Scalable Bayesian sparse learning in high-dimensional model

Xiongwen Ke

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Mathematics and Statistics

Faculty of Science

The University of New South Wales

January 2023

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Ke**

First name: **Xiongwen** Other name/s: **NA**

Abbreviation for degree as given in the University calendar: **PhD**

School: **School of Mathematics and Statistics** Faculty: **Faculty of Science**

Title: Scalable Bayesian sparse learning in high-dimensional model

Abstract

Nowadays, high-dimensional models, where the number of parameters or features can even be larger than the number of observations are encountered on a fairly regular basis due to advancements in modern computation. For example, in gene expression datasets, we often encounter datasets with observations in the order of at most a few hundred and with predictors from thousands of genes. One of the goals is to identify the genes which are relevant to the expression. Another example is model compression, which aims to alleviate the costs of large model sizes. The former example is the variable or feature selection problem, while the latter is the model selection problem.

In the Bayesian framework, we often specify shrinkage priors that induce sparsity in the model. The sparsity-inducing prior will have a high concentration around zero to identify the zero coefficient and heavy tails to capture the non-zero element.

In this thesis, we first provide an overview of the most well-known sparsity-inducing priors. Then we propose to use $L_{\frac{1}{2}}$ prior with a partially collapsed Gibbs sampler to explore the high dimensional parameter space in linear regression models and variable selection is achieved through credible intervals. We also develop a coordinate-wise optimization for posterior mode search with theoretical guarantees. We then extend the PCG sampler to develop a scalable ordinal regression model with a real application in the study of student evaluation of surveys. Next, we move to modern deep learning. A constrained variational Adam(CVA) algorithm has been introduced to optimize the Bayesian neural network and its connection to stochastic gradient Hamiltonian Monte Carlo has been discussed. We then generalize our algorithm to a CVA-EM, which incorporates the spike-and-slab prior to capturing the sparsity of the neural network. Both nonlinear high dimensional variable selection and network pruning can be achieved by this algorithm. We further show that the CVA-EM algorithm can extend to the graph neural networks to produce both sparse graphs and sparse weights. Finally, we discuss the sparse VAE with $L_{\frac{1}{2}}$ prior as potential future work.

Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Xiongwen Ke
12 January, 2023

Copyright Statement

Xiongwen Ke
12 January, 2023

Authenticity Statement

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Xiongwen Ke
12 January, 2023

Abstract

Nowadays, high-dimensional models, where the number of parameters or features can even be larger than the number of observations are encountered on a fairly regular basis due to advancements in modern computation. For example, in gene expression datasets, we often encounter datasets with observations in the order of at most a few hundred and with predictors from thousands of genes. One of the goals is to identify the genes which are relevant to the expression. Another example is model compression, which aims to alleviate the costs of large model sizes. The former example is the variable or feature selection problem, while the latter is the model selection problem.

In the Bayesian framework, we often specify shrinkage priors that induce sparsity in the model. The sparsity-inducing prior will have a high concentration around zero to identify the zero coefficient and heavy tails to capture the non-zero element.

In this thesis, we first provide an overview of the most well-known sparsity-inducing priors. Then we propose to use $L_{\frac{1}{2}}$ prior with a partially collapsed Gibbs sampler to explore the high dimensional parameter space in linear regression models and variable selection is achieved through credible intervals. We also develop a coordinate-wise optimization for posterior mode search with theoretical guarantees. We then extend the PCG sampler to develop a scalable ordinal regression model with a real application in the study of student evaluation of surveys. Next, we move to modern deep learning. A constrained variational Adam(CVA) algorithm has been introduced to optimize the Bayesian neural network and its connection to stochastic gradient Hamiltonian Monte Carlo has been discussed. We then generalize our algorithm to a CVA-EM, which incorporates the spike-and-slab prior to capturing the sparsity of the neural network. Both nonlinear high dimensional variable selection and network pruning can be achieved by this algorithm. We further show that the CVA-EM algorithm can extend to the graph neural networks to produce both sparse graphs and sparse weights. Finally, we discuss the sparse VAE with $L_{\frac{1}{2}}$ prior as potential future work.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Associate Professor Yanan Fan for her patience and support during my candidature. I could not have asked for better supervisors. I can still remember in my first year, I randomly knocked at the door of her office. During the covid lockdown period, I kept receiving encouragement and useful suggestions from Yanan in weekly meetings, which helped me finish the most important part of my research. Without her, I could not have finished the PhD during this hard time period.

Secondly, I would like to thank my secondary supervisor Prof Josef Dick and my master's thesis supervisor Dr Quoc Thong Le Gia. They provided me with a first taste of research and encourage me to start my PhD.

My sincere thanks also go to my family. I am so grateful for their unconditional support of my study. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am today.

The past four years have been one of the most important periods in my life. I met lots of nice people at the school of mathematics and statistics, UNSW. I had a pleasant time with my fellow PhD students, Leyang Zhao, Kai Yi, Guanting Liu, Yandong Lang, Fiona Kim, Anant Mathur and Yudhi Bunjamin. I still remember all the good moments with them. In particular, thanks to the encouragement from Guanting during my PhD thesis writing period, thanks Kai, who discussed deep learning and Bayesian statistic with me every day in my last semester and thanks also go to Leyang, who built a good friendship with me and accompanied me from I began my master degree to today.

I would like to also thank Dr Xingyan Quan, Dr Yuehua Li and Ziyu Li, who took me to play badminton. I met lots of nice people in the badminton group.

Finally, I am also very thankful to the CSC scholarship for funding me during my candidature.

Contents

Chapter 1

Introduction

1.1 Background

Model and variable selection problems are fundamental aspects of statistics and machine learning. This is especially challenging when the dimension of the model or parameter space is very large. Here the main assumption we work with is that the high dimensional parameters are sparse with many components being exactly or nearly zero, and only the nonzero components contribute to the model. These assumptions are crucial in ensuring the identification of the true underlying sparse model, especially in cases where the relative sample sizes are small.

There has been much work on penalization approaches within the frequentist paradigm. Starting from the convex penalty LASSO [Tibshirani, 1996], followed by non-convex penalization methods for high-dimensional variable selection including smoothly clipped absolute deviation (SCAD) [Fan and Li, 2001], Dantzig selector [Candes and Tao, 2007], minimum concave penalty (MCP) [Zhang et al., 2010] and many variations of these methods [Zou, 2006, Belloni et al., 2011].

The research for Bayesian variable selection [Mitchell and Beauchamp, 1988, George and McCulloch, 1993] began even earlier than the penalization approaches. Bayesian methods

allow probabilistic modelling via MCMC stochastic search strategies and incorporate optimal model averaging predictions. In the early years, the Bayesian approaches were not as popular as the penalization approaches due to the computational bottleneck. After 2010, Bayesian variable selection becomes a hot-spot research topic in the statistics community. The sparse inducing prior proposed by the researchers can be classified into two categories: spike-and-slab priors and global-local shrinkage priors. The spike-and-slab prior [George and McCulloch, 1997, Ishwaran et al., 2005] places a latent binary vector to index the possible subsets of predictors and used to induce mixture priors of two components on the model parameter, one peaked at zero (spike) and the other one a diffuse distribution(slab). The global-local shrinkage priors [Carvalho et al., 2010,Polson and Scott, 2010,Polson and Scott, 2012] place absolutely continuous shrinkage priors on the entire parameter vector that selectively shrinks the small signals. In addition, the application is not limited to the simple linear regression setting [Wang, 2012,Ročková and George, 2016b,Ghosh et al., 2019, Kowal et al., 2019]. For nonlinear models, such as deep neural networks, the variable selection also acts as Bayesian model selection as we can not only select the relevant predictors but also obtain the sparse neural networks.

However, even with the rapid development of modern computational power, implementing Bayesian inference is still very challenging in the context of big data and big models. The main purpose of this thesis is to develop scalable Bayesian model selection algorithms in a high-dimensional setting.

In chapter 1, we will briefly review the sparsity-inducing priors from two categories referred to above and outline some of the current computation algorithms. In chapter 2, we first present the $L_{\frac{1}{2}}$ prior and some of the properties found by our research. Then a novel Markov chain Monte Carlo (MCMC) scheme and optimization method for $L_{\frac{1}{2}}$ prior in high dimensional regression will be proposed. Chapter 3 extends the $L_{\frac{1}{2}}$ prior to ordinal response regression with random effects. In Chapter 4, we introduce sparse deep learning in the Bayesian framework. We introduce the CVA-EM algorithm which can perform network pruning and variable selection in various neural networks. In chapter 5, we give a conclusion and point out the future direction of the research.

1.2 Spike and slab prior

For concreteness and ease of introduction, we focus on the high-dimensional linear regression problem in chapter 1 and chapter 2.

$$Y = X\boldsymbol{\beta} + \sigma\epsilon \tag{1.1}$$

where $Y = (y_1, \ldots, y_n)$ is an n-dimensional response vector, assumed to have been centred to 0 to avoid the need for an intercept. X is an $n \times p$ design matrix consisting of p standardised covariate measurements. $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is a $p \times 1$ vector of unknown coefficients, $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is a vector of i.i.d noise with mean 0 and variance 1, and $\sigma > 0$ is the error standard deviation. The model selection problem arises when it is believed that not all p covariates are relevant to the response. A natural and theoretical optimal [Shao, 1997] approach is the best subset selection, which considers all sub-models and selects the model with the best fit according to some criteria. This method is equivalent to adding L_0 penalty to the loss function. The spike and slab priors are designed to mimic the behaviour of L_0 in probabilistic modelling. These priors index 2^p subset choices by the vector $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)'$.

1.2.1 Spike and slab prior settings

There are three versions of spike and slab prior settings:

- g-prior [Smith et al., 1996, George and Foster, 2000, Liang et al., 2008]:

$$\boldsymbol{\beta}|\sigma^2 \sim N(0, g\sigma^2(X_\gamma^T X_\gamma)^{-1}). \tag{1.2}$$

  The g-prior is not a 'pure' Bayesian procedure, which requires the prior to be formulated without regard to the observed data. This prior depends on the design matrix $X^T X$ and g control how informative this prior will be. The main computational advantage is that it is a conjugate prior in a linear regression setting, resulting in an analytical form of the marginal likelihood. There exist some works [Bové and

Held, 2011,Li and Clyde, 2018] trying to extend the g-prior to the generalized linear model. The computation is more complicated in the generalised linear model (GLM) case. Since the form of the g-prior depends on the likelihood form, it is pretty hard to extend it to a more general model setting. We will not discuss it too much in the thesis.

- Discrete two components mixture prior [Mitchell and Beauchamp, 1988, George and McCulloch, 1997]:

$$\beta_j|\sigma^2, \gamma \sim (1-\gamma_j)\delta_0(\beta_j) + \gamma_i p_1(\beta_j) \tag{1.3}$$

  where $\delta_0(\cdot)$ is the Dirac function at $\beta_j = 0$ (spike). When $\gamma_j = 0$, the jth variable will be excluded from the model, since the prior on β_j is a point mass distribution at 0. The distribution p_1 is a flatter continuous distribution (slab) that assigns more probability mass to large values of the parameters as this prior corresponds to the signals ($\gamma_j = 1$). Bayesian variable selection can be achieved by looking at the γ vectors with the largest joint posterior probabilities $\pi(\gamma|Y)$.

- Continuous two components mixture prior [George and McCulloch, 1993,George and McCulloch, 1997, Narisetty and He, 2014, Ročková and George, 2018]

$$\beta_j|\sigma^2, \gamma \sim (1-\gamma_j)p_0(\beta_j) + \gamma_i p_1(\beta_j). \tag{1.4}$$

  Instead of using a point mass at zero, the continuous construction (1.4) allows a small diffusion from zero when $\gamma_j = 0$. From a computational perspective, the discrete version of type (1.3) will exclude non-selected variables from calculating the likelihood. Consequently, it doesn't allow the simple Gibbs sampling scheme as the continuous version [George and McCulloch, 1993].

- Rescaled spike-and-slab prior variance [Ishwaran et al., 2005]:

$$\begin{aligned}
&\beta_j \mid \gamma_j, \tau_j^2 \sim \mathrm{N}\left(0, \gamma_j \tau_j^2\right), \\
&\gamma_j \mid w \sim (1-w)\delta_{\gamma^*}(\cdot) + w\delta_1(\cdot) \\
&\tau_j^{-2} \sim \mathrm{Gamma}\left(a_1, a_2\right), \\
&w \sim \mathrm{U}(0,1),
\end{aligned} \tag{1.5}$$

where γ^* is the hyper-parameter that should be chosen to be small. The idea of this approach is actually close to the global-local shrinkage prior. Prior to this family can be represented as a scale-mixture of normals with the scale being the product of global parameters and local parameters. For rescaled spike-and-slab prior, the scale is the product of two local parameters($\gamma_j \tau_j^2$). The hyper-parameter w plays a similar role as global parameters. w controls how likely γ_j is 1 or γ^*, and thus it controls how many β_j are pushed toward zero and so the complexity of the model. However, with the global-local shrinkage prior increasingly popular in recent years, people either use the continuous spike and slab prior as (1.4) or the global-local shrinkage prior.

1.2.2 Computation

Among the four versions of spike-and-slab priors, continuous spike-and-slab priors with the type of (1.4) are commonly used for high-dimensional Bayesian variable selection due to the ease of design of the Gibbs sampling scheme [George and McCulloch, 1993, George and McCulloch, 1997] and optimization algorithm for posterior mode search [Ročková and George, 2014, Ročková, 2018, Ročková and George, 2018]. In the most recent decade, variational methods have also become popular in the discrete spike-and-slab prior for high dimensional linear regression [Carbonetto and Stephens, 2012, Huang et al., 2016, Ray and Szabó, 2021] setting. We will give a brief review of these approaches.

1.2.2.1 Gibbs sampler

With the linear regression model (1.1) and continuous Gaussian spike-and-slab prior, $\pi(\beta_i \mid \gamma_i) = \gamma_i \phi(\beta_i \mid v_1) + (1-\gamma_i)\phi(\beta_i \mid v_0)$, the standard Gibbs sampling algorithm would take the following form:

1. The full conditional distribution of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} \mid \boldsymbol{\gamma}, \sigma^2, \mathbf{y}, X \sim N\left(\left(X^\top X + \sigma^2 \boldsymbol{D}_\gamma\right)^{-1} X^\top \mathbf{y}, \sigma^2 \left(X^\top X + \sigma^2 \boldsymbol{D}_\gamma\right)^{-1}\right) \tag{1.6}$$

where $\boldsymbol{D}_\gamma = \text{Diag}\left(\gamma\tau_1^{-2} + (1-\gamma)\tau_0^{-2}\right)$.

2. The full conditional distributions for γ_j are independent across different j and take the following form:

$$P\left(\gamma_j = 1 \mid \boldsymbol{\beta}\right) = \frac{\theta\phi\left(\beta_j, 0, \tau_1^2\right)}{\theta\phi\left(\beta_j, 0, \tau_1^2\right) + (1-\theta)\phi\left(\beta_j, 0, \tau_0^2\right)}. \tag{1.7}$$

This scheme can be extended to the continuous Laplace spike-and-slab prior [Ročková and George, 2018] by using the Normal-Exponential representation of Laplace distribution [Park and Casella, 2008]. However, sampling from a p-dimensional Gaussian distribution for $\boldsymbol{\beta}$ is computationally intensive when p is large. The computational complexity is $O(p^3)$ due to performing the cholesky decomposition of $\left(X^\top X + \sigma^2 \boldsymbol{D}_\gamma\right)$ in each iteration [Rue, 2001] or $O(np^2)$ if the sampling strategy from [Bhattacharya et al., 2016] has been used.

Skinny Gibbs sampler

To reduce the computational burden, [Narisetty et al., 2018] proposed a scalable Gibbs sampling algorithm called the Skinny Gibbs. The Skinny Gibbs sampling splits the β into an active set ($\gamma_j = 1$) and an inactive set ($\gamma_j = 0$). If the oracle model is very sparse, the dimension of the active set is low and can be sampled from a multivariate Normal distribution. The inactive part has a high dimension and is sampled from a Normal distribution with independent marginals. In step one, β is sampled from

$$\begin{aligned}
&\beta|\gamma, \sigma^2, y, X \sim \text{N}\left(\left(X_\gamma^\top X_\gamma + \tau_1^{-2}\boldsymbol{I}\right)^{-1} X_\gamma^\top Y, \sigma^2\left(X_\gamma^\top X_\gamma + \tau_1^{-2}\boldsymbol{I}\right)^{-1}\right) \\
&\beta|\gamma^c, \sigma^2, y, X \sim \text{N}\left(0, \sigma^2\left(\text{Diag}\left(X_{\gamma^c}^\top X_{\gamma^c}\right) + \tau_0^{-2}\boldsymbol{I}\right)^{-1}\right).
\end{aligned} \tag{1.8}$$

However, this modification will lead to the loss of the dependence between the active set and the inactive set. Therefore, an adjustment step in the sampling of γ is added to make sure the resulting posterior distribution has the same desired variable selection consistency properties. In step two, we sample γ_j conditioned on the remaining components of γ

$$\pi\left(\gamma_j = 1 \mid \boldsymbol{\gamma}_{-j}, \boldsymbol{\beta}, Y\right) = \frac{\theta\phi\left(\beta_j, 0, \tau_1^2\right)\exp\left[\beta_j X_j^\top\left(Y - X_{C_j}\boldsymbol{\beta}_{C_j}\right)\right]}{(1-\theta)\phi\left(\beta_j, 0, \tau_0^2\right) + \theta\phi\left(\beta_j, 0, \tau_1^2\right)\exp\left[\beta_j X_j^\top\left(Y - X_{C_j}\boldsymbol{\beta}_{C_j}\right)\right]}$$

where $C_j = \{k : k \neq j, \gamma_k = 1\}$. The joint posterior corresponding to the Skinny Gibbs algorithm is given by

$$\pi(\beta, \gamma \mid X, Y) \propto \exp\left\{-\frac{1}{2\sigma^2}\|Y - X\beta_\gamma\|^2\right\} v^{-|\gamma|} \prod_i \exp\left\{-\frac{1}{2}\left(\beta' D_\gamma \beta + \beta_{\gamma^c}^T \text{Diag}\left(X_{\gamma^c}^\top X_{\gamma^c}\right)\beta_{\gamma^c}\right)\right\}$$

where $v = \tau_1 (1-q) / (q\tau_0)$ and $D_\gamma = \text{Diag}\left(\gamma\tau_1^{-2} + (1-\gamma)\tau_0^{-2}\right)$. As we can see, the modification strategy from [Narisetty et al., 2018] leads to different prior from continuous Gaussian spike-and-slab prior and hence different posterior distribution. They also show that the new posterior has strong model selection consistency properties. The skinny Gibbs can be generalized to many modelling settings where the likelihood or priors involved can be written as mixtures of Normal distributions.

1.2.2.2 The EM Approach to Bayesian Variable Selection

Below we discuss the fast Bayesian model selection strategies based on detecting the marginal posterior mode [Ročková and George, 2014, Wang et al., 2016, Ročková, 2018]. With the continuous Gaussian spike and slab prior and linear regression, we write the full posterior distribution as follows:

$$\pi\left(\boldsymbol{\gamma}, \sigma^2, \boldsymbol{\beta}, \theta \mid Y\right) \propto P\left(Y \mid \boldsymbol{\beta}, \sigma^2\right) \pi(\boldsymbol{\beta} \mid \boldsymbol{\gamma}) \pi(\boldsymbol{\gamma} \mid \theta) \pi(\theta) \pi(\sigma^2)$$

where $\pi\left(\sigma^2\right) \sim \text{IG}(\eta/2, \eta\nu/2)$ and $\theta \sim \text{Beta}(a, b)$. The Bayesian variable selection amounts to finding the maxima of the marginal posterior of model probability:

$$(\hat{\boldsymbol{\gamma}}, \hat{\sigma}^2) = \arg\max_{\boldsymbol{\gamma}, \sigma^2} \log \pi\left(\boldsymbol{\gamma}, \sigma^2 \mid Y\right).$$

The discrete optimization problem can be tackled with an EM data augmentation strategy, which will optimize a converging sequence of surrogate objective functions.

E-step:

By treating (β, θ) as the latent variable and integrating them out with respect to the

conditional posterior, we have the following complete-data surrogate objective function:

$$\begin{aligned} Q\left(\boldsymbol{\gamma}, \sigma^2 \mid \boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right) &= \mathrm{E}_{\pi(\boldsymbol{\beta},\theta|\boldsymbol{Y},\boldsymbol{\gamma}^{(t)},\sigma^{2(t)})} \log \pi\left(\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \theta \mid \boldsymbol{Y}\right) \\ &= C + \frac{1}{2}\sum_{i=1}^{p} \gamma_i \log\left(\frac{v_0}{v_1}\right) - \frac{1}{2}\sum_{i=1}^{p}\left(\frac{\gamma_i}{v_1} + \frac{1-\gamma_i}{v_0}\right) \mathrm{E}_{\pi(\boldsymbol{\beta}|Y,\boldsymbol{\gamma}^{(t)},\sigma^{2(t)})}\left[\beta_i^2 \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right] \\ &+ \sum_{i=1}^{p} \gamma_i \mathrm{E}_{\pi(\theta|\boldsymbol{\gamma}^{(t)})}\left[\log\left(\frac{\theta}{1-\theta}\right) \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}\right] - \frac{n+\eta}{2}\log\sigma^2 \\ &- \frac{\eta v + \mathrm{E}_{\pi(\beta|Y,\boldsymbol{\gamma}^{(t)},\sigma^{2(t)})}\left[\|Y - X\boldsymbol{\beta}\|^2 \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right]}{2\sigma^2} \end{aligned} \tag{1.9}$$

where the constant C above absorbs all the terms that do not depend on the parameters of interest $(\boldsymbol{\gamma}, \sigma^2)$. Note that since $\pi(\theta \mid \boldsymbol{\gamma}) \sim \mathcal{B}(a + |\boldsymbol{\gamma}|, b + p - |\boldsymbol{\gamma}|)$ and $\pi(\beta \mid Y, \gamma, \sigma^2)$ is the multivariable Gaussian as (1.6), we have

$$\begin{aligned} \mathrm{E}_{\pi(\theta|\boldsymbol{\gamma}^{(t)})}\left[\log\left(\frac{\theta}{1-\theta}\right) \mid Y, \boldsymbol{\gamma}^{(t)}\right] &= \psi\left(a + \left|\boldsymbol{\gamma}^{(t)}\right|\right) - \psi\left(b + p - \left|\boldsymbol{\gamma}^{(t)}\right|\right) \\ \mathrm{E}_{\pi(\boldsymbol{\beta}|Y,\boldsymbol{\gamma}^{2(t)},\sigma^{2(t)})}\left[\beta_j^2 \mid Y, \boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right] &= \mu_j\left(\boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right)^2 + \Sigma_{jj}\left(\boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right) \end{aligned}$$

where $\psi(\cdot)$ denotes the digamma function.

M-step:

We update $(\boldsymbol{\gamma}, \sigma^2)$ sequentially by holding the other fixed. By fixing σ^2, updating $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)$ to maximize the objective function (1.9) are independent for each dimension. In fact, in this case, the objective function (1.9) can be regarded as the log-likelihood of a Bernoulli trial with an inclusion probability for each dimension j

$$\pi_j = \left(1 + \exp\left\{\mathrm{E}\left[\log\left(\frac{1-\theta}{\theta}\right) \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right]\right\} \frac{\phi\left(\sqrt{\mathrm{E}\left[\beta_j^2 \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right]}, v_0\right)}{\phi\left(\sqrt{\mathrm{E}\left[\beta_j^2 \mid \boldsymbol{Y}, \boldsymbol{\gamma}^{(t)}, \sigma^{2(t)}\right]}, v_1\right)}\right)^{-1}.$$

We set $\gamma_j^{t+1} = 1$ if and only if $\pi_j > 0.5$. Next by fixing γ and set $\frac{\partial Q}{\partial \sigma^2} = 0$, we have

$$\sigma^{2(t+1)} = \frac{1}{n+\eta}\left\{\eta\nu + Y'Y - 2\boldsymbol{\mu}\left(\boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right)' X'Y + \mathrm{tr}\left[X'X\left(\Sigma\left(\boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right) + \boldsymbol{\mu}\left(\boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right)\boldsymbol{\mu}\left(\boldsymbol{\gamma}^{(t)}, \sigma^{(t)}\right)'\right)\right]\right\}.$$

Remark: In the early paper [Ročková and George, 2014], the roles of β and γ are switched. The EM algorithm targets the modes in the parameter space $\pi\left(\boldsymbol{\beta}, \sigma^2, \theta \mid Y\right)$. The variable selection is done by defining submodel $\boldsymbol{\gamma}$ associated with the posterior mode $(\hat{\boldsymbol{\beta}}, \hat{\theta}, \hat{\sigma}^2)$ such that $\hat{\boldsymbol{\gamma}} = \arg\max_{\gamma} \pi(\boldsymbol{\gamma} \mid \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}, \hat{\sigma})$.

1.2.2.3 Variational inference (VI)

From the Bayesian perspective, the spike-and-slab prior is the most natural way to induce sparsity as it assigns each potential model a probabilistic weight. However, the discrete model selection component γ can make computation challenges. Inference using the spike-and-slab prior generally involves a combinatorial search over 2^p possible models, which leads to a slow mixing. This is the motivation for developing the EM algorithm for spike-and-slab prior. The main disadvantage of EM algorithm is that the uncertainty quantification has been lost as we only obtain a single posterior mode.

A popular scalable alternative variational Inference (VI) [Jordan et al., 1999, Bishop and Nasrabadi, 2006], which minimizes the Kullback–Leibler (KL) divergence between a family of tractable distributions and the posterior. Typically, variational approaches provide accurate estimates of mean parameters but tend to underestimate the posterior variances and the correlation structure of the data.

A common approach for approximated posterior is the mean field variational inference, which assumes that the approximated posterior factorizes over some partition of the parameters. This approach is widely used with discrete spike-and-slab prior of the type (1.3) [Carbonetto and Stephens, 2012,Huang et al., 2016,Ray and Szabó, 2021], where the diffuse distribution p_1 in the slab part can be either Gaussian or Laplace. They assumed that the approximated posterior of (β, γ) can be factorized as

$$q(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{j=1}^{p} q(\beta_j, \gamma_j) = \prod_{j=1}^{p} [w_j N(\mu_j, s_j^2)]^{\gamma_j} [(1 - w_j)\delta_0]^{(1-\gamma_j)} \tag{1.10}$$

with free variational parameters $\mu_j \in \mathrm{R}$, $s_j^2 \in \mathrm{R}^+$ and $w_j \in [0, 1]$. Then we have the

objective function

$$\arg \min_{\boldsymbol{\mu},\boldsymbol{s},\boldsymbol{w},\sigma^2,\theta} E_{q(\boldsymbol{\beta},\boldsymbol{\gamma})}\left[\log \frac{\pi(\beta,\gamma,\sigma^2,\theta|Y)}{\prod_{j=1}^{p} q(\beta_j,\gamma_j)}\right]$$

where σ^2 and θ are hyper-parameters. This approach is the variational EM algorithm, which combines the VI steps on $q(\boldsymbol{\beta},\boldsymbol{\gamma})$ and expectation maximization (EM) estimation steps on (σ^2,θ). In fact, one could consider the EM variable selection approaches we discussed as a special case of variational inference [Neal and Hinton, 1998], where the variational distributions are point masses.

A coordinate ascent algorithm can then be implemented by setting the partial derivatives equal to zero. If the diffuse distribution p_1 is Gaussian, the closed form expression for μ_j and s_j^2 exists. For Laplace distribution, all the variational parameters need to be computed by optimization.

Remark:

1. The optimal choice is to work directly with $q(\boldsymbol{\beta},\boldsymbol{\gamma})$. We only factorize $q(\boldsymbol{\beta},\boldsymbol{\gamma},\sigma^2) = q(\boldsymbol{\beta},\boldsymbol{\gamma})q(\sigma^2)$. In this case, $q(\boldsymbol{\beta})$ is a mixture distribution with 2^p components given by $q(\boldsymbol{\beta}) = \sum_{\boldsymbol{\gamma}\in\{0,1\}^p} q(\boldsymbol{\gamma})q(\boldsymbol{\beta} \mid \boldsymbol{\gamma})$, which is impossible to evaluate for large p.

2. There may exist another choice such that

$$q(\boldsymbol{\beta},\boldsymbol{\gamma}) = q(\boldsymbol{\beta})q(\boldsymbol{\gamma}). \tag{1.11}$$

This factorization does a better job of estimating the posterior variances of regression coefficients by keeping all of the regression coefficients in the same partition. [Ormerod et al., 2017] used this factorization in a linear regression model with Bernoulli Gaussian prior [Soussen et al., 2011], which can be written as

$$\begin{aligned} &Y \mid \boldsymbol{\beta},\boldsymbol{\gamma} \sim N\left(X\boldsymbol{\Gamma}\boldsymbol{\beta},\mathbf{I}\right) \\ &\beta_j \sim N\left(0,v^2\right) \\ &\gamma_j \sim \text{Bernoulli}(\rho), j = 1,\ldots,p, \end{aligned} \tag{1.12}$$

where $\boldsymbol{\Gamma} = \text{diag}(\gamma_1,...,\gamma_p)$. We assume that the noise in the regression is the standard Normal distribution to simplify the analysis. With this hierarchical structure, the conditional

posterior for $\boldsymbol{\beta}$ is

$$\pi(\boldsymbol{\beta}|Y,\boldsymbol{\gamma}) = \pi(\beta_{\mathcal{A}}|Y,\gamma_{\mathcal{A}}) \prod_{j\in\mathcal{A}^c} N(\beta_j, 0, v^2) = N(\boldsymbol{\beta}, \mu_p, \Sigma_p)$$

where $\mathcal{A} = \{j : \gamma_j = 1\}$, $\Sigma_p = (\boldsymbol{\Gamma} X^T X \boldsymbol{\Gamma} + v^2 I_p)^{-1}$ and $\mu_p = \Sigma_p \boldsymbol{\Gamma} X^T Y$. In contrast, for the discrete spike-and-slab prior,

$$\pi(\boldsymbol{\beta}|Y,\boldsymbol{\gamma}) = \pi(\beta_{\mathcal{A}}|Y,\gamma_{\mathcal{A}}) \prod_{j\in\mathcal{A}^c} \delta_0(\beta_j).$$

Since the optimal $q(\boldsymbol{\beta})$ with the factorization 1.10 are of the form

$$q(\boldsymbol{\beta}) \propto \exp\left\{\mathbb{E}_{q(\boldsymbol{\gamma})}\left[\log \pi\left(\boldsymbol{\beta} \mid \boldsymbol{\gamma}, Y\right)\right]\right\} \tag{1.13}$$

we see that there exists closed-form expression for $q(\boldsymbol{\beta})$ with the Bernoulli Gaussian prior setting in (1.12), but it is intractable for discrete spike-and-slab prior.

1.3 Global-local shrinkage prior

Although spike-and-slab priors are intuitively appealing and possess attractive theoretical properties, posterior sampling is challenging as exploring the full posterior using point mass mixture priors is prohibitive due to the combinatorial complexity of updating γ. The idea of global-local shrinkage prior introduced by [Carvalho et al., 2010, Polson and Scott, 2010, Polson and Scott, 2012] alleviates this by replacing binary latent variable γ to continuous shrinkage parameter. It takes the form of a scale mixture of the Normal distribution:

$$\beta_j \mid g^2, \tau_j^2 \sim N(0, g^2\tau_j^2), \quad g^2 \sim \pi_G, \quad \tau_j^2 \sim \pi_L \tag{1.14}$$

where the global shrinkage parameter g^2, which is shared by all the dimensions, controls the sparsity of the model and the local shrinkage parameter τ_j^2 identify signals. The global–local shrinkage framework has motivated the design of many sparse priors in the recent ten years [Armagan et al., 2013, Bhattacharya et al., 2015, Zhang et al., 2020, Shin and Liu, 2021], with varying degrees of success in theoretical and numerical performance.

Among all these global-local shrinkage priors, the horseshoe prior in the early paper from [Carvalho et al., 2010] is still the most popular one.

In this section, we will discuss the global-local shrinkage prior from both the selection method and shrinkage method perspective. The former is the frequentist approach which produces an exact zero by thresholding the parameters while the latter is the Bayesian approach, which pushes small signals towards zero but not exactly zero. The shrinkage methods work well when the true parameter $\boldsymbol{\beta}$ has few large entries and many small nonzero entries. The horseshoe prior will be used as an example.

1.3.1 Dual of global-local shrinkage prior and its penalty

There exists a duality between the sparse regularization approach and the mode of the posterior distribution from global-local shrinkage prior. On the one hand, the sparse regularization approach leads to an optimization problem of the form

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big\{ l(y \mid \boldsymbol{\beta}) + \text{pen}_g(\boldsymbol{\beta}) \Big\} \tag{1.15}$$

where $l(y|\beta)$ is the negative log-likelihood and $\text{pen}_g(\boldsymbol{\beta})$ is a sparse penalty. On the other hand, if $\pi(\beta|g) \propto \exp\Big(-\text{pen}_g(\boldsymbol{\beta})\Big)$ is a proper prior, then we have the well-defined posterior

$$\pi(\beta|Y) \propto \exp\Big(-l(y|\beta) - \text{pen}_g(\boldsymbol{\beta})\Big)$$

with the solution from (1.15) as its posterior mode. Note that there exists improper global-local shrinkage prior, which is the Normal-Jeffrey mixture [Figueiredo, 2003, Bae and Mallick, 2004]. One should be careful when using this prior in Bayesian computation as it can result in improper posterior.

Now, two natural questions arise: What kind of properties should $\text{pen}_g(\boldsymbol{\beta})$ satisfy to produce a sparse solution? Will penalty from every global-local shrinkage prior satisfy these properties?

[Fan and Li, 2001] answered the first question by showing the three ideal properties for sparse penalties:

- Sparsity: Estimator is sparse. A sufficient condition is that $\min_{t\geq 0}\left\{t+\text{pen}'_g(t)\right\}>0$
- Nearly unbiasedness: The resultant estimator is (nearly) unbiased when the true parameter is large. A sufficient condition is $\text{pen}'_g(t)\to 0$ for large t
- Continuity: The estimator is continuous with respect to response y to achieve robust prediction. This is true if and only if $\text{argmin}_{t\geq 0}\left\{t+\text{pen}'_g(t)\right\}=0$

For the second question, a quick answer is no. To give more details, we begin by outlining the result from [Andrews and Mallows, 1974].

If $\pi(\beta_j|g)\propto\exp\left(-\text{pen}_g(\beta_j)\right)$ belongs to the global-local shrinkage family such that

$$\exp\left(-\text{pen}_g(\beta_j)\right)\propto\int_0^\infty g^{-1}\tau_j^{-1}\phi\left(g^{-1}\tau_j^{-1}\beta_j\right)h(\tau_j)d\tau_j$$

where $h(\cdot)$ is a density function. Then $\text{pen}_g(\beta_j)$ is a monotone increasing function for $\beta_j\in[0,\infty)$. Therefore, $\text{pen}'_g(\beta_j)\geq 0$ for $\beta_j\in[0,\infty)$.

Hence, we further need $\text{pen}'_g(0)>0$ to satisfy the sparsity property. This is generally not true for prior belongs to the global-local shrinkage family. A counter example is $\tau_j^2\sim U(1,2)$. We can put a stronger condition on global-local shrinkage prior to guaranteeing the sparsity property. That is if the global-local shrinkage prior $\pi(\beta_j|g)$ has Laplace mixture representation [Bhattacharya et al., 2015], then $\text{pen}_g(\cdot)$ must be a strictly increasing function on $[0,\infty)$ and unbounded [Zou and Li, 2008]. The second and third properties are also not generally true for global-local shrinkage prior. For example, the Lasso satisfies the continuity property but violates the nearly unbiasedness property.

[Song and Liang, 2017] showed that **if the global-local shrinkage prior has a heavier than exponential tail, and allocates a sufficiently large probability mass in a tiny neighbourhood of zero, then its posterior posterior contraction rate are as good as those of the spike-and-slab priors**. Their result implies that the penalty from an optimal design global-local shrinkage prior must satisfy sparsity and nearly unbiasedness properties. On the other hand, it seems that putting a sufficiently large probability mass around zero often leads to $\lim_{t\to}\text{pen}'_\lambda(t)=\infty$. So far as we know, none

of the penalties from global-local shrinkage prior with the optimal posterior contraction rate satisfies the continuity property. Therefore, their sparse penalties result in a non-convex non-Lipschitz optimization problem. An obvious example is the bridge penalty $\text{pen}_\lambda(t) = \lambda|t|^\alpha$ for $0 < \alpha < 1$ with the exponential power prior as its Bayesian counterpart. Algorithms involving derivatives, for example, will fail to perform.

It should be pointed out that the penalty from spike-and-slab Lasso [Ročková and George, 2018] satisfies all three properties. Thus, we believe it is also possible to develop the global-local shrinkage prior which has optimal properties from both sides in the future. Below, we discuss the properties of the penalty from Horseshoe prior as another example.

[Carvalho et al., 2010] showed that, conditional on the global shrinkage parameter g^2, the horseshoe prior density admits tight upper and lower bounds

$$\frac{\log\left(1+\frac{4g^2}{\beta_j^2}\right)}{g(2\pi)^{3/2}} < \pi_{HS}\left(\beta_j \mid g\right) < \frac{2\log\left(1+\frac{2g^2}{\beta_j^2}\right)}{g(2\pi)^{3/2}}.$$

Then the corresponding penalty is $\text{pen}_g(t) = -\log\log\left(1+\frac{2g^2}{t^2}\right)$ and $\text{pen}'_g(t) = \frac{4g^2/|t|^3}{(1+2g^2/t^2)\log(1+2g^2/t^2)}$. Since $\lim_{t\to 0}\text{pen}'_g(t) = \infty$ and $\lim_{|t|\to\infty}\text{pen}'_g(t) = 0$, the penalty from horseshoe satisfies sparsity and nearly unbiasedness properties and violates the continuity property. In fact, both the density and penalty of horseshoe prior are unbounded at zero suggesting a global solution to the optimization problem identically equal to zero. This is not a good property from a selection method perspective.

1.3.2 Horseshoe estimator: a Bayesian approach for sparse signal

To illustrate the behaviour of the Bayes estimator from sparse prior, we carry out the analysis in terms of the simple Normal mean model. This is similar to a linear regression model with $\boldsymbol{X}^T\boldsymbol{X}$ having an orthogonal design. Suppose that we observe data from the probability model $y_i|\beta_i \sim N(\beta_i, 1)$ for $i = 1, 2, ..., n$. Here the primary goal is to estimate the vector of Normal means $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_n)$ and a secondary goal is to simultaneously test if the β_i's are zero. To do that, we consider the Bayesian method by applying the

sparsity inducing prior to $\boldsymbol{\beta}$. We focus on the posterior mean, which is known to be optimal under quadratic loss.

Assuming the global parameter g^2 is fixed, the global-local shrinkage prior can be represented as $\beta_i \mid \tau_i^2, \lambda \sim N\left(0, \tau_i^2 g^2\right)$ and $\tau_i^2 \sim \pi\left(\tau_i^2\right)$. Then the conditional posterior for β_i is $\beta_i \mid y_i, \tau_i^2, g^2 \sim N\left(\frac{g^2\tau_i^2}{1+g^2\tau_i^2} y_i, \frac{g^2\tau_i^2}{1+g^2\tau_i^2}\right)$. Hence, the marginal posterior mean is

$$\hat{\beta}_i = \mathrm{E}\left(\beta_i \mid y_i, g^2\right) = y_i \int_0^\infty \left(1 - \frac{1}{1+g^2\tau_i^2}\right) \pi\left(\tau_i \mid y_i\right) \mathrm{d}\tau_i = [1 - E(\kappa_i \mid y_i)] y_i \tag{1.16}$$

where we set $\kappa_i = \frac{1}{1+g^2\tau_i^2} \in [0,1]$ as a pseudo inclusion probability. The marginal likelihood after parametrizing κ_i is

$$\pi(y_i \mid \kappa_i) = \kappa_i^{1/2} \exp\left(-\frac{\kappa_i y_i^2}{2}\right). \tag{1.17}$$

The posterior density $\pi(\kappa_i \mid y_i, g^2) \propto \pi(y_i \mid \kappa_i)\pi(\kappa_i \mid g^2)$ identifies signals and noises by letting $E(\kappa_i \mid y_i)] \to 0$ and $E(\kappa_i \mid y_i)] \to 1$. By using Tweedie's formula [Efron, 2011], there also exists an alternative representation for 1.16,

$$E(\beta_i \mid y_i, g^2) = y_i + \frac{\mathrm{d}}{\mathrm{d}y_i} \log \pi(y_i). \tag{1.18}$$

The robustness of the posterior mean can be achieved by finding a sparse prior for β_j such that $\lim_{|y_i|\to 0}\left[y_i + \frac{\mathrm{d}}{\mathrm{d}y_i}\log\pi(y_i)\right] = 0$. For horseshoe prior, the prior and posterior of κ_i are

$$\begin{aligned} \pi\left(\kappa_i \mid g^2\right) &\propto \kappa_i^{-1/2}\left(1-\kappa_i\right)^{-1/2} \frac{1}{1+\left(g^2-1\right)\kappa_i} \\ \pi(\kappa_i \mid y_i, g^2) &\propto \frac{\exp\left(-\kappa_i y_i^2/2\right)}{\left(1-\kappa_i\right)^{1/2}} \frac{1}{g^2\kappa_i + 1 - \kappa_i} \end{aligned}.$$

We can see that the marginal likelihood is zero when $\kappa_i = 0$, which does not help identify the signals. The prior of κ_i from horseshoe fixes this issue. It cancels the $\kappa_i^{1/2}$ term in the marginal likelihood (1.17) in calculating posterior. [Carvalho et al., 2010] showed that there exists a closed-form expression of the marginal posterior mean for horseshoe:

$$\hat{\beta}_{\mathrm{HS}} = E\left(\beta_i \mid y_i, g^2\right) = y_i\left\{1 - \frac{2\Phi_1\left(1/2, 1, 3/2, y_i^2/2, 1-1/g^2\right)}{3\Phi_1\left(1/2, 1, 5/2, y_i^2/2, 1-1/g^2\right)}\right\}. \tag{1.19}$$

In addition, they further showed that $\left|E\left(\beta_i \mid y_i, g^2\right) - y_i\right| \leq b_g$ for some $b_g < \infty$ that depends on g and $\lim_{|y_i|\to\infty} E\left(\beta_i \mid y_i, g^2\right) = y_i$.

Finally, we present the theoretical result of horseshoe prior under the "nearly black" vectors $\ell_0[s_p] = \{\boldsymbol{\beta} \in \mathbb{R}^p : \sum_{i=1}^p \mathbf{1}\{\beta_i \neq 0\} \leq s_p\}$ for some s_p, which may be known (non-adaptive case) or unknown (adaptive case). The asymptotic minimax risk rate in ℓ_2 for nearly black vectors is given by [Donoho et al., 1992] to be $s_p \log(p/s_p)$. The minimax rate is a frequentist criterion for evaluating the convergence of the point estimators to the underlying true parameter. [Ghosal et al., 2000] showed that the minimax rate is the upper bound of the posterior contraction rate. [Van Der Pas et al., 2014] showed that the near-minimax rate can be achieved with the horseshoe prior by setting the global shrinkage parameter $g = (s_p/p)\log(p/s_p)$. That is

$$\sup_{\boldsymbol{\beta} \in \ell_0[s_p]} \mathbb{E}_{\mathbf{y}|\boldsymbol{\beta}} \left\| \hat{\beta}_{\mathrm{HS}}(\mathbf{y}) - \boldsymbol{\beta} \right\|^2 \asymp s_p \log(p/s_p)$$

where $a_n \asymp b_n$ means $\lim_{n\to\infty} \frac{a_n}{b_n} = 1$. In practice, the global shrinkage parameter g is unknown due to the unknown of the sparsity level and must either be estimated by an empirical Bayesian approach or handled via a fully Bayesian approach by putting a suitable prior on g. [van der Pas et al., 2017a, van der Pas et al., 2017b] constructed the restriction maximum marginal likelihood estimator for $\hat{g}$ as an empirical Bayesian approach and, for the fully Bayesian approach, they proposed to use the truncated Cauchy distribution as a prior for g. They further showed that both of them can achieve the near minimax rate.

In fact, this optimal property is not a unique feature for horseshoes prior. [Van Der Pas et al., 2016, Ghosh and Chakrabarti, 2017] proved the near minimax rate for a more general class of global-local shrinkage priors under the non-adaptive cases. The conditions they used for the prior are slightly different, but roughly speaking, both of them require the density of global-local shrinkage priors to have heavy tails and sufficient mass near zero. We will not provide the technical details here as it is not intuitively easy to understand. In fact, checking whether the global-local shrinkage prior satisfies these conditions is hard because not all the global-local shrinkage prior has closed-form expression for $\pi(\tau_i^2)$. This is the main reason why we focus on the horseshoe prior in the section.

Chapter 2

Bayesian computation with $L_{\frac{1}{2}}$ prior

In this chapter, we develop a partially collapsed Gibbs sampling scheme, which outperforms existing Markov chain Monte Carlo strategies under the $L_{\frac{1}{2}}$ prior. The posterior consistency of $L_{\frac{1}{2}}$ prior in a high dimensional linear regression setting has also been proved. In addition, we introduce a non-separable bridge penalty function inspired by the fully Bayesian formulation and a novel, efficient, coordinate-descent algorithm. We prove the algorithm's convergence and show that the local minimizer from our optimization algorithm has an oracle property. Simulation studies were carried out to illustrate the performance of these two algorithms. Finally, we propose a partially collapsed variational inference for $L_{\frac{1}{2}}$ prior, which may be a new framework in the future to extend the mean-field family for variational Bayes.

2.1 Bayesian $L_{\frac{1}{2}}$ prior

The bridge estimator [Frank and Friedman, 1993] is the solution to the objective function of the form

$$\arg\min_{\beta} \frac{1}{2}\|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|^{\alpha}$$

with $0 < \alpha < 1, \lambda > 0$. When $\alpha = 1$, the bridge estimator is the same as the LASSO (whose Bayesian counterpart is also known as Laplace or double exponential prior) [Park and Casella, 2008]. The bridge estimator produces a sparser solution than the LASSO penalty ($\alpha = 1$), while the case of $\alpha = 0$ is the NP-hard problem of subset selection. Several papers have studied the statistical properties of bridge regression estimators and their optimization strategies from a frequentist point of view [Knight et al., 2000, Huang et al., 2008, Zou and Li, 2008]. From a Bayesian perspective, the bridge penalty induces the exponential power prior distribution for β of the form

$$\pi(\beta|\lambda, \alpha) \propto \prod_{j=1}^{p} \frac{\lambda^{1/\alpha}}{2\Gamma(1 + 1/\alpha)} \exp\left(-\lambda |\beta_j|^{\alpha}\right), \quad 0 < \alpha < 1.$$

This prior can be expressed as a scale mixture of normals [Polson et al., 2014, Armagan, 2009, West, 1987]

$$\pi\left(\beta|\lambda, \alpha\right) \propto \prod_{j=1}^{p} \int_0^{\infty} \mathcal{N}\left(0, \tau_j^2 \lambda^{-2/\alpha}\right) \pi\left(\tau_j^2\right) d\tau_j^2$$

where $\pi(\tau_j^2)$ is the density for scale parameter τ_j^2. However, there is no closed form expression for $\pi(\tau_j^2)$. [Polson et al., 2014] proposed to work with the conditional distribution of $\tau^2|\beta$ in their MCMC strategy:

$$\pi(\tau^2|\beta) \sim \prod_{j=1}^{p} \frac{\exp\left(-\lambda^{2/\alpha} |\beta_j|^2 \tau_j^2\right) \pi\left(\tau_j^2\right)}{E_{\pi\left(\tau_j^2\right)}\left\{\exp\left(-\lambda^{2/\alpha} |\beta_j|^2 \tau_j^2\right)\right\}}$$

which is an exponentially-tilted stable distribution.

They suggested the use of the double rejection sampling algorithm of [Devroye, 2009] to sample this conditional distribution. However, the approach is very complicated and not easy to scale to high dimensions.

A second approach proposed by [Polson et al., 2014], is based on a scale mixture of triangular (SMT) representation. More recently, an alternative uniform-gamma representation was introduced by [Mallick and Yi, 2018].

When the design matrix X is orthogonal, both the scale mixture of triangles [Polson et al., 2014] and the scale mixture of uniforms [Mallick and Yi, 2018] work well, but both of them suffer from poor mixing when the design matrix is strongly collinear. This is because both sampling schemes need to generate from truncated multivariate Normal distributions, which are still difficult to sample from efficiently in higher dimensions.

In this section, we propose a new data-augmentation strategy, which provides us with a Laplace-gamma mixture representation of the bridge penalty at $\alpha = \frac{1}{2}$, or $L_{(\frac{1}{2})^\gamma}$ prior for $\gamma = 1$, which is further decomposed into a Normal-Exponential-Gamma mixture. This procedure can be generalised to $\alpha = (\frac{1}{2})^\gamma$ for any $\gamma \in \mathbb{N}^+$. The closed-form representation introduces extra latent variables, allowing us to circumvent sampling difficult conditional distribution $\pi(\tau^2|\beta)$ directly.

We will further leverage the conjugacy structure in our model by removing certain conditional components in the Gibbs sampler without disturbing the stationary distribution. This is done by using the partially collapsed Gibbs sampling strategy [Park and Van Dyk, 2009, Van Dyk and Park, 2008, Van Dyk and Jiao, 2015]. Thus improving the speed of convergence of the Markov chain Monte Carlo (MCMC) algorithm.

we begin with the decomposition of the exponential power prior with $L_{(\frac{1}{2})^\gamma}$. We first present results for the case $\gamma = 1$, and then generalise the decompositions to $\gamma > 1$. Throughout this work we refer to the $L_{(\frac{1}{2})^\gamma}$ prior as the $L_{\frac{1}{2}}$ prior for simplicity.

2.1.1 The Laplace mixture representation

The scale mixture of representation was first found by [West, 1987], who showed that a function $f(x)$ is completely monotone if and only if it can be represented as a Laplace

transform of some function $g(\cdot)$:

$$f(x) = \int_0^\infty \exp(-sx)g(s)\mathrm{d}(s).$$

To represent the exponential power prior with $0 < \alpha < 1$ as a Gaussian mixture, we let $x = \frac{t^2}{2}$,

$$\exp\left(-|t|^\alpha\right) = \int_0^\infty \exp\left(-st^2/2\right) g(s)\mathrm{d}s$$

where $\exp\left(-|t|^\alpha\right)$ is the the Laplace transform of $g(s)$ evaluated at $\frac{t^2}{2}$ [Polson and Scott, 2012, Polson et al., 2014]. Unfortunately, there is no general closed-form expression for $g(s)$. However, for the special case $\alpha = \frac{1}{2^\gamma}$ where γ is any positive integer, we can construct a data augmentation scheme to represent $g(s)$. We begin with special case $\alpha = \frac{1}{2}$ in Lemma 2.1.1.

Lemma 2.1.1. *The exponential power distribution with $\alpha = \frac{1}{2}$, of the form $\pi_\lambda(\beta) = \frac{\lambda^2}{4}\exp(-\lambda|\beta|^{\frac{1}{2}})$, can be decomposed as $\beta|s \sim \mathrm{DE}(0, s)$, $s \sim \mathrm{Gamma}(\frac{3}{2}, \frac{\lambda^2}{4})$ mixture or equivalently, $\beta|v \sim \mathrm{DE}(0, \frac{v}{\lambda^2})$, $v \sim \mathrm{Gamma}(\frac{3}{2}, \frac{1}{4})$ mixture.*

Here $\mathrm{DE}(b)$ denotes a Laplace (also known as double exponential distribution with mean 0 and variance $2b^2$. Since the Laplace distribution can be represented by the Normal-Exponential mixture [Andrews and Mallows, 1974], we, therefore, obtain the Normal-Exponential-Gamma mixture representation for the exponential power prior with $\alpha = \frac{1}{2}$. Lemma 2.1.2 provides a recursive relation for the more general $\alpha = \frac{1}{2^\gamma}$ case.

Lemma 2.1.2. *The exponential power distribution with $\alpha = (\frac{1}{2})^\gamma$, ($\gamma \geq 1$, γ is positive integer) of the form $\pi_\lambda(\beta) \propto \frac{\lambda^{2^\gamma}}{2(2^\gamma!)}\exp\left(-\lambda\left|\beta\right|^{\frac{1}{2^\gamma}}\right)$ can be represented as the mixture of the exponential power distribution with $\alpha = (\frac{1}{2})^{gamma-1}$ and Gamma distribution. More specifically, we have*

$$\pi(\beta|s_\gamma) \propto \frac{1}{2(2^{\gamma-1}!)s_\gamma^{2^{\gamma-1}}}\exp\left(-\frac{|\beta|^{\frac{1}{2^{\gamma-1}}}}{s_\gamma}\right), \quad s_\gamma \sim \mathrm{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{\lambda^2}{4}\right)$$

or equivalently

$$\pi(\beta|v_\gamma, \lambda) \propto \frac{\lambda^{2^\gamma}}{2(2^{\gamma-1}!)v_\gamma^{2^{\gamma-1}}}\exp\left(-\frac{\lambda^2|\beta|^{\frac{1}{2^{\gamma-1}}}}{v_\gamma}\right), \quad v_\gamma \sim \mathrm{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{1}{4}\right).$$

Applying Lemmas 2.1.1 and 2.1.2, the main Theorem 2.1.3 provides the analytic expressions under the scale mixture of normal representation for the exponential power prior.

Theorem 2.1.3. *The exponential power prior with $\alpha = (\frac{1}{2})^\gamma$, ($\gamma \geq 1$ and γ is positive integer) can be represented as laplace mixture with $i = 1, \ldots, \gamma - 1$,*

$$\beta|v_1, \lambda \sim \text{DE}\left(\frac{v_1}{\lambda^{2^\gamma}}\right), \quad v_i|v_{i+1} \sim \text{Gamma}\left(\frac{2^i+1}{2}, \frac{1}{4v_{i+1}^2}\right), \quad v_\gamma \sim \text{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{1}{4}\right)$$

which can further be represented as global-local shrinkage prior:

$$\begin{aligned} &\beta|\tau^2, \lambda \sim N\left(\mathbf{0}, \frac{\tau^2}{\lambda^{2^{\gamma+1}}}\right), && \tau^2|v_1 \sim \text{Exp}\left(\frac{1}{2v_1^2}\right), \\ &v_i|v_{i+1} \sim \text{Gamma}\left(\frac{2^i+1}{2}, \frac{1}{4v_{i+1}^2}\right), && v_\gamma \sim \text{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{1}{4}\right), \end{aligned} \tag{2.1}$$

where $i = 1, \ldots, \gamma - 1$, τ is the local shrinkage parameter and $\frac{1}{\lambda^{2^\gamma}}$ is the global shrinkage parameter. The proof can be found in section A.1.

As we can see, the marginal distribution for the local shrinkage parameter τ has no closed-form expression, leading to computational difficulties in existing MCMC schemes [Polson et al., 2014, Mallick and Yi, 2018]. However, we show that for the cases $\alpha = (\frac{1}{2})^\gamma$, introducing latent variables v_i leads to a computationally efficient decomposition. The augmented representation now makes it easier to create efficiently computational algorithms such as MCMC or the EM algorithm for finding posterior modes.

We now provide some intuition as to why the $L_{\frac{1}{2}}$ prior is suitable as a sparsity prior. We carry out the illustration in terms of the pseudo inclusion probability $\kappa_i = \frac{1}{1+g^2\tau_i^2}$ as what we did in section (1.3.2). To analysis the rule of local shrinkage parameter and make a plot, we further assume the global shrinkage parameter $g^2 = 1$.

Then for the exponential power prior with $\alpha = \frac{1}{2}$, the inducing density of κ is given by

$$\pi_{\alpha=\frac{1}{2}}(\kappa) = \frac{1}{8\sqrt{\pi}\kappa^2} \int_0^\infty v^{-1.5} \exp\left(-\frac{1}{4}v + \frac{1}{2v^2} - \frac{1}{2v^2\kappa}\right) dv$$

Figure 2.1 demonstrates the behaviour of the $L_{\frac{1}{2}}$ prior with $\gamma = 1$ (left panel), as well as the Horseshoe prior (middle panel) and the Laplace (Bayesian LASSO) prior (right panel).

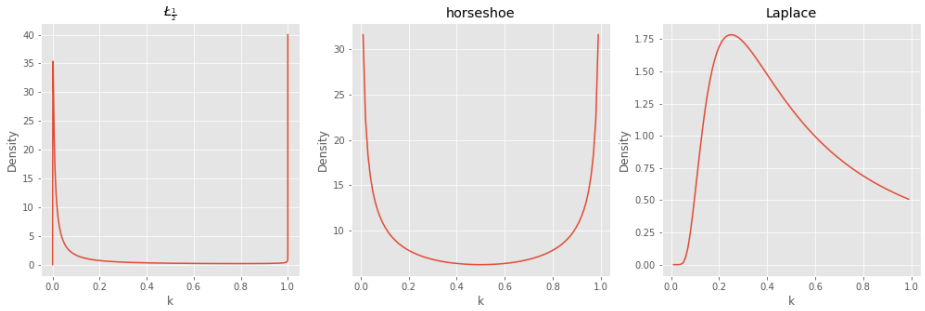


Figure 2.1: The density of κ for $L_{\frac{1}{2}}$ prior with $\gamma = 1$ (left), the Horseshoe prior (middle) and Laplace prior (right). All global shrinkage parameters are set to 1.

The $L_{\frac{1}{2}}$ prior shares many similarities to the Horseshoe prior, both mimic the behaviour of the spike-and-slab prior, putting large probability mass around $\kappa = 0$ and $\kappa = 1$. The $L_{\frac{1}{2}}$ prior starts from zero, then dramatically increase to a first peak at κ_0 near zero and finally follows a U-shape that is very similar to the Horseshoe prior. In addition, its density function is almost zero for $\kappa \in [0.4, 1)$.

