Stony Brook University

Group LASSO for Prediction of Clinical Outcomes in Cancer

A Dissertation presented

by

Xinyu Tian

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Statistics

Stony Brook University

May 2017

(include this copyright page only if you are selecting copyright through ProQuest, which is optional)

Stony Brook University

The Graduate School

Xinyu Tian

We, the dissertation committe for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Xuefeng Wang - Dissertation Advisor
Assistant Professor, Department of Applied Mathematics and Statistics

PeiFen Kuan - Chairperson of Defense
Assistant Professor, Department of Applied Mathematics and Statistics

Wei Zhu - Dissertation Coadvisor
Deputy Chair, Professor, Department of Applied Mathematics and Statistics

Xiaxia Yu - Outside Member
Assistant Professor, Department of Biomedical Informatics

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Group LASSO for Prediction of Clinical Outcomes in Cancer

by

Xinyu Tian

Doctor of Philosophy

in

Applied Mathematics and Statistics

Statistics

Stony Brook University

2017

High-dimensional datasets are now ubiquitous in biomedical research. Feature selection is an essential step in mining high-dim data to reduce noise, avoid overfitting and improve the interpretation of statistical models. In the last few decades, numerous feature selection methods and algorithms have been proposed for various response types, connections in predictors and requirements on sparsities; and penalized methods, such as LASSO and its variations, are among the most efficient and popular ones in this area. In addition, genomic features, such as gene expressions, are usually connected through an underlying biological network, which is an important supplement to the model in improving performance and interpretability. In this study, we first extend the group LASSO to a network-constrained classification model and develop a modified proximal gradient algorithm for the model fitting. In this algorithm, group lasso regularization is used to induce model sparsity, and a network constraint is imposed to induce the smoothness of the coefficients using underlying network structure. The applicability of the proposed method is verified by analyzing both numerical examples and real gene expression data in TCGA.

We further work on the feature selection problem with Bayesian hierarchical structure. R. Tibshirani, who introduced LASSO in 1996, also proposed that linear LASSO can be considered as a Bayesian model with Laplace prior

on coefficient parameters, which shed lights on the feature selection problem in Bayesian models. Compared to frequentist approaches, Bayesian model copes better with complex hierarchical structures of the data. On one hand, we compare the performance of Laplace, horseshoe and Gaussian priors in linear Bayesian models with extensive simulations. On the other, we extend the projection predictive feature selection scheme to group-wise selection and benchmark its feature selection performance and prediction accuracy with standard Bayesian methods. All Bayesian posterior parameters are estimated using Hamiltonian Monte Carlo implemented in Stan.

Contents

List of Figures

List of Tables

List of Abbreviations

ACC	Overall accuracy rate
AUC	Area Under Curve
BMA	Bayesian model averaging
BUGS	Bayesian inference Using Gibbs Sampling
DIC	Deviance information criterion
HMC	Hamiltonian Monte Carlo
JAGS	Just Another Gibbs Sampling
KL	Kullback Leibler
LOO-CV	Leave-one-out cross validation
LPD	Logarithm of the predictive density
MCMC	Markov Chain Monte Carlo
MLPD	Mean logarithm of the predictive density
MSE	Mean squared error
NPV	Negative predictive value
PPV	Positive predictive value
PRMSE	Prediction root-mean-squared error
ROC	Receiver operating characteristic
RSSE	Root-sum-square error
SEN	True positive rate or sensitivity
SNR	Signal-to-noise ratio
SPC	True negative rate or specificity
SVM	Support vector machine
WAIC	Widely applicable information criterion

Acknowledgements

I would like to express my deepest gratitude to my advisors, Dr. Xuefeng Wang and Dr. Wei Zhu, for their splendid academic tutorial, caring, and providing me with a convenient atmosphere for research and work. I would never have been able to finish my dissertation without their help.

I am especially indebted to Dr. Ellen Li for her support in these years. I benefit tremendously from working with her not only on acquiring knowledge, technical skills and lab experiences, but also on getting motivated by her huge passion for public health and meticulous and rigorous attitude on work.

My sincere thanks also go to Dr. Pei Fen Kuan, Dr. Xiaxia Yu and Dr. Song Wu for spending their precious time in participating my thesis defense and/or preliminary exam. My work would not be as complete without your advice.

