boldsymbol{\gamma}')}{\sum_{\tilde{\boldsymbol{\gamma}} \in N(\boldsymbol{\gamma})} S(\tilde{\boldsymbol{\gamma}})} \tag{2.9}$$

where S is any positive score function.

2.3 Multiple-try Metropolis

Multiple-try Metropolis algorithm is proposed by Liu et al. (2000) to mitigate the potential slow convergence problem of traditional MH algorithms. Instead of only considering a single proposal, MTM proposes multiple trials each iteration to prevent the chain from being stuck in local modes in a continuous state space. Specifically, suppose π is the target distribution and T is a transition kernel. Further define weight $\omega(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{x})T(\mathbf{x}, \mathbf{y})$. Then a general MTM algorithm involve the following procedures:

1. Sample M i.i.d. proposals $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M$ according to $T(\mathbf{x},)$.

2. Select $\mathbf{y} \in \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}$ with probability proportional to $\omega(\mathbf{y}_j, \mathbf{x}) \quad j = 1, 2, ..., M$.

3. Sample backward set $\{\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_{M-1}^*\}$ according to $T(\mathbf{y},)$ and set $\mathbf{x}_M^* = \mathbf{x}$.

4. Accept the proposal $\mathbf{y}$ with probability α where

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\sum_{j=1}^{M} \omega(\mathbf{y}_j, \mathbf{x})}{\sum_{j=1}^{M} \omega(\mathbf{x}_j^*, \mathbf{y})}\right\}$$

   otherwise stay at $\mathbf{x}$.

This algorithm generates a reversible Markov chain leaving π as the invariant distribution. We note that M is a user-specified number of trials in each iteration

and can be adjusted for different problems and computational budget. Additionally, $\omega(\mathbf{y}_j, \mathbf{x})$ and $\omega(\mathbf{x}_j^*, \mathbf{y})$s can be evaluated in parallel. The standard MH sampler results as a special case when $M = 1$. In Liu et al. (2000), MTM is demonstrated to be more efficient on multimodal state space exploration than traditional MH algorithms through simulation studies. Pandolfi et al. (2010) extends this approach by further incorporating an additional weight $\omega^*(\mathbf{x}, \mathbf{y})$. The original MTM is obtained when $\omega^*(\mathbf{x}, \mathbf{y}) = \omega(\mathbf{x}, \mathbf{y})$.

3

A paired-move multiple-try stochastic search sampler

The MCMC sampler built on SSS may suffer from two problems. First, the chain may be stuck due to substantially low acceptance rates. Suppose that the current state is $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$ is proposed. When $\boldsymbol{\gamma}'$ has better neighborhoods than $\boldsymbol{\gamma}$, $\sum_{\tilde{\boldsymbol{\gamma}} \in N(\boldsymbol{\gamma}')} S(\tilde{\boldsymbol{\gamma}})$ is much larger than $\sum_{\tilde{\boldsymbol{\gamma}} \in N(\boldsymbol{\gamma})} S(\tilde{\boldsymbol{\gamma}})$ leading to a small acceptance rate. Therefore, the sampler may be able to identify inclusion vectors with high posterior probabilities but fail to transition to them. Another concern is computational complexity. We notice that $|\boldsymbol{\gamma}| + |\boldsymbol{\gamma}'|$ remove, $2p - |\boldsymbol{\gamma}| - |\boldsymbol{\gamma}'|$ add, and $|\boldsymbol{\gamma}|(p - |\boldsymbol{\gamma}|) + |\boldsymbol{\gamma}'|(p - |\boldsymbol{\gamma}'|)$ swap neighborhoods are evaluated in each iteration. This $O(p)$ cost is further exacerbated when n is large. Reducing the length of a chain is inevitable when computational budget is limited, resulting in poor mixing and unreliable inferences.

Qamar and Tokdar (2014) introduces a MCMC sampler by combining the idea of neighborhood-based stochastic search and MTM to address the issues described above. We review this sampler in detail in this chapter. Specifically, a paired-move strategy is introduced in Section 3.1 to improve acceptance rates. In Section 3.2,

multiple-try scheme is generalized to discrete model spaces to allow for a flexible and efficient neighborhood search. Adaptive scores for predictors are further incorporated according to the correlation structure of design matrix and previous posterior samples to improve mixing in Section 3.3.

3.1 Paired-move neighborhood search

The paired-move strategy is motivated by the following fact:

$$\begin{aligned} \boldsymbol{\gamma}' \in N_A(\boldsymbol{\gamma}) &\Longleftrightarrow \boldsymbol{\gamma} \in N_R(\boldsymbol{\gamma}') \\ \boldsymbol{\gamma}' \in N_R(\boldsymbol{\gamma}) &\Longleftrightarrow \boldsymbol{\gamma} \in N_A(\boldsymbol{\gamma}') \\ \boldsymbol{\gamma}' \in N_S(\boldsymbol{\gamma}) &\Longleftrightarrow \boldsymbol{\gamma} \in N_S(\boldsymbol{\gamma}') \end{aligned} \tag{3.1}$$

Therefore, a forward move $\boldsymbol{\gamma} \to \boldsymbol{\gamma}'$ and the corresponding backward move $\boldsymbol{\gamma}' \to \boldsymbol{\gamma}$ are paired. We introduce a paired-move reversible neighborhood sampler ($pRNS$) with "add-remove", "remove-add", and "swap-swap" as forward-backward neighborhoods. The $pRNS$ proposal transition function is defined by

$$T(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = w_A T_A(\boldsymbol{\gamma}, \boldsymbol{\gamma}') + w_R T_R(\boldsymbol{\gamma}, \boldsymbol{\gamma}') + w_S T_S(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \tag{3.2}$$

where w_A, w_R, and w_S are probabilities of proposing add, remove, and swap moves respectively and T_A, T_R, and T_S are proposal transition functions as in 2.9 restricted to their corresponding sets of neighborhoods.

Naturally, w_A, w_R, and w_S are positive and sum to 1. These probabilities are allowed to vary with the current model size to encourage the chain moving towards the models with desired sizes. Note that when $d^* = o(p)$, random-walk Gibbs samplers are heavily biased toward attempting adding additional predictors instead of removing undesirable ones. The inefficiency of random selection is addressed by utilizing this paired-move strategy. For simplicity, the following settings are adopted in all simulations in the thesis:

$$w_A(0) = w_R(p) = 1 \quad \text{and} \quad w_A(|\boldsymbol{\gamma}|) = w_R(|\boldsymbol{\gamma}|) = w_S(|\boldsymbol{\gamma}|) = \frac{1}{3} \text{ if } 0 < |\boldsymbol{\gamma}| < p \tag{3.3}$$

The resulting MCMC algorithm adopting *pRNS* is as follows:

1. Select move $m \in \{A, R, S\}$ with probabilities w_A, w_R, and w_S.

2. Construct the forward set of neighborhoods $N_m(\boldsymbol{\gamma})$.

3. Randomly select a proposal state $\boldsymbol{\gamma}' \in N_m(\boldsymbol{\gamma})$ according to $T_m(\boldsymbol{\gamma}, \cdot)$.

4. Construct the backward set of neighborhoods $N'_m(\boldsymbol{\gamma}')$ where $m' \in \{R, A, S\}$ is the backward move correspond to m.

5. Accept the proposal $\boldsymbol{\gamma}'$ with probability α where

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y})[w_{m'}(|\boldsymbol{\gamma}'|)T_{m'}(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y})[w_m(|\boldsymbol{\gamma}|)T_m(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]}\right\} \tag{3.4}$$

   otherwise stay at $\boldsymbol{\gamma}$.

Remark 3.1.1. *It is clear that the Markov chain produced by this algorithm is reversible because it satisfies the detailed balance condition. Since the neighborhoods evaluated in each iteration are restricted to a subset of $N(|\boldsymbol{\gamma}|)$, pRNS efficiently reduces the computational cost. As p becomes larger, noticing that add and swap neighborhoods remain $O(p)$, an additional mechanism is essential to limit the size of neighborhoods which is the main concern of Section 3.2.*

3.2 A paired-move multiple-try stochastic search MCMC algorithm

It is inefficient to evaluate a large number of neighborhoods in each iteration. A flexible computational cost is desired to accommodate for the requirement of inference efficiency and computational budget. One attractiveness of the MTM is that the computational cost can be adjusted by tuning the number of trails M.

3.2.1 A mixed discrete multiple-try sampler

Instead of considering all neighborhoods, a general framework for generating a stochastic set of neighborhoods of the current state $\boldsymbol{\gamma}$ was proposed. To formulate this method, we first define the "toggle function" $\text{tog} : [p] \times \{0,1\}^p \to \{0,1\}^p$ as follows:

$$\text{tog}(i, \boldsymbol{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_i, ..., \gamma_p)) = (\gamma_1, \gamma_2, ..., 1 - \gamma_i, ..., \gamma_p) \tag{3.5}$$

Namely, if i^{th} predictor is included(excluded) in the current state $\boldsymbol{\gamma}$, then $\boldsymbol{\gamma}' = \text{tog}(i, \boldsymbol{\gamma})$ is a neighborhood removing(adding) i^{th} predictor. Note that a swap move between j^{th} and k^{th} predictors is $\text{tog}(k, \text{tog}(j, \boldsymbol{\gamma}))$ (or $\text{tog}(j, \text{tog}(k, \boldsymbol{\gamma}))$) for $\gamma_j + \gamma_k = 1$.