For the more general $\gamma > 1$ case, there is no analytic expression for the density of κ, and it is difficult to numerically illustrate the density.

Intuitively as γ becomes larger, α tends to zero, the density of κ will degenerate to two points $\kappa = 0$ and $\kappa = 1$, because the $L_{\frac{1}{2}}$ prior converges to an improper spike-and-slab prior $\pi(\beta) \propto \exp(-\|\beta\|_0) \propto \frac{1}{1+e^{-1}}\delta_0(\beta) + \frac{e^{-1}}{1+e^{-1}}\delta_{|\beta|>0}(\beta)$, induced by the L_0 norm, and δ denotes the dirac mass function. In other words, as α tends to zero the density of $\pi_\alpha(\kappa)$ will degenerate to two point mass with $P(\kappa = 0) = \frac{e^{-1}}{1+e^{-1}}$ and $P(\kappa = 1) = \frac{1}{1+e^{-1}}$. Under the sparse normal mean model, we have

$$\pi(\beta_i|y_i) \propto \frac{1}{1+e^{-1}}\delta_0(\beta_i) + \frac{e^{-1}}{1+e^{-1}}N(y_i, 1).$$

and the corresponding posterior mean for β_i is given by

$$\hat{\beta}_i = \mathrm{E}\left(\beta_i \mid \mathbf{y_i}\right) = \left[1 - \mathrm{E}\left(\kappa_i \mid \mathbf{y_i}\right)\right] \cdot y_i = \frac{e^{-1}}{1+e^{-1}} y_i$$

with $\mathrm{E}\left(\kappa_i \mid \mathbf{y_i}\right) = P(\kappa = 1) = \frac{1}{1+e^{-1}}$.

2.1.2 Choice of hyper-prior

The hyper-parameter λ controls global shrinkage, and is critical to the success of the $L_{\frac{1}{2}}$ prior. It is possible to fix λ by selecting a value for it via an empirical Bayes approach, e.g., marginal maximum likelihood [Casella, 2001]. However, in extremely sparse cases, the empirical Bayes estimate of the global shrinkage parameter might collapse to 0 [Scott and Berger, 2010, Datta et al., 2013]. Assigning a hyper-prior to λ allows the model to achieve a level of self-adaptivity and can boost performance [Scott and Berger, 2010, Ročková and George, 2018]. Here we propose to assign a half Cauchy prior to $\frac{1}{\sqrt{\lambda}}$. That is

$$\frac{1}{\sqrt{\lambda}} \sim \text{Cauchy}_+(0,1),$$

which has the following scale mixture representation:

$$\lambda \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{b}\right), \quad b \sim \text{InvGamma}\left(\frac{1}{2}, 1\right), \tag{2.2}$$

this allows us to keep the conjugate structure of our Gibbs sampling scheme later.

The main motivation for using this prior is that the density of the global shrinkage parameter $t = \frac{1}{\lambda^{2^\gamma}}$ approaches infinity at the origin. This prior yields a proper posterior, allowing for strong shrinkage, while its thick tail can accommodate a wide range of values. This can be seen by the simple calculation:

$$\pi(t) = \frac{1}{2^{\gamma+1}(1+t^{\frac{1}{2^{\gamma+1}}})t^{1-\frac{1}{2^{\gamma+2}}}}; \quad \pi(z) = \frac{1}{2^{\gamma+1}(1+z^{\frac{1}{2^{\gamma+1}}})z^{1-\frac{1}{2^{\gamma+2}}}}$$

where $z = \frac{1}{t} = \lambda^{2^\gamma}$. Thus $\lim_{t\to 0}\pi(t) = \infty$ and $\lim_{z\to 0}\pi(z) = \infty$.

2.1.3 Posterior consistency and contraction rate

We focus on the high-dimensional case where the number of predictors p is much larger than the number of observations n ($p > n$) and most of the coefficients in the parameter vector $\beta = (\beta_1, \ldots, \beta_p)$ are zero. We consider the exponential power prior with $\alpha = \frac{1}{2^\gamma}$ on β under the following setup,

$$Y|\beta, \sigma^2 \sim \mathcal{N}_n\left(X\beta, \sigma^2\mathbf{I_n}\right), \quad \beta|\lambda \propto \prod_{j=1}^{P} \frac{\lambda^{2^\gamma}}{2(2^\gamma!)} \exp\left(-\lambda\,|\beta_j|^{\frac{1}{2^\gamma}}\right), \quad \pi\left(\sigma^2\right) \propto \sigma^{-2}. \tag{2.3}$$

[Mallick and Yi, 2018] showed strong consistency of the posterior under exponential power prior in the case $p < n$. When $p > n$, the non-invertibility of the design matrix complicates analysis. In the following, we show that under the local invertibility assumption of the Gram matrix $\frac{X^T X}{n}$, the L_2 contraction rates for the posterior of β are nearly optimal and no worse than the convergence rates achieved by spike-and-slab prior [Castillo et al., 2015]. Therefore, there is no estimation performance loss due to switching from spike-and-slab prior to the $L_{\frac{1}{2}}$ prior.

In what follows, we rewrite the dimension p of the model by p_n to indicate that the number of predictors can increase with the sample size n, and similarly, we rewrite λ by λ_n. We use β_0 and σ_0 to indicate true regression coefficients and the true standard deviation. Let S_0 be the set containing the indices of the true nonzero coefficients, where $S_0 \subseteq \{1, .., p_n\}$, then $s_0 = |S_0|$ denote the size of the true model. We use Π_n to emphasise that the prior distribution is sample size dependent. We now state our main theorem on the posterior contraction rates for the exponential power prior based on the following assumptions:

A1. The dimensional is high $p_n \succ n$ and $\log(p_n) = o(n)$.

A2. The true number of nonzero β_0 satisfies $s_0 = o(n/\log p_n)$.

A3. All the covariates are uniformly bounded. In other words, there exists a constant $k > 0$ such that $\lambda_{\max}(X^T X) \leq k n^{\alpha}$ for some $\alpha \in [1, +\infty)$.

A4. There exist constants $v_2 > v_1 > 0$ and an integer $\widetilde{p}$ satisfying $s_0 = o(\widetilde{p})$ and $\widetilde{p} = o(s_0 \log n)$, so that $nv_1 \leq \lambda_{min}(X_s^T X_s) \leq \lambda_{max}(X_s^T X_s) \leq nv_2$ for any model of size $|s| \leq \widetilde{p}$.

A5. $||\beta_0||_\infty = E$ and E is some positive number independent of n

Theorem 2.1.4. *(Posterior contraction rates). Let $\epsilon_n = \sqrt{s_0 \log p_n / n}$ and suppose that assumptions A1-A5 hold. Under the linear regression model with unknown Gaussian noise, we endow β with the exponential power prior $\Pi_n(\beta) = \frac{\lambda_n^{\frac{1}{\alpha}}}{2\Gamma(1+\frac{1}{\alpha})} \exp\left(-\lambda_n |\beta|^{\alpha}\right)$, where $0 < \alpha \leq \frac{1}{2}$ and $\lambda_n \propto (\log p_n)$ and $\sigma^2 \sim \text{InvGamma}(\delta, \delta)$. Then*

$$\Pi_n\left(\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_2 \gtrsim \epsilon_n \mid \boldsymbol{Y}\right) \to 0$$

$$\Pi_n\left(\boldsymbol{\beta} : \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}\boldsymbol{\beta}_0\|_2 \gtrsim \sqrt{n}\epsilon_n \mid \boldsymbol{Y}\right) \to 0$$

$$\Pi_n\left(\sigma^2 : |\sigma - \sigma_0| \gtrsim \sigma_0 \epsilon_n \mid Y\right) \to 0$$

where $\Pi_n(\cdot \mid \boldsymbol{Y})$ *denote the posterior distribution under the prior* Π_n*.*

Theorem 2.1.5. *(Dimensionality). We define the generalized dimension as*

$$\gamma_j(\beta_j) = I(|\beta_j| > a_n) \text{ and } |\gamma(\boldsymbol{\beta})| = \sum_{j=1}^{p_n} \gamma_j(\beta_j)$$

where $a_n = \frac{\sigma_0}{\sqrt{2k}}\sqrt{s_0 \log p_n/n}/p_n$ *and* $\lim_{n\to\infty} a_n = 0$*. Then under the assumptions A1-A5, for sufficient large* $M_3 > 0$*, we have*

$$\sup_{\theta_0} P^n_{\theta_0} \Pi\left(\boldsymbol{\beta} : |\gamma(\boldsymbol{\beta})| > M_3 s_0 \mid \boldsymbol{Y}\right) \to 0$$

The proof of theorem 2.1.4 and 2.1.5 can be found in section A.3

2.2 Markov Chain Monte Carlo

The MCMC-based implementations for global-local shrinkage priors for linear regression usually proceed via block updating β, τ^2 and g^2 using either a Gibbs sampler with parameter expansion or slice sampling strategy. For example, [Scott, 2010] proposed a parameter expansion strategy for the horseshoe prior. On a similar route, [Makalic and Schmidt, 2015] used an Inverse-Gamma scale mixture to represent the half-Cauchy distribution and construct a Gibbs sampling scheme for the horseshoe. However, the Gibbs sampling scheme which relies on the scale mixture of Normal representation of the prior only scales well to large n and small p owing to the need to sample the multivariate Gaussian distribution $\pi(\boldsymbol{\beta} \mid \boldsymbol{\tau}^2, g^2, \sigma^2, Y)$ which has $O(p^3)$ complexity [Rue, 2001] or $O(np^2)$ complexity [Bhattacharya et al., 2016]. [Hahn et al., 2019] proposed an elliptical slice sampler for an arbitrary prior with a linear regression model, which scales very well in n but is less efficient in $p \gg n$. In addition, their scheme requires closed-form expression for the unnormalized $\pi(\beta_j \mid g^2)$, which limits its usage.

Recently, [Johndrow et al., 2020] proposed a what they call a JOB-approximation strategy to sample $\pi(\boldsymbol{\beta} \mid \boldsymbol{\tau}^2, g^2, \sigma^2, Y)$. This approximation works well when p is large relative to n and the truth $\boldsymbol{\beta}$ is sparse or close to sparse. Since their strategy can be inserted into

our MCMC scheme for $L_{\frac{1}{2}}$ prior, we briefly review their approach under our Bayesian $L_{\frac{1}{2}}$ regression setting in 2.3. The conditional posterior of $\boldsymbol{\beta}$ under $L_{\frac{1}{2}}$ prior is

$$\beta \mid \sigma^2, \tau^2, \lambda, Y \sim \mathrm{N_p}\left((X^TX + \boldsymbol{\Gamma}^{-1})^{-1}X^TY, \sigma^2(X^TX + \boldsymbol{\Gamma}^{-1})^{-1}\right) \tag{2.4}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}\left(\frac{\tau_j^2}{\sigma^2\lambda^4}\right)$. As we discuss in section 1.2.2.1, [Narisetty et al., 2018] developed a skinny Gibbs sampler to sample the conditional posterior of $\boldsymbol{\beta}$ with spike-and-slab prior, where they partition $\boldsymbol{\beta}$ into active and inactive sets. They ignored the dependence between the active and inactive sets. A further reduction is achieved by sampling the components of $\boldsymbol{\beta}$ in the inactive set independently of each other. Unlike spike-and-slab priors, the binary latent variable naturally partitions the $\boldsymbol{\beta}$ into active and inactive sets, such partition is not naturally available for global-local shrinkage prior.

The JOB-approximation partition $\boldsymbol{\beta}$ into active and inactive set by user defined thresholding parameter $\delta > 0$ such that $S_\delta = \left\{j : \frac{\tau_j^2}{\sigma^2\lambda^4} > \delta\right\}$. Then the approximated covariance can be written as

$$\Sigma_\delta = \begin{bmatrix} \sigma^2\left(X_{S_\delta}^TX_{S_\delta} + \boldsymbol{\Gamma}_{S_\delta}^{-1}\right)^{-1} & -\sigma^2\boldsymbol{\Gamma}_{S_\delta}X_{S_\delta}^TM_{S_\delta}^{-1}X_{S_\delta^c}\boldsymbol{\Gamma}_{S_\delta^c} \\ -\sigma^2\left(\boldsymbol{\Gamma}_{S_\delta}X_{S_\delta}^TM_{S_\delta}^{-1}X_{S_\delta^c}\boldsymbol{\Gamma}_{S_\delta^c}\right)^T & \sigma^2\boldsymbol{\Gamma}_{S_\delta^c} \end{bmatrix}$$

where $M_{S_\delta} = \left(I_n + X_{S_\delta}\boldsymbol{\Gamma}_{S_\delta}X_{S_\delta}^T\right)$. The main distinction between the JOB-approximation algorithm and the skinny Gibbs in (1.8) is that it preserves the correlations between the variables in S_δ and S_δ^c. Note that variables that are thresholded away at iteration k need not be thresholded away at iteration $k+1$. The computation complexity of the JOB-approximation is $O(\max\left(s_\delta^2, p\right)n)$ in each iteration, where $s_\delta = \sum_{j=1}^p \mathbf{1}\left(\frac{\tau_\mathbf{j}^\mathbf{2}}{\sigma^\mathbf{2}\lambda^\mathbf{4}} > \delta\right)$. The main limitation of the JOB-approximation is that it only works for linear regression. In addition, it is not easy to tune the user-defined thresholding parameter $\delta > 0$. This is very disappointing.

Perhaps, we can consider another approximate MCMC based on the unadjusted Langevin dynamic. The Langevin diffusion, defined by the stochastic differential equation

$$\mathrm{d}\boldsymbol{\beta}(t) = -\frac{1}{2}\nabla U(\boldsymbol{\beta}(t))\mathrm{d}t + \mathrm{d}B_t$$

where $\nabla U(\boldsymbol{\beta}(t))$ is the drift term and B_t denotes p dimensional Brownian motion, has $\pi(\boldsymbol{\beta}) \propto \exp\left(-U(\boldsymbol{\beta})\right)$ as its stationary distribution under mild regularity conditions [Roberts

and Tweedie, 1996]. The unadjusted Langevin algorithm [Parisi, 1981] simulates the Euler approximation but does not use a Metropolis accept-reject step and so the MCMC output produces a biased approximation of the target distribution. That is

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \frac{h}{2}\nabla U(\boldsymbol{\beta}(t)) + \sqrt{h}Z$$

where h is the step size and Z is a vector of p independent standard Gaussian random variables. [Welling and Teh, 2011] proposed the stochastic gradient Langevin dynamic(SGLD), where the gradient component of the unadjusted Langevin algorithm is replaced by a stochastic approximation calculated on a subsample of the full data. Recently, SGLD has become a popular tool for scalable Bayesian inference in the deep-learning community. In this approach, all the parameters are sampled simultaneously by stochastic gradient unadjusted Langevin dynamic. Then in another research [Ahn et al., 2015], which is not well known, they used the SGLD within the Gibbs sampler for the Bayesian matrix factorization model, where only the computation-intensive steps are replaced by SGLD. They showed that their approach can achieve the same level of prediction accuracy as Gibbs sampling an order of magnitude faster.

In many cases, the first-order gradient of the log density can be evaluated cheaply (but not always). For example, in Bayesian $L_{\frac{1}{2}}$ regression setting, $\frac{\partial \log \pi(\boldsymbol{\beta}|Y,\tau^2,\lambda,\sigma^2)}{\partial \boldsymbol{\beta}} = \frac{X^T(Y-X\boldsymbol{\beta})}{\sigma^2} - \frac{\lambda^4\boldsymbol{\beta}}{\tau^2}$, which doesn't involve any matrix inverse.

Since the gradients of log density for $\pi(\beta \mid g^2)$ are often not well defined at zero for most of the global-local shrinkage prior, the SGLD can not be applied directly. By using a scale mixture of Normal representation for the sparse prior, SGLD within Gibbs sampler should be a reasonable solution.

2.2.1 Gibbs sampling approach

For the linear model with Gaussian error, together with the scale mixture of Normal representation of $L_{\frac{1}{2}}$ prior, and the hyper-prior, the fully conditional posterior distributions

are given by

$$\beta \mid \sigma^2, \tau^2, v, \lambda, b, Y \sim \mathrm{N_p}\left((X^TX + \sigma^2\lambda^4 D_{\tau^2}^{-1})^{-1}X^TY, \sigma^2(X^TX + \sigma^2\lambda^4 D_{\tau^2}^{-1})^{-1}\right)$$

$$r_j = \frac{1}{\tau_j^2} \mid \beta, v, \lambda, b, Y \sim \text{InvGaussian}\left(\frac{1}{\lambda^2 v_j|\beta_j|}, \frac{1}{v_j^2}\right), \quad j = 1, ..., p$$

$$\sigma^2 \mid \beta, \tau^2, v, \lambda, b, Y \sim \text{InvGamma}\left(\frac{n}{2}, \frac{1}{2}(Y - X\beta)^T(Y - X\beta)\right)$$

$$\lambda^4 \mid \beta, \tau^2, v, \sigma^2, b, Y \propto (\lambda^4)^{\frac{2p-3.5}{4}} \exp\left(-\lambda^4 \sum_{j=1}^{p} \frac{\beta_j^2}{2\tau_j^2} - \frac{\lambda}{b}\right)$$

$$b \mid \beta, \tau^2, v, \lambda, \sigma^2, Y \sim \text{InvGamma}(1, 1 + \lambda)$$

$$v_j \mid \tau^2, \beta, \sigma^2, \lambda, Y \propto (v_j)^{-\frac{3}{2}} \exp\left(-\frac{\tau_j^2}{2v_j^2} - \frac{1}{4}v_j\right), \quad j = 1, ..., p$$

where $D_{\tau^2} = \text{diag}(\tau_1^2, ..., \tau_p^2)$. Here the full conditional distributions for v and λ do not follow any standard distributions, so would require additional sampling strategies, such as the Metropolised-Gibbs sampler or adaptive rejection sampling [Gilks and Wild, 1992]. The partially collapsed Gibbs sampler (PCG) [Park and Van Dyk, 2009] speeds up convergence by reducing the conditioning in some or all of the component draws of its parent Gibbs sampler, which also provides a closed-form conditional posterior in our case. In the next section, we develop a partially collapsed Gibbs sampler based on the approach of [Park and Van Dyk, 2009], first for the case with $\gamma = 1$, then separately for the case with $\gamma > 1$.

2.2.2 Partially collapsed Gibbs sampling

2.2.2.1 Case 1: $L_{\frac{1}{2}}$ prior with $\gamma = 1$

[Van Dyk and Park, 2008] showed that by applying three basic steps: marginalize, permute and trim, one can construct the PCG sampler which maintains the target joint posterior distribution as the stationary distribution.

Following the prescription in [Van Dyk and Park, 2008], the first step of the PCG sampler is marginalization, where we sample the τ_j^2 and v_j's multiple times in the Gibbs cycle

without perturbing the stationary distribution. In the second step, we permute the ordering of the update which helps us identify trimming strategies. Based on the final trimming step, we are able to construct a sequential sampling scheme to sample the posterior where each update is obtained from a closed-form distribution with some conditional components being removed. We provide the exact sampling scheme below in Algorithm 1:

Algorithm 1 PCG $L_{\frac{1}{2}}$, $\gamma = 1$ update algorithm

Initialize the parameters $\beta, \lambda, v, \tau^2, \sigma^2$ and set $t = 1$, sequentially update the parameters until at $t = T$, via

$S1$. Sample $\beta' \sim \mathrm{N_p}\left((X^TX + \sigma^2\lambda^4 D_{\tau^2}^{-1})^{-1}X^TY, \sigma^2(X^TX + \sigma^2\lambda^4 D_{\tau^2}^{-1})^{-1}\right)$

$S2$. Sample $\lambda' \sim \text{Gamma}\left(2p + 0.5, \sum_{j=1}^{p} |\beta'_j|^{\frac{1}{2}} + \frac{1}{b}\right)$

$S3$. Sample $\frac{1}{v'_j} \sim \text{InvGaussian}\left(\sqrt{\frac{1}{4\lambda'^2|\beta'_j|}}, \frac{1}{2}\right), \quad j = 1, ..., p$

$S4$. Sample $\frac{1}{\tau'^2_j} \sim \text{InvGaussian}\left(\frac{1}{\lambda'^2 v'_j |\beta'_j|}, \frac{1}{v'^2_j}\right), \quad j = 1, ..., p$

$S5$. Sample $\sigma'^2 \sim \text{InvGamma}\left(\frac{n}{2}, \frac{1}{2}(Y - X\beta')^T(Y - X\beta')\right)$

$S6$. Sample $b' \sim \text{InvGamma}\left(1, 1 + \lambda'\right)$

$S7$. Set $\beta, \lambda, v, \tau^2, \sigma^2$ to $\beta', \lambda', v', \tau'^2, \sigma'^2$ and increment $t = t + 1$.

The details of the three steps and derivations for $\pi(\lambda|\beta, \sigma^2, Y)$ and $\pi\left(h_j = \frac{1}{v_j}|\beta_j, \sigma^2, \lambda, Y\right)$ are given in A.2.

2.2.2.2 Case 2: $L_{\frac{1}{2}}$ prior with $\gamma \in \mathbf{N}^+$

We can similarly design a PCG sampler for the more general $\alpha = (\frac{1}{2})^\gamma, \gamma \in \mathbf{N}^+$ case. The construction here is slightly more tedious but within the same spirit as $\gamma = 1$ case. Under $\gamma > 1$, we have an expanded parameter space with $v = \{v_{i,j}\}, i = 1, \ldots, \gamma, j = 1, \ldots, p$,

We omit the details and refer the reader to section A.2 for the intuition under $\gamma = 1$. We describe the algorithm for $\gamma > 1$ below

Algorithm 2 PCG $L_{\frac{1}{2}}, \gamma > 1$ update algorithm

Initialize the parameters $\beta, \lambda, v, \tau^2, \sigma^2$ and set $t = 1$, sequentially update the parameters until at $t = T$, via

$S1$. Sample $\beta' \sim \mathrm{N}_p \left((X^T X + \sigma^2 \lambda^{2^{\gamma+1}} D_{\tau^2}^{-1})^{-1} X^T Y, \sigma^2 (X^T X + \sigma^2 \lambda^{2^{\gamma+1}} D_{\tau^2}^{-1})^{-1} \right)$

$S2$. Sample $\lambda' \sim \mathrm{Gamma} \left(2^\gamma p + 0.5, \sum_{j=1}^{p} \left| \beta'_j \right|^{\frac{1}{2^\gamma}} + \frac{1}{b} \right)$

$S3$. Sample $\frac{1}{v'_{\gamma,j}} \sim \mathrm{InvGaussian} \left(\frac{1}{2\lambda' |\beta'_j|^{\frac{1}{2^\gamma}}}, \frac{1}{2} \right), \quad j = 1, ..., p$

For $i = 1, \ldots, \gamma - 1$,

Sample $\frac{1}{v'_{i,j}} \sim \mathrm{InvGaussian} \left(\frac{1}{2v'_{i+1,j} \lambda |\beta'_j|^{\frac{1}{2^i}}}, \frac{1}{2v'^2_{i+1,j}} \right), \quad j = 1, ..., p$

$S4$. Sample $\frac{1}{\tau'^2_j} \sim \mathrm{InvGaussian} \left(\frac{1}{\lambda'^{2^\gamma} v'_{1,j} |\beta'_j|}, \frac{1}{v'^2_{1,j}} \right), \quad j = 1, ..., p$

$S5$. Sample $\sigma'^2 \sim \mathrm{InvGamma} \left(\frac{n}{2}, \frac{1}{2} (Y - X\beta')^T (Y - X\beta') \right)$

$S6$. Sample$b' \sim \mathrm{InvGamma} \left(1, 1 + \lambda' \right)$

$S7$. Set $\beta, \lambda, v, \tau^2, \sigma^2$ to $\beta', \lambda', v', \tau'^2, \sigma'^2$ and increment $t = t + 1$.

Note that this scheme also works for $\gamma = 0$ (Bayesian LASSO) case. In this case, step 3 is removed and step 4 is replaced by

$S4$. Sample $\frac{1}{\tau'^2_j} \sim \mathrm{InvGaussian} \left(\frac{1}{\lambda |\beta'_j|}, 1 \right), \quad j = 1, ..., p.$

2.3 Optimization

2.3.1 Coordinate descent optimization and non-separable bridge penalty

Whilst the PCG sampler produces full posterior distributions for all the parameters, it does not produce exact zeros for the regression coefficients $\boldsymbol{\beta}$, making variable selection difficult. It may be possible to set a threshold value to separate the important and unimportant variables, however, it is not clear how the threshold can be chosen optimally. On the other hand, it is well known that the penalized likelihood estimators have a Bayesian interpretation as posterior modes under the corresponding prior. Thus, it is possible to construct very flexible sparse penalty functions from a Bayesian perspective, whose solution is the generalized thresholding rule [Ročková and George, 2016a]. We consider a full Bayesian formulation of the exponential power prior, which introduces a non-separable bridge (NSB) penalty

$$\mathbf{pen}(\boldsymbol{\beta}) = -\log \int \prod_j \pi(\beta_j|\lambda)\pi(\lambda)d\lambda$$

with the hyper-parameter λ being integrated out with respect to some suitable choice of hyper-prior. If the goal of inference is to identify important variables very quickly, we propose below a fast optimization strategy which searches for the posterior modes and is capable of producing sparse solutions suitable for variable selection.

Coordinate descent (CD) type of algorithms have been developed for bridge penalties previously, see for example, [Marjanovic and Solo, 2012, Marjanovic and Solo, 2013, Marjanovic and Solo, 2014]. However, the bridge penalty is limited by its lack of ability to adapt to the sparsity pattern across the coordinates. Without knowing the true sparsity, specifying the global shrinkage parameter λ can be challenging. In the spirit of the full Bayesian treatment, by assigning a suitable hyper-prior for λ to the bridge penalty, we can work with the marginalized hyper-parameter λ, which can achieve a level of adaptiveness. A similar approach was used by [Ročková and George, 2016a, Ročková and George, 2018] for the spike-and-slab LASSO penalty. The marginalized log posterior distribution can be

written as:

$$\begin{aligned}\log \pi\left(\beta, b, \sigma^2 \middle| Y\right) &= -\frac{1}{2\sigma^2}\|Y - X\beta\|_2^2 - (n+2)\log\sigma + \log\int \pi(\beta|\lambda)\pi(\lambda|b)d\lambda + \log\pi(b) \\ &= -\frac{1}{2\sigma^2}\|Y - X\beta\|_2^2 - (n+2)\log\sigma - \mathbf{pen_b}(\beta) - \log C_b + \log\pi(b)\end{aligned} \tag{2.5}$$

where b is the hyper-parameter in (2.2) for $\frac{1}{\sqrt{\lambda}}$ and λ is marginalized out. Finally, Jeffreys' prior is used for σ^2 in (2.5). The prior $\pi(\beta|b)$ is now written in terms of the penalty function $\mathbf{pen_b}(\beta) = -\log\int \pi(\beta|\lambda)\pi(\lambda|b)d\lambda - \log C_b$, where C_b is the normalizing constant which depends on b. The posterior mode can then be obtained by maximizing (2.5) with respect to β, b and σ^2

$$(\beta_{MAP}, b_{MAP}, \sigma^2_{MAP}) = \arg\max_{\beta,b,\sigma^2} \log\pi(\beta, b, \sigma^2|Y)$$

which can be achieved by iteratively updating the σ^2, β, b in turn. In fact, the parameter b is not of interest and updating b is also very dangerous to do in high dimension and very sparse problem (see corollary 2.3.2.1), however, it is not possible to analytically integrate it out as it was for λ. In section 5.4, We will discuss a strategy to avoid iteratively updating b.

When σ^2 is unknown, the non-convexity of the Gaussian negative log-likelihood is a well-known difficulty for optimization [Bühlmann and Van De Geer, 2011]. For our model, the iterative optimization algorithm appears to work well in low dimensional cases $n > p$, but often shrinks everything to zero when $n < p$. However, it appears to work very well when σ^2 is known. It is not clear why variance estimation can sometimes fail in high dimensions. Here we suggest the following simple strategy to tackle this issue.

If we treat σ_0^2 as an unknown constant in the penalized regression with separable penalty, the objective function can be re-written as

$$\arg\min_{\beta} \frac{1}{2}\|Y - X\beta\|^2 + \sigma_0^2\lambda\sum_{j=1}^{p}\mathbf{pen}(\beta_j)$$

and define $\lambda' = \sigma_0^2\lambda$ as a new penalty parameter. We assign the same prior to λ' instead of λ in (2.2). Then, if we marginalize over λ', we have a simpler objective function to (1.9) with the same penalty function.

$$\log\pi\left(\beta, b\middle| Y\right) = -\frac{1}{2}\|Y - X\beta\|_2^2 - \mathbf{pen_b}(\beta) - \log C_b + \log\pi(b) \tag{2.6}$$

where $\mathbf{pen_b}(\beta) = -\log \int \pi(\beta|\lambda')\pi(\lambda'|b)d\lambda' - \log C_b = -\log \int \pi(\beta|\lambda)\pi(\lambda|b)d\lambda - \log C_b$ and there is no parameter σ^2 in the objective function. We then follow a two-stage approach. In stage one, we find $\hat{\beta}$ and $\hat{s}$ (the number of nonzero elements in $\hat{\beta}$) by maximizing (2.6). In stage two, after finishing the optimization, we consider the variance estimator:

$$\hat{\sigma}^2 = \frac{\|Y - X\hat{\beta}\|_2^2}{n - \hat{s}}. \tag{2.7}$$

[Fan and Lv, 2011] studied the oracle properties of the non-concave penalized likelihood estimator in the high dimensional setting. [Fan et al., 2012] showed that the variance can be consistently and efficiently estimated by $\hat{\sigma}_\lambda^{2(SCAD)}$ of the form in (2.7). The main difference in our approach here is that rather than using cross-validation to determine λ', we marginalized over λ'.

2.3.2 The KKT condition and coordinate-descent algorithm

For the optimization of β, [Ročková and George, 2016a] gave the following necessary Karush-Kuhn-Tucker (KKT) condition for the global mode:

$$\beta_j = (X_j^T X_j)^{-1} \left(|z_j| - \frac{\partial \mathbf{pen}(\beta)}{\partial |\beta_j|} \right)_+ \text{sign}(z_j) \tag{2.8}$$

where $z_j = X_j^T(Y - X_{-j}\beta_{-j})$ and $(z)_+ = max(0, z)$. The derivatives of $\mathbf{pen}(\beta)$ play a crucial role here. For example, for separable priors where λ is treated as fixed, then under the Laplace priors, the solution is the traditional LASSO estimator

$$\beta_j = (X_j^T X_j)^{-1} \left(|z_j| - \sigma^2\lambda \right)_+ \text{sign}(z_j).$$

For the non-separable prior, we integrate out the hyper-parameter λ with respect to the $L_{(\frac{1}{2})^\gamma}$ prior, which gives us

$$\pi(\beta_1, ..., \beta_p|b) = \frac{(2^\gamma p - 0.5)!}{\sqrt{\pi} b 2^p [(2^\gamma)!]^p} \frac{1}{(\sum_{j=1}^p |\beta_j|^{\frac{1}{2^\gamma}} + 1/b)^{2^\gamma p + 0.5}}.$$

Then

$$\frac{\partial \mathbf{pen}_b(\beta)}{\partial |\beta_j|} = -\frac{\partial \log \pi(\beta|b)}{\partial |\beta_j|} = \frac{(2^\gamma p + 0.5)|\beta_j|^{\frac{1}{2^\gamma} - 1}}{2^\gamma (\sum_{j=1}^p |\beta_j|^{\frac{1}{2^\gamma}} + \frac{1}{b})} = \left(p + \frac{1}{2^{\gamma+1}} \right) \frac{1}{|\beta_j|^{1 - \frac{1}{2^\gamma}}} \frac{1}{\sum_{j=1}^p |\beta_j|^{\frac{1}{2^\gamma}} + \frac{1}{b}},$$

and thus we have

$$\beta_j = (X_j^T X_j)^{-1} \left(|z_j| - \frac{C_1}{|\beta_j| + C_2 |\beta_j|^{1-\frac{1}{2^\gamma}}} \right)_+ \text{sign}(z_j) \tag{2.9}$$

where we set

$$C_1 = \left(p + \frac{1}{2^{\gamma+1}} \right), \quad C_2 = \sum_{i \neq j} |\beta_i|^{\frac{1}{2^\gamma}} + \frac{1}{b}. \tag{2.10}$$

For $\gamma > 0$, the shrinkage term $\frac{\partial \mathbf{pen}_b(\beta)}{\partial |\beta_j|}$ is global-local adaptive, it depends on the information both from itself and from the other coordinates. This is consistent with the framework of global-local shrinkage priors [Polson and Scott, 2012]. While for $\gamma = 0$, the shrinkage term $\frac{\partial \mathbf{pen}_b(\beta)}{\partial |\beta_j|} = \frac{1}{\sum_{j=1}^p |\beta_j| + 1/b}$, does not have the local adaptive properties.

However, the KKT condition given by Equation (2.8) does not apply to penalties with unbounded derivatives. In fact, the set of $\hat{\beta}$ satisfying Equation (2.8) is only a superset of the solution set for our non-separable penalty. We can see that for a non-separable bridge penalty with $\gamma > 0$, the solution $\beta_j = 0$ always satisfies equation (2.9). Here, we derive the true KKT condition for our penalty, which serves as a basis for the CD algorithm.

Theorem 2.3.1. *Let the constants $\tilde{\beta}_j$ and $h(\tilde{\beta}_j)$, be such that they satisfy the following conditions*

$$\begin{aligned} 1 &= (X_j^T X_j)^{-1} C_1 \frac{1 + (1 - \frac{1}{2^\gamma}) C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}}}{\tilde{\beta}_j^2 \left(1 + C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}} \right)^2} \\ h(\tilde{\beta}_j) &= \tilde{\beta}_j + (X_j^T X_j)^{-1} \frac{C_1}{\tilde{\beta}_j + C_2 \tilde{\beta}_j^{(1-\frac{1}{2^\gamma})}}. \end{aligned} \tag{2.11}$$

Then the solutions satisfy

$$\beta_j = T(\beta_{-j}, \beta_j) = \begin{cases} 0 & \text{if } (X_j^T X_j)^{-1} |z_j| < h(\tilde{\beta}_j) \\ 0 & \text{if } (X_j^T X_j)^{-1} |z_j| \geq h(\tilde{\beta}_j) \quad \text{and} \quad \delta_{-j}(\beta_j'') > 0 \\ \text{sign}(z_j) \beta_j'' I(\beta_j \neq 0) & \text{if } (X_j^T X_j)^{-1} |z_j| \geq h(\tilde{\beta}_j) \quad \text{and} \quad \delta_{-j}(\beta_j'') = 0 \\ \text{sign}(z_j) \beta_j'' & \text{if } (X_j^T X_j)^{-1} |z_j| \geq h(\tilde{\beta}_j) \quad \text{and} \quad \delta_{-j}(\beta_j'') < 0 \end{cases} \tag{2.12}$$

where $\delta_{-j}(\beta_j) = L_{-j}(\beta_j) - L_{-j}(0)$,$\beta_j'' \in [\tilde{\beta}_j, (X_j^T X_j)^{-1} |z_j|]$, C_1 and C_2 are as given in (2.10) *and L_{-j} denotes the loss function Equation (2.6) with all except the jth β fixed.*

The proof of the theorem is provided in the section B.1, where in the step one and step two of the proof, we also show the following corollary:

Corollary 2.3.1.1. *$(X_j^T X_j)^{-1}|z_j| \geq h(\tilde{\beta}_j)$ if and only if $\beta_j^{''}$ exists and can be computed by fixed point iteration: $\beta_j^{(k+1)} = \rho(\beta_j^{(k)})$, where*

$$\rho(\beta_j) = (X_j^T X_j)^{-1}|z_j| - (X_j^T X_j)^{-1}\frac{C_1}{\beta_j + C_2\beta_j^{1-\frac{1}{2\gamma}}} \tag{2.13}$$

with the initial condition $\beta_j^{(0)} \in [\tilde{\beta}_j, (X_j^T X_j)^{-1}|z_j|]$. When $(X_j^T X_j)^{-1}|z_j| = h(\tilde{\beta}_j)$, $\beta_j^{''} = \tilde{\beta}_j$.

Remark 1: From Theorem 2.3.1, we see that the global mode is a blend of hard thresholding and nonlinear shrinkage. $h(\tilde{\beta}_j)$ is the selection threshold where $\tilde{\beta}_j$ needs to be solved numerically according to Equation (2.11). Rather than evaluating the selection threshold by solving this difficult non-linear Equation directly, we suggest selecting the variable by checking the convergence of the fixed point iterations.By corollary 2.3.1.1, we can run the fixed point iteration with an initial guess $\beta_j^{(0)} = (X_j^T X_j)^{-1}|z_j|$. If it fails to converge after a long iteration or $\rho(\beta_j^{(k)}) < 0$, we conclude that $\beta_j = 0$.

However, this is still computationally intensive. A single block of update requires running the non-parallel fixed point iteration algorithms p times sequentially. The next lemma allows us to run the fixed point iteration only on a small subset of variables. It is also a useful auxiliary result for us to show the oracle property of the estimator later.

Lemma 2.3.2. *If the regressors have been centered and standardized with $\|X_j\|^2 = n$, for $1 \leq j \leq p$, then by Equation (2.11), for $\gamma \in \mathbb{N}^+$, we always have $2\tilde{\beta}_j \leq h(\tilde{\beta}_j) \leq 3\tilde{\beta}_j$ and*

$$\tilde{\beta}_j^{(2-\frac{1}{2\gamma})} = \frac{C_1 M'(\gamma)}{X_j^T X_j}\frac{1}{\tilde{\beta}_j^{\frac{1}{2\gamma}} + C_2} = M'(\gamma)\frac{p + \frac{1}{2^{\gamma+1}}}{n\tilde{\beta}_j^{\frac{1}{2\gamma}} + nC_2}$$

for $M'(\gamma) \in (\frac{1}{2}, 1]$. If the estimator satisfies $\sum_{i=1}^{p_n} |\hat{\beta}_j|^{\frac{1}{2\gamma}} = O(s_0)$ and $\|\hat{\beta}\|_\infty < \infty$, then $\tilde{\beta}_j^{(2-\frac{1}{2\gamma})} = O(\frac{p_n}{s_0 n + n b^{-1}})$. In addition, there exists another lower bound for the selection threshold $u(\beta_{-j}) < h(\tilde{\beta}_j)$, which is not a function of $\tilde{\beta}_j$.

$$u(\beta_{-j}) = 2\left\{\frac{C_1(X_j^T X_j)^{-1}}{2C_2 + 2[(X_j^T X_j)^{-1}|z_j|]^{\frac{1}{2\gamma}}}\right\}^{\frac{1}{2-\frac{1}{2\gamma}}}. \tag{2.14}$$

The proof of Lemma 2.3.2 is in section 3 of supplementary materials.

Remark 2: Lemma 2.3.2 provides an explicit lower bound for the selection threshold $h(\tilde{\beta}_j)$, we see that $\hat{\beta}_j = 0$ if $(X_j^T X_j)^{-1}|z_j| \leq u(\beta_{-j})$. Therefore, we only need to run the fixed point iterations in $(X_j^T X_j)^{-1}|z_j| > u(\beta_{-j})$ cases. To further speed up, we suggest using the null model $\beta_{Initial} = 0$ as initialization. Then in the early stages of the updates, the lower bound (2.14) will be large, which means we only need to run fixed point iteration in a very small set of variables.

We iteratively updating $\hat{b}$ by taking $\frac{\partial \log \pi(\hat{\boldsymbol{\beta}}, b|Y)}{\partial b} = 0$, which leads to solving the nonlinear equation:

$$\frac{1}{b} = \frac{1}{2b^2} + \frac{2^\gamma p + 0.5}{2b^2 \sum_{j=1}^p |\hat{\beta}_j|^{\frac{1}{2^\gamma}+} + 2b}.$$

Since the equation above implies the inequality

$$\frac{2^\gamma p - 1.5}{2\sum_{j=1}^p |\hat{\beta}_j|^{\frac{1}{2^\gamma}}} \leq b \leq \frac{1}{2} + \frac{2^\gamma p + 0.5}{2\sum_{j=1}^p |\hat{\beta}_j|^{\frac{1}{2^\gamma}}}$$

we can approximate the solution by

$$\hat{b}^{(k)} \approx \frac{2^{\gamma-1} p}{\sum_{j=1}^p |\hat{\beta}_j{}^{(k)}|^{\frac{1}{2^\gamma}}}. \tag{2.15}$$

Remark 3: This approach can be viewed as an empirical Bayes approach as the hyper-parameter b is learned from the data. Of most concern is that the empirical Bayes estimator has the risk of a degenerate solution (e.g., setting all $\beta_j = 0$) in the high dimensional and very sparse setting, resulting in an inappropriate statement of the sparsity of the regression model [Scott and Berger, 2010, Polson and Scott, 2012, Datta et al., 2013]. Next corollary demonstrates this issue.

Corollary 2.3.2.1. *Suppose* $\sum_{i=1}^p |\beta_{0,j}|^{\frac{1}{2^\gamma}} = O(s_0)$ *and* $||\boldsymbol{\beta}_0||_\infty = E$*, using the empirical Bayesian estimator* (2.15) $\hat{b} = \frac{2^{\gamma-1} p}{\sum_{i=1}^p |\hat{\beta}_i|^{\frac{1}{2^\gamma}}}$ *for* b*, then the model will fail to recover the signal if* $\lim_{n\to\infty} \frac{p_n}{n s_0} = \infty$.

Proof. Suppose there exists estimator $\hat{\boldsymbol{\beta}}$ which can recover the signal, then $\sum_{i=1}^p |\hat{\beta}_j|^{\frac{1}{2^\gamma}} = \sum_{i=1}^p |\beta_{0,j}|^{\frac{1}{2^\gamma}} = O(s_0)$. By Lemma 2.3.2, $h(\tilde{\beta}_j) = O\left(\left[\frac{p_n}{n s_0 (1 + 1/2^{\gamma-1} p_n)}\right]^{\frac{1}{2-\frac{1}{2^\gamma}}}\right)$. Since

$\lim_{n\to\infty} \frac{p_n}{ns_0} = \infty$, $h(\tilde{\beta}_j) \to \infty$ as $n \to \infty$. This implies that the model will shrink all the elements to zero, which contradicts the assumption that $\hat{\boldsymbol{\beta}}$ can recover the signal.

This phenomenon is empirically observed in Figure 2.2. This figure shows the solution path of $\boldsymbol{\beta}$ in the same setting as Figure 2.3 in section 2.3.2. Here, seven of the ten nonzero elements have been estimated to be zero. One way to remedy this issue is to set $b \propto \frac{\log p_n}{p_n}$ (see Figure 2.3). In this case, the oracle properties of the $\hat{\beta}$ and $\hat{\sigma}^2$ are guaranteed by Theorem 2.3.5.The exact value of hyper-parameter b(determine the unknown constant) can be determined by the variable screening strategy as discuss in section B.4. Here we provide the details of our CD optimisation algorithm in Algorithm 3:

Algorithm 3 Coordinate descent (CD) optimization

Input $\epsilon > 0$; β^1; $b > 0$; $\gamma > 0$

while $||\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}^{(i-1)}|| > \epsilon$ **do**

for $j \leftarrow 1 \cdots P$ **do**

$z_j = X_j^T(Y - X_{-j}\beta^i_{-j})$

if $(X_j^T X_j)^{-1}|z_j| \leq u(\beta_{-j})$ **then**

$\beta_j^{(i+1)} = 0$

else

$\beta_j^{(i+1,0)} = (X_j^T X_j)^{-1}|z_j|$

$k = 1$

while $|\beta_j^{(i,k)} - \beta_j^{(i,k-1)}| > \epsilon \quad \text{and} \quad \beta_j^{(i,k)} > 0 \quad \text{and} \quad k \leq T$ **do**

$C_2 = ||\boldsymbol{\beta}^i_{-j}||^{\frac{1}{2\gamma}}_{\frac{1}{2\gamma}} + \frac{1}{b}$

$\beta_j^{(i,k+1)} = (X_j^T X_j)^{-1}|z_j| - (X_j^T X_j)^{-1}\frac{C_1}{\beta_j^{(i,k)} + C_2(\beta_j^{(i,k)})^{1-\frac{1}{2\gamma}}}$

$k = k + 1$

end while

if $\beta_j^{(i,k)} < 0 \quad \text{or} \quad k > T \quad \text{or} \quad L_{-j}(0) > L_{-j}(\text{sign}(z_j)\beta_j^{(i,k)})$ **then**

$\beta_j^{(i+1)} = 0$

else

$\beta_j^{(i+1)} = \text{sign}(z_j)\beta_j^{(i,k)}$

end if

end if

end for

end while

Return $\boldsymbol{\beta} = \{\beta_1^{(i)}, \ldots, \beta_p^{(i)}\}$

where ϵ is some suitable small error tolerance, T is the terminal number for the fixed point iteration, and b is the value of the hyper-parameter chosen by the user. $\boldsymbol{\beta}^i_{-j} = (\beta_1^{(i+1)}, ..., \beta_{j-1}^{(i+1)}, \beta_{j+1}^{(i)}, ..., \beta_p^{(i)})$.

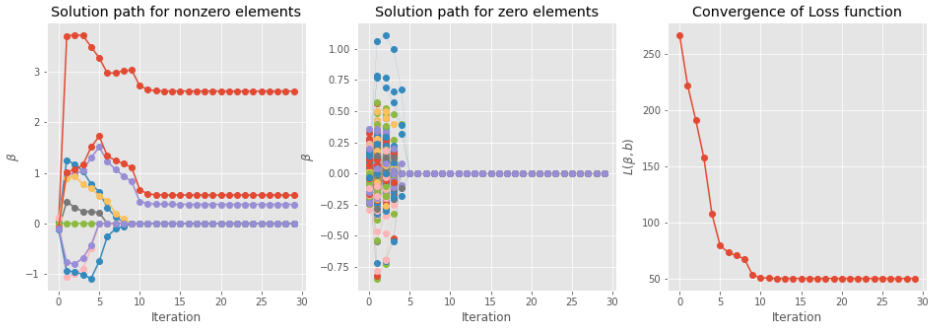


Figure 2.2: Solution paths of the CD algorithm. Data is simulated using $n = 100, p = 1000, s_0 = 10$ with iteratively update b by $\hat{b} = \frac{2^{\gamma-1}p}{\sum_{i=1}^{p} |\hat{\beta}_i|^{\frac{1}{2^\gamma}}}$. Left panel shows the solution paths for the ten non-zero elements of $\boldsymbol{\beta}$, middle panel shows the solutions paths for the 990 zero elements of $\boldsymbol{\beta}$ and the right panel shows the path for the loss function $L(\boldsymbol{\beta})$.

2.3.3 Convergence analysis

In general, convergence of the loss function alone cannot guarantee the convergence of $\left\{\boldsymbol{\beta}^i\right\}_{i\geq 1}$. [Mazumder et al., 2011] showed the convergence of CD algorithms for a subclass of non-convex non-smooth penalties for which the first and second order derivatives are bounded. The authors also observed that for the type of log-penalty proposed by [Friedman, 2012], the CD algorithm can produce multiple limit points without converging.

The derivative of the bridge penalty and its non-separable version are unbounded at zero, suggesting a local solution at zero to the optimization problem, this causes discontinuity in the induced threshed operators. Recently, [Marjanovic and Solo, 2014] proved the convergence of the CD algorithm for the bridge penalty. Here, we show that the CD algorithm with the proposed non-separable bridge penalty also converges. We provide convergence analysis in the next two theorems. Details of the proofs can be found in section B.2.

Theorem 2.3.3. *Suppose the data (Y, X) lies on compact set and the sequence $\left\{\boldsymbol{\beta}^i\right\}_{i\geq 1}$ is generated by $\beta_j^{(i+1)} = T(\boldsymbol{\beta}_{-j}^{(i)}, \beta_j^{(i)})$ where the $T(\cdot)$ map is given by equation (2.12) and $\boldsymbol{\beta}_{-j}^{(i)} = (\beta_1^{(i+1)}, ..., \beta_{j-1}^{(i+1)}, \beta_{j+1}^{(i)}, ..., \beta_p^{(i)})$, then $\|\boldsymbol{\beta}^i - \boldsymbol{\beta}^{i+1}\| \to 0$ as $i \to \infty$.*

Theorem 2.3.4. *Let $\boldsymbol{\beta}^\infty$ be the estimator obtained by the CD algorithm 3. Suppose X*

satisfies A4 with $\|\boldsymbol{\beta}^{\infty}\|_0 \leq \widetilde{p}$*, then* β^{∞} *is a strict local minimizer of* $L(\cdot)$*. In other words, for any* $\mathbf{e} = (e_1, e_2, ..., e_p) \in \mathrm{R}^{\mathrm{P}}$ *and* $\|\mathbf{e}\| = 1$

$$\lim_{\alpha \downarrow 0+} \inf_{\alpha} \left\{ \frac{L\left(\boldsymbol{\beta}^{\infty} + \alpha \boldsymbol{e}\right) - L\left(\boldsymbol{\beta}^{\infty}\right)}{\alpha} \right\} > 0.$$

Proofs for the above theorems are given in Section 7 of supplementary materials.

Empirically, we observe that convergence of the solution paths are pleasingly well behaved. Figure 2.3 demonstrates solution paths of the CD algorithm with different initialization strategies on a simulated sample of high dimensional regression problem with 990 zero entries and 10 nonzero entries in β. The design matrix X is generated with pairwise correlation between x_i and x_j equal to $0.5^{|i-j|}$. Both the response and predictors have been standardized, so $\sigma_0^2 = 1$. Data is simulated with $n = 100$ and $p = 1000$. From figure 2.3, we see that two solution paths finally converge to the same local minima with only two mistakes (they failed to identify one nonzero element and to exclude one zero element).

2.3.4 Oracle properties

In this section, we study the statistical properties of the non-separable bridge estimator. We will show that using $b \propto \frac{\log p_n}{p_n}$, model consistency and asymptotic normality of $\hat{\sigma}^2$ can be achieved.We state some additional assumptions here:

A6. $\sum_{j \in S_0} |\beta_{0j}|^{\frac{1}{2^\gamma}} = O(s_0)$

A7. There exist a constant $0 < c < \infty$ such that $P\left(\left\| \frac{X_{S_0^c}^T X_{S_0}}{n} \right\|_{2,\infty} \leq c \right) \to 1$ as $n \to \infty$

A8. $\min_{j \in S_0} |\beta_{0,j}| \geq \max \left\{ 3\sigma_0 \sqrt{\frac{2 \log s_0}{v_1 \log p_n}}, 3\sigma_0 \left(\frac{\log p_n}{n} \right)^{\frac{1}{2-\alpha}} \right\}$ where $\alpha \in (\frac{1}{2^\gamma}, 2)$

A9. $\min_{j \in S_0} |\beta_{0,j}| \geq \max \left\{ 3\sigma_0 \sqrt{\frac{2 \log s_0}{v_1 \log p_n}}, 3\sigma_0 \left(\frac{p_n}{s_0 n} \right)^{\frac{1}{2-\alpha}} \right\}$ where $\alpha \in (\frac{1}{2^\gamma}, 2)$

Theorem 2.3.5. *Suppose assumptions A1-A4, A6 to A7 are satisfied. Then under the following two cases,*

(1): $\lim_{n \to \infty} \frac{p_n}{n s_0} = 0$*,* $b \geq 1$ *or* $b \propto \frac{\log p_n}{p_n}$ *and assumption A8 holds.*

(2): $\lim_{n \to \infty} \frac{p_n}{n s_0} = \infty$*,* $b = O\left(\frac{\log p_n}{p_n} \right)$ *and assumption A9 holds.*

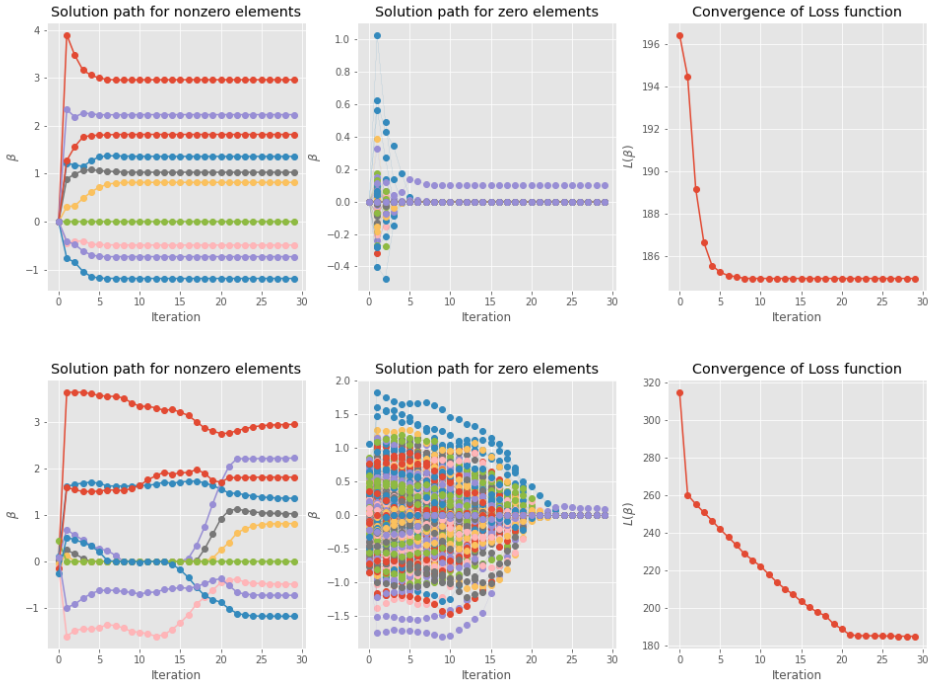


Figure 2.3: Solution paths according to two initialization strategies, $\boldsymbol{\beta}_{initial} = 0$ (top row) and random starting values $\beta_{initial} \sim N_p(0, I_p)$ (bottom row). The left panel shows the solution paths for the (10) non-zero elements of $\boldsymbol{\beta}$, the middle panel shows the solutions paths for the (990) zero elements of $\boldsymbol{\beta}$ and the right panel shows the path for the loss function $L(\boldsymbol{\beta})$.

we have:

(a) **Model consistency** *There is a strictly local minimizer* $\hat{\beta}$ *such that*

$$\#\left\{j : \hat{\beta}_j \neq 0\right\} = S_0$$

with probability tending to 1 and,

(b) **Asymptotic normality** *With the local estimator* $\hat{\boldsymbol{\beta}}$*, the correponding variance estimator* $\hat{\sigma}^2$ *from Equation (2.7) has the property that*

$$(\hat{\sigma}^2 - \sigma_0^2)\sqrt{n} \to N(0, \sigma_0^4 E[\epsilon^4] - \sigma_0^4).$$

Details of proof are given in section B.3.

2.4 Simulation study

In this section, we present some results from a set of simulation studies.

2.4.1 Comparison of PCG sampler effective sample size

In this section, we benchmark the PCG sampler scheme for $L_{\frac{1}{2}}$ prior with $\gamma = 0$ and $\gamma = 1$ against the Gibbs sampler scheme of [Park and Casella, 2008] for the Bayesian LASSO and the Horseshoe prior [Makalic and Schmidt, 2015] in the high dimension sparse regression setting. We included only these two schemes for comparison since in the high dimensional setting, very few MCMC schemes have closed form condition posterior, which are free of Metropolis-Hastings proposal.

We simulate two sets of data with moderate correlation between the covariates, ($\rho_{i,j} = 0.5^{|i-j|}$) and high ($\rho_{i,j} = 0.8^{|i-j|}$) correlation for the design matrix $\boldsymbol{X}$ with $p = 1000$ variables and sample size $n = 100$. The true vector $\boldsymbol{\beta}_0$ is constructed by assigning 10 nonzero elements $(3, 1.5, 2, 1, 1, 0.5, -0.5, 2, -1.2, -1)$ to location $(1, 2, 5, 10, 13, 19, 26, 31, 46, 51)$ and setting all the remaining coefficients to zero. We set the variance of the error to 1 ($\sigma_0^2 = 1$).

For each data set, we ran 10 MCMC chains. Each chain generated 10,000 samples after an initial burn-in of 10,000. We calculated the effective sample sizes (ESS) based on the formula from [Gelman et al., 2013]. In Table 2.1, we report the maximum, minimum, median, average and standard deviation of ESS for β_j across $p = 1000$ dimensions. Since all MCMC schemes considered here are based on the Normal mixture representations, they all need to generate from high dimensional Gaussian distributions. All three schemes have the same $O(np^2)$ computational complexity [Bhattacharya et al., 2016]. Thus, if all their conditional posterior distribution have closed-form expressions, there is not too much difference between their computational times. Thus, we only consider the indication of convergence rates, via their effective sample sizes here.

The effective sample size from the table shows that for the Bayesian LASSO, the PCG

sampler ($\gamma = 0$) performs similarly to the traditional Gibbs sampler under all criteria, where the PCG sampler achieves a slightly higher ESS overall. The performances of both samplers do not appear to be sensitive to the increase in the correlation of the design matrix.

The PCG sampler for $L_{\frac{1}{2}}$ prior ($\gamma = 1$) significantly overperforms the Horseshoe prior (using a Gibbs sampler) with respect to all the criteria. Here increasing the correlation of the design matrix has had a large effect on the ESS of both the Horseshoe Gibbs sampler and the PCG sampler. For the Horseshoe prior, the ESS has decreased by as much as 10-fold in some cases and is particularly dramatic in the non-zero elements of $\boldsymbol{\beta}$. The PCG sampler is also impacted, with the worst-case scenario being a sevenfold decrease in ESS. However, comparing PCG to the Horseshoe prior, it can be seen that the PCG sampler is up to ten times more efficient in terms of ESS.