Finally, I would also like to thank my friends and my parents for their unconditional love and support.

Publications

Tian, X., Wang, X., & Chen, J. (2014). Network-constrained group lasso for high-dimensional multinomial classification with application to cancer subtype prediction. Cancer informatics, 13(Suppl 6), 25.

Son, J. S., Zheng, L. J., Rowehl, L. M., Tian, X., Zhang, Y., Zhu, W., ... & Ir, D. (2015). Comparison of fecal microbiota in children with autism spectrum disorders and neurotypical siblings in the Simons Simplex Collection. PloS one, 10(10), e0137725.

Son, J. S., Khair, S., Pettet III, D. W., Ouyang, N., Tian, X., Zhang, Y., ... & Frank, D. N. (2015). Altered Interactions between the Gut Microbiome and Colonic Mucosa Precede Polyposis in APC Min/+ Mice. PloS one, 10(6), e0127985.

Wang, X., Ji, P., Zhang, Y., LaComb, J. F., Tian, X., Li, E., & Williams, J. L. (2016). Aberrant DNA Methylation: Implications in Racial Health Disparity. PloS one, 11(4), e0153125.

Chapter 1

Network-constrained Group LASSO

Classic multinomial logit model, commonly used in multiclass regression problem, is restricted to few predictors and with no regard to the relationship among variables. Its usage is insufficient for genomic data, where the number of genomic features far exceeds the sample size. Also, genomic features such as gene expressions are usually related to each other via an underlying biological network. Making use of the network information is crucial to improving model performance as well as the biological interpretability. In this Chapter, a classification model based on logistic regression is discussed, which accommodates network information and group LASSO as well. The result has been published in 2014 [1].

1.1 Introduction

In cancer diagnosis, cancer patients with the same diagnostic profile may have different clinical outcomes. This difference probably lies in the limitation of the traditional strategies in tumor type classification, which typically are based

on morphology only. A reliable and precise classification of tumors is essential for successful diagnosis [2]. Modern sequencing and microarray technology has enabled more detailed molecular characterization of cancer samples, leading to the discovery of many cancer subtypes. Depending on the subtype, different treatments will be administered. In conclusion, cancer subtype identification has become an integral part of personalized medicine [3].

The problem of multiclass cancer classification has been approached in many ways, including multinomial logit models [4], Bayesian probit models [5, 6], random forest [7] and support vector machine (SVM) [8–11]. Other discriminationary methods, including linear discriminant analysis (LDA), k-nearest-neighbor (kNN) classifier and classification trees, were also investigated [12]. Among those, SVM was a successful procedure applied to microarray-based cancer diagnosis problems. However, SVM predicts the class label without estimating the underlying probabilities. Multinomial logit model, on the other hand, performs similarly to SVM, but it provides estimation of the probabilities [4]. The probabilistic nature of multinomial logit regression model has many advantages, such as the abilities to set the rejection thresholds freely, and to accommodate the frequency of each class in an unbalanced design [13].

Classic multinomial logit model works well when the number of predictors is small. However, a large number of predictors often leads to model overfitting and even a singular matrix of the normal equations when the number of predictors exceeds the number of observations as commonly seen in genomic studies. To deal with the curse of high-dimensionality as well as to increase the model interpretability, regularized procedures that incorporate a sparsity

penalty have been proposed [14–18]. Among these methods, group LASSO is particularly appropriate for models with multiclass responses, which means all the coefficients linked to a common predictor constitute a group and are forced to shrink to zero simultaneously in the process of variable selection [14].

Although the sparse multinomial logit models are able to achieve variable selection, they can not efficiently utilize prior biological information, such as a network of regulatory relationships between genes or gene-products. Such biological information has been accumulated through years of biomedical research and many databases such as KEGG, Reactome and MIPS have been developed to organize different types of biological network information. Cancer is a complex disease caused by dysregulation of pathways instead of a single gene [19–21]. Thus, the incorporation of the network information can potentially increase the power of identifying cancer subtypes.