To introduce stochasticity, we further define $\eta_i \sim \mathsf{Ber}(\omega(\gamma_i, v_i))$ for $i \in [p]$ with a weight function $\omega : \{0,1\} \times \mathbb{R}^+ \to [0,1]$ taking inputs γ_i and a nonnegative predictor importance score v_i. For simplicity, we do not consider swap moves which will be handled in detail in the next section and focus on a mixed set of neighborhoods only containing add and remove neighborhoods for now. In these settings, the forward set of neighborhoods of $\boldsymbol{\gamma}$ is defined as $N_{mix}(\boldsymbol{\gamma}) = \{\text{tog}(i, \boldsymbol{\gamma}) \mid \eta_i = 1, i \in [p]\}$ and $T_{mix}(\boldsymbol{\gamma}, \cdot)$ is a proposal transition function as in 2.9 restricted to $N_{mix}(\boldsymbol{\gamma})$. An algorithm for this generalized discrete MTM (dMTM) over a model space is:

1. For current state $\boldsymbol{\gamma}$ and $i \in [p]$, independently sample $\eta_i \sim \mathsf{Ber}(\omega(\gamma_i, v_i))$.

2. Form the forward mixed set of neighborhoods of $\boldsymbol{\gamma}$: $N_{mix}(\boldsymbol{\gamma}) = \{\text{tog}(i, \boldsymbol{\gamma}) \mid \eta_i = 1, i \in [p]\}$.

3. Select $\boldsymbol{\gamma}' = \text{tog}(i^*, \boldsymbol{\gamma}) \in N_{mix}(\boldsymbol{\gamma})$ according to $T_{mix}(\boldsymbol{\gamma}, \cdot)$.

4. For the proposed state $\boldsymbol{\gamma}'$ and $j \neq i^* \in [p]$, independently sample $\eta'_j \sim \mathsf{Ber}(\omega(\gamma'_j, v_j))$ and set $\eta'_{i*} = 1$.

5. Form the backward mixed set of neighborhoods of $\boldsymbol{\gamma}'$: $N'_{mix}(\boldsymbol{\gamma}') = \{\text{tog}(j, \boldsymbol{\gamma}') \mid \eta'_j = 1, j \in [p]\}$.

6. Accept the proposal $\boldsymbol{\gamma}'$ with probability α where

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y})[\omega(\gamma'_{i*}, v_{i*})T'_{mix}(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y})[\omega(\gamma_{i*}, v_{i*})T_{mix}(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]}\right\} \tag{3.6}$$

otherwise stay at $\boldsymbol{\gamma}$.

Remark 3.2.1. *Efficient strategies of specifying v_i and ω_i for $i \in [p]$ would enhance the possibilities of including important predictors and excluding undesirable ones. An adaptive configuration is provided in Section 3.3.*

3.2.2 A paired-move multiple-try stochastic search sampler

A paired-move multiple-try stochastic search MCMC algorithm ($pMTM$) is obtained as a special case of the dMTM algorithm under the following configuration of weight function ω:

$$\omega(\gamma_i, v_i; m) = (1 - \gamma_i)f(v_i)\mathbb{1}_{\{m=A\}} + \gamma_i g(v_i)\mathbb{1}_{\{m=R\}} \tag{3.7}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function and $f, g : \mathbb{R}^+ \to [0, 1]$ determine the probabilities of including and removing predictors. Note that here we further take the type of move into account. It is reasonable because it allows for including an important predictor i with high probability (large $f(v_i)$) and being preserved (small $g(v_i)$). Then the $pMTM$ algorithm is given as:

1. Select move $m \in \{A, R, S\}$ with probabilities $w_A(|\boldsymbol{\gamma}|), w_R(|\boldsymbol{\gamma}|)$, and $w_S(|\boldsymbol{\gamma}|)$.

2. (a) If move $m \in \{A, R\}$: for $i \in [p]$, independently sample $\eta_i \sim \mathsf{Ber}(\omega(\gamma_i, v_i; m))$. Define the forward add or remove set as $N_F(\boldsymbol{\gamma}) = \{\mathrm{tog}(i, \boldsymbol{\gamma}) \mid \eta_i = 1, i \in [p]\}$.

   (b) If move $m = S$: for $(a, r) \in \boldsymbol{\gamma}^c \times \boldsymbol{\gamma}$, sample $\eta_a \sim \mathsf{Ber}(\omega(\gamma_a, v_a; A))$, and independently sample $\eta_r \sim \mathsf{Ber}(\omega(\gamma_r, v_r; R))$ (totally sample $|\boldsymbol{\gamma}^c||\boldsymbol{\gamma}|$ Bernoulli random variables). Define the forward swap set as $N_F(\boldsymbol{\gamma}) = \{\mathrm{tog}(a, \mathrm{tog}(r, \boldsymbol{\gamma})) \mid \eta_a = \eta_r = 1, (a, r) \in \boldsymbol{\gamma}^c \times \boldsymbol{\gamma}\}$.

3. Select $\boldsymbol{\gamma}' \in N_F(\boldsymbol{\gamma})$ according to $T_F(\boldsymbol{\gamma}, \cdot)$. If $m \in \{A, R\}$, denote $\boldsymbol{\gamma}' = \text{tog}(i^*, \boldsymbol{\gamma})$; otherwise denote $\boldsymbol{\gamma}' = \text{tog}(a^*, \text{tog}(r^*, \boldsymbol{\gamma}))$ for $m = S$.

4. (a) If move $m = A$: for $j \neq i^*$, sample $\eta'_j \sim \mathsf{Ber}(\omega(\gamma'_j, v_j; R))$ and set $\eta'_{i^*} = 1$. Define the backward remove set as $N_B(\boldsymbol{\gamma}') = \{\text{tog}(j, \boldsymbol{\gamma}') \mid \eta'_j = 1, j \in [p]\}$.

   (b) If move $m = R$: for $j \neq i^*$, sample $\eta'_j \sim \mathsf{Ber}(\omega(\gamma'_j, v_j; A))$ and set $\eta'_{i^*} = 1$. Define the backward add set as $N_B(\boldsymbol{\gamma}') = \{\text{tog}(j, \boldsymbol{\gamma}') \mid \eta'_j = 1, j \in [p]\}$.

   (c) If move $m = S$: for $(a', r') \in (\boldsymbol{\gamma}')^c \times \boldsymbol{\gamma}'$, sample $\eta'_{a'} \sim \mathsf{Ber}(\omega(\gamma'_{a'}, v_{a'}; A))$, and independently sample $\eta'_{r'} \sim \mathsf{Ber}(\omega(\gamma'_{r'}, v_{r'}; R))$ (totally sample $|(\boldsymbol{\gamma}')^c||\boldsymbol{\gamma}'|$ Bernoulli random variables) and set $(\eta'_{a^*}, \eta'_{r^*}) = (1, 1)$. Define the backward swap set as $N_B(\boldsymbol{\gamma}') = \{\text{tog}(a', \text{tog}(r', \boldsymbol{\gamma}')) \mid \eta'_{a'} = \eta'_{r'} = 1, (a', r') \in (\boldsymbol{\gamma}')^c \times \boldsymbol{\gamma}'\}$.

5. For $m \in \{A, R, S\}$, the corresponding backward paired-move is $m' \in \{R, A, S\}$. Accept the proposal $\boldsymbol{\gamma}'$ with probability α where

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \begin{cases} \min\left\{1, \dfrac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y})[w_{m'}(|\boldsymbol{\gamma}'|)\omega(\gamma'_{i^*}, v_{i^*}; m')T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y})[w_m(|\boldsymbol{\gamma}|)\omega(\gamma_{i^*}, v_{i^*}; m)T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]}\right\} & \text{if } m \in \{A, R\} \\ \min\left\{1, \dfrac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y})[\omega(\gamma'_{r^*}, v_{r^*}; A)\omega(\gamma'_{a^*}, v_{a^*}; R)T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y})[\omega(\gamma_{a^*}, v_{a^*}; A)\omega(\gamma_{r^*}, v_{r^*}; R)T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]}\right\} & \text{if } m = S \end{cases} \tag{3.8}$$

   otherwise stay at $\boldsymbol{\gamma}$.

Theorem 3.2.2. *The paired-move multiple-try stochastic search MCMC (pMTM) algorithm with acceptance probability 3.8 satisfies the detailed balance condition leaving the desired target distribution $\pi(\boldsymbol{\gamma} \mid \mathbf{Y})$ invariant.*

Remark 3.2.3. *The proof is provided in Appendix A and is established on a general form of T as in 2.9. If we specify $S(\tilde{\boldsymbol{\gamma}}) \propto \pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y})$, the unnormalized marginal*

posterior probability for $\tilde{\boldsymbol{\gamma}}$, then the acceptance ratio α is

$$\alpha(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \begin{cases} \min\left\{1, \quad \dfrac{w_{m'}(|\boldsymbol{\gamma}'|)\omega(\gamma'_{i*}, v_{i*}; m') \sum\limits_{\tilde{\boldsymbol{\gamma}} \in N_F(\boldsymbol{\gamma})} \pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y})}{w_m(|\boldsymbol{\gamma}|)\omega(\gamma_{i*}, v_{i*}; m) \sum\limits_{\tilde{\boldsymbol{\gamma}} \in N_B(\boldsymbol{\gamma}')} \pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y})}\right\} & \text{if } m \in \{A, R\} \\ \min\left\{1, \quad \dfrac{\omega(\gamma'_{r*}, v_{r*}; A)\omega(\gamma'_{a*}, v_{a*}; R) \sum\limits_{\tilde{\boldsymbol{\gamma}} \in N_F(\boldsymbol{\gamma})} \pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y})}{\omega(\gamma_{a*}, v_{a*}; A)\omega(\gamma_{r*}, v_{r*}; R) \sum\limits_{\tilde{\boldsymbol{\gamma}} \in N_B(\boldsymbol{\gamma}')} \pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y})}\right\} & \text{if } m = S \end{cases} \tag{3.9}$$

All simulations in the thesis are performed under this setting. Note that $\pi(\tilde{\boldsymbol{\gamma}} \mid \mathbf{Y}) \propto \mathcal{L}(\mathbf{Y} \mid \tilde{\boldsymbol{\gamma}})\pi(\tilde{\boldsymbol{\gamma}})$. When the sample size n is large, computing $\mathcal{L}(\mathbf{Y} \mid \tilde{\boldsymbol{\gamma}})$ will be expensive. An alternative choice is using Laplace approximation of the marginal likelihood $\hat{\mathcal{L}}(\mathbf{Y} \mid \tilde{\boldsymbol{\gamma}})$ and hence $S(\tilde{\boldsymbol{\gamma}}) = \hat{\mathcal{L}}(\mathbf{Y} \mid \tilde{\boldsymbol{\gamma}})\pi(\tilde{\boldsymbol{\gamma}})$.