Finally, it should be pointed out that the proposed PCG sampler for the $L_{\frac{1}{2}}$ prior has one limitation in practice. Theoretically, it works for any $\gamma \in \mathbb{N}^+$, but in our python implementation, we found that when $\gamma = 2$, numerical underflow is quite frequently encountered in the high dimension and very sparse setting. The problem worsens when $\gamma > 2$, occurs even in very simple low-dimension problems. When $\gamma \geq 2$, the $L_{\frac{1}{2}}$ prior places increasingly more probability mass around zero. In this case, the full conditional posterior of β, has a covariance matrix of the form $\Sigma = \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2\lambda^{2^{\gamma+1}}D_{\tau^2}^{-1})^{-1}$, and is nearly singular. It seems that the 64-bit floating-point numbers in the `numpy` package from Python is not enough to keep the precision, therefore we restrict our numerical experiments to the case $\gamma = 1$ only.

2.4.2 Performance of signal recovery in PCG and CD algorithms

In this section, we examine how $L_{\frac{1}{2}}$ prior recover signal under high-dimensional settings. We compare both the PCG algorithm and CD optimization algorithm with two well-known Bayesian regression procedures, the Bayesian LASSO [Park and Casella, 2008] and the Horseshoe [Carvalho et al., 2010]. We also consider the penalized likelihood

	$\rho_{i,j} = 0.5^{\|i-j\|}$					$\rho_{i,j} = 0.8^{\|i-j\|}$				
ESS	Max	Min	Median	Average	SD	Max	Min	Median	Average	SD
	Bayesian LASSO Gibbs Sampler					**Bayesian LASSO Gibbs Sampler**				
$\beta_j = 0$	100053	24030	90496	86022	11814	100669	15986	92847	89595	10356
$\beta_j \neq 0$	86707	9806	38738	40636	21737	90138	10208	24451	36125	26440
All β_j	100053	9806	90317	85568	12779	100669	10208	92715	89060	11894
	PCG sampler $\gamma = 0$					**PCG sampler** $\gamma = 0$				
$\beta_j = 0$	100831	26760	91208	87218	10949	101525	21476	92851	89802	10604
$\beta_j \neq 0$	90860	10500	54789	47604	32152	91965	10399	31069	37976	26254
All β_j	100831	10500	91018	86821	12023	101525	10399	92805	89284	12022
	Horseshoe Gibbs Sampler					**Horseshoe Gibbs Sampler**				
$\beta_j = 0$	40536	1418	18854	18178	9194	53821	866	23564	23551	13492
$\beta_j \neq 0$	6772	943	3482	3554	1427	5729	90	2840	3151	1716
All β_j	40536	943	18530	18326	9266	53821	90	23331	23347	13578
	PCG sampler $\gamma = 1$					**PCG sampler** $\gamma = 1$				
$\beta_j = 0$	90564	12298	72514	67809	15355	92276	1614	69610	66323	16082
$\beta_j \neq 0$	37852	7166	20229	20355	10002	47088	1427	6768	13483	14627
All β_j	90564	7166	72093	67334	16022	92276	1427	69380	65794	16906

Table 2.1: Effective sample size of β_j across $p = 1000$ dimensions, for two data sets generated from different degrees of correlation between variables ($\rho_{i,j} = 0.5^{|i-j|}$ and $\rho_{i,j} = 0.8^{|i-j|}$. The final MCMC sample contains 100,000 samples. The maximum, minimum, median average and standard deviation of ESS over 10 MCMC runs and β_j are reported for PCG, LASSO and Horseshoe priors.

procedures: the MC+ penalty [Zhang et al., 2010] and non-separable spike-and-slab-lasso (NSSL) [Ročková and George, 2018].

Similarly, with the previous section, we construct the AR(1) correlation structure for design matrix $\boldsymbol{X}$ with correlation $\rho_{ij} = 0.5^{|i-j|}$. In Section 2.4.2.1, we test the performance of the model in both low and high dimensional settings under the following five settings when the number of nonzero regression coefficients is $s_0 = 10$.

(1) $n = 500, p = 25, \sigma_0^2 = 3$ (2) $n = 500, p = 1000, \sigma_0^2 = 1$ (3) $n = 500, p = 1000, \sigma_0^2 = 1$

(4) $n = 100, p = 1000, \sigma_0^2 = 1$ (5) $n = 100, p = 1000, \sigma_0^2 = 3$

For each of the above scenarios, we repeated the experiment 100 times. Each time, the true vector $\boldsymbol{\beta}_0$ is constructed by assigning 10 nonzero elements $(3, 1.5, 2, 1, 1, 0.5, -0.5, 2, -1.2, -1)$ to random locations and setting all the remaining βs to zero.Similarly, in Section 2.4.2.2, we consider the small sample size and all zero settings i.e., $s_0 = 0$. In this case, $\boldsymbol{\beta}_0 = 0$ for $p = 1000$. We test two scenarios:

(1) $n = 100, p = 1000, \sigma_0^2 = 1$ (2) $n = 100, p = 1000, \sigma_0^2 = 3$.

Finally, we also consider the very challenging case of small sample size and a less sparse problem [Reid et al., 2016] in Section 2.4.2.3 with $s_0 = 20$, under the following two scenarios:

(1) $n = 100, p = 1000, \sigma_0^2 = 1$ (2) $n = 100, p = 1000, \sigma_0^2 = 3$.

The true vector $\boldsymbol{\beta}_0$ with 20 nonzero elements is constructed by duplicating the 10 nonzero elements $(3, 1.5, 2, 1, 1, 0.5, -0.5, 2, -1.2, -1)$ and assigning them to random locations and setting all the remaining βs to zero.

For all the Bayesian models, we assign a non-conjugate Jeffrey's prior to σ^2. The Bayesian LASSO and $L_{\frac{1}{2}}$ prior use the hyper-parameter setting as discussed in Section 2.1.2. For the Horseshoe prior, we assign the hyper-prior $\tau \sim C^+(0, 1)$ to the hyper-parameter as suggested by [Carvalho et al., 2010]. We use posterior means of $\boldsymbol{\beta}$ and σ^2 as the estimator. Since the MCMC cannot provide a sparse solution directly, variables are selected by a hard thresholding approach where we perform the t-test based on the posterior samples, under the null hypothesis $\beta_j = 0$ using 95% as a significant level.

For the optimization procedure, the hyper-parameter b for the NSB penalty is chosen by using the forward/backward variable screening algorithm described in Section B.4. The NSSL [Ročková and George, 2018] is claimed to be cross-validation free. The popular MC+ penalty [Zhang et al., 2010] has two tuning parameter (λ, γ).

We applied the Sparsenet algorithm [Mazumder et al., 2011] to fit the MC+ penalty, which performs cross-validation over a two-dimensional grid of values (λ, γ). For NSSL, the variance is estimated by the iterative algorithm from [Moran et al., 2018]. For NSB and MC+ penalties, we use the variance estimator from Equation (2.7).

In Tables 2.2 - 2.5, we report the simulation results by calculating the average of the following criteria from the 100 experiments: L_2 error (root mean square error), L_1 error (mean absolute error), FDR (false discover rate), FNDR (false non-discover rate), HD (Hamming distance), $\hat{\sigma}^2$ (variance estimator), $\hat{s}$ (estimated model size). We also report

	Full Bayesian procedure				**Optimization procedure**			
	$L_{\frac{1}{2}}$ ($\gamma = 1$)	$L_{\frac{1}{2}}$ ($\gamma = 2$)	Horseshoe	Bayesian LASSO	NSB ($\gamma = 1$)	NSB ($\gamma = 3$)	NSSL	MC+
$n = 500$, $p = 25$, $s_0 = 10$, $\sigma_0^2 = 3$								
L2	0.42(0.03)	**0.38**(0.02)	0.49(0.03)	0.46(0.03)	0.45(0.04)	0.40(0.02)	1.05(0.08)	**0.33**(0.02)
L1	1.64(0.11)	**1.44**(0.09)	1.92(0.13)	1.86(0.15)	1.68(0.20)	1.37(0.15)	2.67(0.41)	**0.92**(0.16)
FDR	1.30	**0.00**	4.37	3.79	50.75	38.00	**1.12**	6.68
FNDR	0.06	6.65	**0.00**	**0.00**	**0.00**	**0.00**	11.13	**0.00**
HD	**0.16**(0.05)	1.10(0.11)	0.50(0.12)	0.43(0.27)	10.42(1.41)	6.41(0.98)	2.02(0.33)	1.00(0.10)
$\hat{s}$	**10.14**	8.9	10.50	10.43	20.42	16.41	8.20	**11.00**
$\hat{\sigma}^2$	**3.01**(0.35)	3.13(0.14)	2.96(0.24)	**3.01**(0.40)	2.96(0.31)	2.97(0.20)	1.69(0.40)	**2.99**(0.21)
$n = 500$, $p = 1000$, $s_0 = 10$,$\sigma_0^2 = 1$								
L2	0.41(0.04)	–	**0.25**(0.04)	1.15(0.19)	0.15(0.02)	**0.13**(0.01)	0.14(0.02)	0.21(0.02)
L1	9.15(1.77)	–	**2.90**(0.88)	26.82(2.98)	0.37(0.06)	**0.35**(0.04)	0.37(0.08)	0.61(0.16)
FDR	4.68	–	**2.16**	47.67	0.73	**0.00**	**0.00**	10.00
FNDR	**0.00**	–	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
HD	0.51(0.10)	–	**0.20**(0.06)	10.4(1.21)	0.03(0.01)	**0.00**(0.00)	**0.00**(0.00)	3.06(0.54)
$\hat{s}$	10.51	–	**10.20**	20.40	10.03	**10.00**	**10.00**	13.06
$\hat{\sigma}^2$	0.70(0.05)	–	**0.81**(0.04)	0.09(0.02)	1.02(0.10)	**1.01**(0.05)	0.97(0.05)	1.02(0.19)
$n = 500$, $p = 1000$, $s_0 = 10$, $\sigma_0^2 = 3$								
L2	0.65(0.13)	–	**0.49**(0.08)	1.45(0.31)	0.81(0.09)	0.41(0.07)	**0.27**(0.08)	0.34(0.09)
L1	13.93(2.01)	–	**4.55**(1.25)	32.17(4.96)	3.95(0.78)	1.22(0.35)	**0.65**(0.20)	0.97(0.28)
FDR	10.73	–	**9.10**	59.16	63.87	14.54	**0.00**	10.77
FNDR	**0.00**	–	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
HD	1.29(0.20)	–	**1.01**(0.17)	15.50(2.33)	18.97(1.52)	2.66(0.49)	**0.14**(0.02)	2.91(0.53)
$\hat{s}$	11.29	–	**11.01**	25.50	28.97	12.66	**9.86**	12.91
$\hat{\sigma}^2$	2.06(0.18)	–	**2.43**(0.16)	1.20(0.15)	2.39(0.13)	2.88(0.10)	2.95(0.01)	**2.97**(0.14)

Table 2.2: Comparison of big sample size ($n = 500$) and the sparse case $s_0 = 10$ among the following priors: $L_{\frac{1}{2}}$, Bayesian LASSO, Horseshoe, non-separable bridge penalty (NSB), non-separable spike-and-slab LASSO (NSSL) and MC+.

the standard derivations of the L_2 error, L_1 error, Hamming distance and the variance estimator $\hat{\sigma}^2$ among 100 experiments in the brackets. We use bold font to highlight the best performance for each criterion among the full Bayesian procedure and optimization procedures respectively.

2.4.2.1 Sparse case with $s_0 = 10$

Tables 2.2 and 2.3 summarize the results of the simulation studies for $s_0 = 10$. In low dimensional problems ($n > p$), all approaches performed well. In high dimensional problems ($n < p$), all except the Bayesian LASSO performed well. This is within our expectation

	Full Bayesian procedure			**Optimization procedure**			
	$L_{\frac{1}{2}}$ ($\gamma = 1$)	Horseshoe	Bayesian LASSO	NSB ($\gamma = 1$)	NSB ($\gamma = 3$)	NSSL	MC+
$n = 100$, $p = 1000$, $s_0 = 10$, $\sigma_0^2 = 1$							
L2	1.19(0.21)	**0.80**(0.15)	3.86(0.66)	0.81(0.17)	0.70(0.15)	0.71(0.19)	**0.64**(0.19)
L1	14.05(2.57)	**9.40**(2.05)	40.27(5.35)	2.06(0.73)	1.65(0.37)	**1.62**(0.40)	1.80(0.41)
FDR	1.11	**1.02**	59.00	0.21	0.21	**0.20**	16.62
FNDR	0.20	0.20	**0.00**	0.10	0.10	0.10	**0.03**
HD	2.10(0.22)	**1.80** (0.19)	12.80(2.01)	1.77(0.15)	1.56(0.14)	**1.50**(0.14)	4.66(0.57)
$\hat{s}$	8.10	9.20	14.60	8.03	7.89	8.54	13.90
$\hat{\sigma}^2$	0.02(0.01)	0.10(0.02)	**0.25**(0.05)	1.30(0.10)	1.44(0.11)	**1.10**(0.11)	**1.10**(0.14)
$n = 100$, $p = 1000$, $s_0 = 10$, $\sigma_0^2 = 3$							
L2	2.01(0.25)	**1.60**(0.20)	4.09(0.90)	1.46(0.22)	1.35(0.20)	1.41(0.19)	**1.33**(0.22)
L1	18.96(2.88)	**13.6**(2.44)	43.33(5.55)	4.93(1.01)	3.91(0.41)	**3.52**(0.56)	4.46(0.59)
FDR	**6.03**	14.37	73.11	35.99	20.00	**0.80**	36.88
FNDR	0.36	**0.20**	0.44	0.19	0.20	0.30	**0.16**
HD	4.04(0.44)	**3.54**(0.41)	21.30(3.54)	6.92(1.10)	4.46(0.77)	**3.26**(0.67)	11.63(1.69)
$\hat{s}$	6.78	**9.46**	22.6	13.09	**10.00**	6.86	18.39
$\hat{\sigma}^2$	0.44(0.1)	**0.77**(0.21)	0.20(0.05)	2.49(0.30)	**3.02**(0.20)	3.68(0.41)	2.91(0.22)

Table 2.3: Comparison of small sample size ($n = 100$) and the sparse case $s_0 = 10$ among the following priors: $L_{\frac{1}{2}}$, Bayesian LASSO, Horseshoe, non-separable bridge penalty (NSB), non-separable spike-and-slab LASSO (NSSL) and MC+.

as [Castillo et al., 2015] showed that the Bayesian LASSO has poor posterior contraction rate in high dimensional problems.

One remarkable thing is the underestimation of the variance by both $L_{\frac{1}{2}}$ and Horseshoe priors when $p > n$. It was shown by [Moran et al., 2018] that using the conjugate prior to σ^2 can result in the underestimation of variance in high-dimension settings. However, as shown in Figure 2.4, when $p >> n$, using the non-conjugate prior for σ^2 will also lead to under-estimation. This is because there exists a spurious sample correlation with the realized noises in some predictors or between the predictor and response, even when they are actually independent [Fan and Lv, 2008]. As a result, the realized noises are explained by the model with extra irrelevant variables, leading to an underestimate of the variance σ^2 [Fan et al., 2012]. It can be seen that the marginal posterior distribution for $\hat{\sigma}^2$ has a heavy long tail and is biased downwards. Since the full conditional posterior of σ^2 has a InvGamma $\left(\frac{n}{2}, \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\right)$ distribution, we conclude that most of the time, the MCMC sampler of β overfits the model. Figure 2.4 also shows that when the sample size becomes large, this downward bias becomes smaller.

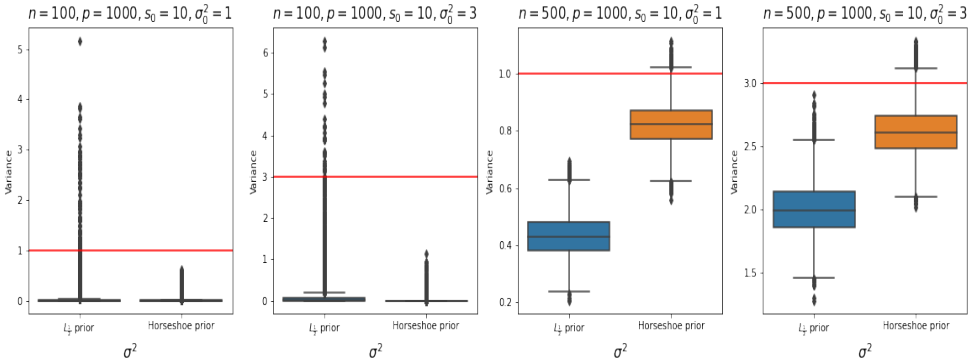


Figure 2.4: Boxplots of the marginal posterior samples of the variance parameter ($\hat{\sigma}$), under the $L_{\frac{1}{2}}$ and Horseshoe priors. The horizontal line indicates the true value of σ^2, $p > n$ cases.

Overall, among all the full Bayesian procedures, the $L_{\frac{1}{2}}$ prior with $\gamma = 2$ has the best performance in terms of L_2 error and L_1 error in the low dimensional case (Table 2.2). For the high dimensional case, the Horseshoe prior slightly outperforms the $L_{\frac{1}{2}}$ prior with $\gamma = 1$. The shrinkage of $L_{\frac{1}{2}}$ prior at $\gamma = 1$ is slightly weaker than the Horseshoe prior, consequently the variance estimates $\hat{\sigma}^2$ from Table 2.2 and 2.3 shows that while both procedures underestimate the variance parameter, the $L_{\frac{1}{2}}$ prior produces more severe underestimation. Unfortunately, the numerical underflow problem prevents us from using the $L_{\frac{1}{2}}$ prior with $\gamma = 2$ in high dimensional settings, we expect it could perform better than the Horseshoe and $L_{\frac{1}{2}}$ with $\gamma = 1$ priors, since the $L_{\frac{1}{2}}$ prior with $\gamma = 2$ has a strong shrinkage for values near zero. From Table 2.2, we see that this prior tends to select less variables and has a slightly higher value for the posterior mean of the variance parameter.

Amongst the optimization procedures, in the low dimensional case, except for the NSSL penalty (which slightly under-performs), no procedure clearly dominated across all situations for all criteria. However, when we look at the high dimensional problem, especially for the small sample size scenario ($n = 100$, Table 2.3), the performances of Bayesian and optimization procedures are comparable, except for the L_1 error where optimization procedure outperforms the full Bayesian procedure.

All optimization procedure successfully targets the true variance and have lower L_2 and

L_1 errors. It should be emphasized again that in the ultra-high dimensional setting, the hyper-parameter b in NSB penalty is determined by a cross-validation-based strategy and we believe this is the key to success. The hyperparameter in the MC+ penalty is always determined by cross-validation in all scenarios.

2.4.2.2 No signal case $s_0 = 0$

Table 2.4 shows the performance of the models under the setting with small sample size $n = 100$, $p = 1000$ and no signal ($s_0 = 0$). It is apparent that the full Bayesian procedure outperforms the optimization procedure under criteria which measure the performance of variable selection (i.e. FDR, FNDR, HD and $\hat{s}$). Both the NSB penalty and MC+ penalties provide reliable variance estimation, but this is not the case for the NSSL penalty. The NSSL selects too many variables and thus underestimates the variance.

Table 2.4 provides further evidence that, compared with the Horseshoe, the $L_{\frac{1}{2}}$ prior with $\gamma = 1$ provides relatively weaker shrinkage at values near zero. We can also see that the $L_{\frac{1}{2}}$ with $\gamma = 1$ again produces smaller variance estimates, although the under estimations are not severe in this setting. In addition, the NSB penalty with $\gamma = 1$ tends to select more variables than NSB penalty with $\gamma = 3$.

2.4.2.3 Less sparse case $s_0 = 20$

Here we consider the less sparse case with $s_0 = 20$ and we set $n = 100$, $p = 1000$ (small sample and high dimensions). We see that whereas all the algorithms in the high dimensional, small sample, but the very sparse setting ($s_0 = 10$, Table 2.3) behave reasonably well and similarly to each other, their performances deteriorate when the sample size is small and true underlying model become less sparse ($n = 100$, $p = 1000$, $s_0 = 20$), see Table 2.5.

Comparing the result from Table 2.3 and 2.5, the full Bayesian procedure shows deterioration of the estimation error (i.e. L_1, L_2 and $\hat{\sigma}^2$). The higher Hamming distance (HD) from

	Full Bayesian procedure			**Optimization procedure**			
	$L_{\frac{1}{2}}$ ($\gamma = 1$)	Horseshoe	Bayesian LASSO	NSB ($\gamma = 1$)	NSB ($\gamma = 3$)	NSSL	MC+
$n = 100$, $p = 1000$, $s_0 = 0$, $\sigma_0^2 = 1$							
L2	0.09(0.01)	**0.08**(0.01)	0.10(0.01)	0.09(0.01)	**0.07**(0.01)	0.72(0.11)	0.14(0.02)
L1	1.96(0.15)	**0.92**(0.10)	1.41(0.24)	**0.21**(0.03)	0.12(0.02)	2.78(0.32)	0.37(0.04)
FDR	**0.00**	**0.00**	**0.00**	42.00	**40.00**	100.00	99.00
FNDR	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
HD	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	3.18(0.18)	**1.42**(0.10)	13.41(1.17)	5.82(0.50)
$\hat{s}$	**0.00**	**0.00**	**0.00**	3.18	**1.42**	13.41	5.82
$\hat{\sigma}^2$	0.75(0.07)	**0.85**(0.07)	0.75(0.07)	0.92(0.03)	**0.95**(0.02)	0.60(0.03)	0.90(0.03)
$n = 100$, $p = 1000$, $s_0 = 0$,$\sigma_0^2 = 3$							
L2	0.10(0.01)	0.13(0.02)	**0.07**(0.01)	**0.13**(0.01)	0.17(0.02)	0.95(0.15)	0.20(0.03)
L1	**1.11**(0.12)	1.21(0.11)	1.94(0.28)	**0.38**(0.04)	**0.38**(0.04)	3.73(0.30)	0.66(0.05)
FDR	**0.00**	**0.00**	**0.00**	**33.33**	41.00	100.00	98.00
FNDR	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
HD	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	3.22(0.20)	**2.58**(0.23)	7.89(0.40)	7.60(0.44)
$\hat{s}$	**0.00**	**0.00**	**0.00**	3.22	**2.58**	7.89	7.60
$\hat{\sigma}^2$	2.50(0.08)	**2.74**(0.10)	2.45(0.23)	**2.80**(0.07)	2.75(0.06)	1.20(0.15)	2.64(0.10)

Table 2.4: Comparison of high dimension and small sample size problem ($p = 1000$, $n = 100$, $s_0 = 0, \sigma_0^2 = 1, 3$) when there exists no signal, among the following priors: $L_{\frac{1}{2}}$, Bayesian LASSO, Horseshoe, non-separable bridge penalty (NSB), non-separable spike-and-slab LASSO (NSSL) and MC+.

all the models indicates that neither the t-statistic test from MCMC nor the thresholding operator from optimization provides reliable variable selection. In this difficult situation, the NSB penalty with $\gamma = 3$ shows superior performance in terms of estimation. Our result is consistent with [Reid et al., 2016], who also investigated a similar less sparse scenario. They recommended using the cross-validation strategy. We followed their recommendation and proposed forward variable screening with a cross-validation strategy to determine hyper-parameter b.

	Full Bayesian procedure			**Optimization procedure**			
	$L_{\frac{1}{2}}$ ($\gamma = 1$)	Horseshoe	Bayesian LASSO	NSB ($\gamma = 1$)	NSB ($\gamma = 3$)	NSSL	MC+
$n = 100$, $p = 1000$, $s_0 = 20$, $\sigma_0^2 = 1$							
L2	4.03(0.50)	**2.06**(0.33)	6.17(1.03)	1.74(0.30)	1.33(0.27)	2.96(0.33)	**1.32**(0.25)
L1	39.01(4.05)	**20.14**(3.24)	68.54(6.71)	8.26(1.60)	**5.70**(0.81)	10.83(1.70)	6.31(0.99)
FDR	**0.00**	**0.00**	**0.00**	47.65	27.00	**0.64**	53.00
FNDR	1.70	**0.70**	1.96	0.26	0.19	1.00	**0.10**
HD	16.11(2.01)	**6.96**(1.05)	19.60(2.55)	18.56(1.88)	**8.98** (0.88)	9.86(0.90)	28.79(2.03)
$\hat{s}$	3.89	**13.04**	0.40	33.40	**25.00**	10.24	46.25
$\hat{\sigma}^2$	0.20(0.01)	0.31(0.01)	**0.55**(0.03)	1.35(0.10)	**1.29**(0.10)	2.41(0.22)	1.40(0.13)
$n = 100$, $p = 1000$, $s_0 = 20$, $\sigma_0^2 = 3$							
L2	4.64(0.60)	**2.93**(0.40)	6.17(1.35)	3.20(0.60)	**2.79**(0.31)	4.16(0.44)	3.23(0.30)
L1	49.41(5.05)	**24.58**(4.00)	70.1(6.80)	16.05(2.31)	**12.67**(2.01)	16.06(2.03)	15.98(2.44)
FDR	**0.00**	0.83	**0.00**	59.06	43.00	**0.00**	56.00
FNDR	1.75	**1.11**	1.97	0.66	**0.60**	1.32	0.66
HD	17.46(2.31)	**11.15**(1.55)	19.7(2.60)	24.46(2.32)	16.92(1.88)	**13.33**(1.67)	34.54(3.48)
$\hat{s}$	2.54	**8.84**	0.30	33.46	**24.62**	6.99	41.94
$\hat{\sigma}^2$	0.25(0.03)	0.43(0.04)	**0.89**(0.20)	**3.32**(0.33)	3.66(0.30)	3.69(0.35)	1.05(0.08)

Table 2.5: Comparison of high dimension and small sample size ($p = 1000$, $n = 100$, $s_0 = 20, \sigma_0^2 = 1, 3$) among the following priors: $L_{\frac{1}{2}}$, Horseshoe, Bayesian LASSO, non-separable bridge penalty (NSB), non-separable spike-and-slab LASSO (NSSL) and MC+.

2.5 Partially collapsed variational inference

Although the coordinate descent algorithm is much faster than MCMC and produces the sparse model directly, it only provides point estimates. Variational inference (VI) [Jordan et al., 1999, Bishop and Nasrabadi, 2006, Wainwright et al., 2008, Blei et al., 2017] is a fast optimization-based approach to perform Bayesian inference. The most common approach for VI is the mean-field assumption where the parameters are partitioned into subsets while these subsets are assumed to be independent in the posterior. There is a strong connection between Gibbs sampling and the mean-field variational inference (MFVI). As long as we can obtain the closed-form conditional posterior in the Gibbs sampling step, the approximated posterior from MFVI will fall into the same distribution family as the conditional posterior without extra assumption. The main problem for mean-field VI is that it can potentially lead to poor solutions if strong dependencies exist between parameters.

Collapsed Gibbs sampling improves upon Gibbs sampling by marginalizing some parameters. With this motivation, [Teh et al., 2006] proposed collapsed variational inference

(CVI) in the context of latent Dirichlet allocation. CVI applies VI in the same collapsed space as the collapsed Gibbs sampling CGS and hence offers a potential improvement upon the VB approach. Mathematically, it has been proven that the evidence lower bound of CVI is always tighter than that of the original VB [Teh et al., 2006]. This means CVI is always a better approximation of the true posterior than VB. One problem is that it is often difficult to conduct precise update computations for CVI, even for relatively simple Bayesian probabilistic models such as latent Dirichlet allocation (LDA) and hierarchical Dirichlet process(HDP). More specifically, taking expectations over $q_{-i}(z_{-i})$ leads to intractable computation. In this case, people often use Taylor expansion as an approximation for the integral function [Teh et al., 2006, Asuncion et al., 2012]. Although empirically, it works well, the convergence of this approach has no theoretical guarantee.

In this section, we propose a novel partially collapsed variational inference approach to model selection in Bayesian $l_{\frac{1}{2}}$ regression, which is parallel to the partially collapsed Gibbs sampling approach.

2.5.1 Coordinate ascent update

We first write down the evidence lower bound (ELBO) in different collapsed or non-collapsed spaces for the Bayesian $L_{\frac{1}{2}}$ regression. We keep the same setting as before except the hyper-prior is adjusted to $\lambda \sim \text{Gamma}(\frac{1}{2}, c)$. We denote Q_1, Q_2, Q_3 as the ELBO for three different collapsed or non-collapsed parameter spaces.

$$\begin{aligned}
\log P(Y|X) &\geq \underbrace{\mathrm{E}_{q(\beta)q(\sigma^2)q(\lambda)}\left[\log \frac{P(Y|X,\beta,\sigma^2)\pi(\beta|\lambda)\pi(\lambda)\pi(\sigma^2)}{q(\beta)q(\sigma^2)q(\lambda)}\right]}_{Q_1(q(\beta),q(\sigma^2),q(\lambda)))} \\
&\geq \underbrace{\mathrm{E}_{q(\beta)q(\sigma^2)q(v)q(\lambda)}\left[\log \frac{p(Y|X,\beta,\sigma^2)\pi(\beta|v,\lambda)\pi(v)\pi(\lambda)\pi(\sigma^2)}{q(\beta)q(\sigma^2)q(v)q(\lambda)}\right]}_{Q_2(q(\beta),q(\sigma^2),q(\lambda),q(v)))} \\
&\geq \underbrace{\mathrm{E}_{q(\beta)q(\sigma^2)q(\tau)q(v)q(\lambda)}\left[\log \frac{p(Y|X,\beta,\sigma^2)\pi(\beta|\tau^2,\lambda)\pi(\tau^2|v)\pi(\lambda)\pi(\sigma^2)}{q(\beta)q(\sigma^2)q(\tau^2)q(v)q(\lambda)}\right]}_{Q_3(q(\beta),q(\sigma^2),q(\lambda),q(\tau^2),q(v)))}
\end{aligned} \tag{2.16}$$

where the inequality is followed by the fact that the more parameters we can integrate out, the tighter the ELBO we can obtain [Teh et al., 2006]. The second bound is exact if we replace $q(v)$ to $q(v|\beta,\lambda)$ and set $q(v|\beta,\lambda) = \pi(v|\beta,\lambda)$. The third bound is exact if we replace $q(\tau^2)$ to $q(\tau^2|\beta,v,\lambda)$ and set $q(\tau^2|\beta,v,\lambda) = \pi(\tau^2|\beta,v,\lambda)$.

During coordinate ascent, we follow the same update order as PCG sampler. At $(t+1)$th iterations:

Step 1: We update $q(\lambda)$ and keep the other term fixed to maximize Q_1:

$$q_{t+1}(\lambda) \sim \text{Gamma}\left(2p + 0.5, \sum_{j=1}^{p} E_{q_t(\beta_j)}[|\beta_j|^{\frac{1}{2}}] + c\right)$$

where $E_{q_t(\beta_j)}[|\beta_j|^{\frac{1}{2}}] = \sigma_{jj}^{\frac{1}{2}} \frac{2\Gamma(3/4)}{\sqrt{\pi}} {}_1F_1\left(-\frac{1}{4}, \frac{1}{2}, -\frac{1}{2}\left(\frac{\mu_j}{\sigma_{jj}}\right)^2\right)$ and ${}_1F_1$ is confluent hypergeometric functions. Then we have

$$Q_1\left(q_{t+1}(\lambda), q_t(\beta), q_t(\sigma^2)\right) \geq Q_1(q_t\left(\lambda\right), q_t(\beta), q_t(\sigma^2)\right).$$

Step 2: We update $q(v)$ and keep the other term fixed to maximize Q_2:

$$q_{t+1}(v) \sim \prod_{j=1}^{P} \text{GIG}\left(\frac{1}{2}, 2E_{q_{t+1}(\lambda)}[\lambda^2] E_{q_t(\beta_j)}[|\beta_j|], \frac{1}{2}\right)$$

or equivalently

$$q_{t+1}\left(\frac{1}{v}\right) \sim \prod_{j=1}^{P} \text{IG}\left(\frac{1}{2}\sqrt{\frac{1}{E_{q_{t+1}(\lambda)}[\lambda^2] E_{q_t(\beta_j)}[|\beta_j|]}}, \frac{1}{2}\right)$$

where $E_{q_{t+1}(\lambda)}[\lambda^2] = \frac{(2p+0.5)(2p+1.5)}{\left[\sum_{j=1}^{p} E_{q_t(\beta_j)}[|\beta_j|^{\frac{1}{2}}] + c\right]^2}$ and $E_{q_t(\beta_j)}[|\beta_j|] = \sigma_{jj}\sqrt{\frac{2}{\pi}} {}_1F_1\left(-\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}\left(\frac{\mu_j}{\sigma_{jj}}\right)^2\right)$. Then we have $Q_2\left(q_{t+1}(v), q_{t+1}(\lambda), q_t(\beta), q_t(\sigma^2)\right) \geq Q_2\left(q_t(v), q_{t+1}(\lambda), q_t(\beta), q_t(\sigma^2)\right)$ and Q_1 is not changed in this step.

Step 3: We update $q(\tau^2)$ and keep the other term fixed to maximize Q_2:

$$q_{t+1}(\tau^2) \sim \prod_{j=1}^{P} \text{GIG}\left(\text{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right], 2E_{q_{t+1}(\lambda)}[\lambda^4] E_{q_t(\beta_j)}[\beta_j^2], \frac{1}{2}\right)$$

or equivalently

$$q_{t+1}\left(\frac{1}{\tau^2}\right) \sim \prod_{j=1}^{P} \text{IG}\left(\sqrt{\text{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right] \frac{1}{E_{q_{t+1}(\lambda)}[\lambda^4] E_{q_t(\beta_j)}[\beta_j^2]}}, \text{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right]\right)$$

where

$$\mathrm{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right] = \frac{1}{4}\frac{1}{E_{q_{t+1}(\lambda)}[\lambda^2]E_{q_t(\beta_j)}[|\beta_j|]} + \frac{1}{4}\left(\frac{1}{E_{q_{t+1}(\lambda)}[\lambda^2]E_{q_t(\beta_j)}[|\beta_j|]}\right)^{1.5}$$

and

$$E_{q_{t+1}(\lambda)}[\lambda^4] = \frac{\Gamma(2p+2.5)}{\Gamma(2p+0.5)}\frac{1}{\left[\sum_{j=1}^{p} E_{q_t(\beta_j)}[|\beta_j|^{\frac{1}{2}}] + c\right]^4}.$$

Then we have

$$Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_t(\sigma^2), q_t(\beta)\right) \geq Q_3\left(q_t(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_t(\sigma^2), q_t(\beta)\right)$$

Q_1 is not changed in this step.

Step 4: We update $q(\sigma^2)$ and keep the other term fixed to maximize Q_1, Q_2 and Q_3

$$q_{t+1}(\sigma^2) \sim \text{InvGamma}\left(\frac{n}{2}, \frac{1}{2}\mathrm{E}_{q_t(\beta)}[\|Y - X\beta\|^2]\right)$$

where $\mathrm{E}_{q_t(\beta)}[\|Y - X\beta\|^2] = \|Y - X\mu_t\|^2 + \mathrm{Tr}(X\Sigma_t X^T)$. Then we have

$$Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \geq Q_1(q_{t+1}\left(\lambda\right), q_t(\sigma^2), q_t(\beta)\right)$$
$$Q_2\left(q_{t+1}(v), q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \geq Q_2\left(q_{t+1}(v), q_{t+1}(\lambda), q_t(\sigma^2), q_t(\beta)\right)$$
$$Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \geq Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_t(\sigma^2), q_t(\beta)\right).$$

Step 5: We update $q(\beta)$ and keep the other term fixed to maximize Q_3

$$q_{t+1}(\beta) \sim \mathrm{N_p}\left(\mu_{t+1}, \Sigma_{t+1}\right)$$

where $\mu_{t+1} = \Sigma_{t+1}X^TY$, $\Sigma_{t+1} = \left(X^TX + \mathrm{E}_{q_{t+1}(\lambda)}[\lambda^4]/\mathrm{E}_{q_{t+1}(\sigma^2)}\left[\frac{1}{\sigma^2}\right]E_{q_{t+1}(\tau^2)}[D_{\tau^2}^{-1}]\right)^{-1}$
and

$$\begin{aligned}\mathrm{E}_{q_{t+1}}\left(\frac{1}{\tau^2}\right) =& \mathrm{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right]\frac{1}{E_{q_{t+1}(\lambda)}[\lambda^4]E_{q_t(\beta_j)}[\beta_j^2]} \\ &+ \left(\mathrm{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right]\frac{1}{E_{q_{t+1}(\lambda)}[\lambda^4]E_{q_t(\beta_j)}[\beta_j^2]}\right)^{1.5} \Big/ \mathrm{E}_{q_{t+1}(v_j)}\left[\frac{1}{v_j^2}\right]\end{aligned}$$

and

$$\mathrm{E}_{q_{t+1}(\sigma^2)}\left[\frac{1}{\sigma^2}\right] = \frac{n}{\mathrm{E}_{q_t(\beta)}[\|Y - X\beta\|^2]}.$$

Then we have

$$\begin{aligned} Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_{t+1}(\beta)\right) &\geq Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_{t+1}(\beta)\right) \\ &\geq Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \end{aligned}$$

where the first inequality is followed by inequality (2.16). Similarly, we also have

$$Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \geq Q_3\left(q_{t+1}(\tau^2), q_{t+1}(v), q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right).$$

However, this doesn't imply that $Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_{t+1}(\beta)\right) \geq Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right)$. If the above inequality is true, the PCVI will improve or maintain the ELBO in the collapsed space at each iteration. In other words, we will always have

$$Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_{t+1}(\beta)\right) \geq Q_1\left(q_t(\lambda), q_t(\sigma^2), q_t(\beta)\right).$$

A simple way to fix this issue is that, after updating $q(\beta)$, we check the difference of the Q_1 directly:

$$\begin{aligned} &Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_{t+1}(\beta)\right) - Q_1\left(q_{t+1}(\lambda), q_{t+1}(\sigma^2), q_t(\beta)\right) \\ =&\frac{n}{2}\left(1 - \frac{\mathrm{E}_{q_{t+1}(\beta)}[\|Y - X\beta\|^2]}{\mathrm{E}_{q_t(\beta)}[\|Y - X\beta\|^2]}\right) + (2p + 0.5)\left(1 - \frac{\sum_{j=1}^{p} E_{q_{t+1}(\beta_j)}[|\beta_j|^{\frac{1}{2}}] + c}{\sum_{j=1}^{p} E_{q_t(\beta_j)}[|\beta_j|^{\frac{1}{2}}] + c}\right) + \frac{1}{2}\log\frac{|\Sigma_{t+1}|}{|\Sigma_t|}. \end{aligned} \tag{2.17}$$

If the update of $q(\beta)$ fails to improve Q_1, we will reject this update and then the iteration will terminate since the update of the other approximated posteriors relies on the update of $q(\beta)$. The variable selection is done by calculating the expectation of pseudo-inclusion probabilities under approximated posterior. We write

$$\hat{\gamma_j} = E_{q(\lambda)q(\tau_j^2)}\left[\frac{1}{1 + \tau_j^2/\lambda^4}\right]. \tag{2.18}$$

The decision rule is then: β_j is nonzero if $\hat{\gamma_j} < \delta$ and zero otherwise, where δ is user-defined threshold.

2.5.2 Computation strategies

There are two main computational burdens for our PCVI in Bayesian $L_{\frac{1}{2}}$ regression:

- Evaluating equation (2.17) requires calculating $\log|\Sigma_t|$ at each iteration, which has $O(p^3)$ computational complexity.
- Evaluating Σ_t involves inverse a non-sparse $p \times p$ matrix, which also has $O(p^3)$ complexity. By using Woodbury matrix identity, the inversion of an $p \times p$ matrix can be transformed to the inversion of a $n \times n$ matrix. However, this strategy is not useful when both p and n are very large.

In practice, we found that even if we don't evaluate the terminate condition (2.17), the algorithm converges very well. Thus, we use $\|\mu_{t+1} - \mu_t\| \leq \epsilon$ as terminate condition instead. Intuitively, we believe that we are optimizing an implicit ELBO bound Q_{PCG}, such that

$$Q_1(q(\beta), q(\sigma^2), q(\lambda))) \geq Q_{\text{PCG}}(q(\beta), q(\sigma^2), q(\lambda), q(\tau^2), q(v))) \geq Q_3(q(\beta), q(\sigma^2), q(\lambda), q(\tau^2), q(v))). \tag{2.19}$$

For the problem of inverting a non-sparse $p \times p$ matrix in evaluating Σ_t, this is the price we pay to keep the dependence structure of approximated posterior of $\boldsymbol{\beta}$ no matter what sparse prior we use. Unless we further assume $q(\beta) = \prod_{j=1}^{p} q_j(\beta_j)$ as previous works [Carbonetto and Stephens, 2012, Huang et al., 2016, Ray and Szabó, 2021]. Then the joint update of Step 5 will be replaced with p component-wise update and Σ_t will be diagonal. In this case, the cost for evaluating equation (2.17) is also cheap.

Here, we propose a two-stage approach to speed up the computation. This idea is motivated by [Narisetty et al., 2018, Johndrow et al., 2020], who partition $\boldsymbol{\beta}$ into the active set and inactive set in MCMC scheme. In stage one, we perform PCVI by assuming that $q(\beta) = \prod_{j=1}^{p} q_j(\beta_j)$. The variable selection is made by using the criteria 2.18. In stage two, we perform PCVI again by assuming that $q(\beta) = q(\beta_{\hat{\mathcal{A}}}) \prod_{j \in \hat{\mathcal{A}}^c} q(\beta_j)$, where $\hat{\mathcal{A}}$ is the active set based on the variable selection in stage one. The computation complexity of our two-stage approach is $O(\max(|\hat{\mathcal{A}}|, p)n)$, which is similar to the coordinate descent algorithms for Lasso and Elastic Net [Friedman et al., 2010].

It is reasonable to carry out the variable selection in stage one. The mean field assumption of $q(\beta) = \prod_{j=1}^{p} q_j(\beta_j)$ may tend to underestimate posterior variances but it provides

good estimates of mean parameters. [Zhang et al., 2016] performed a comparison study on MCMC and variational algorithm for a linear model and show that the mean field variational inference reduces the computational cost without compromising accuracy in both the variable selection and the estimation of the nonzero coefficients. In addition, we only require the variable selection in stage one to satisfy $\hat{\mathcal{A}} \supseteq \mathcal{A}_0$ to partition $\boldsymbol{\beta}$. After stage two, a more refined variable selection approach based on a t-test can be carried out.

2.5.3 Simulation study and discussion of the future work

Simulation study

In the following numerical examples, we consider simulated data with $p = 1000$ and $n = 100$ following the setting from [Ročková and George, 2018] to assess the performances of the PCVI, PCG sampler with $\gamma = 1$, non-separable bridge penalty with $\gamma = 1$, SparseNet algorithm [Mazumder et al., 2011] for MCP penalty [Zhang et al., 2010] and horshshoe [Carvalho et al., 2010]. We generate a data matrix X from a multivariate Gaussian distribution with mean zero and a block diagonal covariance matrix $\boldsymbol{\Sigma} = \text{bdiag}(\widetilde{\Sigma}, \ldots, \widetilde{\Sigma})$, where $\widetilde{\Sigma} = (\tilde{\sigma}_{ij})_{i,j=1}^{50}$ with $\tilde{\sigma}_{ij} = \rho$ if $i \neq j$ and $\tilde{\sigma}_{ii} = 1$. The true vector $\boldsymbol{\beta}_0$ is constructed by assigning regression coefficients $\frac{1}{\sqrt{3}}\{-2.5, -2, -1.5, 1.5, 2, 2.5\}$ to $q = 6$ entries located at $\{1, 51, 101, 151, 201, 251\}$ and setting to zero all the remaining coefficients. In the experiment, we consider both $\rho = 0.9$ and $\rho = 0.6$. We report the simulation results by calculating the average of the following criteria from the 100 repetitions: MSE (mean squared error), FDR (false discovery rate), FNR (false non-discovery rate), HD (Hamming distance), $\hat{s}$ estimated model size) and $\hat{\sigma}^2$ (variance estimate). The result is reported in Table 2.6. We see that the PCVI performs similarly to the horseshoe and PCG sampler. But both of them underestimated the variance σ^2, this is similar to what we observed in a previous simulation study section.

In figure 2.5 and 2.6, we provide a marginal posterior density plot from PCG sampler, the exact Bayesian inference, and PCVI, the approximated Bayesian inference. This time, we modified the simulation set up to $p = 3000$, $n = 200$ and used $\rho = 0.6$ because in this case,

	$\rho = 0.9$						$\rho = 0.6$					
	MSE	FDR	FNR	$\hat{s}$	HAM	$\hat{\sigma}^2$	MSE	FDR	FNR	$\hat{s}$	HAM	$\hat{\sigma}^2$
PCVI	3.22	0.261	0.423	7.1	3.98	0.01	1.10	0.24	0.18	7.2	2.13	0.021
PCG	3.23	0.251	0.427	6.9	3.90	0.015	1.09	0.21	0.14	7.0	2.09	0.028
NSB	3.16	0.260	0.2	6.0	3.10	1.05	0.32	0	0.04	5.8	0.4	1.01
SSL	3.32	0.255	0.257	6.0	3.07	1.07	1.30	0	0.225	4.90	1.36	1.05
Horseshoe	3.21	0.250	0.407	4.63	3.70	0.021	1.07	0.206	0.114	6.71	2.07	0.032
MCP	8.40	0.767	0.580	10.60	11.50	0.20	0.40	0.095	0.04	6.45	0.80	1.1

Table 2.6: Simulation study using 100 repetitions under high dimensional sparse regression setting with $n = 100$,$p = 1000$, $s_0 = 6$, $\sigma_0^2 = 1$

.

both two approaches can perfectly identify the signal and noise. Then it is easy for us to compare the shape of their marginal posterior density under the noise and signals. We see that the PCVI algorithm slightly underestimated the variance of the posterior. When β is nonzero, both of their marginal posteriors can cover the oracle value.

Discussion of the future work

Finally, we would like to discuss some unsolved problems for this research. The PCVI algorithm for $L_{\frac{1}{2}}$ prior is not limited to the linear regression model. For the more general model setting, $q(\boldsymbol{\beta}) \propto \exp\left\{\mathbb{E}_{q(\boldsymbol{\lambda})q(\tau^2)}\left[\log \pi\left(\boldsymbol{\beta} \mid \boldsymbol{\tau}^2, \lambda, Y\right)\right]\right\}$ may loose the closed form expression. We need to further restrict $q(\boldsymbol{\beta})$ to be Gaussian. If the dimension of the model is high, we may assume $q(\beta) = \prod_{j=1}^{p} q_j(\beta_j)$. Updating the variational parameters of $q(\boldsymbol{\beta})$ is done by a gradient-based learning algorithm. However, in this case, we lose the convergence guarantee of the algorithm. The convergence of the PCVI in the sense of loss function in a linear regression setting is guaranteed by keep checking the criteria 2.17, which has closed form at the end of each iteration. In general, the criteria $Q_1\left(q_{t+1}(\lambda), q_{t+1}(\beta)\right) - Q_1\left(q_{t+1}(\lambda), q_t(\beta)\right)$ is intractable due to complexity of the likelihood. In our preliminary numerical experiments, we found that we can also use $\|\mu_{t+1} - \mu_t\| \leq \epsilon$ to replace criteria 2.18 both in linear regression and logistic regression setting, which implies that there may exist a stronger convergence result.

From a broader perspective, partially collapsed variational inference may provide a way to extend the mean-field family if we can show that there exists an implicit ELBO such as

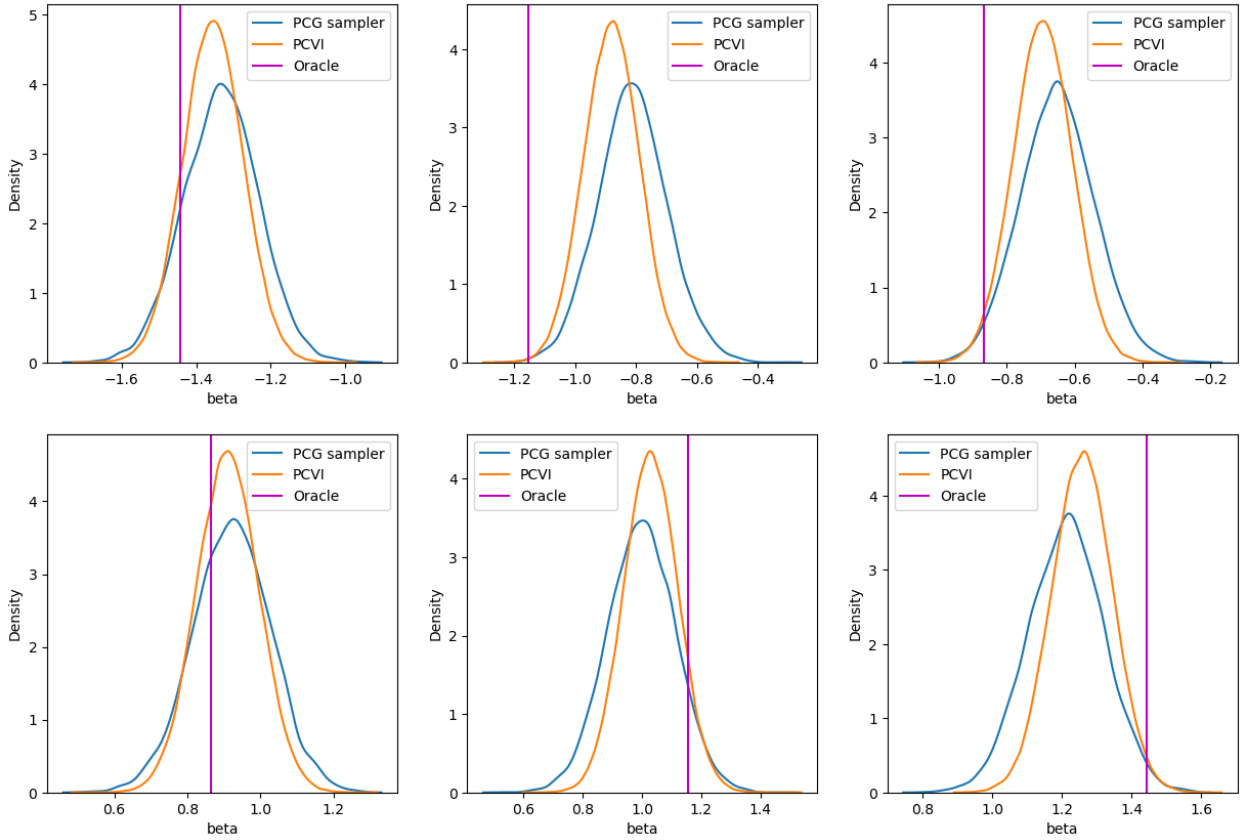


Figure 2.5: Marginal posterior density plot of β under $p = 3000$, $n = 200$ and $\rho = 0.6$, when $\beta_{Oracle} \neq 0$

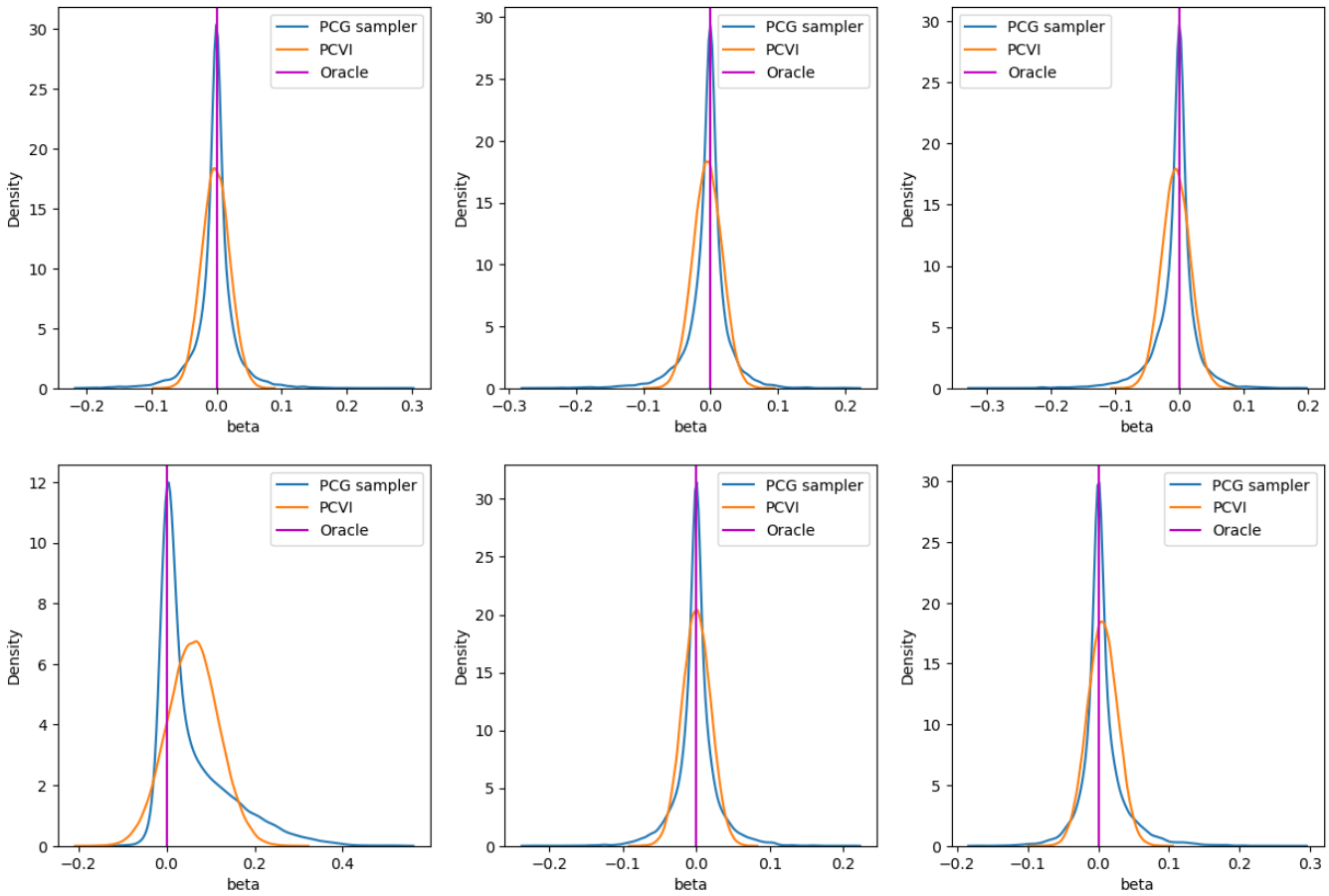


Figure 2.6: Marginal posterior density plot for β indexed from 51 to 56 under $p = 3000$, $n = 200$ and $\rho = 0.6$

(2.19). Then we have the theoretical guarantee that checking criteria 2.17 is not necessary. As pointed out by [Park and Van Dyk, 2009], PCG sampler is a stochastic counterpart to the ECME [Liu and Rubin, 1994] and AECM [Meng and Van Dyk, 1997] algorithms, which use different data augmentation schemes (different marginal posterior distributions) in the conditional maximization steps of the ECM [Meng and Rubin, 1993] algorithm. PCVI may be viewed as a generalization of ECME and AECM algorithms, where the latter use point mass approximation to different marginal posterior distributions.

Another unsolved problem is the lack of theoretical study for the contraction rates for the approximated posterior from PCVI to a sparse truth in ℓ_2 loss and means square prediction error. [Ray and Szabó, 2021] showed the posterior contraction rates of mean-field variational Bayes for high-dimensional linear regression under discrete spike-and-slab prior. At the same time, [Bai et al., 2020b] showed a similar result for the Normal Inverse-Gamma mixture prior. There are three challenges for us to show this result:

- Both of [Ray and Szabó, 2021, Bai et al., 2020b] assumed a fixed variance, but in our setting, the variance is unknown.
- We assign a hyper-parameter to the global shrinkage parameter. So far as we know, under a linear regression setting, the theoretical study of the fully Bayesian approach for global-local shrinkage prior doesn't exist.
- It is pretty hard to extend current the posterior consistency result for mean field variational inference [Yang et al., 2020] to partially collapsed variational inference.

Chapter 3

Scalable inference for Bayesian ordinal linear mixed regression: application to student evaluation of teaching survey data

In this chapter, we consider the analysis of a real dataset involving a large number of surveys from students' evaluation of teaching (SET). We first extend the Bayesian $L_{\frac{1}{2}}$ prior for variable selection to ordinal response data, using the cumulative logistic link, and where the model also contains random-effects parameters. We then provide some results from the real data analysis and conclude with some discussions.

3.1 Cumulative logit regression model

We consider the Bayesian cumulative logit model with random effects. The cumulative logistic model [Agresti, 2001] is perhaps the most commonly used model linking the ordered categorical response variable to a set of covariates, such models can be fitted using both

frequentist and Bayesian methods, see for instance [Zhang and Archer, 2021]. Alternative link functions are also possible such as the seminal work in [Albert and Chib, 1993] using the probit link. Here we work with the cumulative logistic model for its relative ease of interpretation using cumulative log-odds. In addition, we would like to have a model combined with a variable selection method that results in understanding which factors significantly impact the rating of the lecturers.

Denote the $N \times 1$ vector of ordinal response by $\boldsymbol{Y} = (y_1, ..., y_n)'$ and introduce a vector of latent variable $\boldsymbol{Z} = (z_1, ..., z_n)'$ where

$$z_i \sim logistic(\boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{T}_i\boldsymbol{b}, 1) \tag{3.1}$$

where $\boldsymbol{X}_i$ is the ith entry of an $n \times p$ covariate matrix corresponding to the p fixed effects and $\boldsymbol{T}_i$ is the ith entry of an $n \times q$ covariate matrix corresponding to $q \times 1$ vector random effects parameters $\boldsymbol{b}$. The latent variable is related to the observed ordinal response via

$$y_i = \begin{cases} 1, & \text{if } z_i \leq \gamma_1 \\ s, & \text{if } \gamma_{s-1} < z_i \leq \gamma_s, \quad \text{for } s = 2, \ldots, S-1 \\ S, & \text{if } z_i > \gamma_{S-1} \end{cases}$$

where S is the number of ordinal response categories, and $\gamma = (\gamma_1, ..., \gamma_{S-1})$ are unknown cutoff points.