Networks are often represented as graphs, where each vertex indicates a gene or a gene-product and each edge represents a relationship between two connected vertices. The incorporation of network information has been studied in other regression models. A constraint, enrolled by the Laplace matrix of a graph [22], has been proposed to facilitate the selection of predictors in ordinary regression setting, enhancing both the global smoothness over network and the interpretability of the association between selected genes and responses in the context of known biology.

1.2 Multinomial Logit Model and Penalized Likelihood Approach

For data $(y_i, \mathbf{x}_i), i = 1, \ldots, n$ with n observations and p predictors, y_i denotes an observation of the categorical response variable $Y \in \{1, \ldots, k\}$ and $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$ indicates an observation of a p-dimensional vector of predictors. Assuming that y_i follows a multinomial distribution, a multinomial logit model is built with logit link, which is,

$$\pi_{ir} = P(Y = r|\mathbf{x}_i) = \frac{\exp\left(\beta_{r0} + \mathbf{x}_i \beta_{r\cdot}^T\right)}{\sum_{s=1}^{k} \exp\left(\beta_{s0} + \mathbf{x}_i \beta_{s\cdot}^T\right)} = \frac{\exp(\eta_{ir})}{\sum_{s=1}^{k} \exp(\eta_{is})}, \tag{1.1}$$

where $\beta_{r\cdot} = (\beta_{r1}, , \beta_{r2}, \ldots, \beta_{rp})$ and $\eta_{ir} = \beta_{r0} + \mathbf{x}_i \beta_{r\cdot}^T$. We choose category k as the reference category by setting $\beta_{k0} = 0, \beta_{k\cdot} = \mathbf{0}$. Under this choice, the linear predictors $\eta_{ir}, r = 1, \ldots, k$ correspond to the log odds ratio between category r and the reference category k.

We regularize the multinomial logit model using a penalized likelihood approach, in which one maximizes the penalized log-likelihood

$$l_p(\beta) = l(\beta) - \lambda J(\beta), \tag{1.2}$$

over a $(k-1)\times(p+1)$-dimensional parameter vector $\beta = \left(\beta_{10}, \cdots, \beta_{(k-1)0}, \beta_{1\cdot} \cdots, \beta_{k-1,\cdot}\right)^T$. In (1.2),

$$l(\beta) = \sum_{i=1}^{n} \sum_{r=1}^{k} y_{ir} \log \pi_{ir} = \sum_{i=1}^{n} \left(\sum_{r=1}^{k-1} y_{ir} \eta_{ir} - \log \left(\sum_{s=1}^{k} \exp(\eta_{is}) \right) \right)$$

denoting the ordinary log-likelihood of a multinomial logit model. $J(\beta)$ is a

function that penalizes the size of the parameters and regularizes the structure of features. λ, the tuning parameter, determines the strength of the regularization.

Assuming that all predictors are metric and standardized, that is, each predictor has one degree of freedom and the differences in scale will not influence the penalty and thus the variable selection. In the multinomial logit model, we use a vector $\beta_{\cdot j} = (\beta_{1j}, \beta_{2j}, \ldots, \beta_{k-1,j})^T$ of parameters to capture the effect of variable x_j, so that variable selection is achieved only when the $k-1$ parameters are shrunk to zero simultaneously. Since the ordinary LASSO facilitates only parameter selection rather than predictor selection, a group LASSO penalty is applied to penalize the parameters at a group level, defined as

$$J_1(\beta) = \sum_{j=1}^{p} \phi_j \|\beta_{\cdot j}\| = \sum_{j=1}^{p} \phi_j \left(\beta_{1j}^2 + \beta_{2j}^2 + \cdots + \beta_{k-1,j}^2\right)^{\frac{1}{2}}, \qquad (1.3)$$

where ϕ_j is a penalty weight, set as 1 by default. All the parameters in a group $\beta_{\cdot j}$ would be shrunk to zero simultaneously.