Remark 3.2.4. *This framework can be adapted to various settings based on different problems, structures of datasets and computational budgets. Adopting adaptive importance scores for predictors within this framework is discussed in the next section.*

3.3 Adaptive predictor importance

In spectroscopy or gene expression data (see Section 4.3), for example, predictors are typically grouped and highly correlated within a group because of their spatial proximity and hence provide similar predictive power. Weight function $\omega(\gamma_i, v_i; m)$ for $i \in [p]$ and $m \in \{A, R\}$ can efficiently improve mixing for sampling predictor inclusion vectors when the correlation structure of the design matrix is far from independent.

Suppose that f and g in 3.7 are monotone increasing and decreasing functions of v_i respectively. Therefore, as importance scores are updated, predictors with large v_i are promoted within add neighborhoods and demoted in remove neighborhoods.

Denote the length of the MCMC chain as T with a burnin period b_0. Define a $p \times p$ thresholded absolute correlation matrix $\mathbf{C}$ with

$$C_{ij} = |\rho_{ij}| \mathbb{1}_{\{|\rho_{ij}| > \varepsilon\}} \tag{3.10}$$

where $\rho_{ij} = \text{Cor}(\mathbf{X}_i, \mathbf{X}_j)$ is the empirical correlation between predictors $i, j \in [p]$ and $\varepsilon \in (0, 1)$ is a pre-specified threshold. By incorporating the correlation structure of the design matrix and the history of the MCMC chain, adaptive importance scores for predictors are introduced. For all $i \in [p]$ at $(t+1)^{th}$ iteration, $v_i(t+1)$ is updated as follows:

$$v_i(t+1) = v_i(t) + z(i, \boldsymbol{\gamma}) \left(\frac{t}{b_0} \mathbb{1}_{\{t \leqslant b_0\}} + \frac{1}{(t - b_0)^{\zeta}} \mathbb{1}_{\{t > b_0\}} \right) \tag{3.11}$$

with $z : [p] \times \{0, 1\}^p \to [0, 1]$; in particular, $z(i, \boldsymbol{\gamma}) = (1 - \gamma_i)(\sum_{j=1}^p \gamma_j C_{ij} / \sum_{j=1}^p \gamma_j) + \gamma_i$. Qamar and Tokdar (2014) suggests specifying the learning rate $\zeta \in (0.5, 1]$ as 2/3 following convention from stochastic gradient descent. Further modification may be adopting the quantile of $|\rho_{ij}|$s as the threshold to ensure a fixed ratio of entries of $\mathbf{C}$ are zeros. Based on this updating scheme for importance scores, an adaptive version of the *pMTM* can be constructed by defining

$$f(v_i) = \frac{M v_i}{M v_i + p} \qquad g(v_i) = 1 \tag{3.12}$$

Under this configuration, we initialize the *pMTM* sampler with $\boldsymbol{\gamma} = (0, 0, ..., 0)^T$ and $v_i = 1$ for all $i \in [p]$. Accordingly, M can be viewed as a target "neighborhood budget" noting that the expected number of add neighborhoods is $\sum_{i \notin \boldsymbol{\gamma}} f(v_i) \approx M(p - |\boldsymbol{\gamma}|)/(p + M) \approx M$ initially when $M = o(p)$. When the true model size $d = o(p)$, most of the importance scores retain $v_i \approx 1$ and hence the stochastic control of the number of neighborhoods is maintained. Stationarity of the *pMTM* sampler is preserved subject to diminishing adaptation of predictor importance scores [Roberts and Rosenthal (2007)]. This adaptive version of the *pMTM* sampler is denoted as *ada-pMTM*.

4

Numerical studies

In this chapter[1], we study the behaviors of *pMTM* empirically by numerical experiments. In Section 4.1, two examples are provided to illustrate the effectiveness of *pMTM* (and *ada-pMTM*) on model space exploration. Through the comparisons with other methods based on intensive simulation results, we illustrate the state-of-the-art performance of *pMTM* on identifying relevant predictors in 4.2. Two real data examples are presented in 4.3 to examine the prediction performance of *pMTM*. We close this chapter by a discussion of computational efficiency on a toy example in Section 4.4. Except for Section 4.4, all simulations are performed without parallelization.

We first specify the tunning parameters adopted in all simulations in this section. Burnin period b_0 is set be the first 20% of the total length of the chain. We adopt the updating scheme 3.11 with $\zeta = 2/3$ and $\mathbf{C}$ with the 75% quantile of $|\rho_{ij}|$s as the threshold. For simplicity, $g(v_i)$ for all $i \in [p]$ is specified as 1 which means that all remove neighborhoods are included in the forward move set when a remove move is proposed.

[1] All the simulations are run in `R` on a computer with x86×64 Intel(R) Core(TM) i7-3770k.

4.1 Effectiveness of pMTM

In this section, we compare the proposed algorithms with two traditional Gibbs samplers, random-scan Gibbs and systematic-scan Gibbs, based on their efficiencies on exploration of the model space. George and McCulloch (1993) describe a systematic-scan Gibbs sampler by sequentially updating components of $\boldsymbol{\gamma}$ according to $\pi(\gamma_i \mid \gamma_{-i}, \mathbf{Y})$ for $i \in [p]$ in one iteration. A random-scan Gibbs sampler will randomly select an index i first and then update the corresponding γ_i

The algorithms are compared using a simulated dataset based on the number of marginal likelihood evaluations needed to find the true model. Simulated data is based on the dataset used in West et al. (2001) which contain 49 patients and each of which has gene expression data including 3883 genes. In terms of the rank of contributions of different genes to tumor provided in the supporting information 3 in West et al. (2001), we extract `TFF1` (rank 1), `ESR1` (rank 2), `CYP2B6` (rank 3) and `IGFBP2` (rank 5) to form the true predictors. The reason why we didn't choose the 4^{th} gene `TFF3` is that it has a high correlation with `TFF1`.

The simulated dataset is constructed as follows: we first normalized these four genes and further combined with standard multivariate normals to form the design matrix. Then $\boldsymbol{\beta}$ is specified by $\boldsymbol{\beta}_{\boldsymbol{\gamma}} = (1.3, 0.3, -1.2, -0.5)$ for $\boldsymbol{\gamma} = \{1, 2, 3, 4\}$ with a sequence of increasing values of $p = 50, 100, 150, ..., 500$. ε is standard normal with mean 0 and variance 0.5. The values of regression coefficients and variance of noise are exactly same with the example in Section 4.4 of Hans et al. (2007). Hyperparameters are specified as $u = 4$, $v = p - 4$, $M = p/10$. We report the median of results based on 100 synthetic dataset for each value of p in Figure.4.1. Circles and crosses are employed to represent the value larger than $\log(3 \times 10^5)$.

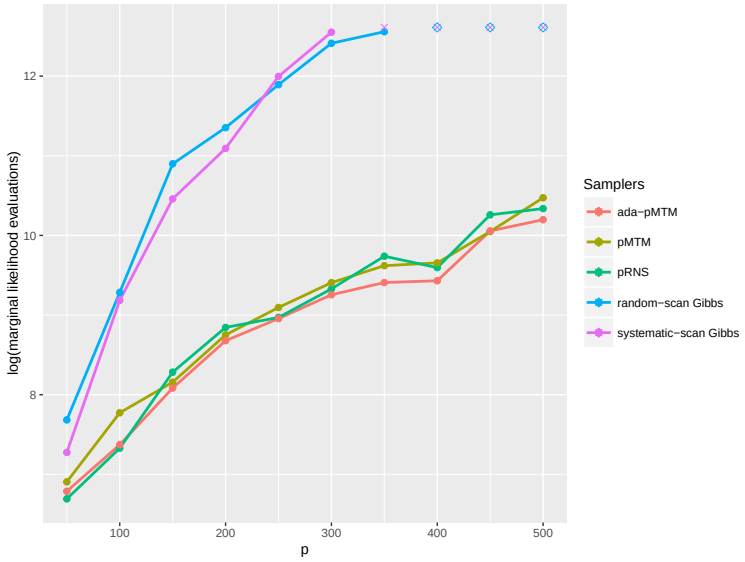

FIGURE 4.1: Logarithm of the median number of marginal likelihood evaluations needed to find the true model

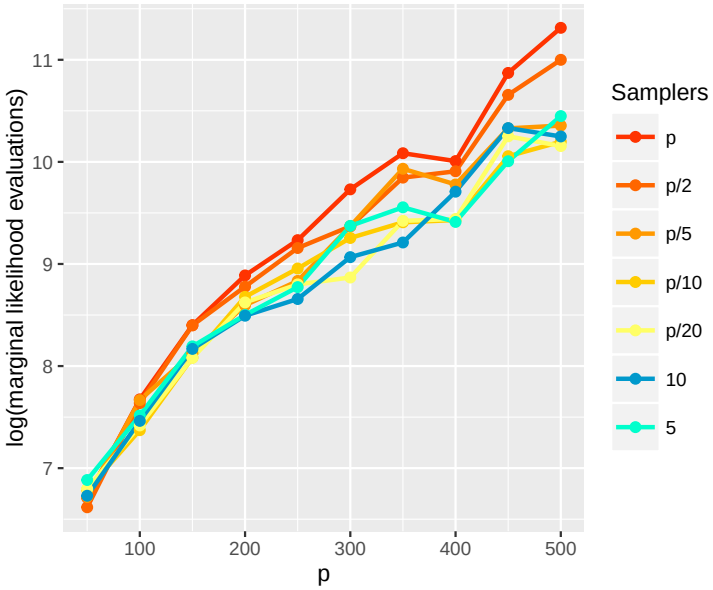

FIGURE 4.2: Logarithm of the median number of marginal likelihood evaluations needed to find the true model for *ada-pMTM* with different expected number of trails (M).