The cumulative probabilities take the form

$$P(y_i \leq s) = P(z_i \leq \gamma_s | \boldsymbol{X}, \boldsymbol{T}) = \frac{\exp(\gamma_s - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{T}_i\boldsymbol{b})}{1 + \exp(\gamma_s - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{T}_i\boldsymbol{b})}$$

and the cumulative logit model is of the form

$$\log\left(\frac{p(y_i \leq s|\boldsymbol{X}, \boldsymbol{T})}{p(y_i > s|\boldsymbol{X}, \boldsymbol{T})}\right) = \gamma_s - \boldsymbol{X}_i\boldsymbol{\beta} - \boldsymbol{T}_i\boldsymbol{b}, \quad s = 1, \ldots, S-1$$

where the above expression provides a log-odds interpretation for the fixed parameters $\boldsymbol{\beta}$.

3.2 Prior specification

For the fixed effect parameters $\boldsymbol{\beta}$, we consider the use of exponential power prior with the form

$$\pi(\boldsymbol{\beta}|\lambda,\alpha) \propto \prod_{j=1}^{p} \frac{\lambda^{1/\alpha}}{2\Gamma(1+1/\alpha)} \exp\left(-\lambda\left|\beta_j\right|^{\alpha}\right), \quad 0<\alpha<1. \tag{3.2}$$

This is the bridge prior [Knight et al., 2000,Polson et al., 2014]. Here we work with $\alpha = \frac{1}{2}$ for the parameters $\boldsymbol{\beta}$, we refer to this as the $L_{\frac{1}{2}}$ prior. It is possible to set a hyperprior for λ to automatically adjust for the sparsity level. In Chapter 2, we proposed to set

$$\frac{1}{\sqrt{\lambda}} \sim \text{Cauchy}_{+}(0,1),$$

as a fully Bayesian approach. We can also tune it manually. For the prior of the cutoff points $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_{S-1})$, we have that for $s = 1, \ldots, S$, $p_s = P(\gamma_{s-1} < z < \gamma_s) = F(\gamma_s) - F(\gamma_{s-1})$, and $\sum_{s=1}^{S} p_s = 1$, so that

$$\gamma_1 = F^{-1}(p_1), \quad \gamma_2 = F^{-1}(p_1+p_2) \quad \ldots, \gamma_{S-1} = F^{-1}(p_1+p_2+\cdots+p_{S-1}),$$

where F is the CDF of the logistic distribution. We can assign a symmetric Dirichlet prior to $(p_1, p_2, ..., p_S)$, with concentration parameter $a > 0$,

$$\pi(p_1, \ldots, p_S|a) = \frac{\Gamma(aS)}{\Gamma(a)^S} \prod_{s=1}^{S} p_s^{a-1}$$

and by change of variable from the equation above, we have

$$\pi(\gamma_1, \gamma_2, ..., \gamma_{S-1}|a, v) = \frac{\Gamma(aS)}{\Gamma(a)^S} \prod_{s=1}^{S} [F(\gamma_s) - F(\gamma_{s-1})]^{a-1} \prod_{s=1}^{S-1} f(\gamma_s)$$

where $f(\cdot)$ is the density of $F(\cdot)$.

Finally for the random effects $\boldsymbol{b}$, we set the prior for $\boldsymbol{b} \sim N(0, \boldsymbol{\Lambda}^{-1})$ with unknown precision matrix $\boldsymbol{\Lambda}^{-1}$. We assign a conjugate prior to $\boldsymbol{\Lambda}$ such that

$$\boldsymbol{\Lambda} \sim \text{Wishart}(\delta, \boldsymbol{P}^{-1})$$

where $\boldsymbol{P}$ is the precision matrix. We used $\delta = 2$ and the identity matrix for $\boldsymbol{P}$ throughout the analysis. The model is largely insensitive to the choices of δ and $\boldsymbol{P}$.

3.3 A partially collapsed Gibbs sampler

We extend the Gaussian linear regression model presented in Chapter 2 to the Bayesian ordinal model with random effects and cumulative logistic link. To derive the PCG sampler, we first require appropriate choices of prior decomposition. Here, we adopt the following decomposition for the $L_{\frac{1}{2}}$ prior on the fixed effects in Equation 3.2, as follows:

$$\boldsymbol{\beta}|\tau_1^2,\ldots,\tau_p^2 \sim N_p\left(\mathbf{0}, \frac{1}{\lambda^4}\boldsymbol{D}_{\tau^2}\right), \quad \boldsymbol{D}_{\tau^2} = \text{diag}\left(\tau_1^2,\ldots,\tau_p^2\right)$$
$$\tau_1^2|v_1^2,\ldots,\tau_p^2|v_p^2 \sim \prod_{k=1}^{p} \text{Exp}\left(\frac{1}{2v_k^2}\right), \quad v_1,\ldots,v_p \sim \prod_{k=1}^{p} \text{Gamma}(\frac{3}{2},\frac{1}{4})$$

where $\boldsymbol{D}_{\tau^2}$ is a diagonal matrix, $\tau_1^2,\ldots,\tau_p^2,\nu_1^2,\ldots,\nu_p^2$ are latent parameters introduced to facilitate the decomposition, when marginalised over these parameters, the prior reverts to the form in (3.2). This Normal-Exponential-Gamma mixture representation allows us to obtain simple, full conditionals that are easy to sample from using the Gibbs sampler.

For latent variables assumed to come from a logistic distribution as in 3.1, it is still not possible to easily obtain conjugate updates in the Gibbs sampler even with the above decomposition. However, [Pingel, 2014] showed that the logistic distribution can be well approximated by a t-distribution, using the degree of freedom $\nu = 6.4$ and scale parameter $\eta = 1.539$ to approximate the standard logistic distribution resulting in an error of 0.0006 for the CDF and 0.0007 for the PDF under the L_∞ norm. Note that of course, in setting up the ordinal model, we are not restricted to using the logistic distribution, we can of course also directly use the t-distribution to model the latent variable z, however, we have chosen to work with the logistic model, since it has better interpretability.

Utilising the Normal-Gamma mixture representation of the t-distribution, we can now express the latent variable z_i as

$$z_i|\boldsymbol{\beta},\boldsymbol{b},w_i \sim N(x_i\boldsymbol{\beta}+T\boldsymbol{b},w_i^{-1}), \quad w_i \sim \text{Gamma}\left(\frac{\nu}{2},\frac{\nu\eta^2}{2}\right)$$

this is then used in deriving the condition posterior for $\pi(\boldsymbol{\beta}|\boldsymbol{b},\boldsymbol{\tau}^2,\boldsymbol{Z})$. Figure 3.1 graphically illustrates the Bayesian ordinal model and the dependence structure of all the parameters and latent variables.

The PCG sampler is given below:

$$
\begin{aligned}
&\text{Sample } \boldsymbol{\beta}|\boldsymbol{b}, \boldsymbol{W}, \boldsymbol{\tau}^2, \boldsymbol{Z}, \lambda \sim \mathrm{N_p}((\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + \lambda^4\boldsymbol{D}_{\boldsymbol{\tau}^2}^{-1})^{-1}\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{Z} - \boldsymbol{T}\boldsymbol{b}), (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + \lambda^4\boldsymbol{D}_{\boldsymbol{\tau}^2}^{-1})^{-1})\\
&\text{Sample } \boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{W}, \boldsymbol{\Lambda} \sim N_q((\boldsymbol{T}^T\boldsymbol{W}\boldsymbol{T} + \boldsymbol{\Lambda})^{-1}\boldsymbol{T}^T\boldsymbol{W}(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta}), (\boldsymbol{T}^T\boldsymbol{W}\boldsymbol{T} + \boldsymbol{\Lambda})^{-1})\\
&\text{Sample } \boldsymbol{\Lambda}|\boldsymbol{b} \sim \text{Wishart}(t+1, (\boldsymbol{P} + \boldsymbol{b}\boldsymbol{b}^T)^{-1})\\
&\text{Sample } \lambda|\beta \sim \text{Gamma}(2p + 0.5, \sum_{k=1}^{p} |\beta_k|^{\frac{1}{2}} + \frac{1}{\phi})\\
&\text{Sample } \phi|\lambda \sim \text{InvGamma}(1, 1+\lambda)\\
&\text{Sample } \frac{1}{v_k}|\beta_k, \lambda \sim \text{InverseGaussian}\left(\sqrt{\frac{1}{4\lambda^2|\beta_k|}}, \frac{1}{2}\right), \quad k = 1, ..., p\\
&\text{Sample } \frac{1}{\tau_k^2}|\beta_k, v_k, \lambda \sim \text{InverseGaussian}\left(\frac{1}{\lambda^2 v_k|\beta_k|}, \frac{1}{v_k^2}\right), \quad k = 1, ..., p\\
&\text{Sample } z_i|y_i, \boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\gamma} \sim \text{Logistic}(x_i\boldsymbol{\beta} + T_i\boldsymbol{b}, 1)1_{\gamma_{s-1} < z_i < \gamma_s}, \text{ if } y_i = s, \text{ for } i = 1, ..., n\\
&\text{Sample } w_i|z_i, \boldsymbol{\beta}, \boldsymbol{b} \sim \text{Gamma}\left(\frac{\nu}{2}, \frac{\nu\eta^2 + (z_i - x_i\boldsymbol{\beta} - T_i\boldsymbol{b})^2}{2}\right), \quad i = 1, ..., n\\
&\text{Sample } \gamma_s \sim \pi(\gamma_s|\gamma_{-s}, \boldsymbol{Y}, \boldsymbol{Z}) \propto f(\gamma_s)1_{l_s < \gamma_s < u_s}
\end{aligned}
\tag{3.3}
$$

Remarks:

1. By using the t-distribution and the Normal-Gamma decomposition to approximate the logistic distribution $\pi(\boldsymbol{Z}|\boldsymbol{b}, \boldsymbol{\beta})$, we have $\pi(\boldsymbol{\beta}|\boldsymbol{W}, \boldsymbol{\tau}^2, \boldsymbol{Z}, \lambda) \propto \pi(z_i|\boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{W})\pi(\boldsymbol{\beta}|\boldsymbol{\tau}^2, \lambda)$ and $\pi(\boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{W}, \boldsymbol{\Lambda}) \propto \pi(\boldsymbol{Z}|\boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{W})\pi(\boldsymbol{b}|\lambda)$. Since all the terms on the right hand side are Gaussian distributions, we have

$$\pi(\boldsymbol{\beta}|\boldsymbol{b}, \boldsymbol{W}, \boldsymbol{\tau}^2, \boldsymbol{Z}, \lambda) \sim \mathrm{N_p}((\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + \lambda^4\boldsymbol{D}_{\boldsymbol{\tau}^2}^{-1})^{-1}\boldsymbol{X}^T\boldsymbol{W}(\boldsymbol{Z} - \boldsymbol{T}\boldsymbol{b}), (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + \lambda^4\boldsymbol{D}_{\boldsymbol{\tau}^2}^{-1})^{-1})$$

and

$$\pi(\boldsymbol{b}|\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{W}, \boldsymbol{\Lambda}) \sim N_q((\boldsymbol{T}^T\boldsymbol{W}\boldsymbol{T} + \boldsymbol{\Lambda})^{-1}\boldsymbol{T}^T\boldsymbol{W}(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta}), (\boldsymbol{T}^T\boldsymbol{W}\boldsymbol{T} + \boldsymbol{\Lambda})^{-1}).$$

2. The conditional posterior

$$\pi(z_j|\boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{Y}) \propto \pi(y_i|z_i, \boldsymbol{\gamma})\pi(z_i|\boldsymbol{\beta}, \boldsymbol{b}) \propto \pi(z_i|\boldsymbol{\beta}, \boldsymbol{b})1_{\gamma_{s-1} < z_i < \gamma_s}, \text{ if } y_i = s.$$

This is a univariate truncated logistic distribution for the full conditional posterior, and therefore can be easily sampled directly by inversion of CDF method.

3. The conditional posterior

$$\begin{aligned}\pi(\gamma_s|\gamma_{s-1},\boldsymbol{\gamma}_{s+1},\boldsymbol{Z},\boldsymbol{Y}) &\propto \pi(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{\gamma})\pi(\gamma_s|\gamma_{s-1},\gamma_{s+1}) \\ &\propto [F(\gamma_s)-F(\gamma_{s-1})]^{a-1}[F(\gamma_{s+1})-F(\gamma_s)]^{a-1}f(\gamma_s)1_{l_s<\gamma_s<u_s}.\end{aligned}$$

Note that

$$\pi\left(c=\frac{F\left(\gamma_s\right)-F\left(\gamma_{j-1}\right)}{F\left(\gamma_{j+1}\right)-F\left(\gamma_{j-1}\right)}\mid\gamma_{j-1},\gamma_{j+1}\right)\sim\text{Beta}(a,a)1_{c_1<c<c_2}$$

where $c_1=\frac{F(l_s)-F(\gamma_{j-1})}{F(\gamma_{j+1})-F(\gamma_{j-1})}$ and $c_2=\frac{F(u_s)-F(\gamma_{j-1})}{F(\gamma_{j+1})-F(\gamma_{j-1})}$. To sample $\pi(\gamma_s|\gamma_{s-1},\gamma_{s+1},\boldsymbol{Z},\boldsymbol{Y})$, we first sample c from truncated Beta distribution above and then get $\gamma_s=F^{-1}[F(\gamma_{j-1})+c(F\left(\gamma_{j+1}\right)-F\left(\gamma_{j-1}\right))]$.

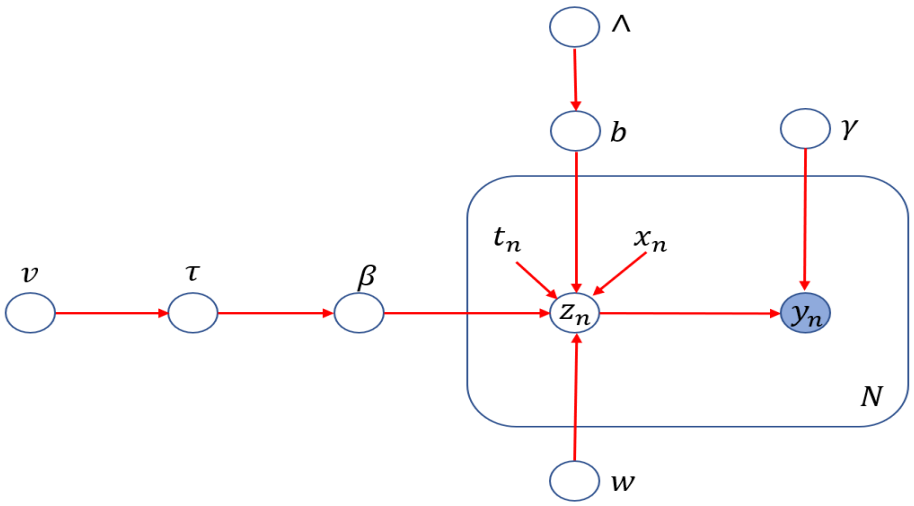


Figure 3.1: A graphical representation of the Bayesian ordinal regression model. The box labelled N represents N nodes of which only a single variable of t_n,x_n,z_n and y_n are shown explicitly. The shaded circle correspond to the response variable y_n.

3.4 Numerical experiment

3.4.1 Simulated data analysis

In this section, we will test the performance of $L_{\frac{1}{2}}$ prior with a simulated data set. For the simulated data set, the number of covariates is set to be $p = 500$, the number of the random effect is set to be $q = 100$, and the sample size $n = 360$. We generated ordinal response Y by assuming an underlying continuous random variable Z, which was generated using the following equation:

$$Z = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{T}b + \boldsymbol{\epsilon}$$

where X is the $n \times p$ covariate matrix for the fixed effect, $\boldsymbol{T}$ is $n \times q$ covariate matrix for the random effect and $\boldsymbol{\epsilon}$ follows an indepdenent standard logistic distribution.

The random effect b is generated by $N(0, \Sigma_1)$ with covariance matrix Σ_1 has AR(1) correlation structure such that $\sigma_{ij} = 0.5^{|i-j|}$.The elements in covariate matrix $\boldsymbol{T}$ is generated by i.i.d $N(0, \frac{1}{q})$. This construction guarantees the boundedness of the variance from $\boldsymbol{T}b$.

We generate covariate matrix $\boldsymbol{X}$ for fixed effect from a multivariate Gaussian distribution with mean zero and a block diagonal covariance matrix $\boldsymbol{\Sigma}_2 = \text{bdiag}(\widetilde{\Sigma}, \ldots, \widetilde{\Sigma})$ with block size 50, where $\widetilde{\Sigma} = (\widetilde{\sigma}_{ij})_{i,j=1}^{50}$ with $\tilde{\sigma}_{ij} = \rho^{|i-j|}$ if $i \neq j$ and $\tilde{\sigma}_{ii} = 1$. We selected ten β indexed with 1, 30, 60, 90, 120, 150, 180, 210, 240 and 270 to have nonzero values. We set $\beta = \log(3)$ for β's indexed 1, 30, 60, 90 and 120. We set $\beta = -\log(3)$ for β's indexed 150, 180, 210, 240 and 270. We let our ordinal response Y take one of six levels and the cutoff points are defined by the quantile of Z evenly. In other words, the propotional of data in each level of response is around 1/6. We consider both weak and strong correlation designs for the fixed effect in the simulation study by setting $\rho = 0.5$ and $\rho = 0.9$

Our goal was to assess whether our model would identify the truly non-zero elements of $\boldsymbol{\beta}$. We standardized the fixed effect of the data prior to fitting our model. We run the MCMC chains for 30000 iterations with the first 10000 iterations as a burn-in period. Since there is no exact zero produced by MCMC for global-local shrinkage prior, variable selection is

done based on t-statistic with 95% confidence level. All the experiments are repeated by 20 times and we report the mean and standard deviation of the results in all the tables.

In the weak correlation design of the fixed effect($\rho = 0.5$), all 20 experiments successfully identify the signal and noise by using 95% confidence level. In addition, 3.1 shows that the estimation of the nonzero β is also reasonably good given the oracle value of the none zero β in the first five elements in the table are -1.1 and the last five elements in the table are 1.1. Figure 3.2 and 3.3 shows the marginal density plot of the β in one experiment with weak corrleation design of the fixed effect ($\rho = 0.5$).Figure 3.2 shows that when the parameter is identified as a signal, the marginal posterior density covers the oracle value very well. Figure 3.3 shows that when the parameter is identified as noise, most of its posterior mass is around zero to guarantee a strong shrinkage. In addition, 3.4 and 3.5 show that the behaviour of the PCG sampler in a sparse ordinal regression setting is quite similar to the sparse linear regression setting in table 2.3. The autocorrelation is small for sparse parameters and large for nonzero parameters.

In the strong correlation design of the fixed effect($\rho = 0.9$), we observe the failure to identify the signal in the experiments. Table 3.2 summarizes the variable selection results in over 20 experiments. It shows that the $L_{\frac{1}{2}}$ prior can consistently identify all the noise in both weak and strong correlation designs for fixed effect. As we can see the False negative is 490 with 0 standard deviations in all 20 experiments in 95% and 90% significance levels. However, it seems that the $L_{\frac{1}{2}}$ prior becomes very conservative and tends to over-shrinkage the parameters in a strong correlation design. Even with a 90% confidence level, we still fail to identify all the signals. This over-shrinkage phenomenon has also been observed in high dimensional linear regression settings for both $L_{\frac{1}{2}}$ prior and horseshoe prior. See table 2.3.

3.4.2 SET Data analysis

In this section, we study a real dataset from an Australian university, where both numerical and text survey responses are available in large quantities. The goal of the study is to

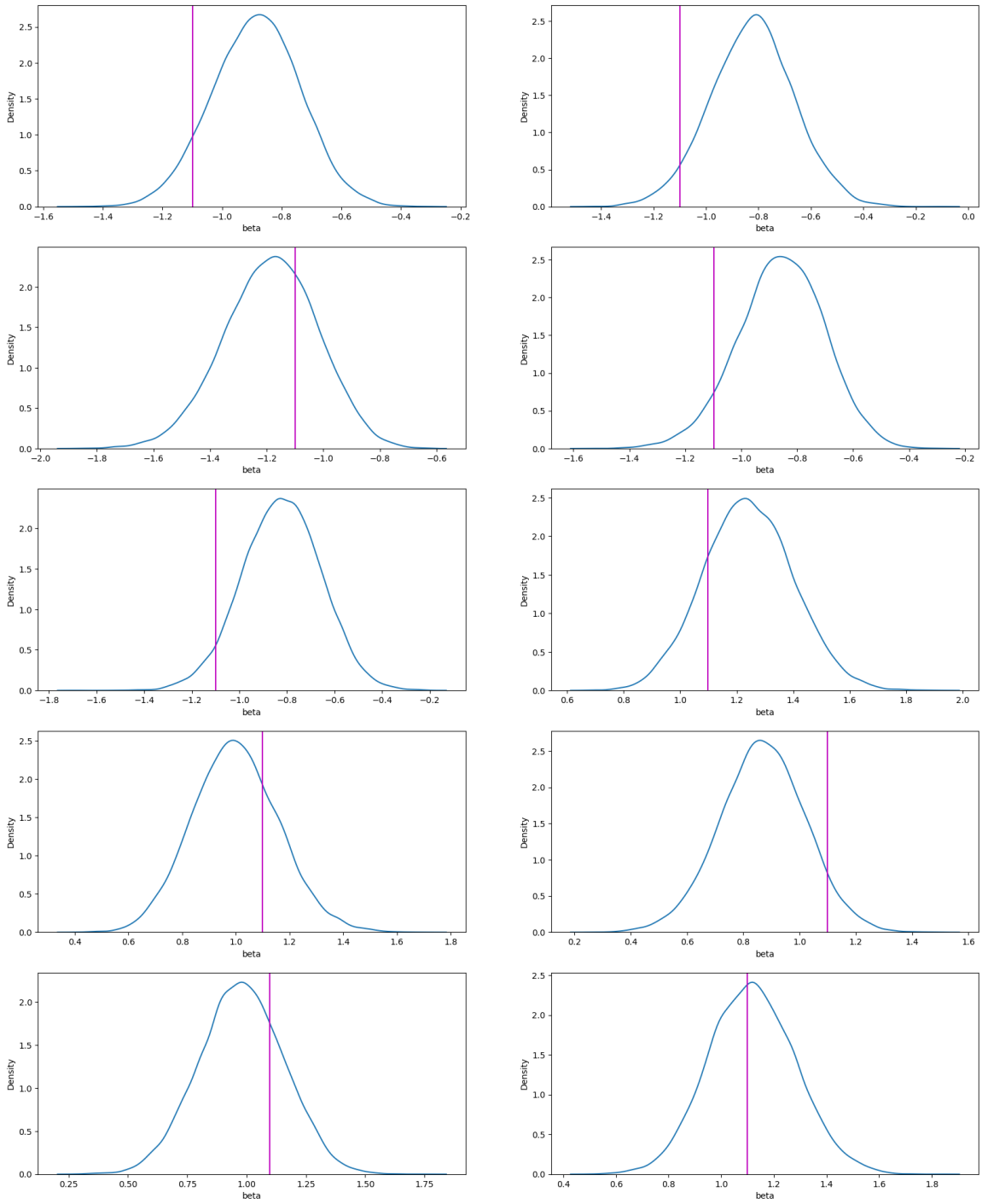


Figure 3.2: The density plot of marginal posterior of β, whose truth are nonzero, in one experiment with weak corrleation design of the fixed effect ($\rho = 0.5$)

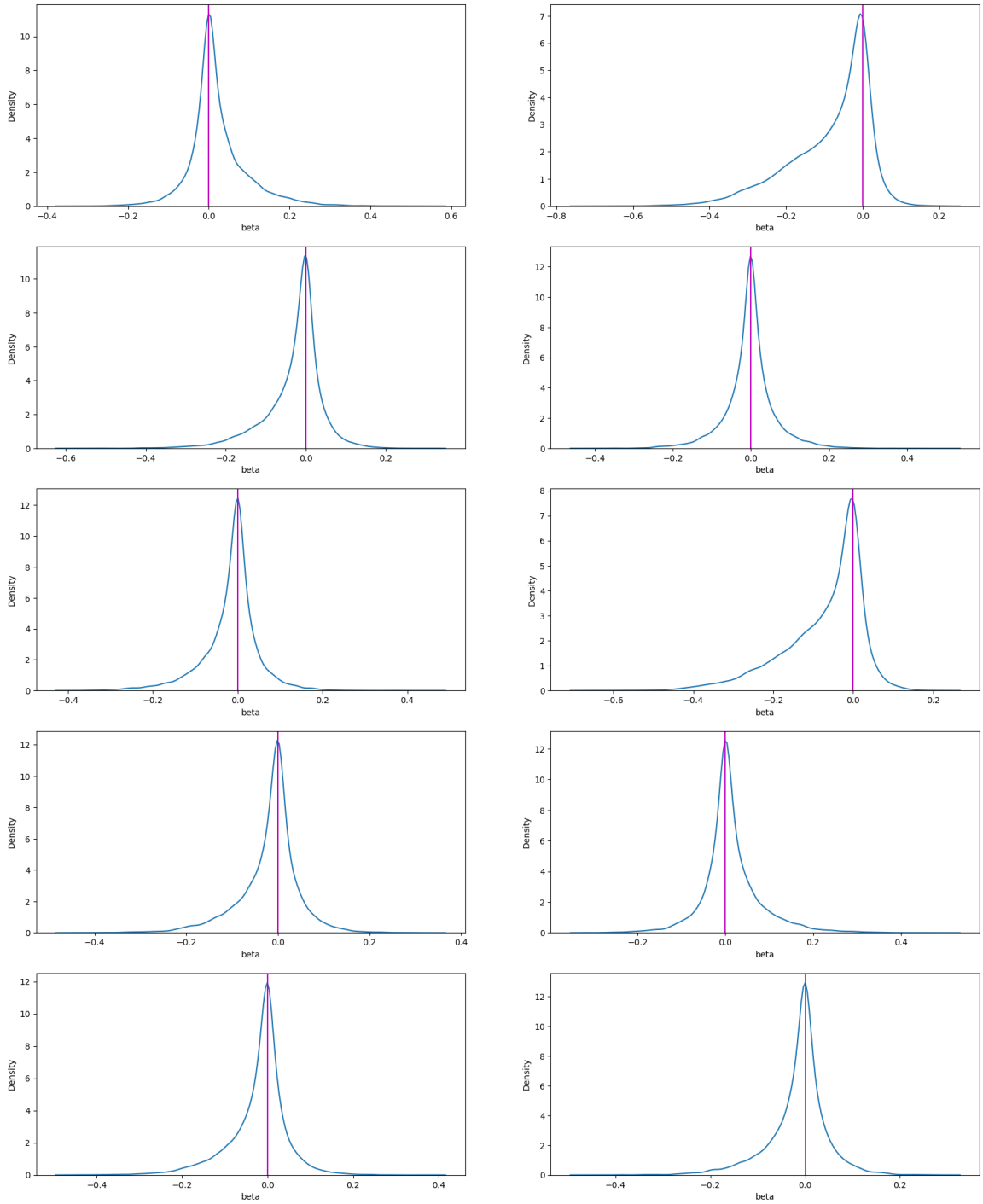


Figure 3.3: The density plot marginal posterior of β selected from index 271 to 281, whose truth are zero, in one experiment with weak correlation design of the fixed effect($\rho = 0.5$).

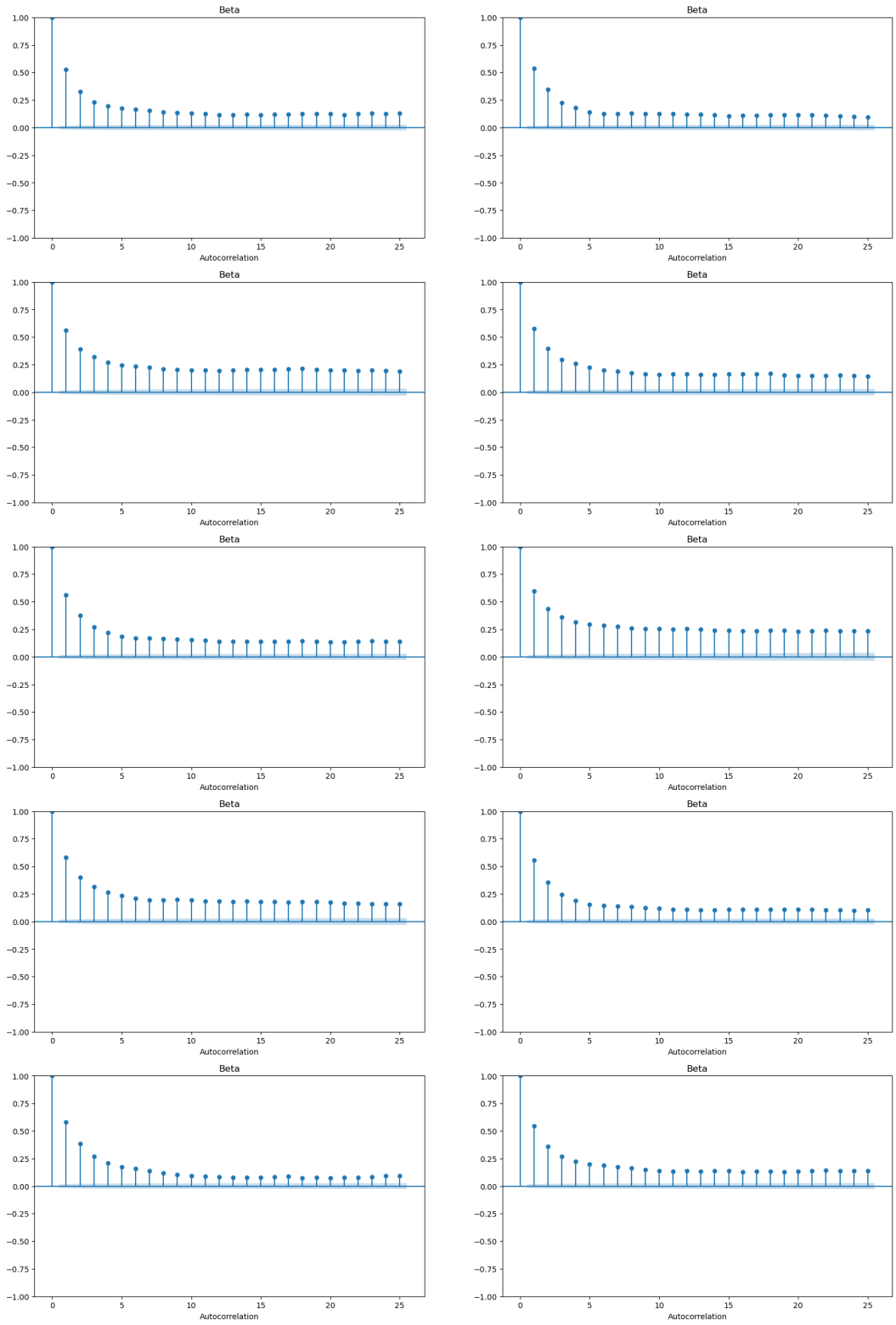


Figure 3.4: The autocorrelation of β, whose truth is nonzero, in one experiment with weak corrleation design of the fixed effect($\rho = 0.5$).

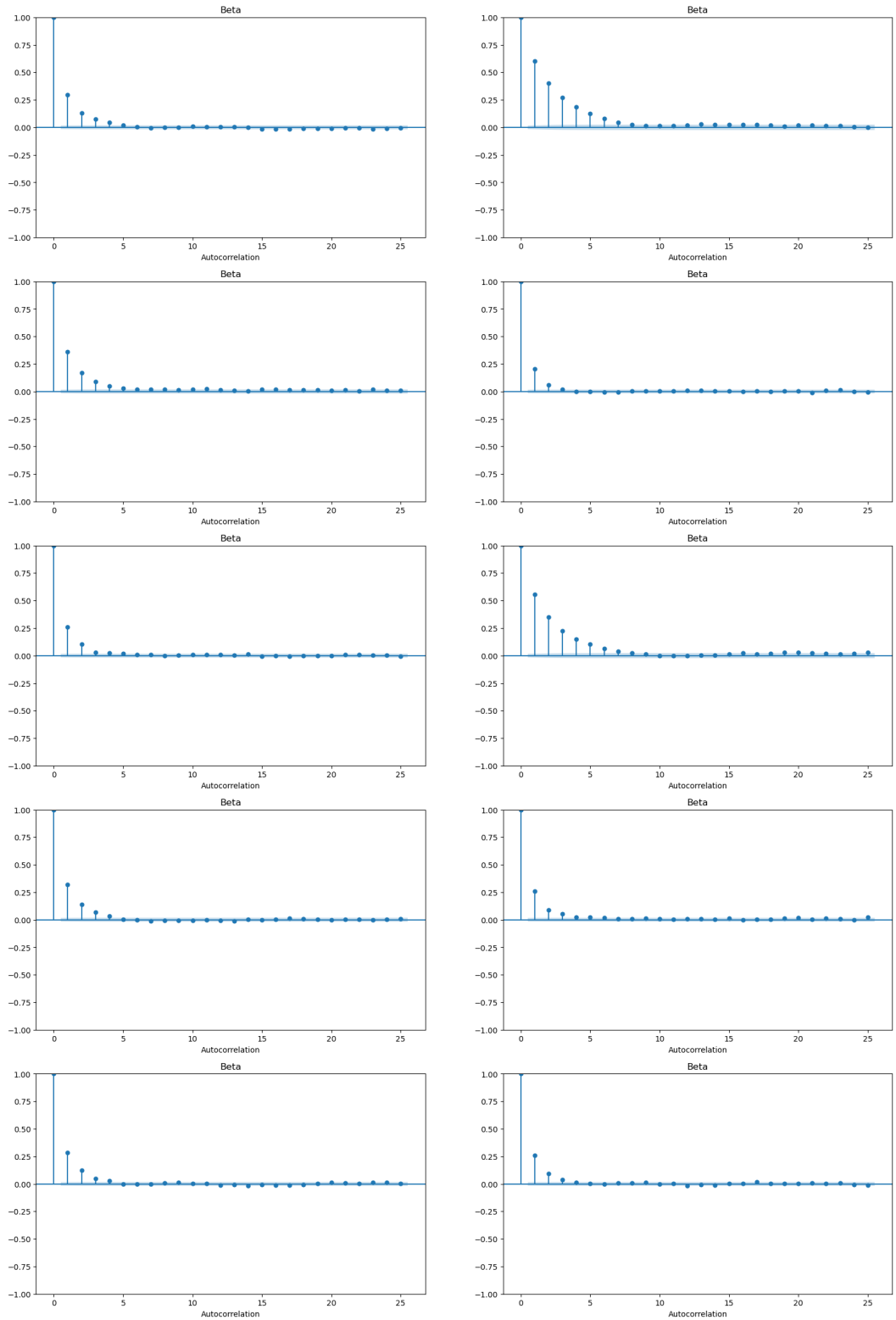


Figure 3.5: The autocorrelation of β selected from index 271 to 281, whose truth are zero, in one experiment with weak correlation design of the fixed effect($\rho = 0.5$).

	$\beta \neq 0$									
	β_1	β_{30}	β_{60}	β_{90}	β_{120}	β_{150}	β_{180}	β_{210}	β_{240}	β_{270}
$\hat{\beta}$	-0.90	-0.84	-1.20	-0.86	-0.81	1.22	1.03	0.90	1.02	1.22
Std	0.05	0.07	0.04	0.04	0.06	0.07	0.03	0.07	0.06	0.03

Table 3.1: In weak correlation design of the fixed effect($\rho = 0.5$), all 20 experiments successfully identify the signal and noise. Here we report the average of the posterior mean of β from ten experiments, whose truth is nonzero. We also show the standard deviation of the estimator from 20 experiments.

	$\rho = 0.5$			$\rho = 0.9$		
	90%	95%	99%	90%	95%	99%
True Positive	10(0)	10(0)	9.5(0.3)	6.2(0.3)	4.9(0.2)	3.3(0.1)
False Positive	1.4(0.2)	0(0)	0(0)	0(0)	0(0)	0(0)
False Negative	488.6(0.2)	490(0)	490(0)	490(0)	490(0)	490(0)
True Negative	0(0)	0(0)	0.5(0.3)	3.8(0.3)	5.1(0.2)	6.7(0.1)

Table 3.2: We report the average of the variable selection result over 20 experiments. In the bracket, we also show the corresponding standard deviation. The columns correspond to the significance level used for the t-statistic.

identify the drivers of higher SET ratings, and whether these are also affected by the gender or language backgrounds of instructors.

3.4.2.1 Data setting

Data from existing SET surveys have been collected over a 7-year time period from a large research and teaching-intensive Australian university, data is collected over several administrative units called faculties, such as Faculties of Science, Arts and Medicine etc. The primary focus is on the response the students provide to the final survey question, which asks students to indicate where on the Likert scale (from a scale of 1 to 6 corresponding to strongly disagree, disagree, moderately disagree, moderately agree, agree, strongly agree) they would rate how satisfied they were with the quality of their lecturer's teaching. Students are also able to provide comments on the best features of the lecturer's teaching through a free text field. The raw data was prepared and cleaned to ensure the data are in the appropriate format for analysis. The final dataset is made up of the following variables for each individual survey: Course ID, Student ID, Lecturer ID, Semester weighted

Step 1	**Keyword Tagging:** Identify all noun phrases from the comments;
Step 2	**Keyword Count:** Produce the frequency counts for the list of noun phrases;
Step 3	**Topic Determination:** In conjunction with expert knowledge;
Step 4	**Seed lists:** Produce a list of seed words for each topic;
Step 5	**Sentence Topic Assignment:** Assign each sentence to the relevant topic;
Step 6	**Sentence Sentiment Scoring:** For each document, assign a sentiment score.

Table 3.3: Text to data: using nouns and noun phrases to determine topics, which are then assigned a sentiment value.

average mark (WAM), Total Students, Lecturer Gender, Lecturer English or non-English speaking background (an indicator for language and ethnic background), Student Gender, Student Culture (local and international student flag), SET Score (on a Likert scale of 1 to 6) and Best Features (free text field).

Free text responses in surveys provide a rich source of data into the psychology of the respondents. As they became easier to collect and store in large quantities, it is an increasingly popular form of feedback mechanism for organisations looking for more insight. Examples include restaurant reviews; hotel customer satisfaction; customer surveys from banking institutions and student evaluations of teaching (SET) surveys in higher education institutions. In large quantities, such data are difficult to analyse within a statistical framework, since they allow the respondents to discuss an unrestricted number of issues, thus there is a need to convert the text to a stand-in quantity of interest, which is the ordered categorical responses.

We developed approaches to convert text to quantitative data in the form of topics and sentiment scores. Table 3.3 summarized the step of our approaches.

We created six topics, which are related to different dimensions of teaching and course quality: assessment (AS); course content (CC); learning environment (LE); staff quality (SQ); teaching and learning resources (TL) and teaching methods (TM) and an additional group collecting all miscellaneous topics (MS). Each noun phrase was assigned to only one topic. The sentences from the students' comments will assign to relevant topic.

From each sentence, a sentiment score was assigned using python's VADER package

[Hutto and Gilbert, 2014], which incorporates a sentiment lexicon that was developed for microblog-like contexts. The sentiment lexicon behind VADER is designed to account for both polarity and intensity expressed in social media contexts and is normally applicable to sentiment analysis in other domains. The compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 and +1. For all these details, see [Kim, 2022].

The final dataset is comprised of 149,292 individual student surveys, collected from 3063 unique courses and lecturers. The data was split across the five main academic faculties as follows: 32,852 in Arts and Social Science (ART); 40,145 in Commerce (COM); 28,272 in Engineering (ENG); 6,630 in Medicine (MED) and 41,393 in Science (SCI). Lecturer and student composition vary across the faculties, for example, ENG has the highest proportion of male students and the lowest proportion of female lecturers while ART, MED and SCI have the highest proportion of local students and COM has the most diverse lecturers from a non-English speaking background. SET data is a high dimension, it contains up to 100 fixed effects variables, and over 3,000 random effects parameters, which we employ to model dependencies between surveys.

3.4.2.2 Consensus Monte Carlo

When the sample size of the data is large, it would be tempting to be able to run parallel computations to reduce the computational burden. [Scott et al., 2016] describes a simple method to compute approximate posterior distributions for big data using distributed computing, they call this consensus Monte Carlo (CMC). Denoting the full set of data by Y and Y_r a subset of Y (or shard r) and let θ denote the set of model parameters, suppose the posterior distribution can be written as

$$\pi(\theta|Y) = \pi(Y|\theta)\pi(\theta) \propto \prod_{r=1}^{R} \pi(Y_r|\theta)\pi(\theta)^{1/R} \tag{3.4}$$

where R is the total number of shards and the prior $\pi(\theta)$ is diluted depending on the number of pieces.

Then each of the R pieces can be computed in parallel using MCMC (or any appropriate method of choice), and if each of the pieces is sufficiently large, it is then reasonable to assume that the posteriors of each piece are approximately normally distributed, leading to an approximately normally distributed full posterior. More concretely, suppose each subset generates draws θ_{ri}, $i = 1, \ldots, N$, the consensus posterior draws θ_i is given as

$$\theta_i = \sum_{r=1}^{R} w_r \theta_{ri} / \sum_{r=1}^{R} w_r.$$

When the Gaussian approximation is used, the weights $w_r = \Sigma_r^{-1}$ are optimal and can be estimated using the sample covariance matrix from each shard. If only a subset of θ is of interest, for example, the fixed effects parameters only, then the weighted samples are obtained marginally only for the parameters of interest.

For our current analyses, the full data can be naturally broken by the administrative unit of faculty. The observations are conditionally independent across the faculties, as the clusters modelled by the random effects occur wholly within each faculty. In this case, the precision matrix $\mathbf{\Lambda}$ for the random effect is block-diagonal. Furthermore, the covariate effects are not expected to be homogeneous across the faculties, for example, one might expect student perceptions to vary between the Arts and Social Sciences faculty where there is a large female presence in both the teaching and student population, compared to the Engineering faculty for example, where there's a larger proportion of males. So the CMC approach is desirable in this setting as it allows us to simultaneously understand both faculty and university-level results, which sometimes had very different outcomes.

To make sure our prior setting satisfies the equation 3.4, instead of assigning a hyper prior to λ, we tune the hyper-parameter λ. In addition, the cutoff points γ are pre-determined by empirical estimation such that

$$\gamma_1 = F^{-1}(p_1), \quad \gamma_2 = F^{-1}(p_1 + p_2) \quad \ldots, \gamma_{S-1} = F^{-1}(p_1 + p_2 + \cdots + p_{S-1}),$$

where $F^{-1}(\cdot)$ is the inverse of standard logistic distribution with p_s as proportional of data with response s. Care should be taken however when comparing the faculty-level results with the CMC results for the same λ, as the shrinkage effect from the priors is different

due to $\pi(\theta)^{1/R}$ from equation 3.4. We set $\lambda = 5$ for each piece, when these are combined via consensus Monte Carlo, the induced equivalent prior would be stronger than those used on the faculty level.

To fit the ordinal regression models, all non-binary covariates (including the sentiment scores) were standardised by subtracting the mean and dividing by 2 standard deviations to allow for easy interpretation of results. [Gelman, 2008] argued that this approach allows comparable interpretation of these covariates to the binary variables in the same model while subtracting the mean allows for the main effects of the interactions to be more easily interpreted.

We ran the PCG sampler described above using 20,000 iterations discarding the initial 10,000 as burn-in. The sampler mixes well and converges quickly. Each shard, corresponding to the five faculties, was run separately using the prior conditions set above and combined using the CMC approach.

3.4.2.3 Summary of findings

Here we provide a brief summary of the findings from the ordinal regression model, for a more detailed discussion, see [Kim, 2022].

First, based on the text responses provided by the students, they are primarily focused on six main topic groups which are all related to different teaching dimensions, despite the fact that they sometimes speak about topics completely unrelated to teaching, such as the instructor's appearance or other apparently unrelated topics. The six theme groups (or topics) related to different dimensions of teaching and course quality: *assessment* which include comments on feedback, exams and other types of assessment tasks. *course content* which includes anything that refers to the content of the course, e.g., the subject matter, concepts and structure of the course, the topics in the course etc. *learning environment* which refers to lecture theatres and the learning experience. *staff quality* included phrases describing the lecturer directly, e.g., good pace, depth knowledge, approachability, clear

teaching and nice guy etc. The topic *teaching and learning resources* refers to the provision of slides, and lecture materials, or videos and visual aids. *course content*, which is related to the subject matter of the course. Finally, *teaching method* covers techniques used for teaching, such as encouraging student participation, the use of real-life examples, and guest speakers. Results from ordinal regression suggest that only the staff quality and teaching methods dimensions contribute directly towards student ratings, as well as the miscellaneous comments. This suggests that students may place verbal significance toward certain teaching dimensions that are not reflected in the numerical score. Additionally, the significance of staff quality demonstrates the important role of personal characteristics, in evaluation scores.

Second, our findings suggest that female lecturers, and lecturers from non-English speaking backgrounds, are more likely to be assessed negatively on irrelevant topics which then has a negative impact on the ratings. The effects of gender and culture vary drastically depending on cohort/faculty and when the data is aggregated such effects may disappear, as the CMC results point out to sometimes conflicting biases. In addition, while female lecturers tend to receive higher sentiment scores, their ratings are more often negative, suggesting that female educators may be held to higher standards than their male counterparts. Finally, male lecturers were more likely to receive more general feedback consisting of 'good' and 'great', while female lecturers were more likely to receive more specific feedback in regards to 'passionate', 'approachable', 'friendly' and 'engaging. These results suggest that often male and female instructors are perceived using different metrics.

Chapter 4

Sparse deep learning

In this chapter, we first develop an adaptive optimization algorithm for a variational Bayesian neural network. We then employ the spike-and-slab prior and generalize our algorithm to the variational EM algorithm to train a sparse neural network. Next we extend our algorithm to a graph neural network(GNN). We introduce an unconstraint and trainable mask to the adjacency matrix of the graph. By jointly training both the weight and mask with our variational EM algorithm, we obtain a sparse graph and weight for GNN. Finally, we discuss a potential future work, which is about using $L_{\frac{1}{2}}$ prior to obtaining a sparse latent variable in the VAE.

4.1 On the optimization and pruning for Bayesian neural network

A deep neural network has the strong power to represent a highly complex nonlinear system. But it is often over-parametric and easy to be overfitted. They are often incapable of correctly assessing the uncertainty in the training data and so make overly confident decisions about the correct class, prediction or action. Bayesian inference [Bishop and Nasrabadi, 2006] provides an elegant way to capture the uncertainty of the neural net-

work via the posterior distribution over model parameters. Unfortunately, the posterior inference is intractable due to the complex likelihood from the neural networks and is not as scalable as traditional approaches such as stochastic gradient descent.

Works focussing on scalable inference for Bayesian deep learning over the last decade can be separated into two streams. One stream use deterministic approximation approach such as variational inference [Graves, 2011,Blundell et al., 2015], dropout [Gal and Ghahramani, 2016], Laplace approximation [Ritter et al., 2018], or expectation propagation [Hernández-Lobato and Adams, 2015]. The other stream involves sampling approaches such as MCMC using stochastic gradient Langevin dynamics (SGLM) [Welling and Teh, 2011,Chen et al., 2014].

Prior to 2019, deep Bayesian neural networks (BNN) generally struggle with predictive accuracy and computational efficiency. Recently, a lot of advances have been made in both directions of research. In the deterministic approach, several authors consider using dimension reduction techniques such as subspace inference [Maddox et al., 2019], rank-1 parameterization [Dusenberry et al., 2020], subnetwork inference [Daxberger et al., 2021] and node-space inference [Trinh et al., 2022]. In the sampling approach, [Zhang et al., 2019] propose to use cycles of learning rates with a high-to-low step size schedule. A large step size in the early stage of the cycle results in aggressive exploration in the parameter space; as the step size decreases, the algorithm begins to collect samples around the local mode.

Apart from the progress within these two streams, [Wilson and Izmailov, 2020] show that deep ensembles [Lakshminarayanan et al., 2017] can be interpreted as an approximate approach to posterior predictive distribution. They combine multiple independently trained SWAG (Gaussian stochastic weight averaging) approximations [Maddox et al., 2019, Izmailov et al., 2018] to create a mixture of Gaussian approximation to the posterior. However, performing variational inference directly on weight space still produces poor predictive accuracy and struggles with computational efficiency. Even a simple mean-field variational inference will involve doubling the number of parameters of the neural network (i.e., mean and variance for each weight), which incurs an extra GPU memory requirement

and 2-5 times the runtime of the baseline neural network [Osawa et al., 2019].

In this chapter, we develop an adaptive optimization algorithm for Gaussian Mean-field variational Bayesian inference that can achieve state-of-the-art predictive accuracy. We further show that when the learning rate is small and the update of the posterior variance has been frozen, the algorithm is equivalent to the SGHMC (Stochastic Gradient Hamiltonian Monte Carlo) with a preconditioning matrix. Therefore, if we exploit the closed-form expression of the gradient for the posterior variances and only keep track of the weights generated by the algorithm (without registering the mean and variance parameter in the model class of Pytorch and only keep track of them in the optimizer module), we can achieve big savings on GPU memory and runtime costs.

Based on the connection to SGHMC, we extend the EM Algorithm for Bayesian variable selection [Ročková and George, 2014, Wang et al., 2016, Ročková, 2018] for linear models to neural networks by replacing the Gaussian prior in BNN with the spike-and-slab group Gaussian prior [Xu and Ghosh, 2015]. Our method is a Variational EM algorithm, which will switch the weight decay factor between small and large based on the magnitude of each group during training. Since by construction, there are no exact zeros, we further find a simple pruning criterion to remove the weights permanently during training. A sparse model will be trained in one shot without additional retraining. Our approach is more computationally efficient than those dynamic pruning strategies that allow regrow [Zhu and Gupta, 2017, Dettmers and Zettlemoyer, 2019, Lin et al., 2020]. We will show that this aggressive approach has no performance loss. Our code is available at GitHub: `https://github.com/z5041294/optimization-and-pruning-for-BNN`

4.1.1 Optimization

Given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, a Bayesian neural network (BNN) is defined in terms of a prior $p(\boldsymbol{w})$ on the p-dimensional weights, as well as the neural network likelihood $p(\mathcal{D}|\boldsymbol{w})$. For example in a GLM neural network setting, we have the likelihood

$$f\left(y \mid \mu(\boldsymbol{x})\right) = \exp\left\{A\left(\mu(\boldsymbol{x})\right) y + B\left(\mu(\boldsymbol{x})\right) + C(y)\right\}$$

where $\mu(x)$ denotes a nonlinear function of x, and $A(\mathring{u})$, $B(\mathring{u})$ and $C(\mathring{u})$ are appropriately defined functions with the fully connected neural network as a nonlinear map

$$\mu(\boldsymbol{x}) = \boldsymbol{w}^{H_n}\psi^{H_n-1}\left[\cdots\psi^1\left[\boldsymbol{w}^1\boldsymbol{x}\right]\right]$$

The ψ^i is the activation function in i hidden layer.

Variational Bayesian methods approximate the true posterior $p(\boldsymbol{w}|\mathcal{D})$ by minimising the KL divergence between the approximate distribution, $q_\theta(\boldsymbol{w})$, and the true posterior. It can be shown that this is equivalent to maximizing the evidence lower bound (ELBO):

$$\begin{aligned}
\theta^\star &= \arg\min_\theta \mathrm{KL}[q_\theta(\boldsymbol{w})\|P(\boldsymbol{w}\mid\mathcal{D})] \\
&= \arg\min_\theta \int q_\theta(\boldsymbol{w})\log\frac{q_\theta(\boldsymbol{w})}{P(\boldsymbol{w})P(\mathcal{D}\mid\boldsymbol{w})}\mathrm{d}\boldsymbol{w} \\
&= \arg\min_\theta \mathrm{KL}[q_\theta(\boldsymbol{w})\|P(\boldsymbol{w})] - \mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D}\mid\boldsymbol{w})] \\
&= \arg\max_\theta \mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D}\mid\boldsymbol{w})] - \mathrm{KL}[q_\theta(\boldsymbol{w})\|P(\boldsymbol{w})] \\
&= \arg\max_\theta \mathcal{L}[\theta]
\end{aligned} \tag{4.1}$$

We consider a Bayesian neural net with Gaussian prior $p(\boldsymbol{w}) \sim N_p(0, \boldsymbol{\Sigma}_0)$ and a Gaussian approximate posterior $q_\theta(\boldsymbol{w}) \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. To make it scale to large-sized models, we assume both the approximate posterior and prior weights $w_j, j = 1, \ldots, p$ are independent, such that $p(w_j) \sim N(0, \delta^{-1})$ and $q_{\theta_j}(w_j) \sim N(\mu_j, \sigma_j^2)$. Then the second KL divergence term in 4.1 has the closed form in

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(q_\theta(\boldsymbol{w})\|p(\boldsymbol{w})) &= \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_0|}{|\boldsymbol{\Sigma}|} - p + \mathrm{tr}\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}\right)\right] + \frac{1}{2}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_0\right)^T\boldsymbol{\Sigma}_0^{-1}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_0\right) \\
&= -\frac{1}{2}\sum_{j=1}^{p}\log\sigma_j^2 + \frac{1}{2}\delta\sum_{j=1}^{p}\sigma_j^2 + \frac{1}{2}\delta\|\boldsymbol{\mu}\|^2 - \frac{p}{2} - \frac{p}{2}\log\delta
\end{aligned} . \tag{4.2}$$

The first term $\mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D}\mid\boldsymbol{w})]$ and its gradient are tractable. However, by using the reparameterization trick [Kingma and Welling, 2013, Rezende et al., 2014], we can obtain a good unbiased estimator for its gradient. We express the random variable $\boldsymbol{w} \sim q_\theta(\boldsymbol{w})$ as a differentiable and invertible transformation of another random variable ϵ given θ

$$w = g_\theta(\epsilon) \tag{4.3}$$

where the distribution of random variable $\epsilon \sim p(\epsilon)$ is independent of θ. Then we have

$$\begin{aligned}\nabla_\theta \mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D} \mid \boldsymbol{w})] &= \frac{\partial}{\partial \theta} \int \log P(\mathcal{D} \mid \boldsymbol{w}) q_\theta(\boldsymbol{w}) dw \\ &= \nabla_\theta \int \log P(\mathcal{D} \mid g_\theta(\boldsymbol{\epsilon})) P(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\ &= \int \nabla_\theta \log P(\mathcal{D} \mid g_\theta(\boldsymbol{\epsilon})) P(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\ &\approx \frac{1}{S} \sum_{i=1}^{S} \nabla_\theta \log P(\mathcal{D} \mid g_\theta(\boldsymbol{\epsilon}_i))\end{aligned} \tag{4.4}$$

With the Gaussian mean field approximation, we have $\boldsymbol{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$. The gradient of ELBO with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} \mathcal{L} &= \mathbf{E}_{q_\theta(\boldsymbol{w})} \left[\nabla_{\boldsymbol{w}} \log p(\mathcal{D} \mid \boldsymbol{w})\right] - \delta \boldsymbol{\mu} \approx -\frac{1}{S} \sum_{i=1}^{S} \mathbf{g}_i - \delta \boldsymbol{\mu} \\ \nabla_{\boldsymbol{\sigma}} \mathcal{L} &= \mathbf{E}_{q_\theta(\boldsymbol{w})} \left[\frac{\boldsymbol{w} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \odot \nabla_{\boldsymbol{w}} \log p(\mathcal{D} \mid \boldsymbol{w})\right] + \frac{1}{\boldsymbol{\sigma}} - \delta \boldsymbol{\sigma} \\ &\approx -\frac{1}{S} \sum_{i=1}^{S} \mathbf{g}_i \odot \frac{\boldsymbol{w}_i - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \frac{1}{\boldsymbol{\sigma}} - \delta \boldsymbol{\sigma}\end{aligned} \tag{4.5}$$

where $\mathbf{g}_i = -\nabla_{\boldsymbol{w}} \log p(\mathcal{D} \mid \boldsymbol{w}_i)$ and $\boldsymbol{w}_i \sim \prod_{j=1}^{p} N(\mu_j, \sigma_j^2)$ are Monte Carlo samples. Rather than using the reparameterization trick, when the approximation posterior $q_\theta(\boldsymbol{w})$ is Gaussian, we can use the following Gaussian gradient identities:

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} \mathbb{E}_{N(\boldsymbol{\mu}, \boldsymbol{\Sigma})}[f(\boldsymbol{w})] &= \mathbb{E}_{N(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \left[\nabla_w f(\boldsymbol{w})\right] \\ \nabla_{\boldsymbol{\Sigma}} \mathbb{E}_{N(\boldsymbol{\mu}, \boldsymbol{\Sigma})}[f(\boldsymbol{w})] &= \frac{1}{2} \mathbb{E}_{N(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \left[\nabla_w^2 f(\boldsymbol{w})\right]\end{aligned} \tag{4.6}$$

to obtain a approximation for $\nabla_{\boldsymbol{\sigma}^2} \mathcal{L}$

$$\nabla_{\boldsymbol{\sigma}^2} \mathcal{L} = \frac{1}{2} \mathbf{E}_{q_\theta(\boldsymbol{w})} \left[\nabla_{\boldsymbol{w}}^2 \log p(\mathcal{D} \mid \boldsymbol{w})\right] + \frac{1}{2\sigma^2} - \frac{1}{2} \delta \approx -\frac{1}{2S} \sum_{i=1}^{S} \mathbf{g}_i^2 + \frac{1}{2\sigma^2} - \frac{1}{2} \delta. \tag{4.7}$$

In addition, by using the Gaussian gradient identities, it is easy to verify that

$$\nabla_{\boldsymbol{\mu}}^2 \mathcal{L} = -\mathbf{E}_{q_\theta(\boldsymbol{w})} \left[\nabla_{\mathbf{w}}^2 \log p(\mathcal{D} \mid \boldsymbol{w})\right] + \delta \approx \frac{1}{S} \sum_{i=1}^{S} \mathbf{g}_i^2 + \delta. \tag{4.8}$$

In practice, to reduce the runtime complexity, we often use $S = 1$ as long as the batch size in one iteration is not too small. An alternative unbiased stochastic gradient estimator

of the $\mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D} \mid \boldsymbol{w})]$ is the score function estimator [Glynn, 1990, Williams, 1992, Kleijnen and Rubinstein, 1996]:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{q_\theta(\boldsymbol{w})}[\log P(\mathcal{D} \mid \boldsymbol{w})] &= \mathbb{E}_{q_\theta(\boldsymbol{w})}[\nabla_\theta \log q_\theta(\boldsymbol{w}) \log P(\mathcal{D} \mid \boldsymbol{w})] \\ &\approx \frac{1}{S}\sum_{i=1}^{S} \nabla_\theta \log q_\theta(\boldsymbol{w}_i) \log P(\mathcal{D} \mid \boldsymbol{w}_i)\end{aligned}$$

where $\boldsymbol{w} \sim q_\theta(\boldsymbol{w})$. This method often requires the various novel control variate techniques for variance reduction [Ranganath et al., 2014]. The main advantage of this approach is that it can be applied directly to discrete variables. However, the score function estimator ignores the gradient information about the function $\log p(\mathcal{D} \mid \boldsymbol{w})$. Even with the control variate techniques, this method will lead to much higher variance compared with the reparameterization trick. Given the unbiased Monte Carlo estimator we have, a straightforward approach used in the early works to optimize the ELBO [Ranganath et al., 2014, Blundell et al., 2015] is using stochastic gradient descent.