In an association study, the graphs or networks depicting relationships among predictors are important priori information, which we may take advantage of. Consider a network represented by a weighted graph $G = (V, E, W)$ with the set of vertices $V = 1, \ldots, p$ corresponding to p predictors, the set of edges $E = \{(j,k) : j \text{ and } k \text{ are linked}\}$ and the set of weights $W = \{w_{jk} : (j,k) \in E\}$. w_{jk} measures the similarity of predictor j and k, with 1 for identity and 0 for complete difference, if normalized to the scale of $[0,1]$. We then construct an

adjacency matrix A by

$$a_{jk} = \begin{cases} w_{jk} & (j,k) \in E \\ 0 & (j,k) \notin E \end{cases},$$

and a degree matrix $D = diag\,(d_1, d_2, \ldots, d_p)$, where $d_j = \sum_{(j,k)\in E} w_{jk}$ is defined as the degree of vertex j. The normalized Laplacian matrix associated with graph G is $L = D - A$, whose the jk^{th} element is defined by

$$L_{jk} = \begin{cases} 1 - \frac{w(j,k)}{d_j} & \text{if } j = k, \text{ and } d_j \neq 0, \\ -\frac{w(j,k)}{\sqrt{d_j d_k}} & (j,k) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

In fact, L is always non-negative definite and can be factorized as $L = SS^T$. By simple algebra, $\beta_{r\cdot} L \beta_{r\cdot}^T$ can be written as

$$\beta_{r\cdot} L \beta_{r\cdot}^T = \sum_{(j,k)\in E} (\beta_{rj} - \beta_{rk})^2 w_{jk},$$

Thus, the network-constrained penalty [22–24], defined as

$$J_2(\beta) = \sum_{r=1}^{k-1} \beta_{r\cdot} L \beta_{r\cdot}^T, \tag{1.4}$$

induces a smooth solution of the vector $\beta_{r\cdot}$ with respect to the labeled weighted graph G.

To sum up, in our regularized model, the penalized log-likelihood function

is given by

$$
\begin{aligned}
l_p(\beta) &= l(\beta) - \lambda J(\beta) && (1.5)\\
&= \sum_{i=1}^{n}\left(\sum_{r=1}^{k-1} y_{ir}\eta_{ir} - \log\sum_{s=1}^{k} e^{\eta_{is}}\right) - \lambda_1\sum_{j=0}^{p}\phi_j\left\|\beta_{\cdot j}\right\| - \lambda_2\sum_{r=1}^{k-1}\beta_{r\cdot}L\beta_{r\cdot}^T && (1.6)
\end{aligned}
$$

of which the second term, the sparse penalty, induces model sparsity and the third term, the network penalty, imposes smoothness over the network. When $\lambda_2 = 0$, the model is reduced to the group LASSO multinomial logit model. The incorporation of the extra tuning parameter expands the parameter search space and directs the search to more biological meaningful regions.

Like ordinary LASSO, group LASSO also suffers from an issue of estimation bias, which is resulted from the fact that all predictors are penalized to the same degree. In order to reduce the bias, we use adaptive group LASSO, which penalizes predictors to different degrees by assigning a weight to each predictor. In our model, the weight is set to be the reciprocal of the L_2 norm of the fitted coefficients in univariate analysis, where we fit the model with each individual predictor only. Denoting $\tilde{\beta}_{\cdot j}$ the univariate estimate, the group LASSO penalty term (1.3) becomes

$$
J_1(\beta) = \sum_{j=1}^{p}\frac{1}{\left\|\tilde{\beta}_{\cdot j}\right\|}\left\|\beta_{\cdot j}\right\|.
$$

1.3 Proximal Gradient Method and Model Fitting

We use the proximal gradient based FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) to fit the model [25]. Consider the optimization of the general penalized log-likelihood $l_p(\beta) = l^*(\beta) - \lambda_1 J_1(\beta)$, composed by a concave and continuously differentiable term $l^*(\beta)$ and a convex penalty term $J_1(\beta)$. The penalized maximum likelihood estimator is defined by

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} l_p(\beta) = \arg\min_{\beta \in \mathbb{R}^d} \left(-l^*(\beta) + \lambda_1 J_1(\beta)\right), \tag{1.7}$$

where

$$l^*(\beta) = \sum_{i=1}^{n} \left(\sum_{r=1}^{k-1} y_{ir}\eta_{ir} - \log \sum_{s=1}^{k} e^{\eta_{is}} \right) - \lambda_2 \sum_{r=1}^{k-1} \beta_{r\cdot} L \beta_{r\cdot}^T.$$

is a smooth function with respect to parameter β.