According to the graph, it is clear that the paired-move strategy can effectively reduce the computational cost comparing to two Gibbs samplers. When $p = 50, 100,$

ada-pMTM is slightly worse than *pRNS* but still competitive. As p becomes larger, *ada-pMTM* dominates all other algorithms which verifies the claim that multiple trials can help the sampler move out of local modes.

To explore how the choice of M influence the efficiency of *ada-pMTM*, we further implemented *ada-pMTM* with different choices of M. Specifically, two groups of M are specified as:

- a function of p: $M = M(p) = p, p/2, p/5, p/10, p/20$, and
- a fixed value: $M = 5, 10$

As displayed in Figure.4.2, a vague pattern appears. It should be clear that the numbers of marginal likelihood evaluation needed for $M = p$ or $p/2$ are larger suggesting the less efficiency for large M while the other 5 choices do not show significant differences. We will use $M = p/10$ throughout the rest of the simulation studies.

4.2 Simulated examples

Intensive simulation studies are presented for assessing the performances of proposed algorithms. In particular, we compare proposed algorithms to random- and systematic- scan Gibbs [George and McCulloch (1993)], *EMVS* [Ročková and George (2014)], *lasso* [Tibshirani (1996)], *adaptive lasso* [Zou (2006)] and *SCAD* [Fan and Li (2001)]. Different structures and degrees (low, moderate and high) of correlation of the design matrix are considered. Specifically, we specify $(n, p) = (100, 1000)$ for all experiments. The signal-to-noise ratio $\|\boldsymbol{\beta}\|_2/\sigma$ is adjusted to guarantee that fitting results for different methods are moderate and comparable. Four structures are described as follows:

1. **Independent design:** This example is originally analyzed by Fan and Lv

(2008) with $t = 4, 5$. We further make this example difficult by setting $t = 1$.

$$\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_p \overset{\text{iid}}{\sim} \mathsf{N}(0, \mathbf{I}_n)$$

$$\beta_i = (-1)^{U_i}(t \log n/\sqrt{n} + |N(0,1)|) \text{ where } U_i \overset{\text{iid}}{\sim} \mathsf{Unif}(0,1) \text{ for } i = 1, 2, ..., 8$$

$$\varepsilon \sim N(0, 1.5^2 \mathbf{I}_n)$$

2. **Compound symmetry:** This example is revised based on the Example 1 in Fan and Lv (2008) where $\|\boldsymbol{\beta}\|_2$ is much smaller here. Every pair of predictors has the same theoretical correlation ρ. We adopt $\rho = 0.3, 0.6$ and 0.9 to allow for different degrees of correlation.

$$\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_p \overset{\text{iid}}{\sim} \mathsf{N}(0, \boldsymbol{\Sigma}) \quad \text{with } \Sigma_{ij} = \rho \text{ for } i \neq j \text{ and } 1 \text{ for } i = j$$

$$\boldsymbol{\beta} = (2.0, 2.5, -2.0, 2.5, -2.5, 0, ..., 0)$$

$$\varepsilon \sim N(0, 1.5^2 \mathbf{I}_n)$$

3. **Autoregression:** This example is modified from the Example 2 in Tibshirani (1996). This type of correlation structure widely exists in time series. Again, we set $\rho = 0.3, 0.6, 0.9$.

$$\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_p \overset{\text{iid}}{\sim} \mathsf{N}(0, \boldsymbol{\Sigma}) \quad \text{with } \Sigma_{ij} = \rho^{|i-j|}$$

$$\boldsymbol{\gamma} = \{1, 2, 3, 4, 5, 20, 35, 60, 90, 150, 151, 300\}$$

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}} = (2, -3, 2, 2, -3, 3, -2, 3, -2, 3, 2, -2)$$

$$\varepsilon \sim N(0, 2^2 \mathbf{I}_n)$$

4. **Group structure:** This example is revised from the simulated experiment 2 in Bottolo et al. (2010) and first analyzed in Nott and Kohn (2005). A group structured correlation exhibits: collinearity exists between $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ for $i = 1, 3, 5$ and linear relationship is presented in group $(\mathbf{X}_7, \mathbf{X}_8, \mathbf{X}_9, \mathbf{X}_{10})$

and $(\mathbf{X}_{11}, \mathbf{X}_{12}, \mathbf{X}_{13}, \mathbf{X}_{14}, \mathbf{X}_{15})$. Whether the algorithm can select the correct predictors and do not select variables from the second block are of interest. The model is simulated as follows:

$\mathbf{Z}, \mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_{15} \overset{\text{iid}}{\sim} \mathsf{N}(0, \mathbf{I}_n)$

$$\mathbf{X}_i = \rho_1 \mathbf{Z} + 2\mathbf{Z}_i \quad \text{for } i = 1, 3, 5, 8, 9, 10, 12, 13, 14, 15$$

$$\mathbf{X}_i = \rho_2 \mathbf{X}_{i-1} + \rho_3 \mathbf{Z}_i \quad \text{for } i = 2, 4, 6$$

$$\mathbf{X}_7 = \rho_4(\mathbf{X}_8 + \mathbf{X}_9 - \mathbf{X}_{10}) + \rho_5 \mathbf{Z}_7$$

$$\mathbf{X}_{11} = \rho_5(\mathbf{X}_{14} + \mathbf{X}_{15} - \mathbf{X}_{12} - \mathbf{X}_{13}) + \rho_5 \mathbf{Z}_{11}$$

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}} = (1.5, 1.5, 1.5, 1.5, -1.5, 1.5, 1.5, 1.5) \text{ with } \boldsymbol{\gamma} = \{1, 3, 5, 7, 8, 11, 12, 13\}$$

where $\rho_i \quad i = 1, 2, ..., 5$ are adjusted to import small, moderate and high correlation into each group.

`R` packages `glmnet`, `parcor`, `ncvreg` and `EMVS` are used for *lasso*, *adaptive lasso*, *SCAD* and *EMVS* respectively. The tunning parameter λ's are specified by minimizing cross-validation errors. All Bayesian methods are implemented with Beta-Binomial prior using same hyperparameters: $u = 10$, $v = p - 10$. *ada-pMTM* is run for 2,000 iterations. We consider the highest probability model (HPM), median probability model (MPM) [Barbieri and Berger (2004)] (i.e., the model containing predictors with inclusion probability larger than 0.5) and Bayesian model averaging (BMA) for proposed algorithms. Recommended default settings for *EMVS* are adopted except for a more elaborate sequences of v_0. The comparisons are based on five metrics: **Model size:** number of predictors selected; **Runtime:** running time for different methods (fixed running time for all MCMC samplers); **FN:** number of false negatives; **FP:** number of false positives; **FDR:** false discovery rate and ℓ_2 **distance:** $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$. For each scenario, 100 synthetic datasets are simulated and the mean of above metrics are reported for assessment.

Several findings based on the simulation results can be summarized as follows.

First, the sizes of models selected by *lasso*, *adaptive lasso*, and *SCAD* are typically greater than the true model sizes leading to small false negatives and large false positives. The sizes of MPMs and HPMs given by Bayesian methods are smaller and closer to the true model sizes. Second, in terms of the ℓ_2 distances between the estimate and true value of $\boldsymbol{\beta}$, *ada-pMTM* and *pMTM* are superior than other methods in most cases. However, in autoregressive settings, neither of the two algorithms exhibits advantages.