Almost at the same time, [Kingma and Ba, 2014] proposed the Adam algorithm, which approximates the diagonal elements of the Hessian matrix by computing the second moment of the gradient. These quantities act as an adaptive learning rate. Adam quickly became extremely popular in the deep-learning community. As this algorithm produces comparable results with SGD and a much faster convergence speed. Table 4 shows the implementation of Adam to find the MLE of the likelihood. It should emphasize that both SGD and Adam used the momentum gradient to make the algorithm robust.

It was found by [Khan et al., 2018, Zhang et al., 2018, Osawa et al., 2019] that, in Gaussian mean-field variational Bayesian inference, the posterior variance of the weight can play the rule of adaptive learning rate. They showed that their approaches are related to natural-gradient descent. With their approaches, the ELBO bound is updated by

$$\begin{aligned}\boldsymbol{\mu}_{t+1} &= \underset{\boldsymbol{\mu}\in\mathbf{R}^p}{\operatorname{argmin}} \left\{ \langle \nabla_{\boldsymbol{\mu}} \mathcal{L}, \boldsymbol{\mu} \rangle + \frac{1}{2l_t}(\boldsymbol{\mu} - \boldsymbol{\mu}_t)^T \mathbf{diag}(\sigma^{-2})^\alpha (\boldsymbol{\mu} - \boldsymbol{\mu}_t)^T \right\} \\ &= \boldsymbol{\mu}_t - l_t(\boldsymbol{\sigma}_t^2)^\alpha \odot \nabla_{\boldsymbol{\mu}_t} \mathcal{L}\end{aligned}$$

where l_t is a learning rate, $\alpha \in \left\{\frac{1}{2}, 1\right\}$ and $\frac{1}{\boldsymbol{\sigma}_t^2}$ is updated with momentum $\frac{1}{\boldsymbol{\sigma}_t^2} = \frac{1-\lambda\gamma}{\boldsymbol{\sigma}_{t-1}^2} + \gamma([\boldsymbol{g}_t \odot \boldsymbol{g}_t] + \frac{\delta}{N})$ ($0 < \gamma < 1$ and λ is another learning rate). When $\alpha = \frac{1}{2}$, this is similar

Algorithm 4 Adam

Input: $\gamma = (\gamma_1, \gamma_2)$

Initialization: $\boldsymbol{m}$

for $t = 1 \cdots$ **do**

$\boldsymbol{g} \leftarrow -\frac{1}{B}\sum_{i\in\mathcal{B}} \nabla_{\boldsymbol{w}} \log(\mathcal{D}_i|\boldsymbol{w})$

$\mathbf{m} \leftarrow \gamma_1 \mathbf{m} + (1-\gamma_1)\,\mathbf{g}$ ▷ momentum gradient

$\mathbf{s} \leftarrow \gamma_2 \mathbf{s} + (1-\gamma_2)\,(\mathbf{g} \circ \mathbf{g})$ ▷ momentum adaptive learning rate

$\hat{\mathbf{m}} \leftarrow \mathbf{m}/\left(1-\gamma_1^t\right)$

$\hat{\mathbf{s}} \leftarrow \mathbf{s}/\left(1-\gamma_2^t\right)$

$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha\hat{\mathbf{m}}/(\sqrt{\hat{\mathbf{s}}} + \delta)$

end for

to the Adam [Kingma and Ba, 2014] algorithm ([Khan et al., 2018] called this approach as variational Adam.) and when $\alpha = 1$, the algorithm is very close to using the Hessian matrix of the ELBO given by $\nabla^2_{\boldsymbol{\mu}}\mathcal{L} = \mathbf{diag}(\sigma^{-2})$ if $\nabla_{\boldsymbol{\sigma}^2}\mathcal{L} = 0$ in equation 4.7.

There are two concerns for these Bayesian adaptive algorithms: First, similar to Adam, the variance of the adaptive learning rate is problematically large in the early stages of the optimization [Liu et al., 2019]. A warm-up stage is recommended for traditional Adam. As an example, consider a ReLu neural network with a single hidden layer and binary cross-entropy loss, then if a normal initialization of the weight with mean zero has been used, then the variance of the adaptive learning will diverge at the beginning of the training period (see detailed discussion in section C.1). Second, when the number of Monte Carlo samples for the gradient is $S = 1$ and the learning rate is small, the injected Gaussian noise may dominate the gradient. Note that when $S = 1$, the Monte Carlo estimator is noisy but unbiased. We see that

$$\begin{aligned}\boldsymbol{w}_t - \boldsymbol{w}_{t-1} &= \boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1} + (\boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon} - \boldsymbol{\sigma}_{t-1} \odot \boldsymbol{\epsilon}') \\ &= -l_t(\boldsymbol{\sigma}_t^2)^{\alpha} \odot \boldsymbol{m}_t + \boldsymbol{\epsilon} \odot \sqrt{\boldsymbol{\sigma}_t^2 + \boldsymbol{\sigma}_{t-1}^2}\end{aligned}$$

where $\boldsymbol{m}_t$ is the momentum of gradient. For $\alpha = \frac{1}{2}$, when the learning rate is small, the gradient may be erased by the injected noise. For $\alpha = 1$, this could also happen when $\boldsymbol{\sigma}_t^2 \ll 1$. Empirically, we observe that both $\boldsymbol{w} \ll 1$ and $\boldsymbol{\sigma}^2 \ll 1$ in Bayesian deep

learning. So a warm-up strategy that starts with a very small learning rate may not work for variational BNN. On the contrary, when the learning rate is large, the size of the injected noise will be relatively small, and the model could overfit. In summary, there is a delicate balance between the size of the gradient and the size of the injected noise. This argument will become more lucid when we show the connection between our algorithm and SGHMC.

4.1.1.1 Constrained variational Adam (CVA)

To remedy the issue highlighted above, we propose a constrained variational Adam (CVA) algorithm, where we reparameterize $\boldsymbol{\sigma}_i$ as a product of global and local parameters such that $\boldsymbol{\sigma}_i = \alpha\boldsymbol{\tau}_i$ where $\boldsymbol{\tau}_i \in (0, 1)$ and $\alpha > 0$. We treat α as a hyper-parameter, which can be used in an annealing scheme. Now, the variance of the adaptive learning rate is upper bounded by $\frac{\alpha^2}{12}$. The updated posterior mean is modified to

$$\begin{aligned}\boldsymbol{\mu}_{t+1} &= \underset{\boldsymbol{\mu}\in\mathbf{R}^p}{\operatorname{argmin}} \left\{ \langle \nabla_{\boldsymbol{\mu}} \mathcal{L}, \boldsymbol{\mu} \rangle + \frac{1}{2l_t} (\boldsymbol{\mu} - \boldsymbol{\mu}_t)^T \mathbf{diag}(\boldsymbol{\tau}^{-1}) (\boldsymbol{\mu} - \boldsymbol{\mu}_t)^T \right\} \\ &= \boldsymbol{\mu}_t - l_t \boldsymbol{\tau}_t \odot \nabla_{\boldsymbol{\mu}_t} \mathcal{L}\end{aligned}.$$

Since $\boldsymbol{\tau}_i \in (0, 1)$ are the parameters we need to learn, the objective function for updating the posterior standard deviation change to:

$$\begin{aligned}\boldsymbol{\tau}_{t+1} &= \underset{\boldsymbol{\tau}\in(0,1)^p}{\operatorname{argmin}} \left\{ \langle \nabla_{\boldsymbol{\tau}} \mathcal{L}, \boldsymbol{\tau} \rangle + \frac{1}{2\eta} \|\boldsymbol{\tau} - \boldsymbol{\tau}_t\|^2 \right\} \\ &= \underset{\boldsymbol{\tau}\in(0,1)^p}{\operatorname{argmin}} \|(\boldsymbol{\tau}_t - \eta \nabla_{\boldsymbol{\tau}} \mathcal{L}) - \tau\|^2 \\ &= \pi_{(0,1)^P}(\boldsymbol{\tau}_t - \eta \nabla_{\boldsymbol{\tau}} \mathcal{L})\end{aligned}$$

where η is a learning rate, $\pi_{(0,1)^P}$ denotes the Euclidean projection onto $(0, 1)^P$. Now we replace the Euclidean distance of $\frac{1}{2}\|\boldsymbol{\tau}_{t+1} - \boldsymbol{\tau}_t\|^2$ to Bregman distance $D_G(\boldsymbol{\tau}_{t+1}, \boldsymbol{\tau}_t)$, where $D_G(\cdot, \cdot)$ is generated by a strictly convex twice-differentiable function $G(\cdot)$

$$D_G(\boldsymbol{\tau}, \boldsymbol{\tau}') := G(\boldsymbol{\tau}) - G(\boldsymbol{\tau}') - \langle \nabla G(\boldsymbol{\tau}'), \boldsymbol{\tau} - \boldsymbol{\tau}' \rangle.$$

If $G = \frac{1}{2}\|\cdot\|^2$, we recover the squared Euclidean distance. This modification leads to a more general version of the gradient descent algorithm which is known as mirror descent.

The idea is that we want to find a distance function which can better reflect the geometry of $(0,1)^p$. Now our objective function can be written in a more general form:

$$\boldsymbol{\tau}_{t+1} = \underset{\boldsymbol{\tau}\in(0,1)^p}{\operatorname{argmin}} \left\{ \langle \nabla_{\boldsymbol{\tau}} \mathcal{L}, \boldsymbol{\tau} \rangle + \frac{1}{\eta} D_G(\boldsymbol{\tau}, \boldsymbol{\tau}_t) \right\} \tag{4.9}$$

The choice of $G(\cdot)$ will define the geometry and the distance metric of the primal space and dual space. Values in $(0,1)$ may be interpreted as a probability. We define the distance $D_G(\boldsymbol{\tau}, \boldsymbol{\tau}_t)$ as negative binary cross entropy loss:

$$G(\tau) = \tau \log \tau + (1-\tau)\log(1-\tau)$$

The parameter in the dual space is defined as

$$\rho = \nabla_\tau G(\tau) = g(\tau) = \log(\tau) - \log(1-\tau) \tag{4.10}$$

with the inverse function from dual space to the primal space, which is the logistic sigmoid function

$$\tau = \nabla_\rho H(\rho) = h(\rho) = g^{-1}(\rho) = \frac{1}{1+\exp(-\rho)} \tag{4.11}$$

The mirror descent allows us to perform gradient descent in the dual space, which is unconstrained in our case, as shown in equation (4.10), and finally move back to the primal space by equation (4.11). Algorithm 5 provides the pseudo-code for the constrained variational Adam. (Appendix C.2 provides a quick introduction to mirror descent and the comparison with the reparameterization trick).

4.1.1.2 Connection to SGHMC

[Mandt et al., 2017] showed that SGD can be viewed as approximate Bayesian inference. Here, we show that by controlling α and β and learning rate l_t, our CVA algorithm can be viewed as a stochastic gradient hamiltonian monte Carlo (SGHMC) [Chen et al., 2014]. Before showing this connection, we first outline the key results from SGHMC.

Suppose we want to sample from the posterior distribution of w given a set of independent observations $x \in \mathcal{D}$:

$$p(w \mid \mathcal{D}) \propto \exp(-U(w))$$

Algorithm 5 Constrained Variational Adam(CVA)

Input: $\beta = (\beta_1, \beta_2)$,$\delta$(weight decay)

N(training size), B(batch size)

Initialization: $\boldsymbol{m}$, $\boldsymbol{\rho}$, $\boldsymbol{\mu}$, $\boldsymbol{\tau}$

for $t = 1 \cdots$ **do**

$l \leftarrow s_1(t)$ ▷ Cyclical learning rate

$\eta \leftarrow s_2(t)$ ▷ learning rate schedule

$\alpha \leftarrow s_3(t)$ ▷ annealing schedule

Sample $\boldsymbol{\epsilon} \sim N(0, \mathbf{1}_p)$

$\boldsymbol{w} \leftarrow \boldsymbol{\mu} + \alpha\boldsymbol{\tau} \odot \boldsymbol{\epsilon}$

$\boldsymbol{g} \leftarrow -\frac{1}{B}\sum_{i\in\mathcal{B}} \nabla_{\boldsymbol{w}} \log(\mathcal{D}_i|\boldsymbol{w}) + \frac{\delta}{N}\boldsymbol{\mu}$

$\boldsymbol{m} \leftarrow \beta_1\boldsymbol{m} + \beta_2\boldsymbol{g}$

$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + (\frac{\eta}{\tau} - \eta\alpha^2\delta\boldsymbol{\tau}) - \eta\alpha\boldsymbol{\epsilon} \odot \boldsymbol{g}$

$\boldsymbol{\tau} \leftarrow 1/(1 + e^{-\boldsymbol{\rho}})$

$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - l\boldsymbol{\tau} \odot \boldsymbol{m}$

end for

where the potential energy function U is given by $U(w) = -\sum_{x\in\mathcal{D}} \log p(x \mid w) - \log p(w)$. [Chen et al., 2014] shows that under the mild condition, the following dynamic

$$\begin{aligned} d\boldsymbol{w}_t &= M^{-1}\boldsymbol{r}_t dt \\ d\boldsymbol{r}_t &= -\alpha\nabla\tilde{U}(w_t)dt - CM^{-1}\boldsymbol{r}_t dt + \sqrt{2\alpha(C-B)}d\boldsymbol{B}_t \end{aligned} \tag{4.12}$$

with

$$\nabla\tilde{U}(w) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|}\sum_{x\in\tilde{\mathcal{D}}} \nabla \log p(x \mid w) - \nabla \log p(w), \tilde{\mathcal{D}} \subset \mathcal{D}$$

yields the stationary distribution $\pi(w, r) \propto \exp\left(-U(w) - \frac{1}{2}r^T M^{-1} r\right)$ if the hyper-parameter B is a good estimation of the variance from the stochastic gradient. In addition, C is another user-specified friction term such that $C > B$, M is the preconditioning matrix and α is any positive number. In practice, we never know the variance of stochastic gradient, in this sense, the SGHMC is an approximate MCMC scheme. As suggested by [Chen et al., 2014], we should set B small. Then when the step size is small, the samples from SGHMC will be close to the exact posterior.

To show the connection of our CVA algorithm with SGHMC, we set the global parameter of posterior standard deviation $\alpha = kl_t^{3/4}$ for some positive k and $\beta_1 = 1 - hl_t^{1/2}$ for some positive h, then based on the following two assumptions:

- the learning rate is small and $l_t \approx l_{t-1}$
- τ is converged

The constrained variational Adam algorithm is equivalent to the following dynamic

$$\begin{aligned} d\boldsymbol{w}_t &= D_{\boldsymbol{\tau}}\boldsymbol{r}_t dt \\ d\boldsymbol{r}_t &= -\beta_2\boldsymbol{g}_t dt - h\boldsymbol{r}_t dt + \sqrt{2+2\beta_1^2}k d\boldsymbol{B}_t \end{aligned} \tag{4.13}$$

where $\boldsymbol{g}_t = -\frac{1}{B}\sum_{i\in\mathcal{B}} \nabla_{\boldsymbol{w}_t} \log(\mathcal{D}_i|\boldsymbol{w}) + \frac{\delta}{N}\boldsymbol{\mu}$ and $D_{\boldsymbol{\tau}}^{-1} = \mathbf{Diag}(\boldsymbol{\tau}^{-1})$ is a preconditioning matrix. By setting $CM^{-1} = h$, $D_\tau = M^{-1}$, $\sqrt{1+\beta_1^2} = \sqrt{C-B}$ and $k^2 = \alpha$, we obtain the parametric form of SGHMC. In C.3, we derive the dynamic 4.13 by setting $\sqrt{l_t} = \Delta t$. The above dynamic produces the distribution proportional to $\exp\left(-\frac{\beta_2 L(\mathbf{w})}{k^2}\right)$ where

$L(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|x_i,\mathbf{w}) - \frac{1}{N}\log p(\mathbf{w})$. By setting $\beta_2 = 1$ and $k = \frac{1}{\sqrt{N}}$, we recover the posterior of the weight $p(\mathbf{w}|\mathcal{D})$.

Based on this connection, it is natural to consider using the cyclical learning rate [Loshchilov and Hutter, 2016, Huang et al., 2017, Zhang et al., 2019] with a high-to-low learning rate in each cycle. The stepsize at iteration k is defined as:

$$\alpha_k = \frac{\alpha_0}{2}\left[\cos\left(\frac{\pi \bmod (k-1, \lceil K/M \rceil)}{\lceil K/M \rceil}\right) + 1\right]$$

where α_0 is the initial stepsize, M is the number of cycles and K is the number of total iterations. A high learning rate at the beginning can quickly find a local mode, where the algorithm at this stage is doing optimization and a small learning rate at the end ensures an accurate simulation around the mode. By defining $\boldsymbol{v}_t = \boldsymbol{r}_{t-1}\Delta t$ as in [Chen et al., 2014], the numerical scheme is shown in algorithm 6.

Remark: The implementation of constrained variational Adam(CVA) algorithm is always based on CVA2 in algorithm 6 with cyclical learning rate. This is because hyper-parameter tuning is much easier in this setting. Another reason is that, provided that we use some decreasing learning rate scheme [Chen et al., 2015], we can enjoy some theoretical guarantees from SGHMC such as the asymptotic unbiasedness of the mean of the approximate posterior. As shown by [Zhang et al., 2019], the cyclical learning rate satisfies the conditions from [Chen et al., 2015].

4.1.1.3 Discussion of cold posterior

From the Gaussian variational inference perspective, α is an annealing parameter, which controls the temperature in the annealing scheme. Unlike the original Bayesian back-propagation [Blundell et al., 2015], we have introduced an extra inductive bias into the approximate posterior by only training the local variance parameter τ and defining the global variance parameter α.

When the size of the model is large, to improve the predictive performance, we use the cold posterior by setting $k = \frac{1}{N}$ rather than $\frac{1}{\sqrt{N}}$. In this case, algorithm 6 will converge

to posterior, which proportional to

$$\exp\left(-\frac{\log p(\mathbf{w}|\mathcal{D})}{T}\right) \tag{4.14}$$

where $T = \frac{1}{N}$ is the temperature. This setting implies that, in high dimensions, the magnitude of posterior variance for individual weight should be very small.

In fact, the purpose of constraining $\tau_i \in (0, 1)$ and using α as annealing parameter is not just for addressing the diverge of adaptive learning rate problem in the early stage of optimization. The usage of annealing parameter α allows us to construct a cold posterior such as 4.14.

Recently, [Wenzel et al., 2020] explored the effect of the cold posterior in Bayesian neural networks. They showed that using the SGHMC to sample the exact posterior $p(\mathbf{w}|\mathcal{D})$ yields systematically worse predictions compared to simple SGD. They further showed that predictive performance is improved significantly through the use of a cold posterior. There are lots of literature on Bayesian neural networks that used cold posterior to support their results. Both from stochastic gradient MCMC side [Li et al., 2016, Heek and Kalchbrenner, 2019, Leimkuhler et al., 2019, Zhang et al., 2019] and Variational inference side [Zhang et al., 2018, Bae et al., 2018, Osawa et al., 2019, Ashukha et al., 2020]. In addition, most of the temperature used in these papers is very low. Paper from [Izmailov et al., 2021] argued against the finding from [Wenzel et al., 2020] argued that the cold posterior is not needed for good predictive performance. However, their numerical result is not convincing. Indeed another paper [Zhang et al., 2019] from the same group used the cold posterior. In fact, the temperature we used in 4.14 follows their approach.

The necessity of using cold posterior to match the predictive performance of SGD is quite annoying. Because in this case, the posterior distribution will concentrate to a point mass. It is not clear to what extent, the uncertainty estimation from posterior probabilities is reliable. It seems that a cold posterior is not necessary when the size of the neural network is small. For example, in fitting the nonlinear regression problem in the experiment, we used a fully connected neural network. However, for convolutional neural networks, the cold posterior appears to be the key to obtaining good predictive accuracy. We found that

when the size of CNN is fairly large, the vanilla variational inference [Blundell et al., 2015] without any tempering will fail to converge during training. So far, the theoretical study of analysis of the behaviour of cold posterior for large-size neural networks doesn't exist.

Algorithm 6 CVA2

Input $\beta = (\beta_1, \beta_2)$, δ(weight decay)
N(training size), B(batch size)
Initialization $\boldsymbol{\rho}$,$\mathbf{w}$,$\boldsymbol{v}$
for $t = 1 \cdots$ **do**
$\Delta t \leftarrow s_1(t)$ ▷ learning rate schedule
$\eta \leftarrow s_2(t)$ ▷ learning rate schedule
$k \leftarrow s_3(t)$ ▷ annealing schedule
$\alpha \leftarrow k * \Delta t^{1.5}$
$\boldsymbol{g} \leftarrow -\frac{1}{B}\sum_{i\in\mathcal{B}} \nabla_{\mathbf{w}} \log(\mathcal{D}_i|\mathbf{w}) + \frac{\delta}{N}\mathbf{w}$
$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + (\frac{\eta}{\boldsymbol{\tau}} - \eta\alpha^2\delta\boldsymbol{\tau}) - \eta\alpha\boldsymbol{\epsilon} \odot \boldsymbol{g}$
$\boldsymbol{\epsilon} \leftarrow N(0, \mathbf{1}_p)$
$\boldsymbol{\tau} \leftarrow 1/(1 + e^{-\boldsymbol{\rho}})$
$\boldsymbol{v} \leftarrow \beta_1\boldsymbol{v} - \beta_2\boldsymbol{g}(\Delta t)^2 + \sqrt{2 + 2\beta_1^2}k(\Delta t)^{1.5}\boldsymbol{\epsilon}$
$\mathbf{w} \leftarrow \mathbf{w} - \boldsymbol{\tau} \odot \boldsymbol{v}$
end for

4.1.2 Pruning

In this section, we introduce an extra binary latent variable γ to control the Gaussian prior to the weight in the neural network to have a small and large variance. This construction resembles the continuous spike and slab Gaussian prior as we discussed in chapter 1. We generalized our CVA algorithm to the Variational EM algorithm (we refer to it as CVA-EM), which allows us to update these latent variables to control the shrinkage of the weights. A small variance of the prior will produces a strong shrinkage of the weight towards zero and vice versa. Since this will not produce the exact sparsity, some simple magnitude pruning rules will be used during training to obtain a sparse neural network.

We implement structure pruning by applying the group spike-and-slab prior to the weight parameters in the neural network. For simplicity, we ignore the subscript for the layers and consider the weights between any two layers. For convolutional layer, let $w_{ij} = (w_{ij1}, ..., w_{ijK^2})$ denote the group of K^2 parameters from the ith input channel to jth output channel, and $K \times K$ is the size of the kernel. Then the prior for w_{ijk} conditioned on a binary inclusion parameter γ_{ij} for the entire group of weights, follows an independent normal distribution such that:

$$\pi\left(w_{ijk} \mid \gamma_{ij}\right) = \begin{cases} \mathrm{N}\left(0, \delta_1^{-1}\right) & \text{if } \gamma_{ij} = 1 \\ \mathrm{N}\left(0, \delta_0^{-1}\right) & \text{otherwise,} \end{cases} \tag{4.15}$$

where $\pi(\gamma_{ij})$ are i.i.d binary distribution such that

$$\pi\left(\gamma_{ij}\right) = \begin{cases} 1 - p_{ij} & \text{if } \gamma_{ij} = 1 \\ p_{ij} & \text{if } \gamma_{ij} = 0. \end{cases}$$

The basic assumption is that a priori, within the same kernel all the weight parameters have the same distribution. For a fully connected layer, we can also make groupings based on whether they share the same input or output unit, for instance, if a group is made based on the input unit, then this is similar to variable/feature selection. Alternatively, we can assign the spike-and-slab-Gaussian prior to all the weights individually, this is related to unstructured pruning.

We now derive the CVA-EM algorithm that can return a MAP estimator of γ_{ij}.

E-Step: First, we run the CVA/CVA2 algorithm to obtain the approximate posterior of weight $q(\mathbf{w} \mid \gamma^{(t-1)}, \mathbf{y})$/the sample from $q(\mathbf{w} \mid \gamma^{(t-1)}, \mathbf{y})$. Then we can write down the objective function $Q\left(\gamma_{ij} \mid \gamma_{ij}^{(t-1)}\right)$ at the tth iteration as the integrated logarithm of the

full posterior with respect to w_{ij}

$$\begin{aligned} Q\left(\gamma_{ij} \mid \gamma_{ij}^{(t-1)}\right) &= \mathbf{E}_{q(\mathbf{w}|\gamma_{ij}^{(t-1)},\gamma_{-(ij)}^{(t-1)},\mathbf{y})} \log \pi(\boldsymbol{\gamma}, \mathbf{w} \mid \mathbf{y}) \\ &= \mathbf{E}_{q(\mathbf{w}|\gamma_{ij}^{(t-1)},\gamma_{-(ij)}^{(t-1)},\mathbf{y})} \log \pi(y \mid w) \\ &- \frac{1}{2} E_{q(\mathbf{w}|\gamma_{ij}^{(t-1)},\gamma_{-(ij)}^{(t-1)},\mathbf{y})} \sum_k [(\delta_1 - \delta_0)\gamma_{ij} + \delta_0] w_{ijk}^2 \\ &+ \frac{K^2}{2} \log[\delta_0 + (\delta_1 - \delta_0)\gamma_{ij}] + \log \pi(\gamma_{ij}) + C \end{aligned}$$

M-Step: Set $\gamma_{ij} = 0$, if $Q\left(\gamma_{ij} = 0 \mid \gamma_{ij}^{(t-1)}\right) \geq Q\left(\gamma_{ij} = 1 \mid \gamma_{ij}^{(t-1)}\right)$,that is,

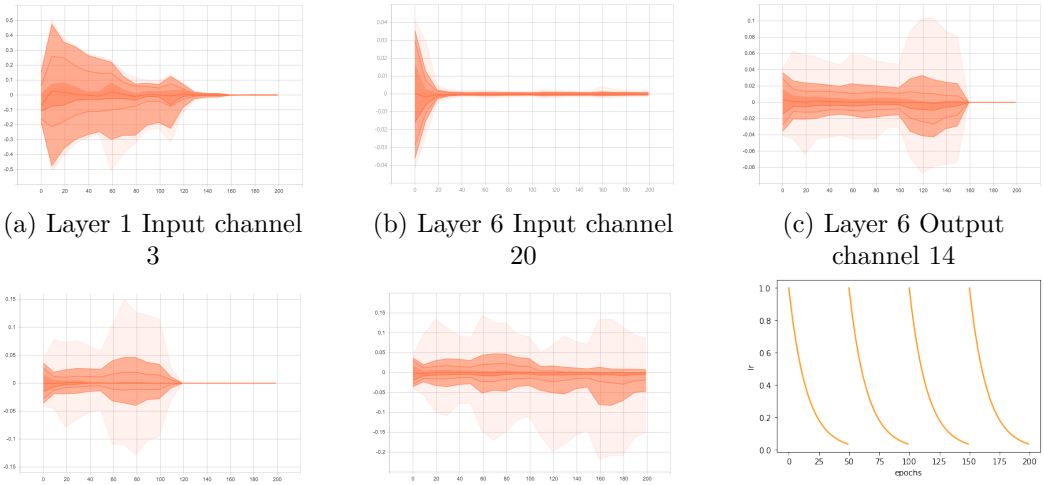
$$\frac{E_{q(\mathbf{w}|\gamma_{ij}^{(t-1)},\gamma_{-(ij)}^{(t-1)},\mathbf{y})}[\sum_k w_{ijk}^2]}{K^2} \leq \frac{1}{\delta_0 - \delta_1} \left[\log \frac{\delta_0}{\delta_1} + \frac{2}{K^2} \log \frac{p_{ij}}{1 - p_{ij}}\right] = \lambda_1$$

and 1 otherwise. Since the term $\log\left(\frac{p_{ij}}{1-p_{ij}}\right) \in \mathbf{R}$ and $0 < p_{ij} < 1$ is the hyper-parameter,instead of turning p_{ij}, we can turn the threshold λ_1 on the right hand side directly. From CVA2 algorithm, we have $E_{q(\mathbf{w}|\gamma_{ij}^{(t-1)},\gamma_{-(ij)}^{(t-1)},\mathbf{y})}[\sum_k w_{ijk}^2] \approx \sum_k w_{ijk}^2$.

The CVA-EM algorithm will adaptively change the weight decay factor for each group based on the magnitude. A large weight decay factor will be assigned to the group of weights with small magnitude via $\frac{\delta_0}{N}$. This will further push them toward zeros. On the contrary, the group with a strong signal will assign a small weight decay factor, via $\frac{\delta_1}{N}$.

Since the strong shrinkage from the spike part of the prior produces no exact zeros, a hard threshold is required to prune the weight and maintain sparsity. Recently there are a large number of dynamic pruning methods that maintain sparse weights through a prune-regrowth cycle during training [Zhu and Gupta, 2017, Bellec et al., 2017, Dettmers and Zettlemoyer, 2019, Mostafa and Wang, 2019, Lin et al., 2020, Kusupati et al., 2020]. Some of these methods can be inserted into our framework directly. For example, we can incorporate dynamic pruning with feedback (DPF) [Lin et al., 2020], which evaluates a stochastic gradient for the pruned model $\widetilde{\mathbf{w}}_t = \mathbf{m}_t \odot \mathbf{w}_t$ (where $\mathbf{m}_t$ is a mask matrix) and apply it to the (simultaneously maintained) dense model

$$\mathbf{w}_{t+1} := \mathbf{w}_t - \gamma_t \mathbf{g}\left(\mathbf{m}_t \odot \mathbf{w}_t\right) = \mathbf{w}_t - \gamma_t \mathbf{g}\left(\widetilde{\mathbf{w}}_t\right) \tag{4.16}$$



(a) Layer 1 Input channel 3

(b) Layer 6 Input channel 20

(c) Layer 6 Output channel 14

(d) Layer 12 Output channel 50

(e) Layer 16 Output channel 100

(f) Cyclical learning rate

Figure 4.1: Sample paths of all weights from select channels of ResNet18 in CIFAR10 during training with cyclical learning rate l_t and $k = \frac{1}{N}$ from algorithm 2. (a)-(e) are grouped by either input and output channels, (f) is the cyclical learning rate used.

and $\mathbf{g}(\mathbf{m}_t \odot \mathbf{w}_t)$ is the gradient evaluated at the sparse model. Although these methods are one-shot (no retraining required), some computational budget for weight regrow is still required.

For linear models, it has been argued that forward or backward selection strategy may be trapped into a bad local mode [Hastie et al., 2009], thus a variable selection algorithm allowing parameter regrow is preferred.

However, [Gur-Ari et al., 2018] argued that only a tiny subspace of the parameters has non-zero gradients. Empirically, when we run the CVA2 algorithm, we find that the gradients will vanish everywhere except in a small number of input and output channels. In addition, during optimization, many of the parameters in the input and out channels will concentrate to zero. This concentration is quite stable even under the two following perturbations from the algorithm:

1. A cyclical learning rate, which can jump from small to large. See Figure 4.1-(f).
2. At each iteration, adding an injected noise $\sqrt{2 + 2\beta_1^2 k(\Delta t)^{1.5}}\boldsymbol{\epsilon}$.

We observe the concentration phenomenon by visualizing the dynamic distributions of the

weights in Figure 4.1, which shows the sample paths of all weights from select channels during optimization. Note that, at this stage, we just ran the CVA2 with a cyclical learning rate. We can see that the contractions occur regardless of whether the grouping is based on input or output channels. In addition, we rarely observe a regrowth of the weight once the whole channel concentrates to zero, see Figure 4.2. This phenomenon is very similar to the posterior contraction in high dimensional sparse Bayesian linear regression [Castillo et al., 2015], where only a sub-model has substantial probability mass. Finally, Figure 4.1(e) shows the behaviour of weights in those channels which are not concentrated around zero.

Motivated by the above observations, we propose a structured pruning rule based on a simple concentration metric, where, within each layer, two loops will be used to scan through all the input channels and output channels.

$$\begin{aligned} &\text{If } \max_{j,k} |W_{ijk}| - \min_{j,k} |W_{ijk}| < \lambda_2 \text{ set } W_i = 0; \\ &\text{If } \max_{i,k} |W_{ijk}| - \min_{i,k} |W_{ijk}| < \lambda_2 \text{ set } W_j = 0; \end{aligned} \tag{4.17}$$

where $\lambda_2 > 0$ is a hard threshold which we need to tune. Apart from this concentration metric, we further suggest using an aggressive pruning strategy, dynamic forward pruning (DFP). That is once the weight has been pruned, it will necessarily be excluded from the final selected model.

Algorithm 7 provides the pseudo-code for implementation. The SoftMask in algorithm 7 will control the shrinkage of the weight. It is used to separate the noise and signal as in sparse linear regression [Castillo et al., 2015]. The group of weights with very small magnitude will be identified as noise and a strong shrinkage (a large weight decay) will be applied to it in the next iteration and vice versa. The HardMask returned by the pruning criterion rule will remove the weight. Although we recommend using DFP, where regrowth is not allowed, we also provide the user with DPF as another option in Algorithm 7. Figure 4.2 shows no significant differences between the DFP and DPF, when comparing their test accuracy and sparsity ratios.

Remark: The pruning criterion from equation (4.17) assumes that if the weights within the input channels or output channels concentrate together, it will concentrate to zero.

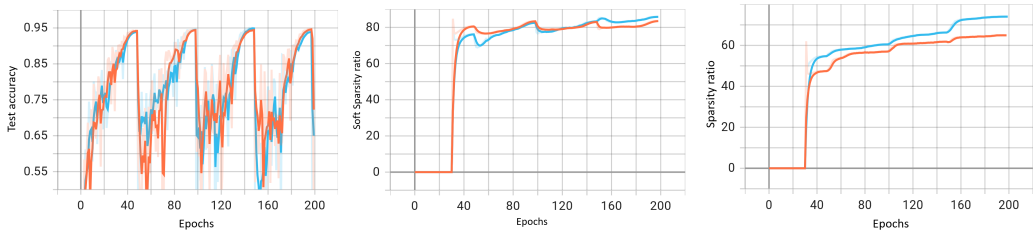


Figure 4.2: Resnet18 pruned using algorithm 3 with the cyclical learning rate under DFP (blue) and DPF (orange), under the same hyper-parameter setting. The left panel shows testing accuracy. The middle panel shows the percentage of the weights, which use a large weight decay factor as suggested by CVA-EM algorithm (soft sparsity ratio). The right panel shows the sparsity ratio of the model during training.

However, unlike CNN, we didn't observe this in feedforward neural networks, we find that even when the problem is known to be sparse, criterion 4.17 can lead to the failure of convergence. Instead, we propose the following L_2 magnitude pruning criterion for fully connect layer:

$$\text{If } \sum_{i=1}^{k_o} w_{ij}^2/k_o \leq \lambda_2, \text{set } w_j = 0 \tag{4.18}$$

where i is output units, j is input units and k_o is the number of output units.

4.1.3 Experiments

4.1.3.1 Experimental setup for image classification task

In this section, we conduct experiments to show the effectiveness of our proposed algorithms in terms of the test accuracy and sparsity ratio by comparing them against both dense and sparse methods in the literature for image classification. Apart from the baseline, which is SGD with momentum, we include two state-of-art Bayesian methods: Rank-1 BNN [Dusenberry et al., 2020], a variational inference approach with batch ensemble [Wen et al., 2020], and an MCMC approach cSGLD [Zhang et al., 2019]. We also include three popular dynamic sparse training methods: SM [Dettmers and Zettlemoyer, 2019]; DSR [Mostafa and Wang, 2019] and DPF [Lin et al., 2020]. **We repeat each experiment three times and average them to obtain stable results**.

Algorithm 7 CVA-EM

Input $\beta = (\beta_1, \beta_2)$, (δ_0, δ_1)(weight decay), N(training size), B(batch size)
λ_1(soft threshold), λ_2(hard threshold), pruning with feedback = False

Initialization $\boldsymbol{\rho}, \boldsymbol{w}, \boldsymbol{v}$, SoftMask $\leftarrow 1$, HardMask $\leftarrow 1$

for $t = 1 \cdots$ **do**

$\delta \leftarrow \delta_1$

δ[SoftMask == 0] $\leftarrow \delta_0$

$\Delta t \leftarrow s_1(t)$; $\eta \leftarrow s_2(t)$; $k \leftarrow T(t)$;

$\alpha \leftarrow k * \Delta t^{1.5}$

$\boldsymbol{g} \leftarrow -\frac{1}{B} \sum_{i \in \mathcal{B}} \nabla_{\mathbf{w}} \log(\mathcal{D}_i|\mathbf{w}) + \frac{\delta}{N}\mathbf{w}$

$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + (\frac{\eta}{\boldsymbol{\tau}} - \eta\alpha^2\delta\boldsymbol{\tau}) - \eta\alpha\boldsymbol{\epsilon} \odot \boldsymbol{g}$

$\boldsymbol{\tau} \leftarrow 1/(1 + e^{-\boldsymbol{\rho}})$

$\boldsymbol{\epsilon} \leftarrow N(0, \mathbf{1}_p)$

$\boldsymbol{v} \leftarrow \beta_1 \boldsymbol{v} - \beta_2 g(\Delta t)^2 + \sqrt{2 + 2\beta_1^2} k(\Delta t)^{1.5}\epsilon$

$\mathbf{w} \leftarrow \mathbf{w} - \tau \odot \boldsymbol{v}$

if $t >$ warm up **then**

SoftMask $\leftarrow$ **EM**$(\mathbf{w}, \lambda_1)$

HardMask $\leftarrow$ **Pruning Criterion**$(\mathbf{w}, \lambda_2)$

$\mathbf{w}$[HardMask] $\leftarrow 0$

if pruning with feedback is False **then**

Freeze $\mathbf{w}$[HardMask]

end if

end if

end for

The Rank-1 BNN has some advantages in this experimental comparison as its training approach is more computationally intensive. Since we don't have enough TPU to implement their method, we extracted their experimental result from the original paper as indicated by * in the table.

For the other methods, we use 200 epochs to train models in CIFAR10 and CIFAR100 datasets with a single RTX3090 node and 200 epochs to train models in ImageNet dataset with a single A100 node. The cyclical learning rate with 4 cycles as shown in Figure 4.1-(f) is used for cSGLD [Zhang et al., 2019] and CVA2. We collect three samples at the end of each cycle, which gives us 12 samples in total for CIFAR10 for CIFAR100 datasets, while for ImageNet, the first 50 epochs are used as a warm-up, therefore, we only collect 9 samples. The posterior predictive distribution is calculated by averaging all the samples as an ensemble.

Since DPF (4.16) has no official code release, but is easy to insert into CVA-EM Algorithm, we implement it by ourselves. With two different pruning rules we can use in CVA-EM, we have the CVA-EM-DFP algorithm and the CVA-EM-DPF algorithm. For short, we still refer them to DFP and DPF in the experiment. For the baseline model, we follow the hyper-parameter settings from [Xie et al., 2022]. Their setting allows us to train a strong baseline for comparison.

4.1.3.2 Results for image classification task

	Dense Method				Sparse Method				
Model	Baseline	Rank-1 BNN	cSGLD	CVA2	SM	DSR	DFP	DPF	Sparsity ratio
ResNet18	95%	-	95.7%	95.5%	92.8%	93.1%	94.7%	94.8%	70%
ResNet18	95%	-	95.7%	95.5%	91.1%	91.2%	94.5%	94.5%	80%
ResNet18	95%	-	95.7%	95.5%	89.7%	90.0%	93.9%	93.8%	90%
WRN-28-10	96.3%	96.5%*	96.2%	96.5%	95.4%	95.8%	95.5%	95.8%	90%
WRN-28-10	96.3%	96.5%*	96.2%	96.5%	95.3%	95.4%	95.3%	95.4%	95%
WRN-28-10	96.3%	96.5%*	96.2%	96.5%	94.5%	94.5%	94.7%	94.5%	99%

Table 4.1: Comparison of test accuracy with different target sparsity ratios in CIFAR10.

Performance for CIFAR10 and CIFFAR100 dataset: We report the testing accu-

Model	Dense method				Sparse method				
	Baseline	Rank-1 BNN	cSGLM	CVA2	SM	DSR	DFP	DPF	Sparsity ratio
ResNet34	78.5%	-	79.7%	79.4%	76.4%	76.6%	77.6%	77.4%	50%
ResNet34	78.5%	-	79.7%	79.4%	73.2%	73.8%	75.4%	75.1%	70%
ResNet34	78.5%	-	79.7%	79.4%	72.1%	72.3%	75.1%	75.3%	90%
WRN-28-10	81.7%	82.4%*	82.7%	83.4%	78.0%	78.1%	78.5%	78.2%	70%
WRN-28-10	81.7%	82.4%*	82.7%	83.4%	77.0%	76.5%	77.7%	77.9%	80%
WRN-28-10	81.7%	82.4%*	82.7%	83.4%	76.5%	76.4%	77.5%	77.4%	90%

Table 4.2: Comparison of test accuracy for different target sparsity ratios for in CIFAR100

Model	Dense method				Sparse method				
	Baseline	Rank-1 BNN	cSGLM	CVA2	SM	DSR	DFP	DPF	Sparsity ratio
ResRet50	76.5%	77.3%*	76.7%	76.8%	73.8%	73.3%	74.5%	74.6%	80%
ResNet50	76.5%	77.3%*	76.7%	76.8%	72.3%	72.0%	73.5%	73.2%	90%

Table 4.3: Comparison of test accuracy for different target sparsity ratios for ImageNet

racy and sparsity ratio for the sparse method in Table 1 and Table 2. For the dense method, compared with two SOTA Bayesian methods and baseline, the CVA2 of Algorithm 1 is very competitive in both CIFAR10 and CIFAR100 datasets.

When pruning is applied, there is almost no performance loss with both DFP and DPF approaches compared with the dense methods in ResNet18, and a much higher sparsity ratio can be achieved by WRN-28-10 in the CIFAR10 dataset. For CIFAR100 dataset, we found that it is harder to target the high sparsity ratio while at the same time keep the performance loss negligible. Overall, our pruning algorithms can achieve high sparsity ratio while sacrificing a small amount in accuracy.

Finally, there is no evidence that using dynamic forward pruning with (no regrowth) suffers performance loss compared to DPF.

Performance for ImageNet: As shown in Table 3, CVA2 produced a modest performance gain compared with the baseline but performed worse than Rank-1 BNN. For cSGLM [Zhang et al., 2019], we failed to reproduce the 77% predictive accuracy as claimed in the original paper. We believe the performance for both our CVA2 and cSGLM can be further improved by tuning the hyper-parameters carefully. For the sparse method, our pruning algorithms outperformed the other two methods SM and DSR. Again, there is

	Example 1				Example 2				Example 3			
Method	MSE	FDR	FNDR	$\hat{S}$	MSE	FDR	FNDR	$\hat{S}$	Accuracy	FDR	FNDR	$\hat{S}$
SPLBNN	1.4	0	0	4	2.43	0	0	5	91.2%	0	0	5
SVBNN	1.6	38.3%	0	6.3	3.12	51%	0	10.2	86%	39.9%	0	8.2
DFP	1.32	0	0	4	2.46	0	0	5	94.70%	0	0	5
DPF	1.29	0	0	4	2.49	0	0	5	94.50%	0	0	5

Table 4.4: Comparison of sparse methods for three simulated examples

still no evidence that using DFP will incur performance loss.

4.1.3.3 Simulated Examples: variable selection for nonlinear regression

[Sun et al., 2021] applied the Spike-and-Slab Gaussian prior to pruning the neural network (SPLBNN). They use a Laplace approximation-based approach to approximate the marginal posterior inclusion probability, which requires the user to train the dense model to find the local mode first followed by pruning and retraining the sparse model, finally they use the Bayesian evidence to elicit sparse DNNs in multiple runs with different initialization. Another related approach is variational BNN with Spike-and-Slab prior [Bai et al., 2020a] (SVBNN). Both approaches are not scalable to large models, but they work well in high dimensional sparse nonlinear regression, so we carry out a comparison with these two approaches in this setting. Here, we follow the same data-generating process and examples as described in [Sun et al., 2021]:

- Simulate $e, z_1, ..., z_p$ independently from the truncated standard normal distribution on the interval $[-10, 10]$.
- Set $x_i = \frac{e+z_i}{\sqrt{2}}$ for $i = 1, ..., p$

Then, all the predictors x_i fall into a compact set and are mutually correlated with a correlation coefficient of about 0.5. Based on this setting, we generate three toy examples from [Sun et al., 2021] where each example consists of 10000 training samples and 1000 testing samples. To fit the data and make a comparison, we follow the network structure described in [Sun et al., 2021].

Example 1:

$$y = \tanh\left(2\tanh\left(2x_1 - x_2\right)\right) + 2\tanh\left(\tanh\left(x_3 - 2x_4\right)\right.$$
$$\left. - \tanh\left(2x_5\right)\right) + +0x_6 + \cdots + 0x_{1000} + \varepsilon,$$

Network Structure: 1000-5-3-1 with ReLu activation

Example 2:

$$y = \frac{5x_2}{1 + x_1^2} + 5\sin\left(x_3 x_4\right) + 2x_5 + 0x_6 + \cdots + 0x_{2000} + \varepsilon$$

Network Structure: 2000-6-4-3-1 with ReLu activation

Example 3:

$$y = \begin{cases} 1 & e^{x_1} + x_2^2 + 5\sin\left(x_3 x_4\right) + 0x_5 + \cdots + 0x_{1000} > 3 \\ 0 & \text{otherwise.} \end{cases}$$

Network Structure: 2000-6-4-3-1 with ReLu activation

where $\epsilon \sim N(0, 1)$. We only compare our algorithm with the method from [Sun et al., 2021] and [Bai et al., 2020a]. The other two sparse methods we used in the image task didn't work in these three examples. We use equation (4.18) as the pruning criterion in Algorithm 3. The performance of variable selection is measured by the false discover rate (FDR) and false non-discover rate (FNDR). The predictive mean square error is used for examples 1 and 2 and the prediction accuracy is used for example 3. $\hat{S}$ is the number of the selected variables returned by the competing methods. We repeat the experiments 10 times and report the averaged results in Table 4.

Performance analysis: In terms of variable selection, SPLBNN, DFP and DPF perfectly selected all the relevant variables and removed all irrelevant variables in all three examples. SVBNN always selected more variables but was able to keep the FNDR at zero. The predictive performance for examples 1 and 3 are good for SPLBNN, DFP and DPF. But for example 2, the mean square error is high (compared to the oracle value of 1), A better network structure may improve this performance. One of the surprising results is that

DFP can still identify all relevant predictors and was never trapped in a bad local mode. Compared with the SPLBNN from [Sun et al., 2021], which requires the train-prune-fine tune process multiple times, our algorithm is very aggressive and efficient, it outperforms the SPLBNN in examples 1 and 3.

4.2 Extentsion to sparse graph neural network

4.2.1 Basic background

Graph neural networks (GNNs) have achieved state-of-the-art results in processing data that has a geometric structure and can be represented as a graph. The key design element of GNNs is the use of so-called message passing, where information (features) on the graph nodes will be updated iteratively by exchanging information with their neighbours. The updated information will be aggregated and embedded by feed-forward propagation. Many variants of GNN have been proposed and have achieved state-of-the-art results on both node and graph classification tasks [Welling and Kipf, 2016, Velickovic et al., 2017, Xu et al., 2018].

The performance of GNN highly depends on the quality of the graph. Noise from the graphs often leads to unsatisfactory representations and prevents us from fully understanding the mechanism underlying the system. Recent researches suggest that a small perturbation in graph structure can easily result in wrong predictions for most GNNs [Dai et al., 2018, Zhu et al., 2019, Zhang and Zitnik, 2020]. One possible reason is that a small noise will be propagated to neighbourhoods by the message-passing rule from GNN, which magnifies the impact of the noise. Thus, to address this issue, we need some graph regularization techniques. A natural solution is to put the sparsity both on the edge of the graph and the features on the nodes. The former corresponds to a sparse representation of the graph and the latter is the structure pruning of the neural network.

For the sparse representation of the graph, the best solution is to put a ℓ_0 penalty to the

adjacency matrix of the graph. But this will lead to the NP-hard problem. There exist Bayesian approaches [Ye and Ji, 2021,Elinas et al., 2020], which mask the nonzero elements from the adjacency matrix with binary random variables. The variational inference has been used to find the posterior of the binary mask and the weights. However, since the reparameterization trick is not applicable to discrete distributions, a continuous relaxation trick [Jang et al., 2016,Maddison et al., 2016] is required to further approximate the binary random variable.

Recently, [Chen et al., 2021] introduce unconstraint and trainable masks to both nonzero elements in the adjacency matrix and weights. The ℓ_1 penalty has been assigned to the masks during training and the magnitude pruning rule has been used after training to identify the important connections in the graphs and weights in GNNs. The train-pruning process will iterate several times until the target sparsity has been reached. With this motivation, we found that the sparse Bayesian neural network approach we discussed in section 4.1 can be extended to graph neural networks with negligible modification.

4.2.2 Problem formulations

We define the graph as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ represents the vertex set consisting of nodes $\{v_1, \ldots, v_n\}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is an adjacency matrix, which describes the graph topology. If the edge weight between nodes v_i and v_j exists, then $A_{ij} = 1$, otherwise $A_{ij} = 0$. We define the degree matrix $\mathbf{D} = \text{diag}\,(d_1, \ldots, d_n)$ as a diagonal matrix where each entry on the diagonal is equal to the row-sum of the adjacency matrix. Each node v_i in the graph has a p-dimensional feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$. By stacking n feature vectors from all the nodes, we have a feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. Each node belongs to one of C classes and can be represented by a one-hot vector $y_i \in \{0, 1\}^C$.

For example, the two-layer graph convolutional network can be defined as

$$f(\mathbf{A}, \boldsymbol{X}) = \sigma_2 \left(\hat{\boldsymbol{A}}_2 \sigma_1 \left(\hat{\boldsymbol{A}}_1 \boldsymbol{X} W^{(1)} \right) W^{(2)} \right)$$

where $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ are activation function and $\hat{\boldsymbol{A}}_l$ is the message passing matrix on

layer l that depends on the adjacency matrix $\mathbf{A}$. In general, the message-passing matrix $\hat{\boldsymbol{A}}_l$ can also depend on the weight $W^{(l)}$ and some extra trainable parameters [Velickovic et al., 2017]. For the semi-supervised classification task, we have the cross-entropy loss function

$$\mathcal{L}(\mathbf{A}, \boldsymbol{W}) = -\frac{\sum_{i \in \mathcal{V}_{\text{label}}} y_i^T \log f(\mathbf{A}, \boldsymbol{x}_i)}{|\mathcal{V}_{\text{label}}|}$$

where $|\mathcal{V}_{\text{label}}|$ are the number of the nodes which are labelled. The approach from [Chen et al., 2021] has the objective function

$$\underset{\boldsymbol{W}, m_g, m_w}{\arg\min}\, \mathcal{L}(\mathbf{A} \odot m_g, \boldsymbol{W} \odot m_w) + \lambda_1 \|m_g\|_1 + \lambda_2 \|m_w\|_1$$

where m_g and m_w are unconstraint and trainable masks with the same shape as $\boldsymbol{A}$ and $\boldsymbol{W}$. Unlike the approach from [Chen et al., 2021], the CVA-EM algorithm presented in section 4.1 only requires the mask parameter m_g for the adjacency matrix. In fact, we can't see the reason to introduce the mask parameter m_w to the weight $\boldsymbol{W}$. The pruning can be done directly to $\boldsymbol{W}$. Therefore, we have the log-likelihood $\mathcal{L}(\mathbf{A} \odot m_g, \boldsymbol{W})$. The spike and slab prior will be assigned to both parameter m_g and $\boldsymbol{W}$. Running the CVA-EM algorithm, we obtain the sparse graph neural network.

4.2.3 Some numerical result

In this section, we will compare our approach with [Chen et al., 2021] by using three popular semi-supervised graph datasets: Cora, Citeseer and PubMed for node classification. Table 4.5 provides the basic statistics for these three datasets. Following the same network architecture used by [Chen et al., 2021], we stick with two-layer GCN/GIN/GAT networks with 512 hidden units to conduct all our experiments. The optimizer Adam [Kingma and Ba, 2014] has been used as a baseline method. In each experiment, we select the model, which achieves a high sparsity level and comparable predictive accuracy with the baseline method. We repeat experiments ten times and average them to obtain stable results. We refer to our approach as a Sparse Graph Neural Network (SGNN) and approach from [Chen et al., 2021] as Graph Lottery Ticket (GLT).

Here are our key observations:

Dataset	Nodes	Edges	Ave Degree	Features	Classes	Train	Validation	Test
Cora	2,708	5,429	3.88	1,433	7	140	500	1000
Citeseer	3,327	4,732	2.84	3,703	6	120	500	1000
PubMed	19,717	44,338	4.50	500	3	60	500	1000

Table 4.5: Summary of datasets used in the experiments. The nodes with labels have been split into training set, validation set and testing set.

Cora									
Model	GCN			GAT			GIN		
Method	Adam	SGNN	GLT	Adam	SGNN	GLT	Adam	SGNN	GLT
Accuracy	80.0%	79.8%	80.1%	80.0%	81.3%	81.8%	79.5%	80.2%	80%
Graph sparsity	NA	20.1%	18.6%	NA	40.0%	36.9%	NA	6.1%	5.0%
Weight sparsity	NA	65.0%	59.1%	NA	90%	86.5%	NA	22%	20%

Table 4.6: The test accuracy over achieved graph sparsity levels and weight sparsity level of GCN, GIN, and GAT on the Cora dataset

- We find that SBNN and GLT have comparable performances on all the datasets. No one always dominates the others.
- On Citesee and PubMed datasets, putting a sparsity on graphs and weights can improve the test accuracy. This implies that some connections in the graph are not important or even wrong and the model is over-parametrized.
- The difference in the test accuracy for three models in all three datasets is small, GAT slightly over-performs the other two.
- Even within the same dataset and the same network architecture, the graph sparsity and weight sparsity achieved by the three models are quite different.

4.3 Sparse variational autoencoder: a potential future work

Finally, we discuss the sparse VAE with $L_{\frac{1}{2}}$ prior, which could be a potential future work. Given observed samples x from an unknown distribution, the goal of a generative model is to approximate the unknown data distribution $p_{\mathcal{D}}(x)$ and generate the new samples. Furthermore, it is possible to use the learned model to estimate the likelihood ratio of observed or sampled data as well.

Citesee									
Model	GCN			GAT			GIN		
Method	Adam	SGNN	GLT	Adam	SGNN	GLT	Adam	SGNN	GLT
Accuracy	70.5%	70.2%	70.1%	70%	71.4%	71.1%	68.2%	71.2%	70.9%
Graph sparsity	NA	50.1%	48.7%	NA	8.2%	9.8%	NA	38.1%	37.0%
Weight sparsity	NA	93.1%	95.6%	NA	39.2%	36.0%	NA	90.1%	86.6%

Table 4.7: The test accuracy over achieved graph sparsity levels and weight sparsity levels of GCN, GIN, and GAT on the Citesee dataset

PubMed									
Model	GCN			GAT			GIN		
Method	Adam	SGNN	GLT	Adam	SGNN	GLT	Adam	SGNN	GLT
Accuracy	79.9%	79.6%	80.1%	78.5%	80.0%	79.7%	77.9%	79.0%	79.2%
Graph sparsity	NA	50.1%	48.7%	NA	45.1%	46.2%	NA	16.3%	13.5%
Weight sparsity	NA	60.2%	56.0%	NA	90.2%	91.4%	NA	63.8%	59.1%

Table 4.8: The test accuracy over achieved graph sparsity levels and weight sparsity levels of GCN, GIN, and GAT on the PubMed dataset

There are two directions of research. One is Generative Adversarial Networks [Goodfellow et al., 2020], which use adversarial training to minimise some distribution distance. Another class of methods is likelihood-based, which seeks to learn a model that assigns a high likelihood to the observed data samples. This includes autoregressive models, variational autoencoders [Kingma and Welling, 2013], energy-based models [Du and Mordatch, 2019] and, more recently, the very popular diffusion model [Ho et al., 2020], which is a score-based generative model and has been shown as a sub-class of Markovian Hierarchical VAE [Luo, 2022].

Apart from learning the data distribution $p_{\mathcal{D}}(x)$, obtaining an interpretable factorized representation of the independent data generative factors can be useful for a large variety of tasks and domains. The area of this study is referred to as disentangled representation learning [Higgins et al., 2016], where a single latent variable z is only sensitive to changes in single generative factors while being relatively invariant to changes in other factors. (e.g. factors in face image generation can refer to rotation, emotion, the colour of hair, etc.)

Variational auto-encoders (VAE) offer an efficient and elegant way of performing approximate intractable data distribution $p(x)$ and, with its by-product, we obtain an approximate

posterior distribution $q(z|x)$, which maps complicated high-dimensional data to lower dimensional latent variable as summary statistics. However, standard VAEs often produce $q(z|x)$ that are dispersed and lack interpretability. We try to address these issues by replacing the gaussian prior to sparse inducing prior to reducing the dimension of the latent space.