With a positive step size v, the quadratic approximation [25] of $-l_p(\beta)$ at a given point β_0 is

$$\mathcal{Q}_v(\beta, \beta_0) = -l^*(\beta_0) - \nabla l^*(\beta_0)^T(\beta - \beta_0) + \frac{1}{2v}\|\beta - \beta_0\|^2 + \lambda_1 J_1(\beta).$$

$\nabla l^*(\beta)$, the first-order derivative of $l^*(\beta)$, is a $(k-1) \times (p+1)$-dimensional vector, whose element corresponding to β_{rj} is

$$[\nabla l^*(\beta)]_{rj} = \frac{\partial l^*}{\partial \beta_{rj}} = \sum_{i=1}^{n} \frac{\partial l^*}{\partial \eta_{ir}} \frac{\partial \eta_{ir}}{\partial \beta_{rj}} = \sum_{i=1}^{n} (y_{ir} - \pi_{ir}) x_{ij}.$$

The iterations of proximal gradient methods are defined by

$$\hat{\beta}^{(t+1)} = \arg\min_{\beta \in \mathbb{R}^d} \left\{ -l^* \left(\hat{\beta}^{(t)} \right) - \nabla l^* \left(\hat{\beta}^{(t)} \right)^T \left(\beta - \hat{\beta}^{(t)} \right) + \frac{1}{2v} \left\| \beta - \hat{\beta}^{(t)} \right\|^2 + \lambda_1 J_1 (\beta) \right\}, \tag{1.8}$$

which consists of a linear approximation of the negative modified log-likelihood at the current value $\hat{\beta}^{(t)}$, a proximity term and the penalty term.

First, we set $\lambda_1 = 0$, and based on the standard formula for the iterates of gradient methods for smooth optimization, the unpenalized estimator $\tilde{\beta}^{(t+1)}$ has an explicit form

$$\tilde{\beta}^{(t+1)} = \tilde{\beta}^{(t)} + v \nabla l^* \left(\tilde{\beta}^{(t)} \right).$$

Then we move back to the optimization problem with an active penalty. Via Lagrange duality, equation (1.7) can be equivalently expressed by

$$\hat{\beta} = \arg\min_{\beta \in C} \left(-l^* (\beta) \right),$$

where $C = \left\{ \beta \in \mathbb{R}^d | J_1 (\beta) \leq \kappa (\lambda_1) \right\}$ is the constraint region corresponding to $J_1 (\beta)$, and $\kappa (\lambda_1)$ is a tuning parameter that is linked to λ_1 by a one-to-one mapping. Given a search point $u \in \mathbb{R}^d$, the so-called proximal operator associated with $J_1 (\beta)$ is defined as

$$\mathcal{P}_\lambda (u) = \text{argmin}_{\beta \in \mathbb{R}^d} \left(\frac{1}{2} \|\beta - u\|^2 + \lambda J_1 (\beta) \right), \tag{1.9}$$

which is the projection of u onto region C. Then the proximal gradient iterates

defined in (1.8) can be equally expressed by the projection

$$\hat{\beta}^{(t+1)} = \mathcal{P}_{\lambda_1 v}\left(\hat{\beta}^{(t)} + v\nabla l^*\left(\hat{\beta}^{(t)}\right)\right).$$

Next, consider the proximal operator (1.9). Due to the block-separability of this specific penalty, the proximal operator can be written as

$$\mathcal{P}_\lambda\left(\tilde{\beta}\right) = \left(p_\lambda\left(\tilde{\beta}_{\cdot 0}\right), p_\lambda\left(\tilde{\beta}_{\cdot 1}\right), \cdots, p_\lambda\left(\tilde{\beta}_{\cdot p}\right)\right),$$

where

$$p_\lambda\left(\tilde{\beta}_{\cdot 0}\right) = \arg\min_{\beta_{\cdot j}\in\mathbb{R}^{k-1}}\left(\frac{1}{2}\left\|\beta_{\cdot 0} - \tilde{\beta}_{\cdot 0}\right\|^2\right) = \tilde{\beta}_{\cdot 0}$$

$$p_\lambda\left(\tilde{\beta}_{\cdot j}\right) = \arg\min_{\beta_{\cdot j}\in\mathbb{R}^{k-1}}\left(\frac{1}{2}\left\|\beta_{\cdot j} - \tilde{\beta}_{\cdot j}\right\|^2 + \lambda\phi_j\left\|\beta_{\cdot j}\right\|\right), j = 1, \ldots, p. \quad (1.10)$$