Table 4.1: Independent design with $(n, p_0) = (100, 8)$

		p=1000					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	6.92	1.60	0.52	0.06545	0.96247	9.98
	HPM	7.02	1.65	0.67	0.07717	0.97281	
	BMA	-	-	-	-	**0.92419**	
pMTM	MPM	6.80	1.64	0.44	0.05683	0.96258	10.03
	HPM	6.78	1.78	0.56	0.06033	0.99559	
	BMA	-	-	-	-	**0.94179**	
pRNS (equal evaluations)	MPM	7.30	1.53	0.83	0.09555	1.00338	37.15
	HPM	7.96	1.53	1.49	0.15161	1.07369	
	BMA	-	-	-	-	0.98907	
pRNS (equal time)	MPM	7.61	1.56	1.17	0.12756	1.04951	9.09
	HPM	7.77	1.61	1.38	0.14125	1.08198	
	BMA	-	-	-	-	1.02115	
pRNS (equal iterations)	MPM	2.16	6.07	0.23	0.06045	2.86641	0.24
	HPM	2.22	6.12	0.34	0.10738	2.91814	
	BMA	-	-	-	-	2.72615	
EMVS		4.47	3.56	0.03	0.00421	1.52944	13.75
Lasso		21.23	1.34	14.57	0.59485	1.69590	2.07
adaptive lasso		12.13	1.34	5.47	0.36516	1.12010	3.93
SCAD		33.24	0.36	25.60	0.75887	0.99172	6.56

Table 4.2: Compound symmetry with $(n, p, p_0) = (100, 1000, 5)$

		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
		ρ=0.3					
ada-pMTM	MPM	5.15	0.00	0.15	0.02357	**0.44765**	
	HPM	5.13	0.00	0.13	0.02119	**0.44244**	10.74
	BMA	-	-	-	-	0.48411	
pMTM	MPM	4.99	0.24	0.23	0.04086	0.92833	
	HPM	5.18	0.06	0.24	0.03808	0.58596	10.35
	BMA	-	-	-	-	0.88684	
pRNS (equal evaluations)	MPM	5.33	0.00	0.33	0.05167	0.49293	
	HPM	5.45	0.00	0.45	0.06899	0.52963	37.33
	BMA	-	-	-	-	0.52373	
pRNS (equal time)	MPM	5.57	0.00	0.57	0.07907	0.55974	
	HPM	5.65	0.00	0.65	0.08597	0.57467	10.87
	BMA	-	-	-	-	0.56371	
pRNS (equal iterations)	MPM	2.41	2.82	0.23	0.09117	3.88156	
	HPM	2.82	2.98	0.80	0.23667	4.03775	0.21
	BMA	-	-	-	-	3.59336	
EMVS		5.03	0.00	0.03	0.00500	0.83232	13.47
Lasso		15.28	0.00	10.28	0.59420	1.52074	2.04
adaptive lasso		6.12	0.00	1.12	0.13605	0.52061	3.95
SCAD		8.64	0.00	3.64	0.24591	0.45551	4.25
		ρ=0.6					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	5.18	0.01	0.19	0.02946	0.57993	
	HPM	5.20	0.01	0.21	0.03065	**0.56093**	10.50
	BMA	-	-	-	-	0.63864	
pMTM	MPM	5.16	0.11	0.27	0.04093	0.79130	
	HPM	5.20	0.04	0.24	0.03594	0.63270	10.43
	BMA	-	-	-	-	0.85272	
pRNS (equal evaluations)	MPM	5.66	0.00	0.66	0.08798	0.72689	
	HPM	5.71	0.00	0.71	0.09543	0.74347	36.36
	BMA	-	-	-	-	0.74913	
pRNS (equal time)	MPM	5.61	0.00	0.61	0.08156	0.70428	
	HPM	5.73	0.00	0.73	0.09294	0.72342	9.98
	BMA	-	-	-	-	0.74506	
pRNS (equal iterations)	MPM	2.08	3.15	0.23	0.07867	4.17630	
	HPM	2.55	3.12	0.67	0.28433	4.10882	0.26
	BMA	-	-	-	-	3.79928	
EMVS		5.05	0.01	0.06	0.01	1.01934	11.64
Lasso		18.36	0.00	13.36	0.63903	2.03600	2.29
adaptive lasso		7.59	0.01	2.60	0.26217	0.80726	4.44
SCAD		6.81	0.00	1.81	0.15528	0.58029	4.53
		ρ=0.9					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	3.38	1.87	0.25	0.08229	3.02116	
	HPM	3.45	1.93	0.38	0.10679	3.02510	10.22
	BMA	-	-	-	-	**2.79824**	
pMTM	MPM	3.52	1.90	0.42	0.11136	3.15391	
	HPM	3.58	1.96	0.54	0.14769	3.15457	10.88
	BMA	-	-	-	-	**2.95858**	
pRNS (equal evaluations)	MPM	3.98	1.83	0.81	0.18968	3.26200	
	HPM	4.27	1.86	1.13	0.24183	3.34397	36.32
	BMA	-	-	-	-	3.16544	
pRNS (equal time)	MPM	3.86	1.86	0.72	0.17876	3.26003	
	HPM	4.24	1.89	1.13	0.25938	3.42405	10.57
	BMA	-	-	-	-	3.13012	
pRNS (equal iterations)	MPM	1.44	3.96	0.40	0.22317	4.90713	
	HPM	1.87	3.95	0.82	0.37617	4.95334	0.26
	BMA	-	-	-	-	4.51167	
EMVS		2.79	2.51	0.30	0.11138	3.80729	10.26
Lasso		11.54	1.43	7.97	0.60918	4.03553	3.76
adaptive lasso		9.74	0.98	5.72	0.48575	3.10494	7.10
SCAD		4.67	1.95	1.62	0.29208	3.47218	2.95

Table 4.3: Autoregressive correlation with $(n, p, p_0) = (100, 1000, 12)$

		ρ=0.3					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	11.07	1.07	0.14	0.01620	1.75973	
	HPM	11.36	1.15	0.51	0.04634	1.74790	15.19
	BMA	-	-	-	-	1.81494	
pMTM	MPM	10.05	2.16	0.21	0.02417	2.86410	
	HPM	10.67	2.32	0.99	0.09460	2.80901	16.33
	BMA	-	-	-	-	2.87099	
pRNS (equal evaluations)	MPM	12.55	0.00	0.55	0.04031	0.95725	
	HPM	13.08	0.00	1.08	0.07350	0.99831	43.94
	BMA	-	-	-	-	0.96122	
pRNS (equal time)	MPM	11.88	0.21	0.09	0.01115	0.99528	
	HPM	11.84	0.30	0.14	0.01478	1.05328	14.89
	BMA	-	-	-	-	1.03719	
pRNS (equal iterations)	MPM	1.93	10.29	0.22	0.10267	7.90704	
	HPM	2.18	10.27	0.45	0.15617	7.96132	0.27
	BMA	-	-	-	-	7.50963	
EMVS		9.57	2.78	0.35	0.06049	3.07500	27.05
Lasso		42.45	1.03	31.48	0.71510	4.60301	2.28
adaptive lasso		19.55	0.58	8.13	0.37514	2.27123	4.27
SCAD		25.29	0.00	13.29	0.46581	**0.95686**	4.68
		ρ=0.6					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	9.89	2.56	0.45	0.05300	3.40834	
	HPM	10.44	2.78	1.22	0.10091	3.60832	13.24
	BMA	-	-	-	-	3.47301	
pMTM	MPM	9.02	3.59	0.61	0.06996	4.32600	
	HPM	9.46	3.82	1.28	0.12107	4.44422	12.58
	BMA	-	-	-	-	4.14006	
pRNS (equal evaluations)	MPM	12.42	0.16	0.58	0.04133	1.21687	
	HPM	13.24	0.16	1.40	0.09381	1.31555	41.96
	BMA	-	-	-	-	1.22476	
pRNS (equal time)	MPM	12.39	0.32	0.71	0.05421	**1.38717**	
	HPM	13.24	0.32	1.56	0.10572	1.48584	13.51
	BMA	-	-	-	-	**1.37363**	
pRNS (equal iterations)	MPM	2.57	9.89	0.46	0.15567	7.88735	
	HPM	2.78	9.92	0.70	0.21045	7.95898	0.25
	BMA	-	-	-	-	7.53342	
EMVS		7.63	4.81	0.44	0.06244	5.06578	23.88
Lasso		30.88	3.36	22.24	0.65295	5.94437	2.36
adaptive lasso		17.81	3.05	8.86	0.41424	5.00711	4.42
SCAD		30.20	1.69	19.89	0.63170	3.64840	5.79
		ρ=0.9					
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	6.14	6.76	0.90	0.14867	6.69856	
	HPM	7.12	6.74	1.86	0.22984	6.77342	12.65
	BMA	-	-	-	-	6.33825	
pMTM	MPM	5.98	7.20	1.18	0.19519	7.04164	
	HPM	6.63	7.10	1.73	0.25785	7.10383	13.21
	BMA	-	-	-	-	6.57290	
pRNS (equal evaluations)	MPM	7.87	5.36	1.23	0.14951	6.06363	
	HPM	8.31	5.37	1.68	0.18888	6.07039	42.05
	BMA	-	-	-	-	5.98609	
pRNS (equal time)	MPM	7.71	5.70	1.41	0.17207	6.22964	
	HPM	8.26	5.71	1.97	0.21550	6.26212	13.87
	BMA	-	-	-	-	**6.15213**	
pRNS (equal iterations)	MPM	4.38	9.89	2.27	0.51740	8.62847	
	HPM	5.47	9.92	3.39	0.63110	8.84104	0.27
	BMA	-	-	-	-	8.14917	
EMVS		7.37	6.58	1.95	0.25416	6.73390	18.91
Lasso		26.29	5.08	19.37	0.69542	6.51992	2.27
adaptive lasso		13.23	5.58	6.81	0.45352	6.28458	4.93
SCAD		21.48	6.29	15.77	0.70338	6.94900	4.61