4.3.1 Variational Auto-encoders

Assuming that the data we observe are generated by an associated unseen latent variable z, then the data distribution can be written as

$$p_{\mathcal{D}}(x) = \int p(x \mid z)p(z)dz \tag{4.19}$$

VAE attempt to approximate this underlying process with a chosen model $p_\theta(x)$, with parameters θ.Then using the latent variable expansion we discussed above, we have

$$p_\theta(x) = \int p_\theta(x \mid z)p(z)dz \tag{4.20}$$

Directly maximizing the $p_\theta(x)$ is difficult because it involves integrating out the latent variables z in equation (4.20), which is intractable for the complex model. Now we derive evidence lower bound (ELBO) of $\log p_\theta(x)$, which can be used as a proxy objective in optimization.

$$\begin{aligned}
\log p_\theta(x) &= \log \int p_\theta(x, z)dz \\
&= \log \int \frac{p_\theta(x, z)q_\phi(z \mid x)}{q_\phi(z \mid x)}dz \\
&= \log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x, z)}{q_\phi(z \mid x)}\right] \\
&\geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x, z)}{q_\phi(z \mid x)}\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| p(z)\right) \\
&= \mathcal{L}_{ELBO}(x)
\end{aligned}$$

where $q_\phi(z \mid x)$ is a approximate posterior of latent variable z with parameter ϕ that seek to optimize. We further average this over the data distribution $p_{\mathcal{D}}(x)$ to obtain the final optimization objective

$$\mathcal{L}_{ELOB} = \mathbb{E}_{p_{\mathcal{D}(x)}}[\mathcal{L}_{ELBO}(x)] = \underbrace{\mathbb{E}_{p_{\mathcal{D}(x)}}[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)]]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{p_{\mathcal{D}(x)}}[D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))]}_{\text{prior matching term}}$$

The ELBO is composed of two terms; a reconstruction term, which measures how effective the decoder can recover the observation, and a prior matching term, which encourages the encoder really learn a useful message rather than collapse into the observation.VAE will maximize the ELBO above with respect to parameters θ and ϕ. More explicitly, VAE will jointly learn the conditional likelihood $p_\theta(x \mid z)$ to convert the latent variable z into an observation x as a decoder and an approximate posterior of latent space $q_\phi(z \mid x)$ as the encoder. Under ideal conditions, optimizing the ELBO objective using sufficiently flexible model families for $p_\theta(x \mid z)$ and $q_\phi(z \mid x)$ over θ, ϕ will achieve both goals of correctly capturing $p_{\mathcal{D}}(x)$ and performing correct amortized inference. However, with finite model capacity the two goals can be conflicting and subtle tradeoffs and failure modes can emerge from optimizing the ELBO objective.

4.3.2 Sparse coding via Variational EM algorithm

One way to introduce the sparsity into the latent space is to use sparse inducing prior. As before, we consider the $L_{\frac{1}{2}}$ prior, which can be represented as normal mixture distribution:

$$z_j | v_j \sim \mathrm{DE}\left(0, \frac{v_j}{\lambda^2}\right) \quad v_j \sim \mathrm{Gamma}\left(\frac{3}{2}, \frac{1}{4}\right) \quad j = 1, 2, ..., p$$

where $\mathrm{DE}\left(0, \frac{v_j}{\lambda^2}\right)$ denotes a Laplace (also known as double exponential distribution with mean 0 and variance $\frac{2v_j^2}{\lambda^4}$. $\lambda > 0$ is the global shrinkage parameter, which controls the overall sparsity level.

With Laplace mixture prior of latent variable z and $q(v) = \delta(v)$, the evidence lower

bound(ELBO) is

$$\begin{aligned}\mathcal{L}_{ELOB} &= \mathbb{E}_{p_{\mathcal{D}(x)}}[\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]] - \mathbb{E}_{p_{\mathcal{D}(x)}}[\mathbb{E}_{q(v)}[D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| \pi(z|v)\right)]] - D_{\mathrm{KL}}(q(v)\|\pi(v)) \\ &= \mathbb{E}_{p_{\mathcal{D}(x)}}[\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]] - \mathbb{E}_{p_{\mathcal{D}(x)}}[D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| \pi(z \mid v)\right)] + \log \pi(v)\end{aligned}$$

We further assume that the encoder $q_\phi(z \mid x)$ is Gaussian with diagonal covariance. In this case, the prior matching term can be computed analytically:

$$D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| p(z \mid \tau)\right) = \frac{1}{2}\sum_{j=1}^{p} \log \frac{v_j^2}{\sigma_j^2(x)} + \lambda^2 \sum_{j=1}^{p} E_{q(z_j|x)}\left[\frac{|z_j|}{v_j}\right] + C$$

We set $h_j = \frac{1}{v_j}$. Then by differentiating the ELBO with respect to h_j,

$$\frac{\partial \mathcal{L}_{ELOB}}{\partial h_j} = -\frac{N}{h_j} + \lambda^2 \sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|] + \frac{1}{4h_j^2} - \frac{1}{2h_j}$$

Setting $\frac{\partial \mathcal{L}_{ELOB}}{\partial h_j} = 0$, we have

$$\begin{aligned} v_j &= (2N+1) - \sqrt{(2N+1)^2 - 4\lambda^2 \sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|]} \\ &\approx 2\lambda \sqrt{\sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|]} \end{aligned} \tag{4.21}$$

where $E_{q(z_j|x_i)}[|z_j|] = \sigma_j(x_i) \cdot \sqrt{\frac{2}{\pi}} {}_1F_1\left(-\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}\left(\frac{\mu_j(x_i)}{\sigma_j(x_i)}\right)^2\right)$.

The variational EM algorithm is implemented in the follower manner:

E-Step: We update the parameter ϕ and θ based on the gradient descent type algorithm, where the gradient $\frac{\partial \mathcal{L}_{ELOB}}{\partial \phi}$ is estimated by the reparameterization trick.

M-Step: We update v based on equation 4.21.

We see that the smaller the $\sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|]$, the larger the ℓ_1 -type penalty we have for μ_j and σ_j. The gradient estimator from the reparameterization trick is equivalent to the gradient of adaptive ℓ_1 penalty.

$$\begin{aligned} \frac{\partial}{\partial \mu_j} E_{q(z_j|x)}\left[\frac{|z_j|}{v_j}\right] &\approx \frac{\operatorname{sign}(\mu_j + \sigma_j \epsilon)}{2\lambda\sqrt{\sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|]}} \\ \frac{\partial}{\partial \sigma_j} E_{q(z_j|x)}\left[\frac{|z_j|}{v_j}\right] &\approx \frac{\operatorname{sign}(\mu_j + \sigma_j \epsilon)\epsilon}{2\lambda\sqrt{\sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|]}} \end{aligned}$$

where $\epsilon \sim N(0,1)$. By the chain rule $\frac{\partial \mathcal{L}_{ELOB}}{\partial \phi} = \frac{\partial \mu_j}{\partial \phi}\frac{\partial \mathcal{L}_{ELOB}}{\partial \mu_j} + \frac{\partial \sigma_j}{\partial \phi}\frac{\partial \mathcal{L}_{ELOB}}{\partial \sigma_j}$, we see that these adaptive ℓ_1 penalty will finally transfer to the weight parameters ϕ in the encoder neural network, which connect to the output units μ_j and σ_j. Thus, putting a sparse prior on each z_j in the latent space is equivalent to giving a group shrinkage to weights in the output layer of the encoder neural network.

However, one problem is that even with a sparse prior, the gradient descent type algorithm can shrink parameters to a very small value but never produces an exact zero. A simple way to fix this issue is using a magnitude pruning rule after training. That is we set the jth dimension of latent space to zero if

$$\frac{1}{N}\sum_{i=1}^{N} E_{q(z_j|x_i)}[|z_j|] < \delta$$

where δ is the user-defined thresholding.

4.3.3 Discussion of the challenge

The idea of regularizing latent space of VAE arises from [Higgins et al., 2016], followed by the works exploiting the divergence measure from information theory to further improve the algorithm [Zhao et al., 2017, Kim and Mnih, 2018, Chen et al., 2018, Burgess et al., 2018]. However, obtaining a latent space with interpretability is still pretty challenging even on toy datasets and a meaningful, mathematically precise definition of disentanglement remains difficult to find. One way to exploit the interpretation of the latent variable z_j is that, after training, we fix the input of other latent variables and only change the value of the latent variable z_j. By monitoring the generated data with respect to different z_j, we gain some insights into z_j. Another annoying thing is that regularizing the latent space often leads to a deterioration of the generative quality [Zhao et al., 2017].

Recently, [Tonolini et al., 2020, Moran et al., 2022] proposed sparse VAE with spike and slab prior. [Moran et al., 2022] also showed the identifiable of latent sparse z for their approach under some assumptions. However, the dataset [Moran et al., 2022] used in the numerical study was relatively simple. [Tonolini et al., 2020] showed some interesting

results for face generation. They can use latent variables to control the lighting position, smile and Fringe of the face image they generated.

For latent space with $L_{\frac{1}{2}}$ prior, it is also worth performing a theoretical study to understand under what conditions the identifiability exists in the future. The current main challenge for us is the lack of interpretability of the latent space after we achieve a dimensional reduction of the latent space for the image generation tasks. More precisely, we found that the result is not stable. The latent space we obtained often changed every time we run the algorithm. Sometimes, we can obtain some sensible results. Somehow, this suggests a lack of identifiable latent space for the image dataset. Some researches [Kim and Mnih, 2018, Locatello et al., 2020, Fil et al., 2021] showed that the almost perfect performance presented by [Higgins et al., 2016] is difficult to obtain.

Another interesting area is studying the function of protein and RNA. [Riesselman et al., 2018] showed that it is possible to capture higher-order, context-dependent constraints in biological sequences via VAE. They found that the latent space can be used to explore new parts of sequence space. Recently, [Castro et al., 2022] proposed a transformer-based VAE with regularized latent space for protein generation. They used a transformer as an encoder neural network to improve the encoding ability and added an extra penalty term in the objective function to regularize the latent space.

Chapter 5

Conclusion and Future Directions

5.1 Concludion

In this thesis, we studied the use of sparsity-inducing priors to perform scalable model selection in high-dimensional settings. In the introductory chapter 1, we started by discussing some computational strategies and theoretical properties of these priors from two classes, the spike-and-slab prior and the global-local shrinkage prior.

We then focussed on the development of the $L_{\frac{1}{2}}$ prior in chapter 2, where a major contribution is showing that $L_{\frac{1}{2}}$ prior has closed form Laplace mixture decomposition. This data augmentation allowed us to then construct the efficient Partially Collapsed Gibbs (PCG) sampler [Van Dyk and Park, 2008] for $L_{\frac{1}{2}}$ prior with fast convergence and scalable to high dimension model. Then we show that the $L_{\frac{1}{2}}$ prior can lead to a nearly optimal posterior contraction rate and the variable selection consistency as the spike-and-slab prior in a high dimensional setting.

Apart from the novel MCMC scheme, with the motivation of a fully Bayesian approach, by integrating the hyper-parameter λ, we also develop a non-separable bridge penalty in chapter 2. This penalty enjoys the global-local adaptive property [Ročková and George, 2016a]. We develop a fast coordinate descent algorithm to solve this penalized linear

regression problem with convergence guarantees. The oracle properties of the resulting estimator for high dimensional sparse linear regression have also been studied. By using the proximal method [Polson et al., 2015], our algorithm can generalize to a model with a non-Gaussian response.

In chapter 3, we extended the PCG sampler to the cumulative logistic ordinal regression with random effects, applying the method to a real data analysis of student evaluations of teaching surveys. By using the t-distribution approximation trick from [Pingel, 2014] in one of the steps in PCG sampler, we recover the closed-form of update of PCG sampler as we enjoyed in the linear regression model. We then use this model to analyse SET data in higher education institutions in a study to discover factors that influence how students rate their instructor's teaching. It should be pointed out that our PCG sampler for $L_{\frac{1}{2}}$ prior can be easily generalized to any likelihood with Gaussian-mixture representation.

Finally, in chapter 4, we study the challenging problem of sparse deep learning in the Bayesian framework. By using the mirror descent [Fang et al., 2020] technique, we develop a constrained variational Adam(CVA) algorithm, which allows tempering of the posterior of the neural network. The cold posterior from the tempering significantly improves the predictive accuracy of the Bayesian neural network. We further introduce an extra binary latent variable to control the variance of the Gaussian prior to being small or large. The binary latent variable is updated by generalizing the CVA algorithm to CVA-EM. This allows us to obtain a sparse neural network. We show the state-of-the-art performance of our algorithm in image classification and high-dimensional nonlinear regression problems. Then we extend our algorithm to graph neural networks with negligible modification. In this case, the CVA-EM algorithm can not only provide a sparse weight but also a sparse graph. In the numerical experiments, we show that the sparse GNN can outperform or at least be as good as the traditional GNN in terms of predictive accuracy in node classification. We finish this chapter by discussing using the $L_{\frac{1}{2}}$ prior to constructing a sparse VAE. We outline the current challenges in this research.

5.2 Future Directions

Now, we outline some unsolved problems in sparse Bayesian learning, which could be potential future directions.

- In fitting high dimensional sparse linear regression problem, the MCMC approaches for sparse priors tend to underestimate the variance parameter of the noise, no matter whether the sparse prior is $L_{\frac{1}{2}}$, horseshoe or the others. [Moran et al., 2019] showed that using conjugate priors for the variance in a high dimensional linear regression setting can lead to an underestimation of the variance. However, we found that even in non-conjugate priors, the variance is underestimated by MCMC. We also found that the optimization approach with cross-validation did much better in variance estimation. We should try to solve this problem in the Bayesian framework in the future.

- One limitation of our PCG sampler for $L_{\frac{1}{2^\gamma}}$ prior is the numerical underflow in a high dimensional problem when $\gamma \geq 2$. For γ sufficiently large, we are close to the L_0 penalty. We anticipate that such prior may attain sharper than the current nearly optimal contraction rate. It may also lead to a better empirical performance in high dimensional settings than the existing global-local shrinkage priors. We hope that, in the future, we can overcome the numerical underflow issue at least for the $\gamma = 2$ case.

- In proving the posterior consistency of all these global-local shrinkage priors to high dimensional linear regression, the global shrinkage parameter is set to be the training sample size dependence, assuming that we know the sparsity level of the model. In practice, the sparsity level is unknown, a fully Bayesian approach assigns a hyper-prior to the global shrinkage prior. However, so far, the posterior consistency properties of the fully Bayesian approach are lacking in the study.

- Motivated by the PCG sampler, we proposed partially collapsed variational inference (PCVI) for $L_{\frac{1}{2}}$ prior. We believe that the PCVI should not be limited to

this particular case. It should be developed as a general framework for variational inference.

- In chapter 4, we use the spike-and-slab prior to obtaining a sparse BNN. However, by using the Laplace mixture representation of $L_{\frac{1}{2}}$ prior, we can also consider replacing the spike-and-slab prior to $L_{\frac{1}{2}}$ prior in algorithm to train a sparse neural network.

Appendix A

A.1 Proof of Normal-mixture representation

Proof of Lemma 2.1.1

Proof. If $\frac{\lambda^2}{4}\exp(-\lambda|\beta|^{\frac{1}{2}})$ is a Laplace transformation of some non-negative function $f(\cdot)$ evaluated at $x = |\beta|$, then $f(\cdot)$ is the inverse Laplace transformation

$$\frac{\lambda^2}{4}\exp(-\lambda|\beta|^{\frac{1}{2}}) = \int_0^\infty f(t)e^{-t|\beta|}dt$$

where $f(t) = \mathcal{L}^{-1}(\frac{\lambda^2}{4}\exp(-\lambda\sqrt{x}))(t) = \frac{\lambda^3}{8\sqrt{\pi}}t^{-\frac{3}{2}}\exp(-\frac{\lambda^2}{4t})$.

Let $\pi(s) = \frac{2}{s}f(\frac{1}{s})$, then $s \sim \text{Gamma}(\frac{3}{2}, \frac{\lambda^2}{4})$. Following this rearrangement, we can put a Laplace prior on β conditionally on s,

$$\pi(\beta|s) = \frac{1}{2s}e^{-\frac{|\beta|}{s}}.$$

We can construct a Laplace-Gamma mixture representation for exponential power prior with $\alpha = \frac{1}{2}$

$$\frac{\lambda^2}{4}\exp(-\lambda|\beta|^{\frac{1}{2}}) = \int_0^\infty \pi(\beta|s)\pi(s)ds.$$

To get another parametric form, move the hyper-parameter λ outside the gamma distribution, and apply the change of variable in the right hand side of the above equation. Let $v = \lambda^2 s$, then we have

$$\frac{\lambda^2}{4}\exp(-\lambda|\beta|^{\frac{1}{2}}) = \int_0^\infty \pi(\beta|v)\pi(v)dv$$

where $\pi(\beta|v) = \frac{\lambda^2}{2v} e^{-\frac{\lambda^2|\beta|}{v}}$ and $v \sim \text{Gamma}(\frac{3}{2}, \frac{1}{4})$. □

Proof of Lemma 2.1.2

Proof. Write

$$\frac{\lambda^{2^\gamma}}{2(2^\gamma!)} \exp\left(-\lambda |\beta|^{\frac{1}{2^\gamma}}\right) = \int_0^\infty f(t) e^{-t|\beta|^{\frac{1}{2^\gamma - 1}}} dt$$

then $\frac{\lambda^{2^\gamma}}{2(2^\gamma!)} \exp\left(-\lambda |\beta|^{\frac{1}{2^\gamma}}\right)$ is the Laplace transform of of some non-negative function $f(\cdot)$ evaluated at $x = |\beta|^{\frac{1}{2^\gamma - 1}}$, where $f(t) = \frac{1}{\Gamma(\frac{2^\gamma+1}{2})} \frac{1}{2^{2^\gamma+2}} \frac{\lambda^{2^\gamma+1}}{(2^{\gamma-1}!)} t^{-\frac{3}{2}} \exp(-\frac{\lambda^2}{4t})$.

Let $\pi(s) = \frac{2(2^{\gamma-1}!)s^{2^\gamma-1}}{s} f(\frac{1}{s})$, then $s \sim \text{Gamma}(\frac{2^\gamma+1}{2}, \frac{\lambda^2}{4})$. Following this rearrangement, we can put a exponential power prior with $\alpha = \frac{1}{2^{\gamma-1}}$ on β conditional on s_γ then,

$$\pi(\beta|s_\gamma) = \frac{1}{2(2^{\gamma-1}!)s_\gamma^{2^\gamma-1}} \exp\left(-\frac{|\beta|^{\frac{1}{2^\gamma-1}}}{s_\gamma}\right).$$

We can see that the exponential power prior with $\alpha = \frac{1}{2^\gamma}$ is the mixture of exponential power prior with $\alpha = \frac{1}{2^{\gamma-1}}$ and $\text{Gamma}(\frac{2^\gamma+1}{2}, \frac{\lambda^2}{4})$

$$\frac{\lambda^{2^\gamma}}{2(2^\gamma!)} \exp\left(-\lambda |\beta|^{\frac{1}{2^\gamma}}\right) = \int_0^\infty \pi(\beta|s_\gamma)\pi(s_\gamma) ds_\gamma.$$

Using a similar argument as in the proof of Lemma 2.1.2, moving the hyper-parameter λ outside the gamma distribution by using the change of variable trick. We let $v_\gamma = \lambda^2 s_\gamma$, then

$$\pi(\beta|v_\gamma, \lambda) = \frac{\lambda^{2^\gamma}}{2(2^{\gamma-1}!)v_\gamma^{2^\gamma-1}} \exp\left(-\frac{\lambda^2|\beta|^{\frac{1}{2^\gamma-1}}}{v_\gamma}\right), \quad v_\gamma \sim \text{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{1}{4}\right).$$

□

Proof of Theorem 2.1.3

Proof. Starting with the decomposition given in Lemma 2.1.2, we make some further decompositions. Our argument consists of three steps:

Step 1: Set $\lambda' = \frac{\lambda^2}{v_\gamma}$, then

$$\pi(\beta|v_\gamma, \lambda) = \frac{\lambda^{2^\gamma}}{2(2^{\gamma-1}!)v_\gamma^{2^{\gamma-1}}} \exp\left(-\frac{\lambda^2|\beta|^{\frac{1}{2^{\gamma-1}}}}{v_\gamma}\right) = \frac{\lambda'^{2^{\gamma-1}}}{2(2^{\gamma-1}!)} \exp(-\lambda'|\beta|^{\frac{1}{2^{\gamma-1}}})$$

Step 2: Apply Lemma 2.1.2 again with $\frac{\lambda'^{2^{\gamma-1}}}{2(2^{\gamma-1}!)} \exp\left(-\lambda\,|\beta|^{\frac{1}{2^{\gamma-1}}}\right) = \int_0^\infty \pi(\beta|s_\gamma)\pi(s_\gamma)ds_\gamma$, then

$$\pi(\beta|s_{\gamma-1}) = \frac{1}{2(2^{\gamma-2}!)s_{\gamma-1}^{2^{\gamma-2}}} \exp\left(-\frac{|\beta|^{\frac{1}{2^{\gamma-2}}}}{s_{\gamma-1}}\right), \quad s_{\gamma-1} \sim \text{Gamma}(\frac{2^{\gamma-1}+1}{2}, \frac{\lambda'^2}{4})$$

Step 3: By change of variables, set $v_{\gamma-1} = \lambda^4 s_{\gamma-1}$, then

$$\pi(\beta|v_{\gamma-1}, \lambda) = \frac{\lambda^{2^\gamma}}{2(2^{\gamma-2}!)v_{\gamma-1}^{2^{\gamma-2}}} \exp\left(-\frac{\lambda^4|\beta|^{\frac{1}{2^{\gamma-2}}}}{v_{\gamma-1}}\right), \quad v_{\gamma-1} \sim \text{Gamma}\left(\frac{2^{\gamma-1}+1}{2}, \frac{1}{4v_\gamma^2}\right).$$

Using the above three-step argument recursively, then for any $i = 1, 2, ..., \gamma-1$, we have

$$\pi(\beta|v_{\gamma-i}, \lambda) = \frac{\lambda^{2^\gamma}}{2(2^{\gamma-i-1}!)v_{\gamma-i}^{2^{\gamma-i-1}}} \exp\left(-\frac{\lambda^{2^{i+1}}|\beta|^{\frac{1}{2^{\gamma-i-1}}}}{v_{\gamma-i}}\right), \quad v_{\gamma-i} \sim \text{Gamma}\left(\frac{2^{\gamma-i}+1}{2}, \frac{1}{4v_{\gamma-i+1}^2}\right).$$

Consequently, the exponential power prior with $\alpha = \frac{1}{2^\gamma}$ can be written as

$$\beta|v_1, \lambda \sim \text{DE}(\frac{v_1}{\lambda^{2^\gamma}}), \quad v_i|v_{i+1} \sim \text{Gamma}\left(\frac{2^i+1}{2}, \frac{1}{4v_{i+1}^2}\right), \quad v_\gamma \sim \text{Gamma}\left(\frac{2^\gamma+1}{2}, \frac{1}{4}\right)$$

for $i = 1, \ldots, \gamma-1$. Since it is well known that the Laplace distribution can be represented as a normal exponential mixture. □

A.2 Partially collapsed Gibbs sampling

A.2.1 Steps for PCG sampler

Here we give the details of the three steps that comprise the PCG sampler, for sampling the parameters $\boldsymbol{\beta}, \sigma^2, \tau_1^2, \ldots, \tau_p^2, v_1, \ldots, v_p, \lambda$ under the $L_{\frac{1}{2}}$ prior with $\gamma = 1$.

Step M: Marginalise

Step 1: Draw $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\tau^2}, \boldsymbol{v}, \lambda, b, \boldsymbol{Y})$

Step 2: Draw $\boldsymbol{\tau^2}$ from $\pi(\boldsymbol{\tau^2}|\boldsymbol{\beta}, \boldsymbol{v}, \lambda, \sigma^2, b, \boldsymbol{Y})$

Step 3: Draw σ^2 from $\pi(\sigma^2|\boldsymbol{\beta}, \boldsymbol{\tau^2}, v, \lambda, b, \boldsymbol{Y})$

Step 4: Draw b from $\pi(b|\boldsymbol{\beta}, \boldsymbol{\tau^2}, v, \lambda, \sigma^2, \boldsymbol{Y})$

Step 5: Draw $\boldsymbol{\tau}^{2\star}$, $\boldsymbol{v}^\star$ and λ from $\pi(\boldsymbol{\tau}^2, v, \lambda|\boldsymbol{\beta}, \sigma^2, b, \boldsymbol{Y})$

Step 6: Draw $\boldsymbol{\tau}^{2\star}$ and $\boldsymbol{v}$ from $\pi(\boldsymbol{\tau}^2, v|\boldsymbol{\beta}, \sigma^2, \lambda, b, \boldsymbol{Y})$

Step P: Permute

Step 1: Draw $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\tau}^2, v, \lambda, b, \boldsymbol{Y})$

Step 2: Draw $\boldsymbol{\tau}^{2\star}$, $\boldsymbol{v}^\star$ and λ from $\pi(\boldsymbol{\tau}^2, v, \lambda|\boldsymbol{\beta}, \sigma^2, b, \boldsymbol{Y})$

Step 3: Draw $\boldsymbol{\tau}^{2\star}$ and $\boldsymbol{v}$ from $\pi(\boldsymbol{\tau}^2, \boldsymbol{v}|\boldsymbol{\beta}, \sigma^2, \lambda, b, \boldsymbol{Y})$

Step 4: Draw $\boldsymbol{\tau}^2$ from $\pi(\boldsymbol{\tau}^2|\boldsymbol{\beta}, \boldsymbol{v}, \lambda, \sigma^2, b, \boldsymbol{Y})$

Step 5: Draw σ^2 from $\pi(\sigma^2|\boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{v}, \lambda, b, \boldsymbol{Y})$

Step 6: Draw b from $\pi(b|\boldsymbol{\beta}, \boldsymbol{\tau}^2, v, \lambda, \sigma^2, \boldsymbol{Y})$

Step T: Trim

Step 1: Draw $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\tau}^2, v, \lambda, \boldsymbol{Y})$

Step 2: Draw λ from $\pi(\lambda|\boldsymbol{\beta}, \sigma^2, \boldsymbol{Y})$

Step 3: Draw $\boldsymbol{v}$ from $\pi(\boldsymbol{v}|\boldsymbol{\beta}, \sigma^2, \lambda, \boldsymbol{Y})$

Step 4: Draw $\boldsymbol{\tau}^2$ from $\pi(\boldsymbol{\tau}^2|\boldsymbol{\beta}, \boldsymbol{v}, \lambda, \sigma^2, \boldsymbol{Y})$

Step 5: Draw σ^2 from $\pi(\sigma^2|\boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{v}, \lambda, \boldsymbol{Y})$

Step 6: Draw b from $\pi(b|\boldsymbol{\beta}, \boldsymbol{\tau}^2, v, \lambda, \sigma^2, \boldsymbol{Y})$

We use superscript $\star$ to present intermediate quantities that are sampled but not retained as part of the output. Step M is a generalization of the traditional Gibbs sampler with some components being updated multiple times within each iteration.

In the second step we rearranged the ordering of the update in Step M by swapping the ordering of steps 5 and 6 with steps 2,3 and 4. This step does not alter the stationary distribution of the Gibbs sampler. Finally, the intermediate draws of $\tau^{2\star}$ and $v^\star$ are not

necessary if we can sample λ and v in steps 2 and 3 directly from the respective marginal distributions, so trimming the intermediate Steps 2 and 3, we arrive at the last set of updating procedure.

A.2.2 Derivations of conditional posteriors

Since

$$\pi(\lambda|\boldsymbol{\beta},\boldsymbol{\tau}^2,\sigma^2,b,\boldsymbol{Y}) \propto \lambda^{2^\gamma p}\exp\left(-\lambda\sum_{j=1}^{p}|\beta_j|^{\frac{1}{2^\gamma}}\right)\pi(\lambda|b)$$

with $\lambda|b \sim \text{Gamma}(\frac{1}{2},\frac{1}{b})$. The conditional posterior is a gamma distribution with shape parameters $0.5+2^\gamma p$ and $\frac{1}{b}+\sum_{j=1}^{p}|\beta_j|^{\frac{1}{2^\gamma}}$.

Next, we derive the conditional posterior for $\boldsymbol{v}_i$. Observe that given $\boldsymbol{v}_{i+1}$, $\boldsymbol{v}_i$ is conditionally independent of the remaining $\boldsymbol{v}_j$s. In addition, given $\boldsymbol{\beta}$, $\boldsymbol{v}_i$ is also conditionally independent of σ^2 and b. Thus,

$$\pi(v_i|\boldsymbol{\beta},\lambda,v_{i+1},\boldsymbol{Y}) = \prod_{j=1}^{p}\pi(v_{ij}|\beta_j,\lambda,v_{i+1,j},\boldsymbol{Y}) \propto \prod_{j=1}^{p}\pi(\beta_j|v_{ij},\lambda)\pi(v_{ij}|v_{i+1,j}).$$

From Theorem 2.3, we see that

$$\pi(\beta_j|v_{ij},\lambda) \propto \frac{\lambda^{2^i}}{2(2^{i-1}!)v_{ij}^{2^{i-1}}}\exp\left(-\frac{\lambda^2|\beta_j|^{\frac{1}{2^{i-1}}}}{v_{ij}}\right), \quad v_{ij}|v_{i+1,j} \sim \text{Gamma}\left(\frac{2^i+1}{2},\frac{1}{4v_{i+1,j}^2}\right).$$

Thus,

$$\pi(\beta_j|v_{ij},\lambda)\pi(v_{ij}|v_{i+1,j}) \propto v_{ij}^{-\frac{1}{2}}\exp\left(-\frac{v_{ij}}{4v_{i+1,j}^2}-\frac{\lambda^2|\beta_j|^{\frac{1}{2^{i-1}}}}{v_{ij}}\right)$$

$$v_{ij}|\beta_j,\lambda,v_{i+1,j},\boldsymbol{Y} \sim \text{GIG}\left(\frac{1}{2v_{i+1,j}^2},2\lambda^2|\beta_j|^{\frac{1}{2^{i-1}}},\frac{1}{2}\right),$$

since sampling the generalized inverse gaussian distribution is not easy, we can sample $\frac{1}{v_{ij}}$ from the inverse gaussian distribution.

$$h_{ij}=\frac{1}{v_{ij}}|\beta_j,\lambda,v_{i+1,j},\boldsymbol{Y} \sim \text{InvGaussian}\left(\frac{1}{2\lambda v_{i+1,j}|\beta_j|^{\frac{1}{2^i}}},\frac{1}{2v_{i+1,j}^2}\right).$$

A.3 Posterior consistency and contraction rate

Assuming that the observations $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ consisting of n independent observations with product measure $P_\theta^n = P_{\theta,1} \times P_{\theta,2} \ldots \times P_{\theta,n}$, where $\theta \in \Theta$. If the measure $P_{\theta,i}$ with density $f_{\theta,X_i}(Y_i)$ is considered random and distributed according to Π_n, as it is the case in Bayesian inference, then the posterior distribution $\Pi_n(B \mid Y_1, \ldots, Y_n)$ is the conditional distribution of P_θ^n based on the prior Π_n

$$\Pi_n(B \mid Y_1, \ldots, Y_n) = \frac{\int_B \prod_{i=1}^n f_{\theta,X_i}(Y_i)\, d\Pi_n(\theta)}{\int_\Theta \prod_{i=1}^n f_{\theta,X_i}(Y_i)\, d\Pi_n(\theta)},$$

where $B \in \mathcal{B}$ and $\mathcal{B}$ is a σ-field on Θ.

The posterior is said to be weakly consistent if it concentrates on arbitrarily small neighborhoods of the true $P_{\theta_0}^n$, $\theta_0 \in \Theta$, in $P_{\theta_0}^n$-probability tending to 1 as $n \to \infty$ or strongly consistent if this convergence is in the almost-sure sense. We study the rate at which the neighborhoods of $P_{\theta_0}^n$ may decrease to a small ball while still capturing most of the posterior mass, in the weak sense (See Chapter 6 of [Ghosal and Van der Vaart, 2017]). Note that in high dimensional regression setting our prior depends on the number of predictors, and hence sample size dependent, therefore we use Π_n to denote this dependence.

The following notation will be used in this section and in proving the oracle property in a later section: we rewrite the dimension p of the model by p_n to indicate that the number of predictors can increase with the sample size n. We use β_0 and σ_0 to indicate true regression coefficients and the true standard deviation and $\theta_0 = (\beta_0, \sigma_0)$. Let S_0 be the set containing the indices of the true nonzero coefficients, where $S_0 \subseteq \{1, .., p_n\}$, then $s_0 = |S_0|$ denotes the size of the true model. The ε-covering number of the space $\mathcal{F}$ with respect to a metric d is denoted by $N(\varepsilon, \mathcal{F}, d)$, which is the minimal number of d-balls of radius ε needed to cover $\mathcal{F}$. We further let $\boldsymbol{X}_s$ denote the design matrix corresponding to the model S, and use Pf to abbreviate $\int f dP$, finally we use $\gtrsim$ and $\lesssim$ to denote inequality up to a constant.

We make the following assumptions on contraction theorems:

A1. The dimensional is high, with $p_n >> n$ and $\log(p_n) = o(n)$.

A2. The true number of nonzero β_0 is s_0, where $s_0 = o(n/\log p_n)$.

A3. All the covariates are uniformly bounded. In other words, there exists a constant $k > 0$ such that $\lambda_{\max}(X^T X) \leq k n^{\alpha}$ for some $\alpha \in [1, +\infty)$.

A4. There exist constants $v_1 > 0$, $v_2 > 0$ and an integer $\widetilde{p}$ satisfying $s_0 = o(\widetilde{p})$ and $\widetilde{p} = o(s_0 \log n)$, so that $nv_1 \leq \lambda_{min}(X_S^T X_S) \leq \lambda_{max}(X_S^T X_S) \leq nv_2$ for any model of size $|s| \leq \widetilde{p}$.

A5. $||\beta_0||_{\infty} = E$ and E is some positive number independent of n.

A.3.1 Some preliminary results

Before proving the posterior contraction rates, we first state and prove some auxiliary results.

Lemma A.3.1. *Suppose that assumptions A1, A2 and A5 hold, the exponential power prior* $\Pi_n(\beta) = \frac{\lambda_n^{\frac{1}{\alpha}}}{2\Gamma(1+\frac{1}{\alpha})} \exp\left(-\lambda_n |\beta|^{\alpha}\right)$ *with* $0 < \alpha \leq \frac{1}{2}$ *and* $\lambda_n \propto \log p_n$*, satisfies*

$$\int_{|\beta| \geq a_n} \Pi_n(\beta) d\beta \leq p_n^{-(1+u)} \tag{A.1}$$

$$-\log\left(\inf_{\beta \in [-E,E]} \Pi_n(\beta)\right) = O\left(\log p_n\right) \tag{A.2}$$

where u, E *are positive constants and* $a_n = \frac{\sigma_0}{\sqrt{2k}} \sqrt{s \log p_n / n} / p_n$.

Proof. It is easy to see that

$$-\log\left(\inf_{\beta \in [-E,E]} \Pi_n(\beta)\right) = \lambda_n E^{\alpha} - \frac{1}{\alpha} \log \lambda_n + \log 2 + \log \Gamma(1 + \frac{1}{\alpha})$$

with $\lambda_n \propto \log p_n$ the second equality holds.

To prove the Equation A.2, first we show that for any $\alpha \in (0, \frac{1}{2})$, the following equation,

$$\lim_{n \to \infty} \frac{\lambda_n^{\frac{1}{\alpha}}}{2\Gamma(1 + \frac{1}{\alpha})} \int_{-a_n}^{a_n} \exp\left(-\lambda_n |\beta|^{\alpha}\right) d\beta \geq \lim_{n \to \infty} \frac{\lambda_n^2}{2\Gamma(3)} \int_{-a_n}^{a_n} \exp\left(-\lambda_n |\beta|^{\frac{1}{2}}\right) d\beta \tag{A.3}$$

hold. We then prove Equation A.2 holds for the special case $\alpha = \frac{1}{2}$. Then the more general case $0 < \alpha < \frac{1}{2}$ will trivially hold by using Equation A.3.

To prove Equation A.3, it is enough to show

$$\int_0^{a_n} \left\{ \exp\left[-\lambda_n |\beta|^\alpha + (\frac{1}{\alpha} - 2)\log\lambda_n + b \right] - \exp\left[-\lambda_n |\beta|^{\frac{1}{2}} \right] \right\} d\beta \geq 0$$

where $b = \log \frac{\Gamma(3)}{\Gamma(1+\frac{1}{\alpha})}$. Consequently, it is sufficient to show

$$(\frac{1}{\alpha} - 2)\log\lambda_n + b \geq \lambda_n |\beta|^\alpha (1 - |\beta|^{\frac{1}{2}-\alpha})$$

for any $|\beta| \in (0, a_n)$. Since $\lim_{n\to\infty} a_n = 0$, then for n sufficient large we just need,

$$(\frac{1}{\alpha} - 2)\log\lambda_n + b \geq \lambda_n |a_n|^\alpha.$$

We see that the above inequality hold since $\lambda_n |a_n|^\alpha \propto \frac{\log p_n}{p_n^\alpha} \epsilon_n^\alpha \to 0$ as $n \to \infty$

Since the exponential power prior with $\alpha = \frac{1}{2}$ has the Laplace-Gamma mixture representation with $\beta|v \sim \text{Laplace}(0, v)$ and $v \sim \text{Gamma}(\frac{3}{2}, \frac{\lambda^2}{4})$, we write

$$\begin{aligned}
P(|\beta| > a_n) &= \int_0^\infty P(|\beta| > a_n | v) f_\lambda(v) dv \\
&= \int_0^\infty e^{-\frac{a_n}{v}} f_\lambda(v) dv \\
&= \int_0^{\frac{a_n}{C}} e^{-\frac{a_n}{v}} f_\lambda(v) dv + \int_{\frac{a_n}{C}}^\infty e^{-\frac{a_n}{v}} f_\lambda(v) dv \\
&\leq e^{-C} \int_0^{\frac{a_n}{C}} f_\lambda(v) dv + \int_{\frac{a_n}{C}}^\infty e^{-\frac{a_n}{v}} f_\lambda(v) dv \\
&= \frac{e^{-C} M^{\frac{3}{2}}}{\Gamma(\frac{3}{2})} \left(\frac{\lambda^2 a_n}{4C} \right)^{\frac{3}{2}} \exp\left(-\frac{\lambda^2 a_n}{4C} M \right) + \int_{\frac{a_n}{C}}^\infty e^{-\frac{a_n}{v}} f_\lambda(v) dv \\
&\leq \frac{e^{-C} M^{\frac{3}{2}}}{\Gamma(\frac{3}{2})} \left(\frac{\lambda^2 a_n}{4C} \right)^{\frac{3}{2}} \exp\left(-\frac{\lambda^2 a_n}{4C} M \right) + \left(\frac{2}{3} \right)^{\frac{3}{2}} \exp\left(\frac{3}{2} \right) \left(\frac{\lambda^2 a_n}{4C} \right)^{\frac{3}{2}} \exp\left(-\frac{\lambda^2 a_n}{4C} \right)
\end{aligned}$$

where $f_\lambda(v)$ is the density of $\text{Gamma}(\frac{3}{2}, \frac{\lambda^2}{4})$. In the fifth line, the equality is obtained by applying the mean value theorem to the first term with $0 < M < 1$. In the last line, the inequality is obtained by applying the Chernoff bound to the tail of Gamma distribution. The Chernoff bound for Gamma distribution is $\left(\frac{2}{3}\right)^{\frac{3}{2}} \exp\left(\frac{3}{2}\right) \left(\frac{\lambda^2 a_n}{4C}\right)^{\frac{3}{2}} \exp\left(-\frac{\lambda^2 a_n}{4C}\right)$ if $\frac{a_n}{C} \geq \frac{6}{\lambda^2}$, otherwise it is 1. Hence, the value of C we choose should satisfy the constraint $\frac{a_n}{C} \geq \frac{6}{\lambda^2}$.

Now, we set $C = a_n$, then we have

$$\begin{aligned} P(|\beta| > a_n) &\leq \frac{M^{\frac{3}{2}}}{\Gamma(\frac{3}{2})}\left(\frac{\lambda^2}{4}\right)^{\frac{3}{2}}\exp\left(-\frac{\lambda^2}{4}M\right) + \left(\frac{2}{3}\right)^{\frac{3}{2}}\exp\left(\frac{3}{2}\right)\left(\frac{\lambda^2}{4}\right)^{\frac{3}{2}}\exp\left(-\frac{\lambda^2}{4}\right) \\ &\leq \exp\left(-\frac{\lambda^2}{4}M'\right) \\ &= p_n^{-M''\log p_n}, \end{aligned}$$

for some $M' > 0$ and $M'' > 0$. □

Lemma A.3.1 shows that the exponential power prior with $0 < \alpha \leq \frac{1}{2}$ can balance the requirement of putting enough mass around zero and tail thickness. It is easy to verify that the Bayesian LASSO fails to satisfy the properties in Lemma A.3.1 unless assumption A5 changes to $||\beta_0||_\infty = O(\sqrt{s_0 \log p_n/n}/p_n)$ and we set $\lambda_n \propto p_n \log p_n/\sqrt{\frac{s\log p_n}{n}}$ [Song and Liang, 2017]. This assumption is unrealistic as it requires the magnitude of β to asymptotically decrease to zero. Next, we provide an upper bound for the covering number in Euclidean space.

Lemma A.3.2. *For $\|x\|_p \leq M$, $\epsilon < 1$ and $p \geq 1$, for any M, we have*

$$N(\epsilon, \{x \in \mathbb{R}^m : \|x\|_p \leq M\}, \|\cdot\|_p) \leq \left(\frac{3M}{\epsilon}\right)^m$$

The proof for Lemma A.3.2 can be found in Appendix C of [Ghosal and Van der Vaart, 2017].

Now, we derive the analytical expressions based on the Renyi distance for two Gaussian distributions.

Lemma A.3.3. *Consider the root of average Renyi divergence of order $\frac{1}{2}$,*

$$d(P_{\theta_1}^n, P_{\theta_2}^n) = [-n^{-1}\log(\int \sqrt{f_1^n f_2^n}dY)]^{1/2}$$

where f_1^n and f_2^n are their density function. Let $P_{\theta_1}^n$ and $P_{\theta_2}^n$ be the law of the random variable with Multivariate Gaussian distribution $N(g_1(X), \sigma_1^2 I_n)$ and $N(g_2(X), \sigma_2^2 I_n)$ respectively. Then

$$d(P_{\theta_1}^n, P_{\theta_2}^n) = \left[\frac{1}{2}\log\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2} + \frac{\|g_1(X)-g_2(X)\|^2}{4n(\sigma_1^2+\sigma_2^2)}\right]^{1/2}.$$

Proof. To simplify notation and apply change of the variable in the later part of the proof, we set $Y = g(X) + \sigma Z$.

$$
\begin{aligned}
&\int \sqrt{f_1 f_2} dY \\
=&\frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{(\sigma_1^2\sigma_2^2)^{\frac{n}{4}}} \int \exp\left[-\frac{\|Y - g_2(X)\|^2}{4\sigma_2^2} - \frac{\|Y - g_1(X)\|^2}{4\sigma_1^2}\right] dY \\
=&\frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{(\sigma_1^2\sigma_2^2)^{\frac{n}{4}}} \int \exp\left[-\frac{\|Y - g_1(X) + g_1(X) - g_2(X)\|^2}{4\sigma_2^2} - \frac{\|Y - g_1(X)\|^2}{4\sigma_1^2}\right] dY \\
=&\frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{(\sigma_1^2\sigma_2^2)^{\frac{n}{4}}} \exp\left[-\frac{\|g_1(X) - g_2(X)\|^2}{4\sigma_2^2}\right] \\
&\cdot \int \exp\left[-\frac{\|Y - g_1(X)\|^2}{4\sigma_2^2} - \frac{\|Y - g_1(X)\|^2}{4\sigma_1^2} - \frac{2(Y - g_1(X))(g_1(X) - g_2(X))}{4\sigma_2^2}\right] dY \\
=&\left(\frac{\sigma_1^2}{\sigma_2^2}\right)^{\frac{n}{4}} \exp\left[-\frac{\|g_1(X) - g_2(X)\|^2}{4\sigma_2^2}\right] \int \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left[-\frac{\|Z\|^2(\sigma_1^2 + \sigma_2^2)}{4\sigma_2^2} - \frac{2\sigma_1 Z^T(g_1(X) - g_2(X))}{4\sigma_2^2}\right] dZ \\
=&\left(\frac{\sigma_1^2}{\sigma_2^2}\right)^{\frac{n}{4}} \exp\left[-\frac{\|g_1(X) - g_2(X)\|^2}{4\sigma_2^2}\right] \int \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left[-\frac{\|Z\|^2(\sigma_1^2 + \sigma_2^2)}{4\sigma_2^2} - \frac{2\sigma_1 Z^T(g_1(X) - g_2(X))}{4\sigma_2^2}\right] dZ \\
=&\left(\frac{\sigma_1^2}{\sigma_2^2}\right)^{\frac{n}{4}} \exp\left[-\frac{\|g_1(X) - g_2(X)\|^2}{4\sigma_2^2}\right] \exp\left[\frac{\sigma_1^2\|g_1(X) - g_2(X)\|^2}{4\sigma_2^2(\sigma_1^2 + \sigma_2^2)}\right] \int \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left[-\frac{\left\|Z\sqrt{\sigma_1^2 + \sigma_2^2} + \frac{\sigma_1}{\sqrt{\sigma_1^2}}\right.}{4\sigma_2^2}\right. \\
=&\left(\frac{\sigma_1^2}{\sigma_2^2}\right)^{\frac{n}{4}} \exp\left[\frac{\|g_1(X) - g_2(X)\|^2}{4(\sigma_1^2 + \sigma_2^2)}\right] \left(\frac{2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^{\frac{n}{2}}
\end{aligned}
$$

Thus,

$$
d(P_{\theta_1}^n, P_{\theta_2}^n) = [-n^{-1}\log(\int \sqrt{f_1^n f_2^n} dY)]^{1/2} = \left[\frac{1}{2}\log\frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1\sigma_2} + \frac{\|g_1(X) - g_2(X)\|^2}{4n(\sigma_1^2 + \sigma_2^2)}\right]^{1/2}.
$$

We can see that both the first and second term are non-negative and $d(P_{\theta_1}^n P_{\theta_2}^n) = 0$ if and only if $g_1 = g_2$ and $\sigma_1 = \sigma_2$. The symmetric property $d(P_{\theta_1}^n, P_{\theta_2}^n) = d(P_{\theta_2}^n, P_{\theta_1}^n)$ is also easy to see. □

Finally, we restate a simple version of Lemma 8.21 from [Ghosal and Van der Vaart, 2017], which will help us to prove Theorem 2.1.4.

Lemma A.3.4. *For any $D > 0$ and $\epsilon_n \geq \frac{1}{\sqrt{n}}$,*

$$
P_{\theta_0}^n \left(\int \frac{P_\theta^n}{P_{\theta_0}^n} d\Pi(\theta) \leq \Pi\left(P_\theta^n : K(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2, V_{2,0}(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2\right) e^{-(1+D)n\epsilon_n^2} \right) \leq \frac{1}{D^2 n\epsilon_n^2}
$$

A.3.2 Proof of the contraction theorem

The key to our proof will be drawing upon the framework of posterior concentration theory for independent but not identical observations from the Theorem 1 of the seminal paper by [Ghosal et al., 2007]. Theorem 1 provides a very general framework for showing posterior concentration in infinite-dimensional models. The contraction rate is described in terms of the average squared Hellinger distance by default. However, as mentioned by [Ning et al., 2020], closeness in terms of the average squared Hellinger distance does not imply that the posterior means of the parameters in the two densities are also close on average in terms of the Euclidean distance. To alleviate the problem, we use the root of average Renyi divergence of order $\frac{1}{2}$ as distance. We will construct a likelihood ratio test and then show that such a test works well for this distance. Finally, we show that the posterior consistency under this new distance implies the the posterior consistency under the Euclidean distance.

Restating Theorem 1 of [Ghosal et al., 2007] here, suppose that for a sequence ϵ_n with $\epsilon_n \to 0$ and $n\epsilon_n^2 \to \infty$, there exists constants $C_1 > 0$, $C_2 > 0$ and a sequence of sieves $F_n \subset F$, where F is the sample space (model space) of the prior distribution, such that if the following three conditions hold:

1. Prior concentration condition:

$$\Pi_n \left(P_\theta^n : K(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2, V_{2,0}(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2 \right) \geq \exp\left(-n\varepsilon_n^2 C_1\right). \tag{A.4}$$

where $K(P_{\theta_0}^n, P_\theta^n) = -P_{\theta_0}^n \log(P_\theta^n / P_{\theta_0}^n)$ and $V_{2,0}(P_{\theta_0}^n, P_\theta^n) = P_{\theta_0}^n |\log(P_{\theta_0}^n / P_\theta^n) - K(P_{\theta_0}^n, P_\theta^n)|^2$

2. Sieve sequence condition:

$$\Pi_n \left(F_n^c\right) \leq \exp\left(-n\varepsilon_n^2 (C_1 + 2)\right) \tag{A.5}$$

for some $\xi \in (0, 1)$ and $j \in \mathbb{N}$

3. Testing conditions:

There exists a sequence of test function ϕ_n such that

$$\begin{aligned} P_{\theta_0}^n \phi_n &\leq \exp\left(-C_2 n\varepsilon_n^2\right) \\ \sup_{\theta \in F_n : d(P_{\theta_0}^n, P_\theta^n) \geq M\epsilon_n} P_\theta^n (1-\phi_n) &\leq \exp\left(-C_2 n\varepsilon_n^2\right) \end{aligned} \tag{A.6}$$

Then for sufficiently large M, $\Pi_n\left(\theta \subset F : d\left(P_\theta^n, P_{\theta_0}^n\right) \geq M\varepsilon_n \mid X_1, \ldots, X_n\right) \to 0$ in $P_{\theta_0}^n-$ probability.

Before begin the proof, we defined the sieve F_n by using the concept of generalized dimension [Bhattacharya et al., 2015, Song and Liang, 2017, Ročková and George, 2018], which approximates the model size. We define the generalized dimension as

$$\gamma_j(\beta_j) = I(|\beta_j| > a_n) \text{ and } |\gamma(\beta)| = \sum_{j=1}^{p_n} \gamma_j(\beta_j)$$

where $a_n = \frac{\sigma_0}{\sqrt{2k}}\sqrt{s_0 \log p_n / n}/p_n$ and $\lim_{n\to\infty} a_n = 0$. Then we can define the sieve F_n as

$$F_n = \{S_\beta : |\gamma(\beta)| \leq C_3 s_0\} \cap \left\{\beta : \sum_{j=1}^{p} |\beta_j|^\alpha \leq C_3 E s_0\right\} \cap \left\{\sigma^2 : \frac{1}{n} \leq \sigma^2 \leq p_n^{C_3 s_0/\delta}\right\}$$

Proof of theorem 3.1

Proof. Part 1. Prior concentration condition:

Let $P_\theta^n \sim N_n(X\beta, \sigma^2 I_n)$ and $P_{\theta_0}^n \sim N_n(X\beta_0, \sigma_0^2 I_n)$. Then we have:

$$\begin{aligned} \log \frac{f_\theta^n}{f_{\theta_0}^n} &= \frac{n}{2}\log\frac{\sigma_0^2}{\sigma^2} + \frac{||Y - X\beta_0||^2}{2\sigma_0^2} - \frac{||Y - X\beta||^2}{2\sigma^2} \\ &= \frac{n}{2}\left(\log\frac{\sigma_0^2}{\sigma^2} + \frac{||Y - X\beta_0||^2}{n\sigma_0^2} - \frac{||Y - X\beta_0 + X\beta_0 - X\beta||^2}{n\sigma^2}\right) \\ &= \frac{n}{2}\left(\log\frac{\sigma_0^2}{\sigma^2} + \frac{||Y - X\beta_0||^2}{n\sigma_0^2} - \frac{||Y - X\beta_0||^2 + ||X\beta_0 - X\beta||^2 + 2(Y - X\beta_0)^T(X\beta_0 - X\beta)}{n\sigma^2}\right) \end{aligned}$$

Thus,

$$-P_{\theta_0}^n \log \frac{f_\theta^n}{f_{\theta_0}^n} = \frac{n}{2}\left(\frac{\sigma_0^2}{\sigma^2} - 1 - \log\frac{\sigma_0^2}{\sigma^2} + \frac{||X(\beta - \beta_0)||_2^2}{\sigma^2 n}\right)$$

Since

$$\log \frac{f_\theta^n}{f_{\theta_0}^n} = \frac{n}{2}\left(\log \frac{\sigma_0^2}{\sigma^2} + \frac{||Y - X\beta_0||^2}{n\sigma_0^2} - \frac{||Y - X\beta_0||^2}{n\sigma^2} - \frac{||X\beta_0 - X\beta||^2}{n\sigma^2} - \frac{2(Y - X\beta_0)^T(X\beta_0 - X\beta)}{n\sigma^2}\right)$$

we set $A = \log \frac{\sigma_0^2}{\sigma^2}$,$B = \frac{||Y-X\beta_0||^2}{n\sigma_0^2}$, $C = -\frac{||Y-X\beta_0||^2}{n\sigma^2}$, $D = -\frac{||X\beta_0-X\beta||^2}{n\sigma^2}$, $E = -\frac{2(Y-X\beta_0)^T(X\beta_0-X\beta)}{n\sigma^2}$

$$P_{\theta_0}^n\left(\log \frac{f_\theta^n}{f_{\theta_0}^n}\right)^2 = \frac{n^2}{4}P_0(A^2 + B^2 + C^2 + D^2 + E^2 + 2AB + 2AC + 2AD + 2AE + 2BC + 2BD + 2BE + 2CD + 2CE + 2DE)$$

with

$$P_{\theta_0}^n A^2 = \left(\log \frac{\sigma_0^2}{\sigma^2}\right)^2, \quad P_{\theta_0}^n B^2 = \frac{n+2}{n},$$
$$P_{\theta_0}^n C^2 = \frac{n+2}{n}\frac{\sigma_0^4}{\sigma^4}, \quad P_{\theta_0}^n D^2 = \frac{||X\beta_0 - X\beta||^4}{n^2\sigma^4},$$
$$P_{\theta_0}^n E^2 = \frac{4\sigma_0^2||X\beta_0 - X\beta||^2}{n^2\sigma^4}, \quad 2P_{\theta_0}^n A(B + C + D + E) = 2\log \frac{\sigma_0^2}{\sigma^2}(1 - \frac{\sigma_0^2}{\sigma^2} - \frac{||X\beta_0 - X\beta||^2}{n\sigma^2}),$$
$$2P_{\theta_0}^n BC = -\frac{2\sigma_0^2}{\sigma^2}\frac{n+2}{n}, \quad 2P_{\theta_0}^n BD = -\frac{2||X\beta_0 - X\beta||^2}{n\sigma^2},$$
$$2P_{\theta_0}^n BE = 0, \quad 2P_{\theta_0}^n CD = \frac{2\sigma_0^2||X\beta_0 - X\beta||^2}{n\sigma^4},$$
$$2P_{\theta_0}^n CE = 0, \quad 2P_{\theta_0}^n DE = 0.$$

By arranging the terms, we have:

$$P_{\theta_0}^n\left(\log \frac{f_\theta^n}{f_{\theta_0}^n}\right)^2 = \frac{n^2}{4}\left[\left(\frac{\sigma_0^2}{\sigma^2} - 1 - \log \frac{\sigma_0^2}{\sigma^2} + \frac{||X\beta_0 - X\beta||_2^2}{n\sigma^2}\right)^2 + \frac{4\sigma_0^2||X\beta_0 - X\beta||_2^2}{n^2\sigma^4} + \frac{2}{n}\left(1 - \frac{\sigma_0^2}{\sigma^2}\right)^2\right]$$

Since $V_{2,0}(P_{\theta_0}^n, P_\theta^n) = P_{\theta_0}^n\left(\log \frac{f_\theta^n}{f_{\theta_0}^n}\right)^2 - K(P_{\theta_0}^n, P_\theta^n)^2$,

$$V_{2,0}(P_{\theta_0}^n, P_\theta^n) = \frac{\sigma_0^2||X\beta_0 - X\beta||_2^2}{\sigma^4} + \frac{n}{2}\left(1 - \frac{\sigma_0^2}{\sigma^2}\right)^2$$

For $0 < \frac{\sigma_0^2}{\sigma^2} \leq 2$, considering the Taylor series centered at 1, we have

$$\log \frac{\sigma_0^2}{\sigma^2} = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}(\frac{\sigma_0^2}{\sigma^2} - 1)^n}{n} = \left(\frac{\sigma_0^2}{\sigma^2} - 1\right) - \frac{1}{2}\left(\frac{\sigma_0^2}{\sigma^2} - 1\right)^2 + \frac{1}{3}\left(\frac{\sigma_0^2}{\sigma^2} - 1\right)^3 - \frac{1}{4}\left(\frac{\sigma_0^2}{\sigma^2} - 1\right)^4 + ...$$

We see that

$$\frac{\sigma_0^2}{\sigma^2} - 1 - \log \frac{\sigma_0^2}{\sigma^2} \leq \frac{1}{2}\left(\frac{\sigma_0^2}{\sigma^2} - 1\right)^2$$

Define the two events $\mathcal{A}_1$ and $\mathcal{A}_2$ as follows:

$$\mathcal{A}_1 = \left\{ \sigma^2 : \left(\frac{\sigma_0^2}{\sigma^2} - 1 \right)^2 \leq \epsilon_n^2 \right\}$$
$$\mathcal{A}_2 = \left\{ (\beta, \sigma^2) : \frac{||X(\beta - \beta_0)||_2^2}{\sigma^2} \leq \frac{n^\alpha \epsilon_n^2}{p_n} \right\}$$

Condition on $\mathcal{A}_1 \cap \mathcal{A}_2$, we see that

$$\begin{aligned} K(P_{\theta_0}^n, P_\theta^n) &= \frac{n}{2} \left(\frac{\sigma_0^2}{\sigma^2} - 1 - \log \frac{\sigma_0^2}{\sigma^2} + \frac{||X(\beta - \beta_0)||_2^2}{\sigma^2 n} \right) \leq \frac{n\epsilon_n^2}{4} + \frac{n\epsilon_n^2}{6} \leq n\epsilon_n^2 \\ V_{2,0}(P_{\theta_0}^n, P_\theta^n) &= \frac{\sigma_0^2 ||X\beta_0 - X\beta||_2^2}{\sigma^4} + \frac{n}{2} \left(1 - \frac{\sigma_0^2}{\sigma^2} \right)^2 \\ &\leq \frac{n^\alpha \epsilon_n^2 (1 + \epsilon_n)}{p_n} + \frac{n\epsilon_n^2}{2} \\ &\leq n\epsilon_n^2 \end{aligned}$$

The last inequality is by assumption A1. Then we have:

$$\Pi \left(P_\theta^n : K(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2, V_{2,0}(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2 \right) \geq \Pi(\mathcal{A}_2 | \mathcal{A}_1) \Pi(\mathcal{A}_1).$$

First, we give a lower bound to $\Pi(\mathcal{A}_1)$. Since $\sigma^2 \sim \text{InvGamma}(\delta, \delta)$, we have

$$\begin{aligned} \Pi(\mathcal{A}_1) &\geq \frac{\delta^\delta}{\Gamma(\delta)} \int_{\frac{\sigma_0^2}{1+\epsilon_n}}^{\sigma_0^2} e^{-\delta/\sigma^2} (\sigma^2)^{-\delta-1} d\sigma^2 \\ &\geq \frac{\delta^\delta}{\Gamma(\delta)} \frac{\epsilon_n \sigma_0^2}{1 + \epsilon_n} \exp \left[-\frac{(1 + \epsilon_n)\delta}{\sigma_0^2} \right] (\sigma_0^2)^{-\delta-1} \\ &\geq \exp \left(-n\epsilon_n^2 \right) \end{aligned}$$

Next, we derive the lower bound for $\Pi(\mathcal{A}_1 | \mathcal{A}_1)$. Condition on $\mathcal{A}_1$ and by assumption A3, we have

$$\frac{||X(\beta - \beta_0)||_2^2}{\sigma^2} \leq \frac{\lambda_{\max}(X^T X)}{\sigma^2} \|\beta - \beta_0\|^2 \leq \frac{2kn^\alpha}{\sigma_0^2} \|\beta - \beta_0\|^2.$$

Define $\mathcal{A}_2^* = \left\{\beta : \|\beta - \beta_0\|^2 \leq \frac{\sigma_0^2\epsilon_n^2}{2kp_n}\right\}$, we have $\Pi(\mathcal{A}_2|\mathcal{A}_1) \geq \Pi(\mathcal{A}_2^*|\mathcal{A}_1)$. Thus,

$$\begin{aligned}
\Pi(\mathcal{A}_2|\mathcal{A}_1) &\geq \Pi(\|(\beta-\beta_0)\|_2^2 \leq \frac{\sigma_0^2\epsilon_n^2}{2kp_n}) \\
&\geq \Pi_{S_0}\left(||\beta_{S_0} - \beta_{0S_0}||_2^2 \leq \frac{\sigma_0^2\epsilon_n^2}{2kp_n}\frac{s_0}{p_n}\right)\Pi_{S_0^c}\left(||\beta_{s_0^c}||_2^2 \leq \frac{\sigma_0^2\epsilon_n^2}{2kp_n}\frac{p_n - s_0}{p_n}\right) \\
&\geq \prod_{j\in S_0}\Pi\left(|\beta_{0j} - \beta_j| \leq \frac{\sigma_0\epsilon_n}{p_n\sqrt{2k}}\right)\prod_{j\in S_0^C}\Pi\left(|\beta_j| \leq \frac{\sigma_0\epsilon_n}{p_n\sqrt{2k}}\right) \\
&\geq \left(\frac{2\sigma_0\epsilon_n}{p_n\sqrt{2k}}\inf_{\beta\in[-E,E]}\Pi_\lambda(\beta)\right)^{s_0}\left(1 - \int_{|\beta|\geq a_n}\Pi_\lambda(\beta)d\beta\right)^{p_n - s_0} \\
&\geq (2\epsilon_n)^{s_0}\exp\left(-2n\epsilon_n^2\right)(1 - p_n^{-(1+u)})^{p_n - s_0} \\
&\geq \exp\left(-Mn\epsilon_n^2\right),
\end{aligned}$$

for some $M > 0$. The fourth and fifth inequality are from Lemma A.3.1. The final inequality is by using the fact that the second term $\exp\left(-n\epsilon_n^2\right)$ dominates the first term and the third term is asymptotically 1.