With $(u)_+ = max(u, 0)$, the explicit solution to the proximal operator (1.10) can be derived from the Karush-Kuhn-Tucker conditions:

$$p_\lambda\left(\tilde{\beta}_{\cdot j}\right) = \left(1 - \frac{\lambda\phi_j}{\left\|\tilde{\beta}_{\cdot j}\right\|}\right)_+ \tilde{\beta}_{\cdot j}, j = 1, \ldots, p.$$

Set $\tilde{\beta}^{(t+1)} = \hat{\beta}^{(t)} + v\nabla l^*\left(\hat{\beta}^{(t)}\right)$, then the solution to the optimization problem (1.8) can be expressed as

$$\begin{aligned}\hat{\beta}^{(t+1)} &= \mathcal{P}_{\lambda_1 v}\left(\tilde{\beta}^{(t+1)}\right) = \left(p_{\lambda_1 v}\left(\tilde{\beta}_{\cdot 0}^{(t+1)}\right), p_{\lambda_1 v}\left(\tilde{\beta}_{\cdot 1}^{(t+1)}\right), \cdots, p_{\lambda_1 v}\left(\tilde{\beta}_{\cdot p}^{(t+1)}\right)\right) \\ &= \left(\tilde{\beta}_{\cdot 0}^{(t+1)}, \left(1 - \frac{\lambda\phi_j}{\left\|\tilde{\beta}_{\cdot 1}^{(t+1)}\right\|}\right)_+ \tilde{\beta}_{\cdot 1}^{(t+1)}, \cdots, \left(1 - \frac{\lambda\phi_j}{\left\|\tilde{\beta}_{\cdot p}^{(t+1)}\right\|}\right)_+ \tilde{\beta}_{\cdot p}^{(t+1)}\right)\end{aligned}$$

To summarize, the basic idea of proximal gradient methods is: First, remove the L_1 penalty of the objective function (1.6) and then optimize the smooth part by taking a step toward its ML estimator via first-order methods, which creates a search point. Second, project this search point onto the constraint region C in order to account for the non-smooth penalty term. To accelerate the convergence rate, we extrapolate the current and the previous iterations with the help of deliberately chosen acceleration factors a_t [18],

$$\hat{\alpha}^{(t)} = \hat{\beta}^{(t)} + \frac{a_{t-1} - 1}{a_t} \left(\hat{\beta}^{(t)} - \hat{\beta}^{(t-1)} \right).$$

The extrapolate point $\hat{\alpha}^{(t)}$, instead of the current iterate $\hat{\beta}^{(t)}$, is used as a starting point to generate a search point, which is then projected on the penalty region.

To select the tuning parameters λ_1 and λ_2, we use cross-validation, where we divide the data set into training and test data set. The model is trained on the training data set and prediction error is then assessed on the test data set. We search on a grid of λ_1, λ_2 values and choose the value of λ_1, λ_2 that minimize the cross-validated errors. We use the Brier score, a measurement of the accuracy of probabilistic predictions defined as the Euclidean distance between sample response and its estimated distribution probabilities, to measure the prediction error.

1.4 Simulation

The purpose of the simulation is to show that the structure-constrained model dominates the alternative models that do not use such prior information in terms of parameter estimation and prediction. For each scenario presented, we simulate a training set and an independent test set both with 200 samples. We first select the optimal tuning parameters through a 5-fold cross validation on the training set. With the selected tuning parameters, a final model is built on the whole training set, and then tested on the test set. For each setting, we run 50 simulations, and calculate several criteria to evaluate the performance of the proposed model.

1.4.1 Simulation Settings

We consider a small model and a large model. Each model has 4 response categories. First of all, we construct a predictor matrix. The numbers of total predictors are 20 for small and 200 for large models, and the numbers of relevant ones are 4 and 10 respectively. The predictors are continuous and follow a multivariate normal distribution with mean $\mathbf{0}$ and the $p \times p$ correlation matrix

$$\Sigma = \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{p-1} \\ \rho & 1 & \rho & \cdots & \rho^{p-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \rho^{p-3} & \cdots & 1 \end{pmatrix}$$

where $\rho = 0.5$.