Table 4.4: Group structure with $(n, p, p_0) = (100, 1000, 8)$

		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
small correlation							
ada-pMTM	MPM	8.14	0.00	0.14	0.01511	**0.36531**	
	HPM	8.09	0.00	0.09	0.01000	**0.34287**	14.57
	BMA	-	-	-	-	0.42152	
pMTM	MPM	8.08	0.12	0.20	0.02039	0.53798	
	HPM	8.20	0.06	0.26	0.02622	0.48917	15.10
	BMA	-	-	-	-	0.68383	
pRNS (equal evaluations)	MPM	8.35	0.00	0.35	0.03584	0.44451	
	HPM	8.58	0.00	0.58	0.05574	0.49687	40.72
	BMA	-	-	-	-	0.47022	
pRNS (equal time)	MPM	8.36	0.00	0.36	0.03717	0.46472	
	HPM	8.48	0.00	0.48	0.04929	0.48844	14.70
	BMA	-	-	-	-	0.48073	
pRNS (equal iterations)	MPM	1.90	6.32	0.22	0.10167	3.91957	
	HPM	1.98	6.32	0.30	0.10804	3.97245	0.23
	BMA	-	-	-	-	3.74767	
EMVS		8.09	0.05	0.14	0.01511	5.37749	12.57
Lasso		37.51	0.00	29.51	0.76589	1.61847	1.92
adaptive lasso		10.65	0.00	2.65	0.20397	0.50107	4.09
SCAD		14.54	0.00	6.54	0.27504	0.40595	3.50
moderate correlation							
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	8.19	0.09	0.28	0.02999	**0.47587**	
	HPM	8.19	0.09	0.28	0.02977	**0.47389**	14.39
	BMA	-	-	-	-	0.49938	
pMTM	MPM	7.76	0.85	0.61	0.08053	1.18489	
	HPM	7.71	0.85	0.56	0.07571	1.06106	14.88
	BMA	-	-	-	-	1.27868	
pRNS (equal evaluations)	MPM	8.47	0.00	0.47	0.04796	0.48850	
	HPM	8.64	0.00	0.64	0.06330	0.53733	38.86
	BMA	-	-	-	-	0.52055	
pRNS (equal time)	MPM	8.48	0.04	0.52	0.05139	0.52188	
	HPM	8.65	0.04	0.69	0.06679	0.55481	15.22
	BMA	-	-	-	-	0.54350	
pRNS (equal iterations)	MPM	2.29	6.25	0.54	0.20833	3.94940	
	HPM	2.44	6.19	0.63	0.22275	3.94975	0.22
	BMA	-	-	-	-	3.83011	
EMVS		7.60	1.22	0.82	0.11750	5.68208	13.15
Lasso		38.81	0.83	31.64	0.78922	2.88405	2.09
adaptive lasso		13.34	0.55	5.89	0.36971	1.46768	4.04
SCAD		18.02	0.20	10.22	0.38697	0.77203	4.60
high correlation							
		Model size	*FN*	*FP*	*FDR*	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$	*Runtime*
ada-pMTM	MPM	7.46	2.79	2.25	0.30544	3.01815	
	HPM	7.42	2.78	2.20	0.29978	3.00770	12.91
	BMA	-	-	-	-	3.02119	
pMTM	MPM	7.55	3.07	2.62	0.34795	3.27811	
	HPM	7.50	3.05	2.55	0.34374	3.25323	13.03
	BMA	-	-	-	-	3.23014	
pRNS (equal evaluations)	MPM	7.78	2.07	1.85	0.23894	2.50479	
	HPM	7.89	2.06	1.95	0.24684	2.51739	38.62
	BMA	-	-	-	-	2.50086	
pRNS (equal time)	MPM	7.46	2.52	1.98	0.26960	2.82655	
	HPM	7.47	2.50	1.97	0.26696	**2.80090**	13.88
	BMA	-	-	-	-	**2.79805**	
pRNS (equal iterations)	MPM	3.79	5.96	1.75	0.47052	4.28314	
	HPM	3.81	6.11	1.92	0.48923	4.37975	0.27
	BMA	-	-	-	-	4.09175	
EMVS		7.99	2.85	2.84	0.35481	6.93585	10.36
Lasso		25.17	3.00	20.17	0.76385	3.67644	2.00
adaptive lasso		12.55	3.26	7.81	0.58423	3.84908	3.90
SCAD		15.76	3.01	10.77	0.58730	3.36388	3.72

4.3 Real data examples

In this section, the prediction performance of the samplers are illustrated by two real data examples. For each example, we run *ada-pMTM* for totally 1,000 iterations. Other settings are kept same as those in Section 4.2. **Running time**, **predictive MSE**, and **model size** for each method are reported.

4.3.1 Near-infrared (NIR) Spectroscopy Analysis of Biscuit Doughs

NIR spectroscopic technique is routinely employed for food quality control. An NIR reflectance spectrum of a food sample is recorded to predict the chemical composition of the food. The dataset [Brown et al. (2001), Osborne et al. (1984)] contains 72 samples of biscuit dough piece with various compositions of dry flour, fat, sucrose, and water. The percentages of these four ingredients are regarded as responses. For each sample, 700 measurements from NIR spectroscopy are recorded and regarded as predictors. The correlation of first 100 predictors are plotted in the left graph in Figure 4.3. Clearly, all predictors are highly correlated. In terms of the experimental design, there are 40 samples in the training set while the other 32 samples are in the test set. Two samples are reported as outliers and are removed in our analysis. For each response, a simple linear regression model is fitted to the training set. Variable selection is performed using different methods for 10 times to reduce stochasticity. Results are reported in Table 4.3.1, 4.3.1, 4.3.1, and 4.3.1. HPMs and MPMs given by Bayesian methods have smaller sizes. The predictive MSEs based on BMA using *ada-pMTM* are often smaller compared to other methods.

4.3.2 Riboflavin Production with Bacillus Subtilis

The dataset [Bühlmann et al. (2014)] consists of 71 samples of riboflavin (vitamin B_2) production rate with Bacillus subtilis and associated 4088 gene expression levels. Predicting the rate in terms of these gene expressions is of interest. In terms of the

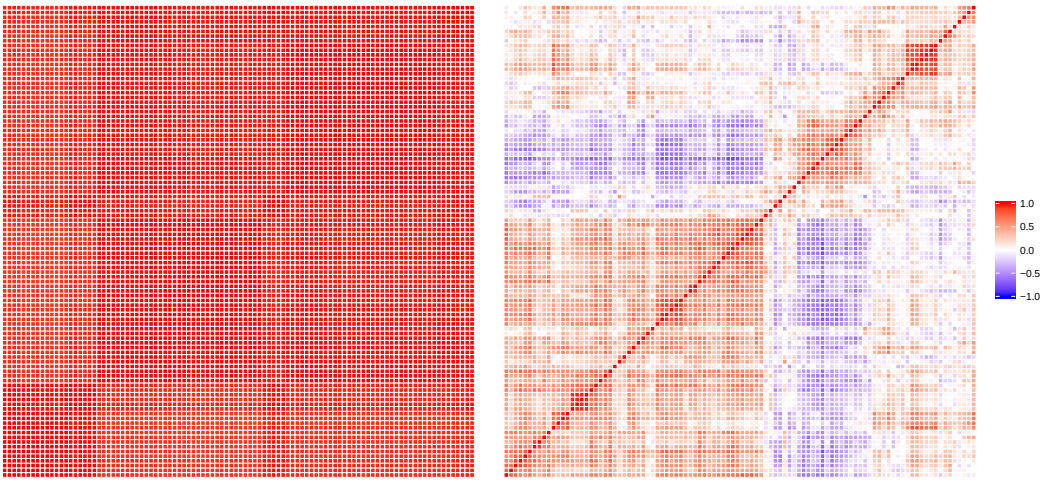

FIGURE 4.3: Heat maps of correlation matrices of the first 100 predictors. Left: NIR spectroscopy example with all 700 predictors highly correlated. Right: Riboflavin gene expression example with a grouped correlation structure.

Table 4.5: The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with fat as response.

		Model size	*Predictive MSE*	*Runtime*
ada-pMTM	MPM	0.3	3.4638	4.23
	HPM	3.7	0.1802	
	BMA	-	0.0883	
pMTM	MPM	0.2	3.6440	4.11
	HPM	4.0	0.1568	
	BMA	-	0.0886	
pRNS	MPM	0.0	3.9310	4.35
	HPM	2.2	0.1724	
	BMA	-	0.0799	
EMVS		0.0	0.0806	3.98
Lasso		12.4	**0.0518**	2.09
adaptive lasso		3.0	0.0717	2.73
SCAD		3.3	0.1525	2.61
Null		-	3.9310	-

right graph of 4.3, the design matrix has a grouped correlation structure. The dataset is first randomly partitioned into a training set and a test set with 80% and 20% of samples. Variable selection is then performed using different methods. These procedures are repeated for 10 times to reduce stochasticity. The prediction results are summarized in 4.3.2. The predictive MSE given by BMA using *pMTM* is the smallest among all methods.

Table 4.6: The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with sucrose as response.

		Model size	*Predictive MSE*	*Runtime*
ada-pMTM	MPM	0.0	14.8664	
	HPM	6.2	1.4327	4.12
	BMA	-	**0.8323**	
pMTM	MPM	0.0	14.8664	
	HPM	4.9	1.3305	4.15
	BMA	-	0.8744	
pRNS	MPM	0.0	14.8664	
	HPM	4.5	1.4981	4.22
	BMA	-	0.9004	
EMVS		0.0	**0.8568**	4.01
Lasso		22.0	1.8263	2.07
adaptive lasso		4.1	1.3916	2.64
SCAD		4.0	1.6785	2.70
Null		-	14.8664	-

Table 4.7: The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with dry flour as response.

		Model size	*Predictive MSE*	*Runtime*
ada-pMTM	MPM	0.0	6.4136	
	HPM	2.8	0.6175	4.19
	BMA	-	**0.4355**	
pMTM	MPM	0.0	6.4136	
	HPM	3.1	0.6075	4.13
	BMA	-	0.4563	
pRNS	MPM	0.0	6.4136	
	HPM	2.2	1.4335	4.31
	BMA	-	0.4472	
EMVS		0.0	0.4554	4.09
Lasso		11.9	1.0025	1.96
adaptive lasso		4.0	**0.4390**	2.35
SCAD		3.3	0.6801	2.57
Null		-	6.4136	-

4.4 Computational efficiency

Scalability is another attractiveness of the proposed algorithms. In this section, we compare the computational efficiency of the proposed algorithms run with different number of cores on a simulated dataset with independent design in Section 4.2 for $n = 10^3$ and $p = 2 \times 10^4$. `apply` function is used for *pMTM* and *ada-pMTM* with single core when evaluating marginal likelihoods. *pMTM, ada-pMTM-4, 8* represent *pMTM* or *ada-pMTM* run on 4 or 8 clusters. For parallelization, datasets are first distributed to multiple clusters and then `parLapply` in `parallel` package is used.