Combining all the result above, we have

$$\Pi\left(P_\theta^n : K(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2, V_{2,0}(P_{\theta_0}^n, P_\theta^n) \leq n\varepsilon_n^2\right) \geq \Pi(\mathcal{A}_2|\mathcal{A}_1)\Pi(\mathcal{A}_1) \geq \exp\left(-C_1 n\epsilon_n^2\right)$$

where $C_1 = M_1 + M_2$.

Part 2. Sieve sequence condition:

First,we show the existence of sieve set F_n such that

$$\Pi_n\left(F_n^c\right) \leq \exp\left(-n\varepsilon_n^2(C_1+2)\right),$$

we choose $C_3 = M(C_1+3)$ with constant $M > 0$ and define F_n as:

$$F_n = \{S_\beta : |\gamma(\beta)| \leq C_3 s_0\} \cap \left\{\beta : \sum_{j=1}^p |\beta_j|^\alpha \leq C_3 E s_0\right\} \cap \left\{\sigma^2 : \frac{1}{n} \leq \sigma^2 \leq p_n^{C_3 s_0/\delta}\right\}$$

It is easy to see that:

$$\begin{aligned}
\Pi_n\left(F_n^c\right) &= \Pi(|\gamma(\beta)| > C_3 s_0) + \Pi\left(\{S_\beta : |\gamma(\beta)| \le C_3 s_0\} \cap \left\{\beta : \sum_{j=1}^{p} |\beta_j|^\alpha > C_3 E s_0\right\}\right) \\
&\quad + \Pi(\sigma^2 < \frac{1}{n}) + \Pi(\sigma^2 > p_n^{C_3 s_0/\delta}) \\
&\le \Pi(|\gamma(\beta)| > C_3 s_0) + \Pi\left(\left\{\beta : \sum_{j=1}^{p} |\beta_j|^\alpha > C_3 E s_0\right\} \Big| \{S_\beta : |\gamma(\beta)| \le C_3 s_0\}\right) \\
&\quad + \Pi(\sigma^2 < \frac{1}{n}) + \Pi(\sigma^2 > p_n^{C_3 s_0/\delta}) \\
&= (a) + (b) + (c) + (d)
\end{aligned}$$

We now show $(a) + (b) + (c) + (d) \le 4\exp\left(-n\varepsilon_n^2(C_1+3)\right)$. For term (a), we see that

$$\Pi(|\gamma(\beta)| > C_3 s_0) = P(\text{Binomial}(p_n, v_n) \ge C_3 s_0)$$

where $v_n = \int_{|\beta| \ge a_n} \Pi_\lambda(\beta) d\beta$. By Lemma A.3 in [Song and Liang, 2017], we have

$$\Pi(|\gamma(\beta)| > C_3 s_0) \le 1 - \Phi(\sqrt{2 p_n H\left[v_n, (C_3 s_0 - 1)/p_n\right]}) \le \frac{\exp\left\{-p_n H\left[v_n, (C_3 s_0 - 1)/p_n\right]\right\}}{\sqrt{2\pi}\sqrt{2 p_n H\left[v_n, (C_3 s_0 - 1)/p_n\right]}}$$

where $p_n H[v_n, (C_3 s_0 - 1)/p_n] = (C_3 s_0 - 1)\log\frac{C_3 s_0 - 1}{p_n v_n} + (p_n - C_3 s_0 + 1)\log\frac{p_n - C_3 s_0 + 1}{p_n - p_n v_n}$. Therefore, to show $\Pi\left(|\gamma(\beta)| > C_3 s_0\right) \le \exp\left(-(C_1+3) n\epsilon_n^2\right)$, it sufficient to show that

$$p_n H[v_n, (C_3 s_0 - 1)/p_n] \ge (C_1 + 3) n\epsilon_n^2$$

For the first term of $p_n H[v_n, (C_3 s_0 - 1)/p_n]$, by Lemma A.3.1, $v_n \le p_n^{-(1+u)}$, we have:

$$\begin{aligned}
(C_3 s_0 - 1)\log\frac{C_3 s_0 - 1}{p_n v_n} &\ge (C_3 s_0 - 1)\log(C_3 s_0 - 1) p_n^u \\
&\ge u C_3 s_0 \log p_n - u\log p_n + (C_3 s_0 - 1)\log(C_3 s_0 - 1) \\
&\ge u C_3 n\epsilon_n^2 - u\log p_n \\
&= u M (C_1 + 3) n\epsilon_n^2 - u\log p_n \\
&\ge 0.5 u M (C_1 + 3) n\epsilon_n^2.
\end{aligned}$$

The last inequality will hold for n sufficient large because the term $n\epsilon_n^2$ always dominates $\log p_n$. We set $M = \frac{2}{u}$, then we have

$$(C_3 s_0 - 1)\log\frac{C_3 s_0 - 1}{p_n v_n} \ge (C_1 + 3) n\epsilon_n^2.$$

For the second term of $p_n H[v_n, (C_3 s_0 - 1)/p_n]$, we have

$$0 < (p_n - C_3 s_0 + 1) \log \frac{p_n - C_3 s_0 + 1}{p_n - p_n v_n} \approx C_3 s_0 - \frac{(C_3 s_0)^2}{p_n} < n\epsilon_n^2.$$

Consequently,

$$p_n H[v_n, (C_3 s_0 - 1)/p_n] \geq (C_1 + 3) n \epsilon_n^2.$$

For term (b), when n is large, it is sufficient to show that

$$\Pi\left(\left\{\beta : \sum_{j:|\beta_j|>a_n} |\beta_j|^\alpha > \frac{3}{4} C_3 E s_0\right\} \Big| \{S_\beta : |\gamma(\beta)| \leq C_3 s_0\}\right) \leq \exp\left(-n\varepsilon_n^2 (C_1 + 3)\right)$$

Since $\sum_{j=1}^{C_3 s_0} |\beta_j|^\alpha$ is Gamma distributed with shape parameter $\frac{C_3 s_0}{\alpha}$ and scale parameter λ_n,applying the Chernoff bound to Gamma distribution and choosing E as any positive constant such that $E > 4$, we have

$$\pi(\sum_{j=1}^{C_3 s_0} |\beta_j|^\alpha > \frac{3}{4} C_3 E s_0) \leq e^{-3 C_3 s_0 \log p_n}.$$

Therefore,

$$\begin{aligned}
\Pi\left(\left\{\beta : \sum_{j:|\beta_j|>a_n} |\beta_j|^\alpha > \frac{3}{4} C_3 E s_0\right\} \Big| \{S_\beta : |\gamma(\beta)| \leq C_3 s_0\}\right) &\leq \sum_{s=1}^{C_3 s_0} \binom{p_n}{s} \pi(\sum_{j=1}^{s} |\beta_j|^\alpha > \frac{3}{4} C_3 E s_0) \\
&\leq C_3 s_0 \binom{p_n}{C_3 s_0} \pi(\sum_{j=1}^{C_3 s_0} |\beta_j|^\alpha > \frac{3}{4} C_3 E s_0) \\
&\leq e^{C_3 s_0 (\log p_n + 1)} e^{-3 C_3 s_0 \log p_n} \\
&\leq e^{-C_3 s_0 \log p_n}
\end{aligned}$$

where the second is obtained by the upper bound $\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$ for $\forall 1 \leq k \leq n$.

For term (c)

$$\Pi\left(\sigma^2 < \frac{1}{n}\right) = \Pi\left(\frac{1}{\sigma^2} > n\right) \leq e^{-n\delta} \leq e^{-C_3 s_0 \log p_n \delta}$$

where the first inequality is followed by using the Chernoff bound to Gamma distribution with shape parameter δ and rate parameter δ and the second inequality is followed by assumption A2.

For term (d):

$$\begin{aligned}\Pi(\sigma^2 > p_n^{C_3 s_0/\delta}) &= \int_{p_n^{C_3 s_0/\delta}}^{\infty} \frac{\delta^\delta}{\Gamma(\delta)} \left(\sigma^2\right)^{-\delta-1} e^{-\delta/\sigma^2} d\sigma^2 \\ &\leq \int_{p_n^{C_3 s_0/\delta}}^{\infty} \frac{\delta^\delta}{\Gamma(\delta)} \left(\sigma^2\right)^{-\delta-1} d\sigma^2 \\ &= \frac{\delta^{\delta+1}}{\Gamma(\delta)} p_n^{-C_3 s_0} \\ &= \frac{\delta^{\delta+1}}{\Gamma(\delta)} e^{-C_3 s_0 \log p_n}\end{aligned}$$

Part 3. Testing conditions:

When the variance is unknown, techniques from [Ghosal et al., 2007, Ghosal and Van der Vaart, 2017] can not be applied directly. Instead, we follow [Ning et al., 2020]'s approach by constructing the likelihood ratio test with the sieve F_n broken up into small pieces. More precisely, we perform tests in small covering pieces such that

$$\|\beta_1 - \beta\|_2 \leq \frac{\sqrt{c}\epsilon_n}{\sqrt{v_1 n}}, \quad |\sigma_1^2 - \sigma^2| \leq \frac{1}{n^2} \tag{A.7}$$

where (β, σ^2) and (β_1, σ_1^2) are parameters which belong to the probability density f_θ^n and $f_{\theta_1}^n$ respectively.

In each piece, we consider testing $H_0 : f_\theta^n = f_{\theta_0}^n$ against $H_1 : f_\theta^n = f_{\theta_1}^n$ with $\theta_1 \in F_n$ and $d(P_{\theta_1}^n, P_{\theta_0}^n) \geq M\epsilon_n$. To do that, we use the likelihood ratio test $\phi_n = 1\left\{\frac{f_\theta^n}{f_{\theta_0}^n} \geq 1\right\}$ which is the most powerful Neyman-Pearson test.

Since $d(P_{\theta_1}^n, P_{\theta_0}^n) = [-n^{-1}\log(\int \sqrt{f_{\theta_1}^n f_{\theta_0}^n} dY)]^{1/2} \geq M\epsilon_n$, $\int \sqrt{f_{\theta_1}^n f_{\theta_0}^n} dY \leq e^{-nM^2\epsilon_n^2}$ then by Markov inequality

$$\begin{aligned}\mathbb{E}_{f_{\theta_0}^n} \phi_n = \mathbb{E}_{f_{\theta_0}^n} \left(\sqrt{f_{\theta_1}^n / f_{\theta_0}^n} \geq 1\right) \leq \int \sqrt{f_{\theta_1}^n f_{\theta_0}^n} dY \leq e^{-nM^2\epsilon_n^2} \\ \mathbb{E}_{f_{\theta_1}^n} (1 - \phi_n) = \mathbb{E}_{f_{\theta_1}^n} \left(\sqrt{f_{\theta_0}^n / f_{\theta_1}^n} \geq 1\right) \leq \int \sqrt{f_1^n f_2^n} dY \leq e^{-nM^2\epsilon_n^2}\end{aligned}$$

For the type II error, by Cauchy-Schwartz inequality

$$\mathbb{E}_{f_\theta^n} (1 - \phi_n) \leq \left\{\mathbb{E}_{f_{\theta_1}^n} (1 - \phi_n)\right\}^{1/2} \left\{\mathbb{E}_{f_{\theta_1}^n} \left(f_\theta^n / f_{\theta_1}^n\right)^2\right\}^{1/2}$$

Thus, if we can show that $\mathbb{E}_{f_{\theta_1}^n}\left(f_\theta^n/f_{\theta_1}^n\right)^2 \leq e^{cn\epsilon_n^2}$ for very small c, then our work finish.

To see this, observe that

$$\begin{aligned}\mathbb{E}_{f_{\theta_1}^n}\left(f_\theta^n/f_{\theta_1}^n\right)^2 &= \left(\frac{\sigma^2}{2\sigma^2-\sigma_1^2}\right)^{\frac{n}{2}}\left(\frac{\sigma^2}{\sigma_1^2}\right)^{\frac{n}{2}}\exp\left(\frac{\|g(X)-g_1(X)\|^2}{2\sigma^2-\sigma_1^2}\right)\\ &= \left(\frac{1}{2-\frac{\sigma_1^2}{\sigma^2}}\right)^{\frac{n}{2}}\left(\frac{\sigma^2}{\sigma_1^2}\right)^{\frac{n}{2}}\exp\left(\frac{\|X\beta-X\beta_1\|^2}{\sigma^2+\sigma_1^2}\frac{1+\frac{\sigma_1^2}{\sigma^2}}{2-\frac{\sigma_1^2}{\sigma^2}}\right)\end{aligned} \tag{A.8}$$

Because $|\sigma_1^2-\sigma^2| \leq \frac{1}{n^2}$ and $\frac{1}{\sigma^2} \leq n$,we obtained the upper bounded

$$\begin{aligned}\mathbb{E}_{f_{\theta_1}^n}\left(f_\theta^n/f_{\theta_1}^n\right)^2 &\leq \left(\frac{1}{1-1/n}\right)^n\exp\left(\frac{\|X\beta-X\beta_1\|^2}{\sigma^2+\sigma_1^2}\frac{1+\frac{\sigma_1^2}{\sigma^2}}{2-\frac{\sigma_1^2}{\sigma^2}}\right)\\ &\leq \left(\frac{1}{1-1/n}\right)^n\exp\left(\frac{2\|X\beta-X\beta_1\|^2}{\sigma^2+\sigma_1^2}\right)\\ &\leq \left(\frac{1}{1-1/n}\right)^n\exp\left(\frac{2nv_1\|\beta-\beta_1\|^2}{\sigma^2+\sigma_1^2}\right)\\ &\leq \left(\frac{1}{1-1/n}\right)^n\exp\left(n^2v_1\|\beta-\beta_1\|^2\right)\\ &\leq \exp\left(cn\epsilon_n^2\right)\end{aligned}$$

where the third inequality is by assumption A3, the fourth inequality is obtained when σ_1^2 and σ^2 are sufficient close(n sufficient large). To complete the construction of the test, we need to show that $\log N^* \lesssim n\epsilon_n^2$, where N^* is the number of covering pieces for sieve F_n satisfying $(A.7)$. We see that

$$\begin{aligned}\log N^* &\leq \log N\left(\frac{\sqrt{c}\epsilon_n}{\sqrt{v_1 n}}, \{\beta\in F_n\}, \|\cdot\|_2\right) + \log N\left(\frac{1}{n^2}, \left\{\sigma^2 : \sigma^2 \in F_n\right\}, |\cdot|\right)\\ &\leq \log N\left(\frac{\sqrt{c}\epsilon_n}{\sqrt{v_1 n}}, \left\{\beta : |\gamma(\beta)| \leq C_3 s_0, \|\beta\|_2 \leq (C_3 E s_0)^{\frac{1}{\alpha}}\right\}, \|\cdot\|_2\right)\\ &+ \log N\left(\frac{1}{n^2}, \left\{\sigma^2 : \sigma^2 \leq p_n^{C_3 s_0/\delta}\right\}, |\cdot|\right)\\ &\leq \log C_3 s_0 + \log\binom{p_n}{C_3 s_0} + C_3 s_0 \log 3\sqrt{v_1 n}(C_3 E s_0)^{\frac{1}{\alpha}} + \log 3 + 2\log n + \frac{C_3 s_0 \log p_n}{\delta}\\ &\lesssim n\epsilon_n^2\end{aligned}$$

where in the first term of the second inequality, we are using the fact that $\|\beta\|_2 \leq \|\beta\|_1 \leq \|\beta\|_\alpha \leq (C_3 E s_0)^{\frac{1}{\alpha}}$ with $0 < \alpha < 1$. We already show that $\Pi_n \left(d(P_\theta^n, P_{\theta_0}^n) \geq M\varepsilon_n \mid X_1, \ldots, X_n \right) \to 0$ in $P_{\theta_0}^n-$ probability. Now we want to argue this implies the posterior consistency in terms of Euclidean distance.

By lemma 3, we see that $d(P_\theta^n, P_{\theta_0}^n) \leq M\varepsilon_n$ implies

$$\frac{1}{2} \log \frac{\sigma^2 + \sigma_0^2}{2\sigma\sigma_0} \leq M^2 \epsilon_n^2, \quad \frac{\|X\beta - X\beta_0\|}{4n(\sigma^2 + \sigma_0^2)} \leq M^2 \epsilon_n^2.$$

The first inequality above implies that

$$2 \leq \frac{\sigma}{\sigma_0} + \frac{\sigma_0}{\sigma} \leq 2e^{2M^2\epsilon_n^2}.$$

For n sufficient large, there exists c such that $\frac{\sigma}{\sigma_0} = 1 + c < e^{2M^2\epsilon_n^2}$, we have $c < 2M^2\epsilon_n^2$. This implies that $|\sigma - \sigma_0| \leq 2\sigma_0 M^2 \epsilon_n^2$. By assumption A4 and Theorem 3.2 for n sufficient large, the second inequality implies $\|\beta - \beta_0\|_2 \leq 5\sigma_0^2 M^2 \epsilon_n^2 / v_1$. Consequently, we have

$$\Pi_n \left(\|\beta - \beta_0\|_2 \gtrsim \varepsilon_n \mid X_1, \ldots, X_n \right) \to 0$$
$$\Pi_n \left(|\sigma - \sigma_0| \gtrsim \sigma_0 \varepsilon_n \mid X_1, \ldots, X_n \right) \to 0$$

in $P_{\theta_0}^n-$ probability. □

Proof of theorem 3.2

Proof. We define the set of event $Z_n = \left\{ \int\int \frac{f_\theta^n}{f_{\theta_0}^n} d\Pi(\beta)\Pi(\sigma^2) \geq e^{-(1+D+C_1)n\epsilon_n^2} \right\}$ for some positive constant D and $T = \{\beta : |\gamma(\beta)| \leq C_3 s_0\}$. Then

$$\Pi(T^c|Y) = \frac{\int\int_{T^c} \frac{f_\theta^n}{f_{\theta_0}^n} d\Pi(\beta)\Pi(\sigma^2)}{\int\int \frac{f_\theta^n}{f_{\theta_0}^n} d\Pi(\beta)\Pi(\sigma^2)}.$$

We have that:

$$P_{\theta_0}^n \Pi(T^c|Y) \leq P_{\theta_0}^n \Pi(T^c|Y) 1_{Z_n} + P_{\theta_0}^n(Z_n^c)$$

For the first term of right hand side, we have:

$$P_0^n \Pi(T^c|Y) 1_{Z_n} \leq e^{(1+D+C_1)n\epsilon_n^2} \int\int\int_{T^c} \frac{f_\theta^n}{f_{\theta_0}^n} d\Pi(\beta)\Pi(\sigma^2) dP_0^n \leq \int_{T^c} \Pi(\beta) = \Pi(|\gamma(\beta)| \leq C_3 s_0)$$

In part 2 of the proof of Theorem 3.1, we see that

$$\Pi(|\gamma(\beta)| \leq C_3 s_0) \leq \exp\left(-n\varepsilon_n^2(C_1+2)\right)$$

For the second term of the right hand side, by applying Lemma A.3.4, we have

$$P_{\theta_0}^n(Z_n^c) \leq \frac{1}{D^2 n\epsilon_n^2}.$$

Combining the two results, we have:

$$P_{\theta_0}^n \Pi(T^c|Y) \leq \exp\left(-n\varepsilon_n^2(C_1+2)\right) + \frac{1}{D^2 n\epsilon_n^2}$$

consequently, as $n \to \infty$, $P_{\theta_0}^n \Pi(T^c|Y) \to 0$. □

Appendix B

B.1 The KKT condition

Proof of Theorem 5.1

Proof. When $\beta_j \neq 0$, taking $\frac{\partial L}{\partial \beta_j} = 0$ yields

$$|\beta_j| = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}|z_j| - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{|\beta_j| + C_2|\beta_j|^{1-\frac{1}{2^\gamma}}} \tag{B.1}$$

and $\text{sign}(z_j) = \text{sign}(\beta_j)$. Thus, without loss of generality, we assume $\beta_j \geq 0$.

Step 1: We first show that there exists a hard threshold in the KKT condition. When $\beta_j \neq 0$ is the solution, define

$$\psi(\beta_j) = \beta_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{1-\frac{1}{2^\gamma}}}.$$

Then

$$\begin{aligned}
\psi^{'}(\beta_j) =& 1 - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{1 + (1-\frac{1}{2^\gamma}) C_2 \beta_j^{-\frac{1}{2^\gamma}}}{\beta_j^2 \left(1 + C_2 \beta_j^{-\frac{1}{2^\gamma}}\right)^2} \\
\psi^{''}(\beta_j) =& (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{C_2(1-\frac{1}{2^\gamma})\frac{1}{2^\gamma}(\beta_j^{-\frac{1}{2^\gamma}} + C_2 \beta_j^{-\frac{1}{2^{\gamma-1}}}) + 2[1 + (1-\frac{1}{2^\gamma}) C_2 \beta_j^{-\frac{1}{2^\gamma}}]^2}{(\beta_j + C_2 \beta_j^{1-\frac{1}{2^\gamma}})^2} > 0,
\end{aligned} \tag{B.2}$$

therefore, $\psi(\beta_j)$ is minimised at $\tilde{\beta}_j$ where $\psi^{'}(\tilde{\beta}_j) = 0$. i.e., $\tilde{\beta}_j$ satisfy the following equation

$$1 = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{1 + (1 - \frac{1}{2^\gamma}) C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}}}{\tilde{\beta}_j^2 \left(1 + C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}}\right)^2} \tag{B.3}$$

with minimum value

$$h(\tilde{\beta}_j) = \tilde{\beta}_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\tilde{\beta}_j + C_2 \tilde{\beta}_j^{(1-\frac{1}{2^\gamma})}}.$$

Re-writing Equation (B.1) as

$$(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| = \beta_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{1-\frac{1}{2^\gamma}}},$$

we see that $\psi(\beta_j) = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|$ has no solution when $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| < h(\tilde{\beta}_j)$, in this case, $\beta_j = 0$. For $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| \geq h(\tilde{\beta}_j)$, there exists at least one solution. As $\psi(\beta_j)$ is strictly increasing when $\beta_j \geq \tilde{\beta}_j$, there exists $\beta_j^{''} \in [\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]$ such that $\psi(\beta_j^{''}) = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|$.

Step 2: Next, we show the convergence of fixed point iteration at $\beta_j^{''} \in [\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]$. Let

$$\rho(\beta_j) = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{1-\frac{1}{2^\gamma}}},$$

we need to check that

$$\rho([\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]) \in [\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|].$$

$\rho([\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]) \leq (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|$ is follows from the definition of $\rho(\cdot)$. We see that for $\beta_j \in [\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]$

$$h(\tilde{\beta}_j) = \tilde{\beta}_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\tilde{\beta}_j + C_2 \tilde{\beta}_j^{(1-\frac{1}{2^\gamma})}} \geq \tilde{\beta}_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{(1-\frac{1}{2^\gamma})}}.$$

Using the fact that $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| > h(\tilde{\beta}_j)$, we have

$$(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| \geq \tilde{\beta}_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{(1-\frac{1}{2^\gamma})}}.$$

Thus, $\rho([\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]) \geq \tilde{\beta}_j$. Now we show that $\rho(\cdot)$ is a a contraction mapping on $[\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]$. We have $\rho^{'}(\beta_j) = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{1 + (1 - \frac{1}{2^\gamma}) C_2 \beta_j^{-\frac{1}{2^\gamma}}}{\beta_j^2 \left(1 + C_2 \beta_j^{-\frac{1}{2^\gamma}}\right)^2} > 0$ and $\rho^{''}(\beta_j) =$

$-\psi^{''}(\beta_j) < 0$. Since $\rho^{'}(\tilde{beta}_j) = 1$, $\rho^{'}(\beta_j) < 1$ for $\beta_j > \tilde{\beta}_j$. Therefore, $\rho(\cdot)$ is a a contraction mapping on $[\tilde{\beta}_j, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|]$. Since $[\tilde{\beta}_j, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|]$ is complete, by Banach fixed point theorem, there exists one and only one fixed point $\beta_j^{''} \in [\tilde{\beta}_j, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|]$ such that

$$\beta_j^{''} = (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| - (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}\frac{C_1}{\beta_j^{''} + C_2\beta_j^{''1-\frac{1}{2^\gamma}}} \tag{B.4}$$

Step 3:Up to now, we showed that when $(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| > h(\tilde{\beta}_j)$, there exists a $\beta_j^{''} \in [\tilde{beta}_j, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|]$ such that the KKT condition is satisfied. We haven't precluded the potential solutions $\beta_j^{'} \in (0, \tilde{\beta})$, which also satisfies the KKT condition. Now we want to claim that $\beta_j^{''}$ is the only local minimum for $\beta \in (0, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|]$ given the parameters in other dimension are fixed. We write the descent function

$$\begin{aligned}\delta_{-j}(|\beta_j|) = L_{-j}(\beta_j) - L_{-j}(0) &= \frac{1}{2}\|Y_{-j} - X_j\beta_j\|^2 - \frac{1}{2}\|Y_{-j}\|^2 + (2^\gamma p + 0.5)\log\left(1 + \frac{|\beta_j|^{\frac{1}{2^\gamma}}}{C_2}\right) \\ &= \frac{(X_j^TX_j)}{2}\beta_j^2 - |z_j||\beta_j| + (2^\gamma p + 0.5)\log\left(1 + \frac{|\beta_j|^{\frac{1}{2^\gamma}}}{C_2}\right)\end{aligned} \tag{B.5}$$

where $Y_{-j} = Y - X_{-j}\beta_{-j}$. Then it is easy to see that

$$\delta_{-j}^{''}(|\beta_j|) = (X_j^TX_j)\psi^{'}(|\beta_j|), \quad \delta_{-j}^{'''}(|\beta_j|) = (X_j^TX_j)\psi^{''}(|\beta_j|) > 0$$

where the derivative is taken with respect to $|\beta_j|$. Since $\delta_{-j}^{''}(\tilde{\beta}_j) = 0$ and $\delta_{-j}^{'''}(|\beta_j|) > 0$, we see that $\delta_{-j}^{''}(|\beta_j|) < 0$ for $|\beta_j| \in (0, \tilde{\beta}_j)$ and $\delta_{-j}^{''}(|\beta_j|) > 0$ for $|\beta_j| \in (\tilde{\beta}, (X_j^TX_j)^{-1}|z_j|)$. Thus, if there exists $\beta_j^{'} \in (0, \tilde{\beta})$ such that $\delta_{-j}^{'}(\beta_j^{'}) = 0$, then $\beta_j^{'}$ can only be the local maximum. $\beta_j^{''}$ is the only local minimum.

Step 4. To find the global minimum when $(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| > h(\tilde{\beta}_j)$, we need to check the value of descent function $\delta_{-j}(\beta_j^{''})$ because we couldn't preclude the case that 0 is the global minimum. To see the existence of this potential case, we provide a constructive proof. Suppose $(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| > h(\tilde{\beta}_j)$, plug in (B.4) into (B.5), then we have

$$\delta_{-j}(\beta_j^{''}) = -\frac{(X_j^TX_j)}{2}\beta_j^{''2} - \frac{C_1}{1 + C_2\beta_j^{''-\frac{1}{2^\gamma}}} + (2^\gamma p + 0.5)\log\left(1 + \frac{\beta_j^{''\frac{1}{2^\gamma}}}{C_2}\right)$$

we have $\delta_{-j}(\beta_j'') > 0$ if

$$(2^\gamma p + 0.5)\log\left(1 + \frac{\beta_j''^{\frac{1}{2^\gamma}}}{C_2}\right) > \frac{(X_j^T X_j)}{2}\beta_j''^2 + \frac{C_1}{1 + C_2\beta_j''^{-\frac{1}{2^\gamma}}}.$$

Thus, there exists sufficient small m such that for $\beta'' < m$, the inequality above will hold. We rewrite Equation (B.4) as

$$f(\beta_j'', |z_j|) = \beta_j'' - (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| + (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}\frac{C_1}{\beta_j'' + C_2\beta_j''^{1-\frac{1}{2^\gamma}}} = 0.$$

Since $\frac{\partial f}{\partial \beta_j''} > 0$ for $\beta_j'' \in (\tilde{\beta}, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|)$, by implicit function theorem, there exists $g \in \mathrm{C}^1$ such that $g(|z_j|) = \beta_j''$. In addition, the fixed point iteration guarantees that g is a one to one map for $\beta \in (\tilde{\beta}, (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j|)$. By Equation $(B.4)$, it is easy to see that g is a monotonic increasing function. Hence, for $|z_j| < g^{-1}(m)$ the inequality will hold. By Lemma 5.2 in the main paper, we have $2\tilde{\beta}_j \leq h(\tilde{\beta}_j) \leq 3\tilde{\beta}_j$ and

$$\tilde{\beta}_j^{(2-\frac{1}{2^\gamma})} = \frac{C_1 M'(\gamma)}{\boldsymbol{X}_j^T\boldsymbol{X}_j}\frac{1}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2}$$

where $M'(\gamma) \in (\frac{1}{2}, 1]$. By setting b sufficient small, it is possible to construct $h(\tilde{\beta}_j)$ such that

$$h(\tilde{\beta}_j) < (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| < (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}g^{-1}(m).$$

□

Proof of Lemma 5.2

Proof. From Equation (10) in the main paper, we plug $\frac{(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}C_1}{\tilde{\beta}_j\left(1+C_2\tilde{\beta}_j^{-\frac{1}{2^\gamma}}\right)} = \frac{\tilde{\beta}_j\left(1+C_2\tilde{\beta}_j^{-\frac{1}{2^\gamma}}\right)}{1+(1-\frac{1}{2^\gamma})C_2\tilde{\beta}_j^{-\frac{1}{2^\gamma}}}$ into $h(\tilde{\beta}_j)$, then

$$\begin{aligned} h(\tilde{\beta}_j) &= \tilde{\beta}_j + (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}\frac{C_1}{\tilde{\beta}_j + C_2\tilde{\beta}_j^{(1-\frac{1}{2^\gamma})}} = \tilde{\beta}_j + \frac{\tilde{\beta}_j\left(1 + C_2\tilde{\beta}_j^{-\frac{1}{2^\gamma}}\right)}{1 + (1-\frac{1}{2^\gamma})C_2\tilde{\beta}_j^{-\frac{1}{2^\gamma}}} \\ &= 2\tilde{\beta}_j + \frac{\frac{1}{2^\gamma}C_2}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + (1-\frac{1}{2^\gamma})C_2}\tilde{\beta}_j \leq 2\tilde{\beta}_j + \frac{\frac{1}{2^\gamma}}{(1-\frac{1}{2^\gamma})}\tilde{\beta}_j \leq 3\tilde{\beta}_j. \end{aligned}$$

By arranging $1 = (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{1+(1-\frac{1}{2^\gamma})C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}}}{\tilde{\beta}_j^2 \left(1+C_2 \tilde{\beta}_j^{-\frac{1}{2^\gamma}}\right)^2}$, we have

$$\tilde{\beta}_j^2 = \frac{C_1}{\boldsymbol{X}_j^T \boldsymbol{X}_j} \frac{\tilde{\beta}_j^{\frac{1}{2^\gamma}}}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2} \left(1 - \frac{\frac{1}{2^\gamma} C_2}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2}\right)$$

$1 - \frac{\frac{1}{2^\gamma} C_2}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2} = M'(\gamma)$ for $M'(\gamma) \in (\frac{1}{2}, 1]$. Therefore,

$$\tilde{\beta}_j^{(2-\frac{1}{2^\gamma})} = \frac{C_1 M'(\gamma)}{\boldsymbol{X}_j^T \boldsymbol{X}_j} \frac{1}{\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2} = M'(\gamma) \frac{p + \frac{1}{2^{\gamma+1}}}{n\tilde{\beta}_j^{\frac{1}{2^\gamma}} + nC_2}$$

Since we have $\tilde{\beta}_j < (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j^{(i)}|$, it follows directly that

$$2 \left\{ \frac{C_1 (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}}{2C_2 + 2[(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j|]^{\frac{1}{2^\gamma}}} \right\}^{\frac{1}{2-\frac{1}{2^\gamma}}} \leq h(\tilde{\beta}_j).$$

□

B.2 Convergence analysis

Theorem 5.3 will follow directly by the Lemmas below.

Lemma B.2.1. *The sequence $\{\beta^i\}_{i \geq 1}$, returned by the CD algorithm is a bounded sequence.*

Proof. Given the initial sequence β^1, then the entire sequence $\{\beta^i\}_{i \geq 1}$ belongs to the sublevel set $\{\beta : L(\beta) \leq L(\beta^1)\}$. This is easy to see as the convergence of loss function $L(\beta)$ is guaranteed by Theorem 5.1, which shows that we can achieve the steepest descent in each coordinate. Since for any $j \in \{1, 2, .., p\}$, $\log(|\beta_j|^{\frac{1}{2^\gamma}}) \leq L(\beta) \leq L(\beta^1)$, this implies that $\beta_j^{(i)}$ is bounded for any $j \in \{1, 2, .., p\}$. This completes the proof. □

Lemma B.2.2. *Define the descent function as:*

$$\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = L_{-j}(\beta_j^{(i)}) - L_{-j}(\beta_j^{(i+1)})$$

where $L_{-j}(\cdot)$ is the loss function for β_j with all other parameters held fixed. Then $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$ if and only if $\beta_j^{(i)} = \beta_j^{(i+1)}$.

Proof. We show that if $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$, then $\beta_j^{(i)} = \beta_j^{(i+1)}$. For the reverse direction, the result is trivial.

Case 1. $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}|z_j^{(i)}| \geq h(\tilde{\beta}_j^{(i)})$ and $\delta_{-j}(\beta_j^{(i)''}) < 0$. In this case, $\beta_j^{(i+1)}$ is the unique univariate global minimizer of $L_{-j}(\cdot)$. Thus, $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$ implies $\beta_j^{(i)} = \beta_j^{(i+1)}$.(The uniqueness of global minimum is shown in the step 3 and step 4 of the proof of theorem 5.1).

Case 2. $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}|z_j^{(i)}| \geq h(\tilde{\beta}_j^{(i)})$ and $\delta_{-j}(\beta_j^{(i)''}) = 0$. In this case, the unique univariate global minimum attains at both $\text{sign}(z_j^{(i)})\beta_j^{(i)''}$ and 0. By the rule of $T(\cdot)$ map (see Theorem 5.3), we have $\beta_j^{(i+1)} = \text{sign}(z_j^{(i)})\tilde{\beta}_j^{(i)} I(\beta_j^{(i)} \neq 0)$. If $\beta_j^{(i)} = 0$, then $\beta_j^{(i+1)} = \beta_j^{(i)} = 0$. If $\beta_j^{(i)} \neq 0$, $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$ implies $\beta_j^{(i)} = \beta_j^{(i+1)} = \text{sign}(z_j^{(i)})\tilde{\beta}_j^{(i)}$.

Case 3.$(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}|z_j^{(i)}| \geq h(\tilde{\beta}_j^{(i)})$ and $\delta_{-j}(\beta_j^{(i)''}) > 0$. In this case, 0 is the unique univariate global minimizer of $L_{-j}(\cdot)$. Thus, $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$ implies $\beta_j^{(i)} = \beta_j^{(i+1)} = 0$.

Case 4 $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1}|z_j^{(i)}| < h(\tilde{\beta}_j^{(i)})$. In this case, 0 is the unique univariate global minimizer of $L_{-j}(\cdot)$. Thus, $\triangle(\beta_j^{(i)}, \beta_j^{(i+1)}) = 0$ implies $\beta_j^{(i)} = \beta_j^{(i+1)} = 0$. □

Lemma B.2.3. *Suppose there exists two convergent subsequences $\{\boldsymbol{\beta}^{n_i}\}_i \subset \left\{\boldsymbol{\beta}^i\right\}_i$ and $\left\{\boldsymbol{\beta}^{n_i+1}\right\}_i \subset \left\{\boldsymbol{\beta}^i\right\}_i$ such that $\boldsymbol{\beta}^{n_i} \to \beta^*$ and $\boldsymbol{\beta}^{n_i+1} \to \beta^{**}$. Then their differences converge to zero:*

$\boldsymbol{\beta}^{n_i+1} - \boldsymbol{\beta}^{n_i} \to \beta_j^{**} - \beta_j^* = 0$.

Proof. First, we want to check the continuity of some functions. We write $z_j = z(\beta_{-j}) = \boldsymbol{X}_j^T(\boldsymbol{Y} - \boldsymbol{X}\beta_{-j})$ and $h(\tilde{\beta}_j) = h(\tilde{\beta}_j(C_2(\beta_{-j})))$. Then the continuity of $z(\beta_{-j})$ is clear.

We see that both $C_2(\beta_{-j}) = \|\beta_{-j}\|^{\frac{1}{2^\gamma}}_{\frac{1}{2^\gamma}} + 1/b$ and $h(\tilde{\beta}_j) = \tilde{\beta}_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\tilde{\beta}_j + C_2 \tilde{\beta}_j^{(1-\frac{1}{2^\gamma})}}$ are continuous maps. From step 1 of the proof of theorem 5.1, $\tilde{\beta}_j$ is the unique solution of the equation

$$f_1(C_2, \beta_j) = 1 - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} C_1 \frac{1 + (1 - \frac{1}{2^\gamma}) C_2 \beta_j^{-\frac{1}{2^\gamma}}}{\beta_j^2 \left(1 + C_2 \beta_j^{-\frac{1}{2^\gamma}}\right)^2} = 0.$$

Since $\frac{\partial f_1}{\partial \beta_j} = \psi^{''}(\beta_j) > 0$, then by implicit function theorem, $\tilde{\beta}_j(C_2)$ is continuous. Therefore, the the composition function $h(\tilde{\beta}_j(C_2(\beta_{-j})))$ is continuous. Consequently, if $\boldsymbol{\beta}^{n_i} \to \beta^*$, we also have $z_j^{(n_i)} \to z_j^{(*)}$ and $h(\tilde{\beta}_j^{(n_i)}) \to h(\tilde{\beta}_j^{(*)})$.

Next, we want to show that $\beta_j^{**} = T(\beta_{-j}^*, \beta_j^*)$. We focus on the jth coordinates for arbitrary $j \in \{1, 2, .., p\}$. Without loss of generality, we assume $z_j^* \geq 0$. The result from the negative side can be obtained by symmetry. We consider the problem in three cases:

Case 1. $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j^{(*)} < h(\tilde{\beta}_j^{(*)})$. By convergence of the subsequence $\boldsymbol{\beta}^{n_i}$, there exists sufficient large n' such that for any $i > n'$, we have $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j^{(n_i)} < h(\tilde{\beta}_j^{(n_i)})$. In this case, $\beta_j^{(n_i+1)} \equiv 0$. We have $\beta_j^{**} = T(\beta_{-j}^*, \beta_j^*) = \beta_j^*$.

Case 2. $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j^{(*)} \geq h(\tilde{\beta}_j^{(*)})$. By convergence of the subsequence $\boldsymbol{\beta}^{n_i}$, there exists sufficiently large n' such that for any $i > n'$, we have $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j^{(n_i)} \geq h(\tilde{\beta}_j^{(n_i)})$. By step 2 and step 3 of the proof of theorem 5.1, there exists local minimizer $\beta_j'' \in (\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j]$ and β_j'' is the unique solution of the equation

$$f_2(\beta_{-j}, \beta_j) = \beta_j - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j + (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{\beta_j + C_2 \beta_j^{1-\frac{1}{2\gamma}}} = 0.$$

From the step 1 of the proof of theorem 5.1, we also see that $\frac{\partial f_2}{\partial \beta_j} = \psi^{'}(\beta_j) > 0$ for any $\beta_j \in (\tilde{\beta}_j, (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j]$, by implicit function theorem, there exists continuous and differential function $g(\beta_{-j}) = \beta_j^{''}$ for $z_j = X_j^T (Y - X\beta_{-j}) \in ((\boldsymbol{X}_j^T \boldsymbol{X}_j) h(\tilde{\beta}_j^{n_i}), +\infty)$. In addition, $g(\cdot)$ is one-to-one map. Hence, if we have $\boldsymbol{\beta}_{-j}^{n_i} \to \beta_{-j}^*$ and $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} z_j^{(*)} \geq h(\tilde{\beta}_j^{(*)})$, this implies $g(\boldsymbol{\beta}_{-j}^{n_i}) = \beta_j^{(n_i)''} \to g(\beta_{-j}^*) = \beta_j^{*''}$. By continuity of $\delta_{-j}(\cdot)$, we also have

$$\delta_{-j}(g(\boldsymbol{\beta}_{-j}^{n_i})) = \delta(\beta_j^{(n_i)''}) \to \delta_{-j}(g(\beta_{-j}^*)) = \delta(\beta_j^{*''}).$$

Now, we can further consider three sub cases:

a. $\delta_{-j}(\beta_j^{*''}) < 0$. There exists a sufficiently large n'' such that for any $i > n'' > n'$, we have $\delta_{-j}(\beta_j^{(n_i)''}) < 0$. Then $\beta_j^{(n_i+1)} = T(\beta_{-j}^{n_i}, \beta_j^{n_i}) = \text{sign}(z_j^{(n_i)}) \beta_j^{(n_i)''}$. Taking the limit on both sides, $\beta_j^{**} = \text{sign}(z_j^*) \beta_j^{*''} = T(\beta_{-j}^*, \beta_j^*)$.

b. $\delta_{-j}(\beta_j^{*''}) > 0$. There exists a sufficiently large n'' such that for any $i > n'' > n'$, we have $\delta_{-j}(\beta_j^{(n_i)''}) > 0$. Then $\beta_j^{(n_i+1)} = T(\beta_{-j}^{n_i}, \beta_j^{n_i}) = \beta_j^{n_i} = 0$. We have $\beta_j^{**} = T(\beta_{-j}^*, \beta_j^*) = 0$.

c. $\delta_{-j}(\beta_j^{*''}) = 0$. There exists a sufficiently large n'' such that for any $i > n'' > n'$,

$\delta_{-j}(\beta_j^{(n_i)''})$ will approach zero from only one side. Suppose this is not true, then $\beta_j^{(n_i+1)} \to \left\{0, \text{sign}(z_j^*)\beta_j^{*''}\right\}$. In other words, $\beta_j^{(n_i+1)}$ will converge to a limit set rather than a limit point which contradicts the assumption in the lemma.

If $\delta_{-j}(\beta_j^{(n_i)''})$ approach zero from the right side. Then,

$$\beta_j^{(n_i+1)} = T(\beta_{-j}^{n_i}, \beta_j^{n_i}) = \text{sign}(z_j^{(n_i)})\tilde{\beta}_j^{(n_i)} I(\beta_j^{(n_i)} \neq 0) = 0 = \beta_j^{n_i}.$$

Therefore we have, $\beta_j^{**} = \beta_j^* = 0$, which also implies $\beta_j^{**} = T(\beta_{-j}^*, \beta_j^*) = 0$.

If $\delta_{-j}(\beta_j^{(n_i)''})$ approach zero from the left side. Then, $\beta_j^{(n_i+1)} = T(\beta_{-j}^{n_i}, \beta_j^{n_i}) = \text{sign}(z_j^{(n_i)})\beta_j^{(n_i)''} \neq 0$ and $\beta_j^{(n_i)} = T(\beta_{-j}^{n_i-1}, \beta_j^{n_i-1}) = \text{sign}(z_j^{(n_i-1)})\beta_j^{(n_i-1)''} \neq 0$. This implies that $\beta_j^* \neq 0$ and $\beta_j^{**} \neq 0$. By the rule of $T(\cdot)$,

$$T(\beta_{-j}^*, \beta_j^*) = \text{sign}(z_j^*)\beta_j^{*''} I(\beta_j^* \neq 0) = \text{sign}(z_j^*)\beta_j^{*''}.$$

Taking the limit of $\beta_j^{(n_i+1)} = \text{sign}(z_j^{(n_i)})\beta_j^{(n_i)''}$, we have $\beta_j^{**} = \text{sign}(z_j^*)\beta_j^{*''}$.Therefore $\beta_j^{**} = T(\beta_{-j}^*, \beta_j^*)$. Finally, we use Lemma 6 to finish the proof. By continuity of descent function, we have

$$\triangle(\beta_j^{(n_i)}, \beta_j^{(n_i+1)}) \to \triangle(\beta_j^*, \beta_j^{**})$$

since Theorem 5.1 guarantees the non-increasing of the loss function, $\triangle(\beta_j^{(n_i)}, \beta_j^{(n_i+1)})$ is decreasing and bounded below by zero. Therefore,

$$\triangle(\beta_j^{(n_i)}, \beta_j^{(n_i+1)}) \to \triangle(\beta_j^*, \beta_j^{**}) = 0.$$

We have already shown that $\beta_j^{**} = T(z_j^*, \beta_j^*)$, then by Lemma 6, $\triangle(\beta_j^*, \beta_j^{**}) = 0$ if and only if $\beta_j^* = \beta_j^{**}$. □

Proof of Theorem 5.4

Proof.

$$\begin{aligned} L\left(\beta^\infty + \alpha e\right) - L\left(\beta^\infty\right) =& \frac{1}{2}\alpha^2 \|Xe\|_2^2 - \alpha e^T X^T \left(Y - X\beta^\infty\right) \\ &+ (2^\gamma P + 0.5)\left[\log\left(\sum_{j\in\hat{S}} \left|\beta_j^\infty + \alpha e_j\right|^{\frac{1}{2\gamma}} + \sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2\gamma}} + 1/b\right)\right. \\ &\left. - \log\left(\sum_{j\in\hat{S}} \left|\beta_j^\infty\right|^{\frac{1}{2\gamma}} + 1/b\right)\right] \end{aligned}$$

We decompose the perturbation e into two disjoint set: $e = e_{\hat{S}} + e_{\hat{S}^c}$ where $e_{\hat{S}} = eI(j \in \hat{S})$ and $e_{\hat{S}^c} = eI(j \in \hat{S}^c)$. Then, we have

$$\begin{aligned}
\|Xe\|_2^2 &= \|Xe_{\hat{S}} + Xe_{\hat{S}^c}\|_2^2 \\
&\geq \|Xe_{\hat{S}}\|_2^2 - 2\|X^TX\|\|e_{\hat{S}}\|_2\|e_{\hat{S}^c}\|_2 + \|Xe_{\hat{S}^c}\|_2^2 \\
&\geq \|Xe_{\hat{S}}\|_2^2 - 2\|X^TX\|\|e_{\hat{S}^c}\|_2 + \|Xe_{\hat{S}^c}\|_2^2 \\
&\geq \|Xe_{\hat{S}}\|_2^2 - 2\|X^TX\|\|e_{\hat{S}^c}\|_1 + \|Xe_{\hat{S}^c}\|_2^2 \\
&\geq \|Xe_{\hat{S}}\|_2^2 - 2\|X^TX\|\|e_{\hat{S}^c}\|_1 \\
&\geq nv_1\|e_{\hat{S}}\|_2^2 - 2\|X^TX\|\|e_{\hat{S}^c}\|_1
\end{aligned}$$

where the last inequality is followed by assumption A4. By mean value theorem, there exists $t \in (0,1)$ such that

$$\log(C+x) = \log(C) + \frac{x}{C+tx}.$$

Plug in $C = \sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + 1/b$ and $x = \sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}}$, we have

$$\begin{aligned}
&\log\left(\sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + \sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}} + 1/b\right) - \log\left(\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 1/b\right) \\
=&\log\left(\sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + 1/b\right) - \log\left(\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 1/b\right) + \frac{\sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}}}{\sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + t\sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}} + 1} \\
\geq&\log\left(\sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + 1/b\right) - \log\left(\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 1/b\right) + \frac{\sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}}}{2\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 2/b}
\end{aligned}$$

where the last inequality holds for α sufficiently small. Combining all the upper bounds we obtained above, we have

$$\frac{L(\beta^\infty + \alpha e) - L(\beta^\infty)}{\alpha} \geq \psi_1(\alpha) + \psi_2(\alpha)$$

where

$$\begin{aligned}
\psi_1(\alpha) =& -\alpha\|X^TX\|\|e_{\hat{S}^c}\|_1 - e_{\hat{S}^c}^T X_{\hat{S}^c}^T(Y - X\beta^\infty) + \frac{1}{\alpha}\frac{(2^\gamma P + 0.5)\sum_{j\in\hat{S}} |\alpha e_j|^{\frac{1}{2^\gamma}}}{2\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 2/b} \\
\psi_2(\alpha) =& \frac{nv_1\alpha}{2}\|e_{\hat{S}}\|_2^2 - e_{\hat{S}}^T X_{\hat{S}}^T(Y - X\beta^\infty) + \\
&\frac{2^\gamma P + 0.5}{\alpha}\left[\log\left(\sum_{j\in\hat{S}} |\beta_j^\infty + \alpha e_j|^{\frac{1}{2^\gamma}} + 1/b\right) - \log\left(\sum_{j\in\hat{S}} |\beta_j^\infty|^{\frac{1}{2^\gamma}} + 1/b\right)\right].
\end{aligned}$$

Now we treat each term separately,

$$\begin{aligned}\psi_1(\alpha) &\geq -\alpha\|X^TX\|\sum_{j\in\hat{S}^c}|e_j| - \sum_{j\in\hat{S}^c}|e_j||z_j^\infty| + \frac{1}{\alpha}\frac{(2^\gamma P+0.5)\sum_{j\in\hat{S}}|\alpha e_j|^{\frac{1}{2^\gamma}}}{2\sum_{j\in\hat{S}}|\beta_j^\infty|^{\frac{1}{2^\gamma}}+2/b}\\ &\geq -\sum_{j\in\hat{S}^c}(\|X^TX\|+h_j^\infty)|e_j| + \frac{1}{\alpha}\frac{(2^\gamma P+0.5)\sum_{j\in\hat{S}}|\alpha e_j|^{\frac{1}{2^\gamma}}}{2\sum_{j\in\hat{S}}|\beta_j^\infty|^{\frac{1}{2^\gamma}}+2/b}\\ &> 0 \quad \text{if and only if} \quad \alpha < \min_{j\in\hat{S}^c}\left\{\frac{1}{|e_j|}\left(\frac{\lambda}{\|X^TX\|+h_j^\infty}\right)^{\frac{1}{1-\frac{1}{2^\gamma}}}\right\}\end{aligned}$$

where $z_j^\infty = X_j^T(Y - X\beta^\infty)$, $\lambda = \frac{(2^\gamma P+0.5)}{2\sum_{j\in\hat{S}}|\beta_j^\infty|^{\frac{1}{2^\gamma}}+2/b}$ and the second inequality follows by Theorem 5.1. the KKT condition. Again, by KKT condition in Theorem 5.1, we have

$$\lim_{\alpha\to 0}\psi_2(\alpha) = -e_{\hat{S}}^T X_{\hat{S}}^T(Y - X\beta^\infty) + \frac{P+\frac{1}{2^{\gamma+1}}}{\sum_{j\in\hat{S}}|\beta_j^\infty|^{\frac{1}{2^\gamma}}+1/b}\sum_{j\in\hat{S}}\frac{e_j}{|\beta_j^\infty|^{1-\frac{1}{2^\gamma}}} = 0.$$

□

B.3 Oracle properties

Proof of Theorem 5.5

We only show the case (2): $\lim_{n\to\infty}\frac{p_n}{ns_0} = \infty$, $b = O\left(\frac{\log p_n}{p_n}\right)$ and assumption A9 holds. The argument for case (1) is exactly the same, so we will omit the details.