Secondly, we simulate the network structure of predictors. We divide the predictors into a few subsets (subnetworks). Taking the small model for example, 20 predictors are divided into 5 subnetworks evenly, and all the 4 relevant predictors belong to the first subnetwork. Ideally, we assume that predictors within each subnetwork are fully connected, and there is no connection between subnetworks. That is, the corresponding adjacency matrix is a block diagonal matrix, with the main diagonal blocks being all-ones square matrices, and the off-diagonal blocks are zero matrices. We label this scenario as 'Ideal structure'. To study the effects of model misspecification, we also simulate incorrect prior information, where the large adjacency coefficients are randomly drawn from (0.4, 1) and the small ones from (0, 0.6) without respect to the relevant variables.

Finally, we also simulate the coefficients, which can be represented by a $3 \times p$ matrix where rows are indexed by all categories of the response but the reference, and columns are indexed by the predictors. The columns with respect to the irrelevant predictors are filled with zeros. The entries of the the relevant columns have three settings: 'identical', 'similar' or 'random'. For 'identical' case, coefficients of relevant predictors in each category are identical. For 'similar' case, coefficients in each category have the same sign but different values, suggesting that all the relevant predictors impact the response in the same direction but different magnitude. Their absolute values are independently drawn from the set $\{0.05, 0.10, \cdots, 0.50\}$ and the sign of each category is random. In the case of 'random' coefficients, each coefficient is independently selected from a set of positive and negative val-

ues $\{-0.50, -0.45, \cdots, -0.05, 0.05, \cdots, 0.50\}$. The 'random' case violates the model assumption, where we expect the coefficients be similar within each subnetwork.

Under the multinomial logistic model, the actual class probabilities are calculated based on the coefficient matrix and the values of predictors for each observation. Then the class label is drawn randomly on the basis of calculated probabilities. We set the intercept to be zero to generate more balanced sample. However, due to the randomness in the data generation, the numbers of observations vary tremendously across categories.

Simulation Results

To see the improved performance of using prior structure information in terms of parameter estimation and prediction accuracy, we compare the variants of the proposed model, network-constrained multinomial logit model with group LASSO penalty (NGL-MLM) and the one with adaptive group LASSO penalty (NGL-MLMa) to two traditional multinomial logit models with LASSO (L-MLM) and group LASSO (GL-MLM) respectively, implemented in the package of glmnet in R [14]. To measure the estimation accuracy, the mean squared error (MSE) between true parameter values and the estimated ones, is used. In addition, the performance of prediction on test data is evaluated with 'Brier score', the Euclidean distance between sample response and the estimated distribution probabilities, and 'prediction accuracy', the proportion of correctly predicted class labels.

We first simulate 'ideal' network structure, that is, all the relevant variables

come from a fully connected subnetwork. Figure 1.1 shows the estimation performance of various models. As expected, the structure information improves estimation significantly, especially for large models, which is particular relevant for real applications. The estimation of the adaptive method (NGL-MLMa) outperforms others substantially. In case of random coefficients, where prior network does not provide any useful information, the proposed model is comparable to models without using the network information (L-MLM, GL-MLM), and sometimes even better. Figure 1.2 shows that the prediction accuracy is also higher for the proposed model in almost all scenarios. When Brier score is used (Figure 1.3), similar trend follows: the network-constrained model always performs better when we simulate ideal and similar coefficients, and is comparable to traditional models without using structure information in case of random coefficients.

To investigate the impact of incorrect prior network, we simulate a medium-sized model with 100 predictors, of which 10 are relevant. The performance of our model is still satisfactory due to the flexible tuning parameter on the smoothness penalty term (Figure 1.4). The prediction accuracy of NGL-MLM is comparable to that of GL-MLM. In terms of parameter estimation, the structure-constrained model performs better even if the structure information is incorrect. This may be due to the fact that network constraint provides better capability of shrinking the entire coefficients of a subnetwork to zeros.