Table 4.8: The fitting and prediction results of different methods (averaged over 10 replicates) for NIR spectroscopy dataset with water as response.

		Model size	*Predictive MSE*	*Runtime*
ada-pMTM	MPM	0.0	1.6231	4.09
	HPM	4.1	0.1614	
	BMA	-	**0.1021**	
pMTM	MPM	0.0	1.6231	4.12
	HPM	4.8	0.2032	
	BMA	-	0.1073	
pRNS	MPM	0.0	1.6231	4.15
	HPM	3.7	0.1568	
	BMA	-	0.1151	
EMVS		0.0	**0.1006**	3.98
Lasso		8.4	0.2487	1.87
adaptive lasso		5.0	0.2438	2.23
SCAD		3.0	0.2704	2.47
Null		-	1.6231	-

Table 4.9: The fitting and prediction results of different methods (averaged over 10 replicates) for riboflavin dataset.

		Model size	*Predictive MSE*	*Runtime*
ada-pMTM	MPM	2.2	0.5105	37.41
	HPM	6.2	0.4035	
	BMA	-	0.2793	
pMTM	MPM	3.0	0.4068	36.10
	HPM	7.6	0.3620	
	BMA	-	**0.2075**	
pRNS	MPM	2.8	0.4113	37.68
	HPM	5.2	0.3480	
	BMA	-	0.2551	
EMVS		0.0	**0.2170**	14.54
Lasso		22.8	0.2429	4.53
adaptive lasso		11.1	0.2406	6.45
SCAD		16.7	0.3013	7.22

All algorithms are implemented on the same 10 synthetic datasets at each value of M. A graph of the mean numbers of evaluations of marginal likelihood within 10 seconds against M/p is provided in Figure.4.4.

The line for *pRNS* is a constant since the algorithm does not involve M. *ada-pMTM* needs to update scores for predictors and hence it evaluates less marginal likelihoods than *pMTM*. 2 communications are required at each iteration and hence parallelization with 4 or 8 clusters when $M = p/5$ is not beneficial. When M becomes larger, computing time overwhelms communication time resulting in the dominance of algorithms implemented with 8 clusters.

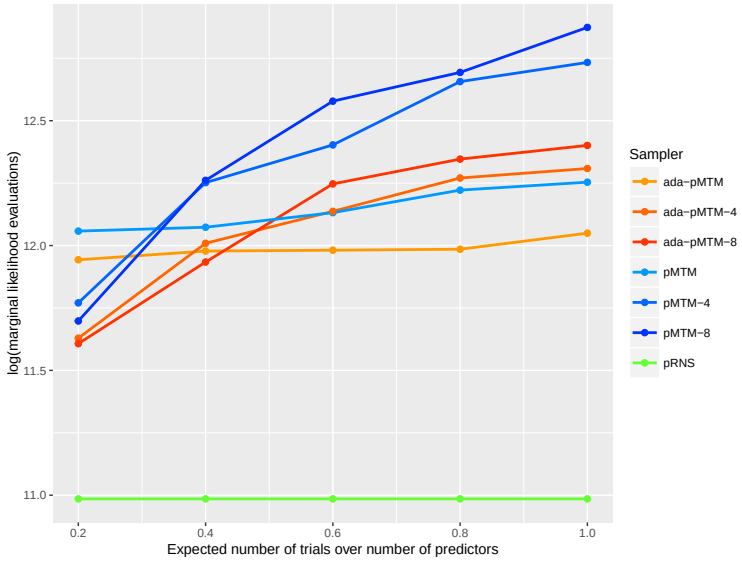

FIGURE 4.4: Logarithm of the mean number of marginal likelihood evaluations within 10 seconds

5

Discussion

A paired-move multiple-try Metropolis MCMC sampler for Bayesian variable selection is introduced and its behaviors are studied. Extensive simulation studies demonstrate the effectiveness of *pMTM* especially for "large p small n" scenario. Efficient model space exploration with less computational cost is achieved by incorporating the paired-move and multiple-try strategies. Comparing to *SSS*, a more flexible computational budget can be determined manually based on data and purpose instead of considering all neighborhoods. In this work, the expected computational budget M is specified as $p/10$. However, the optimal choice of M is still not fully explored. Intuitively, the optimal M may depend on dimensions and correlation structure of the design matrix.

Reproducibility is a key issue in scientific research [Peng (2011); Collins and Tabak (2014); Collaboration et al. (2015)]. Research based on statistical computations is expected to be able to be replicated. In the context of inference using MCMC techniques, both of the following two elements are required for reproducibility:

1. convergence of the Markov chain: To ensure the samples are indeed drawn from the target distribution, we require the chain nearly converging to the

equilibrium.

2. enough posterior samples: Bayesian inference is mostly based on posterior samples. Therefore, enough posterior samples drawn from a converged chain are required to make accurate inference.

Considering running the proposed algorithms under fixed running time, chains produced by *pMTM* and *ada-pMTM* can rapidly converge to equilibrium with a small number of posterior samples while *pRNS* can generate a large number of samples but may be stuck in some local modes. Therefore, implementing each of these algorithms in a short period of time may fail to simultaneously satisfy the two requirements. A hybrid algorithm, combining *pRNS* and *ada-pMTM*, that take advantages of both is worthwhile developing.

To facilitate the application of our method to even huge datasets, one may further accelerate *pMTM* by subsampling [Balan et al. (2014); Quiroz et al. (2015)] which is randomly selecting a mini-batch of samples at each iteration for computing marginal likelihoods. Another possible approach is to partition the design matrix first either using sample space partitioning [Wang et al. (2014)] or feature space partitioning [Wang et al. (2016)] and then apply *pMTM* on each subset of data.

Appendix A

Proof of Theorem 3.2.2

Theorem. *The paired-move multiple-try stochastic search MCMC (pMTM) algorithm with acceptance probability 3.8 satisfies the detailed balance condition leaving the desired target distribution* $\pi(\boldsymbol{\gamma} \mid \mathbf{Y})$ *invariant.*

Proof. Let $A(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ be the actual transition probability for moving from $\boldsymbol{\gamma}_1$ to $\boldsymbol{\gamma}_2$. If $m \in \{A, R\}$: Let $\omega_i = \omega(\gamma_i, v_i; m)$ and $\tilde{\omega}_i = \omega(\gamma_i', v_i; m')$ denote the probabilities of the forward and backward move for predictor $i \in [p]$. Then for $\boldsymbol{\gamma}' = \text{tog}(i^*, \boldsymbol{\gamma})$,

$$
\begin{aligned}
&\pi(\boldsymbol{\gamma} \mid \mathbf{Y}) A(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \\
&= \pi(\boldsymbol{\gamma} \mid \mathbf{Y}) w_m(|\boldsymbol{\gamma}|) \sum_{\substack{\eta, \eta' \in \{0,1\}^p \\ \eta_{i^*} = \eta'_{i^*} = 1}} \left[\omega_{i^*} \left\{ \prod_{j \neq i^*} \omega_j^{\eta_j} (1 - \omega_j)^{1 - \eta_j} \tilde{\omega}_j^{\eta'_j} (1 - \tilde{\omega}_j)^{1 - \eta'_j} \right\} \right. \\
&\left. \times T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \min \left\{ 1, \frac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y}) [w_{m'}(|\boldsymbol{\gamma}'|) \tilde{\omega}_{i^*} T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y}) [w_m(|\boldsymbol{\gamma}|) \omega_{i^*} T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]} \right\} \right] \\
&= \sum_{\substack{\eta, \eta' \in \{0,1\}^p \\ \eta_{i^*} = \eta'_{i^*} = 1}} \left[\left\{ \prod_{j \neq i^*} \omega_j^{\eta_j} (1 - \omega_j)^{1 - \eta_j} \tilde{\omega}_j^{\eta'_j} (1 - \tilde{\omega}_j)^{1 - \eta'_j} \right\} \right. \\
&\left. \times \min \left\{ \pi(\boldsymbol{\gamma} \mid \mathbf{Y}) [w_m(|\boldsymbol{\gamma}|) \omega_{i^*} T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')], \pi(\boldsymbol{\gamma}' \mid \mathbf{Y}) [w_{m'}(|\boldsymbol{\gamma}'|) \tilde{\omega}_{i^*} T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})] \right\} \right] \quad \text{(A.1)}
\end{aligned}
$$