Proof. Let $\hat{\beta}^T = \left([(X_{s_0}^TX_{s_0})^{-1}X_{s_0}^Ty]^T, 0^T\right)$ be the oracle estimator. We want to show that, with probability tending to 1, the oracle estimator $\hat{\beta}$ is a strictly local minimizer of $L(\cdot)$. By theorem 5.1, it suffices to show that, with probability tending to 1, $\hat{\beta}$ satisfies the KKT conditions:

$$\begin{aligned}&(X_j^TX_j)^{-1}|z_j| < h(\tilde{\beta}_j) \quad \text{for} \quad j\in S_0^c\\ &(X_j^TX_j)^{-1}|z_j| > h(\tilde{\beta}_j) \quad \text{for} \quad j\in S_0\\ &\beta_j = \left[(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}|z_j| - (\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1}\frac{C_1}{|\beta_j|+C_2|\beta_j|^{1-\frac{1}{2^\gamma}}}\right]\text{sign}(z_j) \quad \text{for} \quad j\in S_0\\ &L_{-j}(\beta_j) < L_{-j}(0) \quad \text{for} \quad j\in S_0\end{aligned}$$

plug in $\hat{\beta}^T = \left([(X_{s_0}^T X_{s_0})^{-1} X_{s_0}^T y]^T, 0^T\right)$ into the KKT condition, we see that

$$(X_j^T X_j)^{-1}|z_j| = \begin{cases} \left|\beta_{0,j} + \sigma_0[(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j\right| & \text{if } j \in S_0 \\ \left|\frac{\sigma_0 X_j^T \epsilon}{n} - \sigma_0 \frac{X_j^T X_{S_0}}{n}(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon\right| & \text{if } j \in S_0^c \end{cases} \tag{B.6}$$

by mean value theorem, we have

$$\begin{aligned} \sum_{j \in S_0} |\hat{\beta}_j|^{\frac{1}{2^\gamma}} &= \sum_{j \in S_0} |\beta_{0,j} + \sigma_0[(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j|^{\frac{1}{2^\gamma}} \\ &= \sum_{j \in S_0} |\beta_{0,j}|^{\frac{1}{2^\gamma}} + \frac{\sigma_0}{2^\gamma} \sum_{j \in S_0} \frac{[(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j}{|\beta_{0,j} + \sigma_0 t_j [(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j|^{1 - \frac{1}{2^\gamma}}} \end{aligned} \tag{B.7}$$

from some $t = (t_1, ..., t_p)$ such that $t_j \in (0, 1)$. Consider the events

$$\begin{aligned} \mathcal{A}_1 &= \left\{ \left\| \frac{X_{s_0}^T \epsilon}{n} \right\|_\infty \leq 2\sqrt{\frac{2 \log s_0}{n}} \right\} \\ \mathcal{A}_2 &= \left\{ \left\| \frac{X_{s_0^c}^T \epsilon}{n} \right\|_\infty \leq 2\sqrt{\frac{2 \log p_n}{n}} \right\}. \end{aligned}$$

By standard Gaussian tail bounds [Wainwright, 2019], we have

$$\begin{aligned} \lim_{n \to \infty} P\left(\left\| \frac{X_{s_0}^T \epsilon}{n} \right\|_\infty \geq 2\sqrt{\frac{2 \log s_n}{n}} \right) &\leq 2 \lim_{n \to \infty} e^{-\log s_n} = 0 \\ \lim_{n \to \infty} P\left(\left\| \frac{X_{s_0^c}^T \epsilon}{n} \right\|_\infty \geq 2\sqrt{\frac{2 \log(p_n - s_n)}{n}} \right) &\leq 2 \lim_{n \to \infty} e^{-\log(p_n - s_n)} = 0 \end{aligned}.$$

By assumption A4, $X_{S_0}^T X_{S_0}$ is invertible, then under event $\mathcal{A}_1$, we have

$$\begin{aligned} \|(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon\|_\infty &\leq \|(X_{S_0}^T X_{S_0})^{-1}\|_\infty \|X_{S_0}^T \epsilon\|_\infty \\ &\leq \sqrt{s_0} \|(X_{s_0}^T X_{S_0})^{-1}\|_2 \|X_{S_0}^T \epsilon\|_\infty \\ &\leq \sqrt{\frac{s_0}{v_1}} \left\| \frac{X_{s_0}^T \epsilon}{n} \right\|_\infty \\ &\leq 2\sqrt{\frac{2 s_0 \log s_0}{n v_1}} \\ &\leq 2\sqrt{\frac{2 \log s_0}{v_1 \log p_n}} \end{aligned}$$

where the last inequality is followed by assumption A2. In addition, for n sufficient large, under event $\mathcal{A}_1$ and assumption A5, we have $\tilde{\beta}_j < |\hat{\beta}_j| = (X_j^T X_j)^{-1}|z_j| < 2E$, which

implies that $\tilde{\beta}_j^{\frac{1}{2^\gamma}} + C_2 = O(\sum_{j\in S_0} |\hat{\beta}_j|^{\frac{1}{2^\gamma}} + b^{-1})$. By Lemma 5.2, we have

$$h(\tilde{\beta}_j) = O\left(\left(\frac{p + \frac{1}{2^{\gamma+1}}}{n\tilde{\beta}_j^{\frac{1}{2^\gamma}} + nC_2}\right)^{\frac{1}{2-\frac{1}{2^\gamma}}}\right) = O\left(\left(\frac{p}{ns_0 + np_n/\log p_n}\right)^{\frac{1}{2-\frac{1}{2^\gamma}}}\right) = O\left(\left(\frac{\log p_n}{n}\right)^{\frac{1}{2-\frac{1}{2^\gamma}}}\right)$$

where $\sum_{j\in S_0} |\hat{\beta}_j|^{\frac{1}{2^\gamma}} = O(\sum_{j\in S_0} |\beta_{0,j}|^{\frac{1}{2^\gamma}}) = O(s_0)$ by Equation B.7 and assumption A6. Define the event

$$\mathcal{A} = \mathcal{A}_1 \cap \mathcal{A}_2 \cap \left\{ \left\| \frac{X_{S_0^c}^T X_{S_0}}{n} \right\|_{2,\infty} \leq c \right\}$$

then by assumption A7, $\lim_{n\to\infty} P(\mathcal{A}) = 1$. Therefore, it suffices to show that for sufficiently large n under the event $\mathcal{A}$ the KKT condition will statisfy. For $j \in S_0^c$ and n sufficient large, under event $\mathcal{A}$, we have

$$\begin{aligned}
&\max_{j\in S_0^c} \left| \frac{\sigma_0 X_j^T \epsilon}{n} - \sigma_0 \frac{X_j^T X_{S_0}}{n} (X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon \right| \\
&\leq \sigma_0 \left\| \frac{X_{s_0^c}^T \epsilon}{n} \right\|_\infty + \sigma_0 \left\| \frac{X_{S_0^c}^T X_{S_0}}{n} \right\|_{2,\infty} \left\| (X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon \right\|_2 \\
&\leq 2\sigma_0 \sqrt{\frac{2\log p_n}{n}} + \sigma_0 c \sqrt{\mathrm{Trace}((X_{S_0}^T X_{S_0})^{-1})} \\
&\leq 2\sigma_0 \sqrt{\frac{2\log p_n}{n}} + \sigma_0 c \sqrt{\frac{s_0}{nv_1}} \\
&\leq M\sigma_0 \sqrt{\frac{\min(\log p_n, s_0)}{n}} \\
&\leq M\sigma_0 \sqrt{\frac{\log p_n}{n}} \\
&\leq M\sigma_0 \left(\frac{\log p_n}{n}\right)^{\frac{1}{2-\frac{1}{2^\gamma}}} \\
&= O(h(\tilde{\beta}_j)).
\end{aligned}$$

For $j \in S_0$ and n sufficient large, under event $\mathcal{A}$, we have

$$\begin{aligned}
&\min_{j\in S_0} \left|\beta_{0,j} + \sigma_0[(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j\right| \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - \sigma_0 \|(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon\|_\infty \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - \sigma_0 \|(X_{S_0}^T X_{S_0})^{-1}\|_\infty \|X_{S_0}^T \epsilon\|_\infty \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - \sigma_0 \sqrt{s_0} \|(X_{s_0}^T X_{S_0})^{-1}\|_2 \|X_{S_0}^T \epsilon\|_\infty \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - \sigma_0 \sqrt{\frac{s_0}{v_1}} \left\| \frac{X_{s_0}^T \epsilon}{n} \right\|_\infty \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - 2\sigma_0 \sqrt{\frac{2 s_0 \log s_0}{n v_1}} \\
&\geq \min_{j\in S_0} |\beta_{0,j}| - 2\sigma_0 \sqrt{\frac{2 \log s_0}{v_1 \log p_n}} \\
&> \sigma_0 \left(\frac{\log p_n}{n}\right)^{\frac{1}{2-\frac{1}{2^\gamma}}} \\
&= O(h(\tilde{\beta}_j))
\end{aligned}$$

where the last inequality is followed by assumption A8. Next, we want to check that, for $j \in S_0$ and n sufficient large, under event $\mathcal{A}$, we have

$$\hat{\beta}_j = \left[(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| - (\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} \frac{C_1}{|\hat{\beta}_j| + \hat{C}_2 |\hat{\beta}_j|^{1-\frac{1}{2^\gamma}}} \right] \operatorname{sign}(z_j) \tag{B.8}$$

where $\hat{C}_2 = \sum_{j\in S_0} |\hat{\beta}_j|^{\frac{1}{2^\gamma}} + 1/b$. Since $(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1} |z_j| = \left|\beta_{0,j} + \sigma_0[(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon]_j\right| = |\hat{\beta}_j|$, it suffices to show

$$\lim_{n\to\infty} \frac{C_1}{n|\hat{\beta}_j| + \hat{C}_2 n |\hat{\beta}_j|^{1-\frac{1}{2^\gamma}}} = 0.$$

We see that under event $\mathcal{A}_1$, $\lim_{n\to\infty} \hat{\beta}_j = \beta_{0,j}$, thus it suffices to show

$$\lim_{n\to\infty} \frac{p_n}{n|\beta_{0,j}| + C_{0,2} n |\beta_{0,j}|^{1-\frac{1}{2^\gamma}}} = 0$$

where $C_{0,2} = \sum_{j\in S_0} |\beta_{0,j}|^{\frac{1}{2^\gamma}} + 1/b$. By assumption A6, $\lim_{n\to\infty} \frac{p_n}{n s_0} = \infty$ and $b = O\left(\frac{\log p_n}{p_n}\right)$, we have:

$$\frac{p_n}{n|\beta_{0,j}| + C_{0,2} n |\beta_{0,j}|^{1-\frac{1}{2^\gamma}}} \leq \frac{p_n}{n(\min_{j\in S_0} |\beta_{0,j}| s_0 + c p_n / \log p_n)} = O\left(\frac{\log p_n}{n}\right) \to 0 \quad \text{if} \quad n \to \infty$$

where c is some positive constant. Finally, we want to check that, for $j \in S_0$ and n sufficient large, under event $\mathcal{A}$, we have $L_{-j}(\beta_j) < L_{-j}(0)$.

$$\delta_{-j}(|\hat{\beta}_j|) = L_{-j}(\hat{\beta}_j) - L_{-j}(0) = \frac{(X_j^T X_j)}{2}\hat{\beta}_j{}^2 - |z_j||\hat{\beta}_j| + (2^\gamma p + 0.5)\log\left(1 + \frac{|\hat{\beta}_j|^{\frac{1}{2^\gamma}}}{C_2}\right). \quad \text{(B.9)}$$

Since $(X_j^T X_j)^{-1}|z_j| > h(\tilde{\beta}_j)$ for $j \in S_0$, plug in (8.3) into (8.4), then we have

$$\delta_{-j}(|\hat{\beta}_j|) = -\frac{(X_j^T X_j)}{2}\hat{\beta}_j^2 - \frac{C_1}{1 + C_2\hat{\beta}_j^{-\frac{1}{2^\gamma}}} + (2^\gamma p + 0.5)\log\left(1 + \frac{\hat{\beta}_j^{\frac{1}{2^\gamma}}}{C_2}\right).$$

We see that to show $\delta_{-j}(|\hat{\beta}_j|) < 0$, for n sufficient large and under event $\mathcal{A}$, that is equivalent to show

$$\frac{(2^\gamma p + 0.5)}{n}\log\left(1 + \frac{\hat{\beta}_j^{\frac{1}{2^\gamma}}}{C_2}\right) < \frac{1}{2}\hat{\beta}_j^2$$

By assumptions A5,A6 and $b \propto \frac{\log p_n}{p_n}$, taking $n \to \infty$, for the left hand side, we have

$$\frac{(2^\gamma p + 0.5)}{n}\log\left(1 + \frac{\hat{\beta}_j^{\frac{1}{2^\gamma}}}{C_2}\right) = O\left(\frac{\log p_n}{n}\right) \to 0,$$

while for the right hand side, $\frac{1}{2}\hat{\beta}_j^2 = O(1)$. The result follows immediately. Now we show the asymptotic normality of $\hat{\sigma}^2 - \sigma_0^2$. Under the event $\mathcal{A}$, as $n \to \infty$

$$\hat{\sigma}^2 - \sigma_0^2 \to \frac{\sigma_0^2}{n - s_0}(I_n - X_{S_0}(X_{S_0}^T X_{S_0})^{-1}X_{S_0}^T) - \sigma_0^2.$$

Hence, we have

$$\frac{\hat{\sigma}^2 - \sigma_0^2}{\sqrt{n}} \xrightarrow{\mathcal{D}} N(0, \sigma_0^4 E[\epsilon^4] - \sigma_0^4).$$

□

B.4 Forward and backward variable screening

In this section, we provide details of the proposed backward and forward variable screening algorithms. The backward screening algorithm does not use cross-validation, and is recommended for when the sample size is moderate to large. For small sample size, a forward variable screening algorithm cross-validation is recommended.

Consider a sequence of L parameters $g = \{g_1 < g_2 < ... < g_L\}$ with $b_l = g_l * \frac{\log p}{p}$, the CD optimization begins with hyper-parameter $b_1 = g_1 \frac{\log p}{p}$. In default, we set $g = \{1, 2, ..., 100\}$. The first output $\hat{\beta}^{(1)}$ is then used as a warm start for the 2nd iteration of the CD optimisation with $b_2 = g_2 \frac{\log p}{p}$. The algorithm continues in this fashion and we can visualise the entire solution path of $\boldsymbol{\beta}$ with respect to the hyper-parameter b. Since the function $b \to T_b(\cdot)$ is continuous, this assures that a small perturbation in b will lead to only small changes in the solution for $\boldsymbol{\beta}$, this suggests that using the solution of the previous optimization for the next one is a reasonable strategy. The backward screening algorithm is provided in Algorithm 8.

Algorithm 8 Backward variable screening

Set K, γ, $b = \{b_1 < b_2 < ... < b_L\}$, $\boldsymbol{\beta}^{(0)} = 0$

for $i \leftarrow 1 \cdots L$ **do**

$\boldsymbol{\beta}^{(i)} = \text{CD algorithm}(\boldsymbol{Y}, \boldsymbol{X}, b = b_i, \gamma, \boldsymbol{\beta}_{initial} = \boldsymbol{\beta}^{(i-1)})$

$\sigma^{2(i)} = \frac{\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}^{(i)}\|_2^2}{n - s^{(i)}}$

end for

Return $\boldsymbol{\beta} = (\boldsymbol{\beta}^{(1)}, ..., \boldsymbol{\beta}^{(L)}), \sigma^2 = (\sigma^{2(1)}, ..., \sigma^{2(L)})$

Roughly speaking, the higher the value of b, the more the estimated $\hat{\boldsymbol{\beta}}$ are set to zero. Hence we use the term backward variable screening. The algorithm starts with a big model and gradually removes insignificant coefficients. When the sample size is relative large, the solution path stabilizes very quickly with increasing values of b (See Figure B.1). In fact, we found that, when the sample size is large, we can fit the model directly by setting $\frac{1}{b} = 0.0001$, but we still recommend backward variable screening because we don't know how large the sample size needs to be. When the sample size is small, the solution path stabilizes slowly, it may ultimately threshold everything to zero, requiring cross-validation to identify the best solution. Figure B.2 shows the solutions paths from a simulated example with $n = 100, p = 1000, s_0 = 100$ and $\sigma^2 = 3$. We can see that the model quickly shrinks all coefficients to zero. In such cases, we found that a forward screening strategy, together with cross-validation, works better. Although cross-validation is computationally intensive, it was found by [Reid et al., 2016] that it guarantees a robust

performance for small sample size and less sparse models. Our numerical experiments are consistent with the numerical results in their paper.

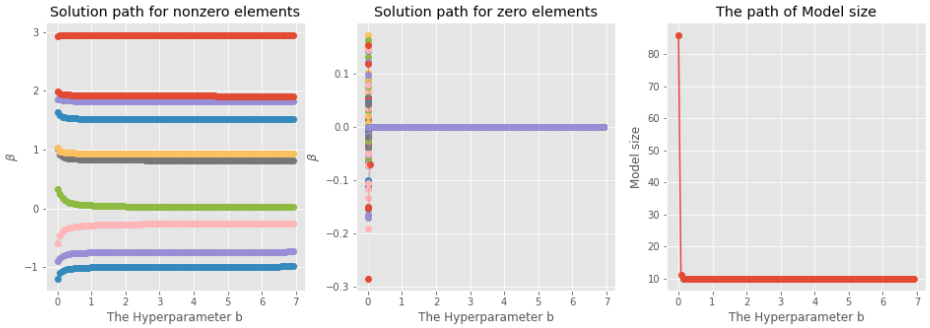


Figure B.1: Solution paths of $\boldsymbol{\beta}$, as b is varied. Nonzero elements (left); zero elements (middle) and $\hat{s}$ (right). The data is simulated using $n = 500$, $p = 1000$, $s_0 = 10$ and $\sigma_0^2 = 3$.

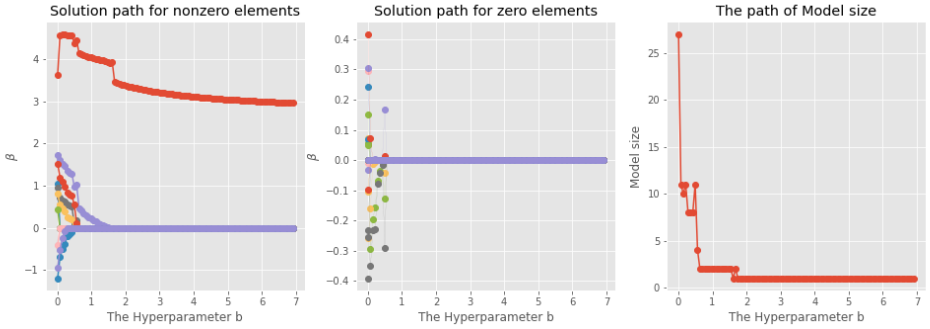


Figure B.2: Solution paths of $\boldsymbol{\beta}$, as b is varied. Nonzero elements (left); zero elements (middle) and $\hat{s}$ (right). The data is simulated using $n = 100$, $p = 1000$, $s_0 = 10$ and $\sigma_0^2 = 3$.

Here we develop a forward variable screening algorithm with K-fold cross-validation. We first reparametrize the hyper-parameter b to $t = \frac{1}{b}$. Then we consider a sequence of L increasing parameters $T = \{t_1 < t_2 < ... < t_L\}$ with $t_1 = 0$ and $t_L = \frac{p}{\log p}$. In default, 100 evenly spaced points have been used. We use the same re-initialization strategy as backward variable screening. The optimization output from the previous value of b is used as a warm start for optimization at current b. For each b, we obtain K solutions of $\boldsymbol{\beta}$ from the K-fold cross-validation. Once we have the optimal value of b, we fit the model to the

entire data set with forward strategies. Empirically, we found the forward strategies did better than the backward strategy when the sample size is small and the model is less sparse.

The sure independence screening by [Fan and Lv, 2008] can be viewed as a backward strategies. They greatly simplified the original ultra-high dimensional problem into a low-dimensional one and then fitted the model with LASSO or SCAD penalty. In another direction, [Wang, 2009] investigated the forward regression in ultra-high dimensional problems. Our forward variable screening strategies can be viewed as a smooth version of forward regression. [Wang, 2009] used the BIC criterion to select the optimal model size. They show screening consistency for their approach. Here, we determine the optimal model by cross-validation. For infinite sample size, the performance of our forward and backward screening algorithms are guaranteed by the oracle properties of the non-separable bridge penalty. We provide the description of forward screening algorithm in Algorithm 9.

Algorithm 9 Forward variable screening with cross-validation

Set $K, \gamma, T = \{t_1 < t_2 < ... < t_L\}$

Split the data into K-fold with $\boldsymbol{Y}_{validation} = (\boldsymbol{Y}_{-1}, ..., \boldsymbol{Y}_{-K})$ and $\boldsymbol{X}_{validation} = (\boldsymbol{X}_{-1}, ..., \boldsymbol{X}_{-K})$

for $k \leftarrow 1 \cdots K$ **do**

$\boldsymbol{\beta}^{(k,0)} = 0$

for $i \leftarrow 1 \cdots L$ **do**

$\boldsymbol{\beta}^{(k,i)} = \text{CD algorithm}(\boldsymbol{Y}_{-k}, \boldsymbol{X}_{-k}, b = \frac{1}{t_i}, \gamma, \boldsymbol{\beta}_{initial} = \boldsymbol{\beta}^{(k,i-1)})$

end for

for $i \leftarrow 1 \cdots L$ **do**

$\text{Err}^{(i)}_{valid} = \frac{1}{N} \sum_{k=1}^{K} \|\boldsymbol{Y}_{-k} - \boldsymbol{X}_{-k}\boldsymbol{\beta}^{(k,i)}\|_2^2$

end for

end for

$m = \textbf{argmin}_i \ \text{Err}^{(i)}_{valid}$

$\hat{\boldsymbol{\beta}} = 0$

for $i \leftarrow 1 \cdots m$ **do**

$\hat{\boldsymbol{\beta}} = \text{CD algorithm}(\boldsymbol{Y}, \boldsymbol{X}, b = \frac{1}{t_i}, \gamma, \boldsymbol{\beta}_{initial} = \hat{\boldsymbol{\beta}})$

$\hat{\sigma}^2 = \frac{\|\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\|_2^2}{n - \hat{s}}$

end for

Return $\hat{\beta}$, $\hat{\sigma}^2$

Appendix C

C.1 The divergence of the variance for adaptive learning rate

The main issue with using the posterior variance or posterior standard deviation as adaptive learning is that it has an undesirably large variance in the early stage of model training. The natural gradient descent, in mean-field Gaussian variational inference updates the posterior precision with momentum [Zhang et al., 2018, Osawa et al., 2019]:

$$\frac{1}{\sigma_t^2} = \frac{1-\lambda\gamma}{\sigma_{t-1}^2} + \gamma(-\mathbb{E}_{N_p(\mu,\Sigma)}\left[\nabla_{\mathbf{w}}^2 \log p(\mathcal{D} \mid \mathbf{w})\right] + \frac{\delta}{N}) \approx \frac{1-\lambda\gamma}{\sigma_{t-1}^2} + \gamma([g_t \odot g_t] + \frac{\delta}{N}) \quad \text{(C.1)}$$

By using $-\mathbb{E}_{N_p(\mu,\Sigma)}\left[\nabla_{\mathbf{w}}^2 \log p(\mathcal{D} \mid \mathbf{w})\right] \approx \frac{1}{S}\sum_{i=1}^{S} g_i \odot \frac{\epsilon_i}{\sigma}$ with $S = 1$, there exists another approximation [Khan et al., 2018]:

$$\frac{1}{\sigma_t^2} \approx \frac{1-\lambda\gamma}{\sigma_{t-1}^2} + \gamma(g_t \odot \frac{\epsilon}{\sigma_{t-1}} + \frac{\delta}{N}) \quad \text{(C.2)}$$

where $g_t = -\nabla_{\mathbf{w}} \log p(\mathcal{D} \mid \mathbf{w}) = \nabla_w l(w)$ and $\epsilon \sim N(0, I_p)$.

Remark: One potential problem for using equation (C.2) is that, if the Hessian matrix of the neural network is not guaranteed to be positive definite, and if the damping parameter is not large enough, it is possible to have negative adaptive learning rate when we use equation (C.2) as an approximation.

For the special case, when $t = 1$, similar to Adam [Kingma and Ba, 2014], if we don't use the initialization σ_0^2 as the past memory, then the first terms of both equations (C.1) and (C.2) do not exist. The upper bound of the posterior variance for the Monte Carlo estimator is

$$\mathbf{Var}(\sigma_1^2) = \mathbf{Var}\left[\frac{1}{-\mathbb{E}_{N_p(\mu,\Sigma)}\left[\nabla_{\mathbf{w}}^2 \log p(\mathcal{D} \mid \mathbf{w})\right] + \delta/N}\right] \leq \frac{1}{12\delta^2}.$$

This indicates that one way to control the variance of the estimator is to set the damping parameter non-negligibly large.

However, a large damping parameter induces a large bias in the adaptive learning rate and slows down the optimization process. From a Bayesian perspective, a large damping parameter means a normal prior with a small variance, which introduces an informative prior.

The next lemma provides a simple example to convince people that the variance of $\hat{\sigma}^2$ will diverge at the beginning of the training.

Lemma

Considering a ReLu neural network with a single hidden layer and binary cross-entropy loss, then if a normal initialization of the weight $w \sim N(0, \rho^2)$ has been used, the variance of the $\hat{\sigma}^2$ estimator based on equation (C.1) and (C.2) will diverge at the beginning of the training if $\delta \to 0$.

Proof

We denote $w_{k,j}^{(1)}$ as the weight connecting input x_j to k hidden unit sampled by $N(0, \rho)$. Then

$$\nabla_{w_{k,j}^{(1)}} l(w) = \sum_{l=1}^{N} \sigma_{Relu}^{'}(h_l^1) w_{1,k}^{(2)} \nabla_{h_l^{(2)}} l(w) x_{j,l}$$

where $\sigma_{Relu}^{'}(\cdot)$ is the derivative of Relu function, which is 0 or 1, $\nabla_{h_l^{(2)}} l(w) = -y_l(1 - \sigma_{sigmoid}(h_l^{(2)})) + (1 - y_l)\sigma_{sigmoid}(h_l^{(2)})$, which is bounded between $[-1, 1]$ and N is the batch size.

If we use equation (C.1) as estimator for $\hat{\sigma}^2$, then we have

$$\lim_{\delta \to 0} \hat{\sigma}_1^2 = \lim_{\delta \to 0} \frac{1}{\nabla_w l(w) \odot \nabla_w l(w) + \delta/N} \to \frac{1}{\nabla_w l(w) \odot \nabla_w l(w)} \tag{C.3}$$

If we use equation (C.2) as estimator for $\hat{\sigma}^2$, then we have

$$\lim_{\delta \to 0} \hat{\sigma}_1^2 = \lim_{\delta \to 0} \frac{1}{\frac{\epsilon \nabla_w l(w)}{\sigma_t} + \delta/N} \to \frac{\sigma_t}{\epsilon \nabla_w l(w)} = \frac{\rho}{\epsilon \nabla_w l(w)} \tag{C.4}$$

Note that both $\sigma^{'}_{Relu}(\cdot)$ and $\nabla_{h_l^{(2)}} l(w)$ are both upper bounded and lower bounded, therefore, for equation (C.3), it suffices to show that

$$\text{Var}\left[\frac{1}{(\sum_{l=1}^N w_{1,k}^{(2)} x_{j,l})^2}\right] = \frac{1}{(\sum_{l=1}^N x_{j,l})^2} \text{Var}\left[\frac{1}{w_{1,k}^{(2)}}\right] = \frac{1}{(\sum_{l=1}^N x_{j,l})^2} \frac{1}{\rho^4} \text{Var}\left[\left|\frac{1}{\epsilon^2}\right|\right] = \infty.$$

Since we have the inequality $\text{Var}(X) \geq \text{Var}(|X|)$, for equation (C.4), it suffices to show that

$$\text{Var}\left[\left|\frac{\rho}{\sum_{l=1}^N x_{j,l} \epsilon w_{1,k}^{(2)}}\right|\right] = \frac{1}{(\sum_{l=1}^N x_{j,l})^2} \text{Var}\left[\left|\frac{\rho}{\epsilon w_{1,k}^{(2)}}\right|\right] = \frac{1}{(\sum_{l=1}^N x_{j,l})^2} \text{Var}\left[\left|\frac{1}{\epsilon^2}\right|\right] = \infty.$$

Since $\frac{1}{(\epsilon)^2} \sim$ Scale-inv- $\mathcal{X}^2(1,1)$ has divergent variance, our proof is completed.

Remark: The similar argument can also apply to $\hat{\sigma}_1$ because we also have $\text{Var}\left[\left|\frac{1}{\epsilon}\right|\right] = \infty$. By assuming $g_1 \sim N(0, \rho^2)$, [Liu et al., 2019] shows the divergence of the adaptive learning rate for the Adam algorithm at $t = 1$.Here, we modify this assumption to a more realistic case $w \sim N(0, \rho^2)$.

C.2 Mirror descent

C.2.1 Bregman divergences and convex duality

Now we introduce the concept of convex conjugate functions. The convex conjugate function for a function G is defined to be:

$$H(\rho) := \sup_{\tau}\{\langle \tau, \rho\rangle - G(\tau)\}$$

If G is lower semi-continuous, G is the convex conjugate of H, implying a dual relationship between G and H. Further, if we assume G is strictly convex and twice differentiable, then so is H. Note also that if $g = \nabla G$ and $h = \nabla H$, $g = h^{-1}$.

Let $\rho = g(\tau)$, then the dual Bregman divergence induced by the strictly convex differentiable function H is

$$B_H\left(\rho, \rho'\right) = H(\rho) - H\left(\rho'\right) - \left\langle \nabla H\left(\rho'\right), \rho - \rho'\right\rangle$$

It is straightforward to show that the divergences between primal space and dual sparse are mutually reciprocal in the sense:

$B_H\left(\rho, \rho'\right) = B_G\left(h\left(\rho'\right), h(\rho)\right) = B_G\left(\tau', \tau\right)$ and $B_G\left(\tau, \tau'\right) = B_H\left(g\left(\tau'\right), g(\tau)\right) = B_H\left(\rho', \rho\right)$

The above property implies that if we can find a Bregman divergence such that its dual space is unconstrained, then we can perform optimization in dual space and finally move back to the original space.

For more details on Bregman divergences and convex duality, please refer to [Raskutti and Mukherjee, 2015] and [Amari and Cichocki, 2010].

C.2.2 Updating τ with Mirror descent

We have the following objective function

$$\underset{\tau\in(0,1)^p}{\operatorname{argmin}} \left\{ \langle \nabla_{\boldsymbol{\tau}} \mathcal{L}, \tau \rangle + \frac{1}{\eta} D_G(\tau, \tau_t) \right\}$$

Finding the minimum by differentiation yields the step:

$$g\left(\tau_{t+1}\right) = g\left(\tau_t\right) - \eta \nabla_{\boldsymbol{\tau}} \mathcal{L}$$

where $g(\tau_t) = \rho_t$. In addition, for Gaussian mean-field approximation, we have

$$\begin{aligned}
\nabla_{\boldsymbol{\tau}} \mathcal{L} = 2\alpha\sigma \nabla_{\boldsymbol{\sigma^2}} \mathcal{L} &= \alpha\sigma \mathbf{E}_{N_p(\boldsymbol{\mu},\boldsymbol{\sigma})} \left[\nabla^2_{\mathbf{w}} \log p(\mathcal{D} \mid \mathbf{w}) \right] + \alpha \mathbf{diag}(\sigma_j^{-1}) - \alpha\sigma\delta \\
&= \alpha \mathbf{E}_{N(0,1_p)} \left[\epsilon \odot \nabla_w l(w) \right] + \mathbf{diag}(\tau_j^{-1}) - \alpha^2 \tau \delta
\end{aligned}$$

More explicitly, we have

$$\begin{aligned}
\rho_{t+1} &= \rho_t + \eta \left(\frac{1}{\tau_t} - \alpha^2 \delta \tau_t \right) - \eta\alpha E_{N(0,1_p)} [\epsilon \odot \nabla_\tau l(w)] \\
&\approx \rho_t + \eta \left(\frac{1}{\tau_t} - \alpha^2 \delta \tau_t \right) - \eta\alpha\epsilon \odot g_{t+1}
\end{aligned}$$

C.2.3 Mirror descent vs Reparametrization trick

Now let's compare the difference between the Mirror descent and the reparametrization trick.

Since $\mathcal{L}(\tau) = \mathcal{L}(g^{-1}(\rho)) = \mathcal{L}(h(\rho))$, by applying the chain rule, we have

$$\nabla_\rho \mathcal{L}(h(\rho)) = \nabla_\rho h(\rho) \nabla_\tau \mathcal{L}$$

which implies that the natural gradient in dual space is equal to the gradient in original space, $\nabla_\tau \mathcal{L} = [\nabla_\rho h(\rho)]^{-1} \nabla_\rho \mathcal{L}(h(\rho))$.

Mirror descent thus first moves into dual (unconstrained) space, performs a natural gradient descent update there,

$$\rho_{t+1} = \rho_t - \eta \nabla_\tau \mathcal{L} = \rho_t - \eta [\nabla_\rho h(\rho)]^{-1} \nabla_\rho \mathcal{L}(h(\rho))$$

and then moves back, $\tau_{t+1} = g^{-1}(\rho_t - \eta\nabla_\tau\mathcal{L}) = h(\rho_t - \eta\nabla_\tau\mathcal{L})$

Reparametrization trick rewrites the problem in dual space and performs gradient descent there

$$\rho_{t+1} = \rho_t - \nabla_\rho\mathcal{L}(h(\rho)) = \rho_t - \nabla_\rho\mathcal{L}(h(\rho))$$

and also moves back finally $\tau_{t+1} = g^{-1}(\rho_t - \nabla_\rho\mathcal{L}(h(\rho)))$

The main advantage of mirror descent is that from a computation perspective, it is a first-order method using gradient information from primal space, but it is actually equivalent to performing natural gradient descent in dual space to explore the geometry structure [Raskutti and Mukherjee, 2015].

C.3 Connection to SGHMC

We now give a precise connection between the constrained Variational Adam algorithm and the Stochastic gradient Hamiltonian Monte Carlo (SGHMC). Based on the following two realistic assumptions:

- when the learning rate is small and $l_t \approx l_{t-1}$
- The update of the local parameter of posterior standard deviation τ is either frozen or has already converged

We can write:

$$\begin{aligned} W_t - W_{t-1} &= \mu_t - \mu_{t-1} + \alpha(\tau_t\epsilon - \tau_{t-1}\epsilon') \\ &= -D_\tau m_t l_t + \alpha\epsilon\sqrt{\tau_t^2 + \tau_{t-1}^2} \\ &\approx -D_\tau m_t l_t + \alpha\sqrt{2}D_\tau\epsilon \\ &= l_t D_\tau(-m_t + \frac{\alpha\sqrt{2}\epsilon}{l_t}) \end{aligned}$$

we set $dt = \Delta t = \sqrt{l_t}$, then, $dW_t = D_\tau r_t dt$.

Next, we define: $r_t = (-m_t + \frac{\sqrt{2}\alpha\epsilon}{l_t})\sqrt{l_t}$ and $r_{t-1} = (-m_{t-1} + \frac{\sqrt{2}\alpha\epsilon'}{l_{t-1}})\sqrt{l_{t-1}}$. Then

$$\begin{aligned}
r_t - r_{t-1} &\approx (-m_t + m_{t-1})\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon - \sqrt{2}kl_t^{1/4}\epsilon' \qquad (l_t \approx l_{t-1}) \\
&= (1-\beta_1)m_{t-1}\sqrt{l_t} - \beta_2\tilde{g}_t\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon - \sqrt{2}kl_t^{1/4}\epsilon' \\
&= (1-\beta_1)(\sqrt{2}kl_t^{1/4}\epsilon' - r_{t-1}) - \beta_2 g_t\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon - \sqrt{2}kl_t^{1/4}\epsilon' \\
&= -(1-\beta_1)r_{t-1} - \beta_2 g_t\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon - \sqrt{2}\beta_1 kl_t^{1/4}\epsilon' \\
&= -(1-\beta_1)r_{t-1} - \beta_2 g_t\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon\sqrt{1+\beta_1^2} \\
&= -hr_{t-1}\sqrt{l_t} - \beta_2 g_t\sqrt{l_t} + \sqrt{2}kl_t^{1/4}\epsilon\sqrt{1+\beta_1^2} \\
&\approx -hr_{t-1}\sqrt{l_t} - \beta_2 g_t\sqrt{l_t} + 2kl_t^{1/4}\epsilon
\end{aligned}$$

where $\beta_1 = 1 - hl_t^{1/2}$. Therefore, the dynamic of r_t can be described as:

$$dr_t = -hr_{t-1}dt - \beta_2 g_t dt + \sqrt{2+2\beta_1^2}kdB_t$$

There are two sources of noise from the above dynamic:

- The injected noise from the Monte Carlo gradient.
- The gradient is calculated by a min-batch of data

The friction term $-hr_{t-1}dt$ in the momentum can minimize the effect on the dynamics [Chen et al., 2014]. From a physical perspective, it helps decrease the extra energy produced by the noise and finally preserves the energy level.

References

[Agresti, 2001] Agresti, A. (2001). *Analysis of ordinal categorical data*. N. J: Wiley.

[Ahn et al., 2015] Ahn, S., Korattikara, A., Liu, N., Rajan, S., and Welling, M. (2015). Large-scale distributed bayesian matrix factorization using stochastic gradient mcmc. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 9–18.

[Albert and Chib, 1993] Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

[Amari and Cichocki, 2010] Amari, S.-i. and Cichocki, A. (2010). Information geometry of divergence functions. *Bulletin of the polish academy of sciences. Technical sciences*, 58(1):183–195.

[Andrews and Mallows, 1974] Andrews, D. F. and Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):99–102.

[Armagan, 2009] Armagan, A. (2009). Variational bridge regression. In *Artificial Intelligence and Statistics*, pages 17–24.

[Armagan et al., 2013] Armagan, A., Dunson, D. B., and Lee, J. (2013). Generalized double pareto shrinkage. *Statistica Sinica*, 23(1):119.

[Ashukha et al., 2020] Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. (2020). Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*.

[Asuncion et al., 2012] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2012). On smoothing and inference for topic models. *arXiv preprint arXiv:1205.2662*.

[Bae et al., 2018] Bae, J., Zhang, G., and Grosse, R. (2018). Eigenvalue corrected noisy natural gradient. *arXiv preprint arXiv:1811.12565*.

[Bae and Mallick, 2004] Bae, K. and Mallick, B. K. (2004). Gene selection using a two-level hierarchical bayesian model. *Bioinformatics*, 20(18):3423–3430.

[Bai et al., 2020a] Bai, J., Song, Q., and Cheng, G. (2020a). Efficient variational inference for sparse deep learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 33:466–476.

[Bai et al., 2020b] Bai, J., Song, Q., and Cheng, G. (2020b). Nearly optimal variational inference for high dimensional regression with shrinkage priors. *arXiv preprint arXiv:2010.12887*.

[Bellec et al., 2017] Bellec, G., Kappel, D., Maass, W., and Legenstein, R. (2017). Deep rewiring: Training very sparse deep networks. *arXiv preprint arXiv:1711.05136*.

[Belloni et al., 2011] Belloni, A., Chernozhukov, V., and Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806.

[Bhattacharya et al., 2016] Bhattacharya, A., Chakraborty, A., and Mallick, B. K. (2016). Fast sampling with gaussian scale mixture priors in high-dimensional regression. *Biometrika*, page asw042.

[Bhattacharya et al., 2015] Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110(512):1479–1490.

[Bishop and Nasrabadi, 2006] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

[Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

[Bové and Held, 2011] Bové, D. S. and Held, L. (2011). Hyper-g priors for generalized linear models. *Bayesian Analysis*, 6(3):387–410.

[Bühlmann and Van De Geer, 2011] Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

[Burgess et al., 2018] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*.

[Candes and Tao, 2007] Candes, E. and Tao, T. (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35(6):2313–2351.

[Carbonetto and Stephens, 2012] Carbonetto, P. and Stephens, M. (2012). Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian analysis*, 7(1):73–108.

[Carvalho et al., 2010] Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.

[Casella, 2001] Casella, G. (2001). Empirical bayes gibbs sampling. *Biostatistics*, 2(4):485–500.

[Castillo et al., 2015] Castillo, I., Schmidt-Hieber, J., Van der Vaart, A., et al. (2015). Bayesian linear regression with sparse priors. *Annals of Statistics*, 43(5):1986–2018.

[Castro et al., 2022] Castro, E., Godavarthi, A., Rubinfien, J., Givechian, K., Bhaskar, D., and Krishnaswamy, S. (2022). Transformer-based protein generation with regularized latent space optimization. *Nature Machine Intelligence*, 4(10):840–851.

[Chen et al., 2015] Chen, C., Ding, N., and Carin, L. (2015). On the convergence of stochastic gradient mcmc algorithms with high-order integrators. *Advances in neural information processing systems*, 28.

[Chen et al., 2018] Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31.

[Chen et al., 2014] Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR.

[Chen et al., 2021] Chen, T., Sui, Y., Chen, X., Zhang, A., and Wang, Z. (2021). A unified lottery ticket hypothesis for graph neural networks. In *International Conference on Machine Learning*, pages 1695–1706. PMLR.

[Dai et al., 2018] Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. (2018). Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR.

[Datta et al., 2013] Datta, J., Ghosh, J. K., et al. (2013). Asymptotic properties of bayes risk for the horseshoe prior. *Bayesian Analysis*, 8(1):111–132.

[Daxberger et al., 2021] Daxberger, E., Nalisnick, E., Allingham, J. U., Antorán, J., and Hernández-Lobato, J. M. (2021). Bayesian deep learning via subnetwork inference. In *International Conference on Machine Learning*, pages 2510–2521. PMLR.

[Dettmers and Zettlemoyer, 2019] Dettmers, T. and Zettlemoyer, L. (2019). Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*.

[Devroye, 2009] Devroye, L. (2009). Random variate generation for exponentially and polynomially tilted stable distributions. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 19(4):1–20.

[Donoho et al., 1992] Donoho, D. L., Johnstone, I. M., Hoch, J. C., and Stern, A. S. (1992). Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society: Series B (Methodological)*, 54(1):41–67.

[Du and Mordatch, 2019] Du, Y. and Mordatch, I. (2019). Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*.

[Dusenberry et al., 2020] Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. (2020). Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pages 2782–2792. PMLR.

[Efron, 2011] Efron, B. (2011). Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614.

[Elinas et al., 2020] Elinas, P., Bonilla, E. V., and Tiao, L. (2020). Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. *Advances in Neural Information Processing Systems*, 33:18648–18660.

[Fan et al., 2012] Fan, J., Guo, S., and Hao, N. (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):37–65.

[Fan and Li, 2001] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

[Fan and Lv, 2008] 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(5):849–911.

[Fan and Lv, 2011] Fan, J. and Lv, J. (2011). Nonconcave penalized likelihood with np-dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484.

[Fang et al., 2020] Fang, H., Harvey, N., Portella, V., and Friedlander, M. (2020). Online mirror descent and dual averaging: keeping pace in the dynamic case. In *International conference on machine learning*, pages 3008–3017. PMLR.

[Figueiredo, 2003] Figueiredo, M. A. (2003). Adaptive sparseness for supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 25(9):1150–1159.

[Fil et al., 2021] Fil, M., Mesinovic, M., Morris, M., and Wildberger, J. (2021). Beta-vae reproducibility: Challenges and extensions. *arXiv preprint arXiv:2112.14278*.

[Frank and Friedman, 1993] Frank, L. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

[Friedman et al., 2010] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

[Friedman, 2012] Friedman, J. H. (2012). Fast sparse regression and classification. *International Journal of Forecasting*, 28(3):722–738.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

[Gelman, 2008] Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in medicine*, 27(15):2865–2873.

[Gelman et al., 2013] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

[George and Foster, 2000] George, E. and Foster, D. P. (2000). Calibration and empirical bayes variable selection. *Biometrika*, 87(4):731–747.

[George and McCulloch, 1993] George, E. I. and McCulloch, R. E. (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

[George and McCulloch, 1997] George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica sinica*, pages 339–373.

[Ghosal et al., 2000] Ghosal, S., Ghosh, J. K., Van Der Vaart, A. W., et al. (2000). Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531.

[Ghosal and Van der Vaart, 2017] Ghosal, S. and Van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press.

[Ghosal et al., 2007] Ghosal, S., Van Der Vaart, A., et al. (2007). Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, 35(1):192–223.

[Ghosh and Chakrabarti, 2017] Ghosh, P. and Chakrabarti, A. (2017). Asymptotic optimality of one-group shrinkage priors in sparse high-dimensional problems. *Bayesian Analysis*, 12(4):1133–1161.

[Ghosh et al., 2019] Ghosh, S., Yao, J., and Doshi-Velez, F. (2019). Model selection in bayesian neural networks via horseshoe priors. *J. Mach. Learn. Res.*, 20(182):1–46.

[Gilks and Wild, 1992] Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348.

[Glynn, 1990] Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

[Goodfellow et al., 2020] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

[Graves, 2011] Graves, A. (2011). Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.

[Gur-Ari et al., 2018] Gur-Ari, G., Roberts, D. A., and Dyer, E. (2018). Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*.

[Hahn et al., 2019] Hahn, P. R., He, J., and Lopes, H. F. (2019). Efficient sampling for gaussian linear regression with arbitrary priors. *Journal of Computational and Graphical Statistics*, 28(1):142–154.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[Heek and Kalchbrenner, 2019] Heek, J. and Kalchbrenner, N. (2019). Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*.

[Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR.

[Higgins et al., 2016] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[Huang et al., 2017] Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*.

[Huang et al., 2008] Huang, J., Horowitz, J. L., Ma, S., et al. (2008). Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, 36(2):587–613.

[Huang et al., 2016] Huang, X., Wang, J., and Liang, F. (2016). A variational algorithm for bayesian variable selection. *arXiv preprint arXiv:1602.07640*.

[Hutto and Gilbert, 2014] Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

[Ishwaran et al., 2005] Ishwaran, H., Rao, J. S., et al. (2005). Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773.

[Izmailov et al., 2018] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

[Izmailov et al., 2021] Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. (2021). What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR.

[Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

[Johndrow et al., 2020] Johndrow, J., Orenstein, P., and Bhattacharya, A. (2020). Scalable approximate mcmc algorithms for the horseshoe prior. *Journal of Machine Learning Research*, 21(73).

[Jordan et al., 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[Khan et al., 2018] Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International Conference on Machine Learning*, pages 2611–2620. PMLR.

[Kim, 2022] Kim, F. (2022). *Analysis of student evaluation of teaching surveys: assessing evidence of implicit bias using numerical and text data.* Phd thesis, University of new south wales.

[Kim and Mnih, 2018] Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Kleijnen and Rubinstein, 1996] Kleijnen, J. P. and Rubinstein, R. Y. (1996). Optimization and sensitivity analysis of computer simulation models by the score function method. *European Journal of Operational Research*, 88(3):413–427.

[Knight et al., 2000] Knight, K., Fu, W., et al. (2000). Asymptotics for lasso-type estimators. *The Annals of statistics*, 28(5):1356–1378.

[Kowal et al., 2019] Kowal, D. R., Matteson, D. S., and Ruppert, D. (2019). Dynamic shrinkage processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(4):781–804.

[Kusupati et al., 2020] Kusupati, A., Ramanujan, V., Somani, R., Wortsman, M., Jain, P., Kakade, S., and Farhadi, A. (2020). Soft threshold weight reparameterization for learnable sparsity. In *International Conference on Machine Learning*, pages 5544–5555. PMLR.

[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

[Leimkuhler et al., 2019] Leimkuhler, B., Matthews, C., and Vlaar, T. (2019). Partitioned integrators for thermodynamic parameterization of neural networks. *arXiv preprint arXiv:1908.11843*.

[Li et al., 2016] Li, C., Chen, C., Carlson, D., and Carin, L. (2016). Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[Li and Clyde, 2018] Li, Y. and Clyde, M. A. (2018). Mixtures of g-priors in generalized linear models. *Journal of the American Statistical Association*, 113(524):1828–1845.

[Liang et al., 2008] 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(481):410–423.

[Lin et al., 2020] Lin, T., Stich, S. U., Barba, L., Dmitriev, D., and Jaggi, M. (2020). Dynamic model pruning with feedback. *arXiv preprint arXiv:2006.07253*.

[Liu and Rubin, 1994] Liu, C. and Rubin, D. B. (1994). The ecme algorithm: a simple extension of em and ecm with faster monotone convergence. *Biometrika*, 81(4):633–648.

[Liu et al., 2019] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

[Locatello et al., 2020] Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2020). A sober look at the unsupervised learning of disentangled representations and their evaluation. *arXiv preprint arXiv:2010.14766*.

[Loshchilov and Hutter, 2016] Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[Luo, 2022] Luo, C. (2022). Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*.

[Maddison et al., 2016] Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

[Maddox et al., 2019] Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32.

[Makalic and Schmidt, 2015] Makalic, E. and Schmidt, D. F. (2015). A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182.

[Mallick and Yi, 2018] Mallick, H. and Yi, N. (2018). Bayesian bridge regression. *Journal of applied statistics*, 45(6):988–1008.

[Mandt et al., 2017] Mandt, S., Hoffman, M. D., and Blei, D. M. (2017). Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*.

[Marjanovic and Solo, 2012] Marjanovic, G. and Solo, V. (2012). On l_q optimization and matrix completion. *IEEE Transactions on signal processing*, 60(11):5714–5724.

[Marjanovic and Solo, 2013] Marjanovic, G. and Solo, V. (2013). On exact l q denoising. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6068–6072. IEEE.

[Marjanovic and Solo, 2014] Marjanovic, G. and Solo, V. (2014). $l\_\{q\}$ sparsity penalized linear regression with cyclic descent. *IEEE Transactions on Signal Processing*, 62(6):1464–1475.

[Mazumder et al., 2011] Mazumder, R., Friedman, J. H., and Hastie, T. (2011). Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138.

[Meng and Rubin, 1993] Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278.

[Meng and Van Dyk, 1997] Meng, X.-L. and Van Dyk, D. (1997). The em algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–567.

[Mitchell and Beauchamp, 1988] Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032.

[Moran et al., 2019] Moran, G. E., Ročková, V., and George, E. I. (2019). Variance prior forms for high-dimensional bayesian variable selection. *Bayesian Analysis*, 14(4):1091–1119.

[Moran et al., 2018] Moran, G. E., Ročková, V., George, E. I., et al. (2018). Variance prior forms for high-dimensional bayesian variable selection. *Bayesian Analysis*, pages 1091–1119.

[Moran et al., 2022] Moran, G. E., Sridhar, D., Wang, Y., and Blei, D. (2022). Identifiable deep generative models via sparse decoding. *Transactions on Machine Learning Research*.

[Mostafa and Wang, 2019] Mostafa, H. and Wang, X. (2019). Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pages 4646–4655. PMLR.

[Narisetty and He, 2014] Narisetty, N. N. and He, X. (2014). Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817.

[Narisetty et al., 2018] Narisetty, N. N., Shen, J., and He, X. (2018). Skinny gibbs: A consistent and scalable gibbs sampler for model selection. *Journal of the American Statistical Association*.

[Neal and Hinton, 1998] Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

[Ning et al., 2020] Ning, B., Jeong, S., Ghosal, S., et al. (2020). Bayesian linear regression for multivariate responses under group sparsity. *Bernoulli*, 26(3):2353–2382.

[Ormerod et al., 2017] Ormerod, J. T., You, C., and Müller, S. (2017). A variational bayes approach to variable selection. *Electronic Journal of Statistics*, 11(2):3549–3594.

[Osawa et al., 2019] Osawa, K., Swaroop, S., Khan, M. E. E., Jain, A., Eschenhagen, R., Turner, R. E., and Yokota, R. (2019). Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32.

[Parisi, 1981] Parisi, G. (1981). Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384.

[Park and Casella, 2008] Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

[Park and Van Dyk, 2009] Park, T. and Van Dyk, D. A. (2009). Partially collapsed gibbs samplers: Illustrations and applications. *Journal of Computational and Graphical Statistics*, 18(2):283–305.

[Pingel, 2014] Pingel, R. (2014). Some approximations of the logistic distribution with application to the covariance matrix of logistic regression. *Statistics and Probability Letters*, 85(63-68).

[Polson and Scott, 2010] Polson, N. G. and Scott, J. G. (2010). Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian statistics*, 9(501-538):105.

[Polson and Scott, 2012] Polson, N. G. and Scott, J. G. (2012). Local shrinkage rules, lévy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):287–311.

[Polson et al., 2015] Polson, N. G., Scott, J. G., and Willard, B. T. (2015). Proximal algorithms in statistics and machine learning. *Statistical Science*, 30(4):559–581.

[Polson et al., 2014] Polson, N. G., Scott, J. G., and Windle, J. (2014). The bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):713–733.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR.

[Raskutti and Mukherjee, 2015] Raskutti, G. and Mukherjee, S. (2015). The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457.

[Ray and Szabó, 2021] Ray, K. and Szabó, B. (2021). Variational bayes for high-dimensional linear regression with sparse priors. *Journal of the American Statistical Association*, pages 1–12.

[Reid et al., 2016] Reid, S., Tibshirani, R., and Friedman, J. (2016). A study of error variance estimation in lasso regression. *Statistica Sinica*, pages 35–67.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

[Riesselman et al., 2018] Riesselman, A. J., Ingraham, J. B., and Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822.

[Ritter et al., 2018] Ritter, H., Botev, A., and Barber, D. (2018). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

[Roberts and Tweedie, 1996] Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363.

[Ročková, 2018] Ročková, V. (2018). Particle em for variable selection. *Journal of the American Statistical Association*, 113(524):1684–1697.

[Ročková and George, 2014] Ročková, V. and George, E. I. (2014). EMVS: The EM approach to bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846.

[Ročková and George, 2016a] Ročková, V. and George, E. I. (2016a). Bayesian penalty mixing: the case of a non-separable penalty. In *Statistical Analysis for High-Dimensional Data*, pages 233–254. Springer.

[Ročková and George, 2016b] Ročková, V. and George, E. I. (2016b). Fast bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 111(516):1608–1622.

[Ročková and George, 2018] Ročková, V. and George, E. I. (2018). The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444.

[Rue, 2001] Rue, H. (2001). Fast sampling of gaussian markov random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):325–338.

[Scott, 2010] Scott, J. G. (2010). Parameter expansion in local-shrinkage models. *arXiv preprint arXiv:1010.5265*.

[Scott and Berger, 2010] Scott, J. G. and Berger, J. O. (2010). Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, pages 2587–2619.

[Scott et al., 2016] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11:78–88.

[Shao, 1997] Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica sinica*, pages 221–242.

[Shin and Liu, 2021] Shin, M. and Liu, J. S. (2021). Neuronized priors for bayesian sparse linear regression. *Journal of the American Statistical Association*, pages 1–16.

[Smith et al., 1996] Smith, M., Kohn, R., et al. (1996). Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75(2):317–344.

[Song and Liang, 2017] Song, Q. and Liang, F. (2017). Nearly optimal bayesian shrinkage for high dimensional regression. *arXiv preprint arXiv:1712.08964*.

[Soussen et al., 2011] Soussen, C., Idier, J., Brie, D., and Duan, J. (2011). From bernoulli–gaussian deconvolution to sparse signal restoration. *IEEE Transactions on Signal Processing*, 59(10):4572–4584.

[Sun et al., 2021] Sun, Y., Song, Q., and Liang, F. (2021). Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, pages 1–15.

[Teh et al., 2006] Teh, Y., Newman, D., and Welling, M. (2006). A collapsed variational bayesian inference algorithm for latent dirichlet allocation. *Advances in neural information processing systems*, 19.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[Tonolini et al., 2020] Tonolini, F., Jensen, B. S., and Murray-Smith, R. (2020). Variational sparse coding. In *Uncertainty in Artificial Intelligence*, pages 690–700. PMLR.

[Trinh et al., 2022] Trinh, T. Q., Heinonen, M., Acerbi, L., and Kaski, S. (2022). Tackling covariate shift with node-based bayesian neural networks. In *International Conference on Machine Learning*, pages 21751–21775. PMLR.

[Van Der Pas et al., 2016] Van Der Pas, S., Salomond, J.-B., and Schmidt-Hieber, J. (2016). Conditions for posterior contraction in the sparse normal means problem. *Electronic journal of statistics*, 10(1):976–1000.

[van der Pas et al., 2017a] van der Pas, S., Szabó, B., and van der Vaart, A. (2017a). Adaptive posterior contraction rates for the horseshoe. *Electronic Journal of Statistics*, 11(2):3196–3225.

[van der Pas et al., 2017b] van der Pas, S., Szabó, B., and van der Vaart, A. (2017b). Uncertainty quantification for the horseshoe (with discussion). *Bayesian Analysis*, 12(4):1221–1274.

[Van Der Pas et al., 2014] Van Der Pas, S. L., Kleijn, B. J., and Van Der Vaart, A. W. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2):2585–2618.

[Van Dyk and Jiao, 2015] Van Dyk, D. A. and Jiao, X. (2015). Metropolis-hastings within partially collapsed gibbs samplers. *Journal of Computational and Graphical Statistics*, 24(2):301–327.

[Van Dyk and Park, 2008] Van Dyk, D. A. and Park, T. (2008). Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103(482):790–796.

[Velickovic et al., 2017] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *stat*, 1050:20.

[Wainwright, 2019] Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

[Wainwright et al., 2008] Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

[Wang, 2009] Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104(488):1512–1524.

[Wang, 2012] Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4):867–886.

[Wang et al., 2016] Wang, J., Liang, F., and Ji, Y. (2016). An ensemble EM algorithm for bayesian variable selection. *arXiv preprint arXiv:1603.04360*.

[Welling and Kipf, 2016] Welling, M. and Kipf, T. N. (2016). Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

[Welling and Teh, 2011] Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer.

[Wen et al., 2020] Wen, Y., Tran, D., and Ba, J. (2020). Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*.

[Wenzel et al., 2020] Wenzel, F., Roth, K., Veeling, B. S., Świątkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*.

[West, 1987] West, M. (1987). On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648.

[Williams, 1992] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

[Wilson and Izmailov, 2020] Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.

[Xie et al., 2022] Xie, Z., Wang, X., Zhang, H., Sato, I., and Sugiyama, M. (2022). Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International Conference on Machine Learning*, pages 24430–24459. PMLR.

[Xu et al., 2018] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

[Xu and Ghosh, 2015] Xu, X. and Ghosh, M. (2015). Bayesian variable selection and estimation for group lasso. *Bayesian Analysis*, 10(4):909–936.

[Yang et al., 2020] Yang, Y., Pati, D., and Bhattacharya, A. (2020). α-variational inference with statistical guarantees. *The Annals of Statistics*, 48(2):886–905.

[Ye and Ji, 2021] Ye, Y. and Ji, S. (2021). Sparse graph attention networks. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhang et al., 2010] Zhang, C.-H. et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942.

[Zhang et al., 2018] Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018). Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5852–5861. PMLR.

[Zhang et al., 2016] Zhang, L., Guindani, M., Versace, F., Engelmann, J. M., and Vannucci, M. (2016). A spatiotemporal nonparametric bayesian model of multi-subject fmri data. *The Annals of Applied Statistics*, 10(2):638–666.

[Zhang et al., 2019] Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2019). Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*.

[Zhang and Zitnik, 2020] Zhang, X. and Zitnik, M. (2020). Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275.

[Zhang and Archer, 2021] Zhang, Y. and Archer, K. J. (2021). Bayesian penalized cumulative logit model for high-dimensional data with an ordinal response. *Statistics in Medicine*, 40(6):1453–1481.

[Zhang et al., 2020] Zhang, Y. D., Naughton, B. P., Bondell, H. D., and Reich, B. J. (2020). Bayesian regression using a prior on the model fit: The r2-d2 shrinkage prior. *Journal of the American Statistical Association*, pages 1–13.

[Zhao et al., 2017] Zhao, S., Song, J., and Ermon, S. (2017). Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*.

[Zhu et al., 2019] Zhu, D., Zhang, Z., Cui, P., and Zhu, W. (2019). Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407.

[Zhu and Gupta, 2017] Zhu, M. and Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.

[Zou, 2006] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

[Zou and Li, 2008] Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509.