If $m = S$: Note that a swap move can be viewed as a composition of a remove and an add move. Denote the probability of a forward move for a pair of predictors $(a, r) \in \boldsymbol{\gamma}^c \times \boldsymbol{\gamma}$ as $\omega_a \omega_r$ where $\omega_a = \omega(\gamma_a, v_a; A)$ and $\omega_r = \omega(\gamma_r, v_r; R)$. Likewise for backward move probabilities, let $\tilde{\omega}_{a'} = \omega(\gamma'_{a'}, v_{a'}; A)$ and $\tilde{\omega}_{r'} = \omega(\gamma'_{r'}, v_{r'}; R)$ for $(a', r') \in (\boldsymbol{\gamma}')^c \times \boldsymbol{\gamma}'$. Note that $(*) : w_S(|\boldsymbol{\gamma}|) = w_S(|\boldsymbol{\gamma}'|)$ since $|\boldsymbol{\gamma}| = |\boldsymbol{\gamma}'|$. Then for $\boldsymbol{\gamma}' = \text{tog}(a^*, \text{tog}(r^*, \boldsymbol{\gamma}))$, we have

$$
\begin{aligned}
&\pi(\boldsymbol{\gamma} \mid \mathbf{Y}) A(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \\
&= \pi(\boldsymbol{\gamma} \mid \mathbf{Y}) w_S(|\boldsymbol{\gamma}|) \sum_{\substack{(a,r) \in \boldsymbol{\gamma}^c \times \boldsymbol{\gamma} \\ (a',r') \in (\boldsymbol{\gamma}')^c \times \boldsymbol{\gamma}' \\ \eta_{a^*} = \eta_{r^*} = 1 \\ \eta'_{a^*} = \eta'_{r^*} = 1}} \Bigg[\omega_{a^*} \omega_{r^*} \Bigg\{ \prod_{(a,r) \neq (a^*, r^*)} \omega_a^{\eta_a} (1 - \omega_a)^{1 - \eta_a} \omega_r^{\eta_r} (1 - \omega_r)^{1 - \eta_r} \\
&\times \prod_{(a',r') \neq (a^*, r^*)} \tilde{\omega}_{a'}^{\eta'_{a'}} (1 - \tilde{\omega}_{a'})^{1 - \eta'_{a'}} \tilde{\omega}_{r'}^{\eta'_{r'}} (1 - \tilde{\omega}_{r'})^{1 - \eta'_{r'}} \Bigg\} \\
&\times T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \min \left\{ 1, \frac{\pi(\boldsymbol{\gamma}' \mid \mathbf{Y}) [\tilde{\omega}_{r^*} \tilde{\omega}_{a^*} T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})]}{\pi(\boldsymbol{\gamma} \mid \mathbf{Y}) [\omega_{a^*} \omega_{r^*} T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')]} \right\} \Bigg] \\
&\overset{(*)}{=\!=} \sum_{\substack{(a,r) \in \boldsymbol{\gamma}^c \times \boldsymbol{\gamma} \\ (a',r') \in (\boldsymbol{\gamma}')^c \times \boldsymbol{\gamma}' \\ \eta_{a^*} = \eta_{r^*} = 1 \\ \eta'_{a^*} = \eta'_{r^*} = 1}} \Bigg[\Bigg\{ \prod_{(a,r) \neq (a^*, r^*)} \omega_a^{\eta_a} (1 - \omega_a)^{1 - \eta_a} \omega_r^{\eta_r} (1 - \omega_r)^{1 - \eta_r} \\
&\times \prod_{(a',r') \neq (a^*, r^*)} \tilde{\omega}_{a'}^{\eta'_{a'}} (1 - \tilde{\omega}_{a'})^{1 - \eta'_{a'}} \tilde{\omega}_{r'}^{\eta'_{r'}} (1 - \tilde{\omega}_{r'})^{1 - \eta'_{r'}} \Bigg\} \\
&\times \min \left\{ w_S(|\boldsymbol{\gamma}|) \pi(\boldsymbol{\gamma} \mid \mathbf{Y}) [\omega_{a^*} \omega_{r^*} T_F(\boldsymbol{\gamma}, \boldsymbol{\gamma}')], w_S(|\boldsymbol{\gamma}'|) \pi(\boldsymbol{\gamma}' \mid \mathbf{Y}) [\tilde{\omega}_{r^*} \tilde{\omega}_{a^*} T'_B(\boldsymbol{\gamma}', \boldsymbol{\gamma})] \right\} \Bigg]
\end{aligned}
\tag{A.2}
$$

Note that the expressions A.1 and A.2 are symmetric in $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$ and hence $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}) A(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \pi(\boldsymbol{\gamma}' \mid \mathbf{Y}) A(\boldsymbol{\gamma}', \boldsymbol{\gamma})$ which is the detailed balance condition. □

Bibliography

Balan, A. K., Chen, Y., and Welling, M. (2014), "Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget," in *ICML*, vol. 32 of *JMLR Workshop and Conference Proceedings*, pp. 181–189, JMLR.org.

Barbieri, M. M. and Berger, J. O. (2004), "Optimal predictive model selection," *Annals of Statistics*, pp. 870–897.

Bogdan, M., van den Berg, E., Sabatti, C., Su, W., and Candès, E. J. (2015), "SLOPEadaptive variable selection via convex optimization," *The annals of applied statistics*, 9, 1103.

Bottolo, L., Richardson, S., et al. (2010), "Evolutionary stochastic search for Bayesian model exploration," *Bayesian Analysis*, 5, 583–618.

Brown, P. J., Fearn, T., and Vannucci, M. (2001), "Bayesian wavelet regression on curves with application to a spectroscopic calibration problem," *Journal of the American Statistical Association*, 96, 398–408.

Bühlmann, P., Kalisch, M., and Meier, L. (2014), "High-dimensional statistics with a view toward applications in biology," *Annual Review of Statistics and Its Application*, 1, 255–278.

Clyde, M. A., Ghosh, J., and Littman, M. L. (2011), "Bayesian adaptive sampling for variable selection and model averaging," *Journal of Computational and Graphical Statistics*.

Collaboration, O. S. et al. (2015), "Estimating the reproducibility of psychological science," *Science*, 349, aac4716.

Collins, F. S. and Tabak, L. A. (2014), "NIH plans to enhance reproducibility," *Nature*, 505, 612.

Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., and West, M. (2004), "Sparse graphical models for exploring gene expression data," *Journal of Multivariate Analysis*, 90, 196–212.

Fan, J. and Li, R. (2001), "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, 96, 1348–1360.

Fan, J. and Lv, J. (2008), "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 849–911.

George, E. I. and McCulloch, R. E. (1993), "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88, 881–889.

Hans, C. (2009), "Bayesian lasso regression," *Biometrika*, pp. 835–845.

Hans, C., Dobra, A., and West, M. (2007), "Shotgun stochastic search for large p regression," *Journal of the American Statistical Association*, 102, 507–516.

Kass, R. E. and Wasserman, L. (1995), "A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion," *Journal of the american statistical association*, 90, 928–934.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008), "Mixtures of g priors for Bayesian variable selection," *Journal of the American Statistical Association*, 103, 410–423.

Liu, J. S., Liang, F., and Wong, W. H. (2000), "The multiple-try method and local optimization in Metropolis sampling," *Journal of the American Statistical Association*, 95, 121–134.

Madigan, D., York, J., and Allard, D. (1995), "Bayesian graphical models for discrete data," *International Statistical Review/Revue Internationale de Statistique*, pp. 215–232.

Mitchell, T. J. and Beauchamp, J. J. (1988), "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, 83, 1023–1032.

Nott, D. J. and Kohn, R. (2005), "Adaptive sampling for Bayesian variable selection," *Biometrika*, 92, 747–763.

Osborne, B. G., Fearn, T., Miller, A. R., and Douglas, S. (1984), "Application of near infrared reflectance spectroscopy to the compositional analysis of biscuits and biscuit doughs," *Journal of the Science of Food and Agriculture*, 35, 99–105.

Pandolfi, S., Bartolucci, F., and Friel, N. (2010), "A generalization of the Multiple-try Metropolis algorithm for Bayesian estimation and model selection." in *AISTATS*, vol. 9, pp. 581–588.

Park, T. and Casella, G. (2008), "The bayesian lasso," *Journal of the American Statistical Association*, 103, 681–686.

Peng, R. D. (2011), "Reproducible research in computational science," *Science*, 334, 1226–1227.

Polson, N. G., Scott, J. G., et al. (2012), "On the half-Cauchy prior for a global scale parameter," *Bayesian Analysis*, 7, 887–902.

Polson, N. G., Scott, J. G., and Windle, J. (2014), "The bayesian bridge," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 713–733.

Qamar, S. and Tokdar, S. T. (2014), "Additive Gaussian Process Regression," *arXiv preprint arXiv:1411.7009*.

Quiroz, M., Villani, M., and Kohn, R. (2015), "Scalable MCMC for large data problems using data subsampling and the difference estimator," *Riksbank Research Paper Series*.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997), "Bayesian model averaging for linear regression models," *Journal of the American Statistical Association*, 92, 179–191.

Roberts, G. O. and Rosenthal, J. S. (2007), "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms," *Journal of applied probability*, pp. 458–475.

Ročková, V. and George, E. I. (2014), "EMVS: The EM approach to Bayesian variable selection," *Journal of the American Statistical Association*, 109, 828–846.

Scott, J. G., Berger, J. O., et al. (2010), "Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem," *The Annals of Statistics*, 38, 2587–2619.

Tibshirani, R. (1996), "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 91–108.

Wang, X., Peng, P., and Dunson, D. B. (2014), "Median selection subset aggregation for parallel inference," in *Advances in Neural Information Processing Systems*, pp. 2195–2203.

Wang, X., Dunson, D., and Leng, C. (2016), "DECOrrelated feature space partitioning for distributed sparse regression," *arXiv preprint arXiv:1602.02575*.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R., and Nevins, J. R. (2001), "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences*, 98, 11462–11467.

Yang, Y., Wainwright, M. J., Jordan, M. I., et al. (2016), "On the computational complexity of high-dimensional Bayesian variable selection," *The Annals of Statistics*, 44, 2497–2532.

Yuan, M. and Lin, Y. (2006), "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68, 49–67.

Zellner, A. (1986), "On assessing prior distributions and Bayesian regression analysis with g-prior distributions," *Bayesian inference and decision techniques: Essays in Honor of Bruno De Finetti*, 6, 233–243.

Zhang, C.-H. et al. (2010), "Nearly unbiased variable selection under minimax concave penalty," *The Annals of statistics*, 38, 894–942.

Zou, H. (2006), "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, 101, 1418–1429.

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 301–320.