In summary, our structure-constrained multinomial logit model has better performance in terms of parameter estimation and prediction when the prior network knowledge is at least partially correct and the performance is compa-

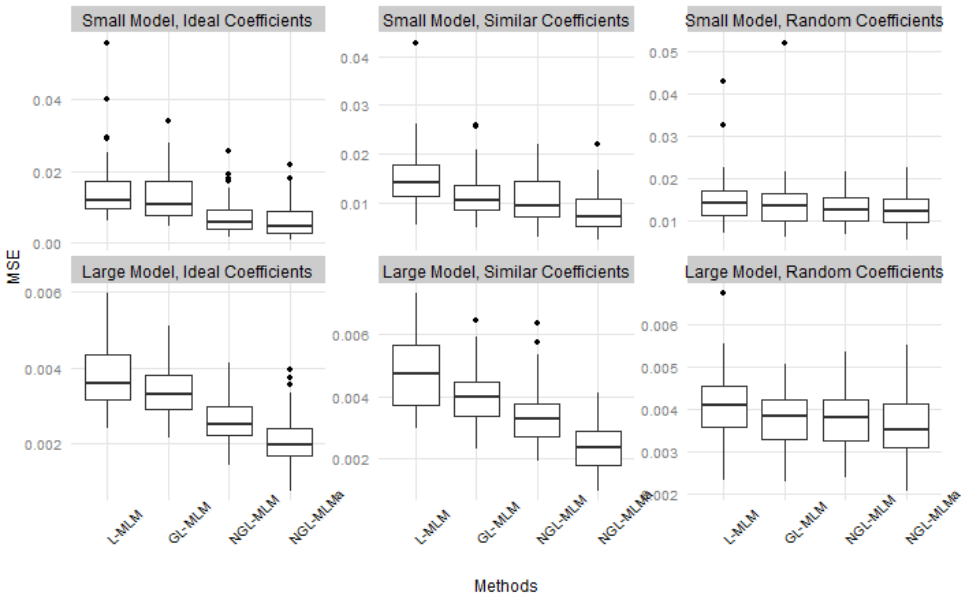

Figure 1.1. MSE of parameter estimation under ideal structure information for small and large models with ideal, similar and random coefficients.

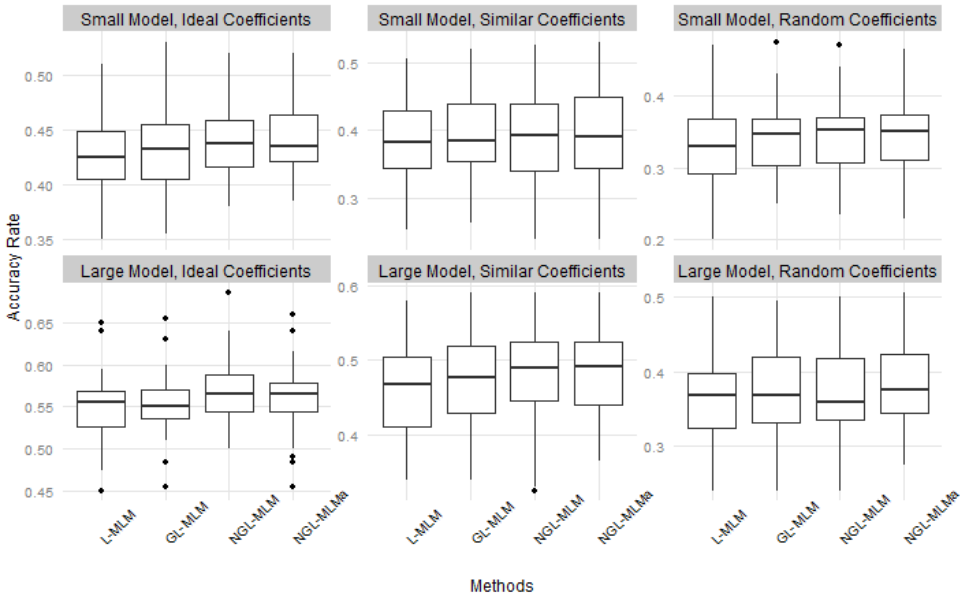

Figure 1.2. Prediction accuracy rate for small and large models with ideal, similar and random coefficients under ideal structure information.

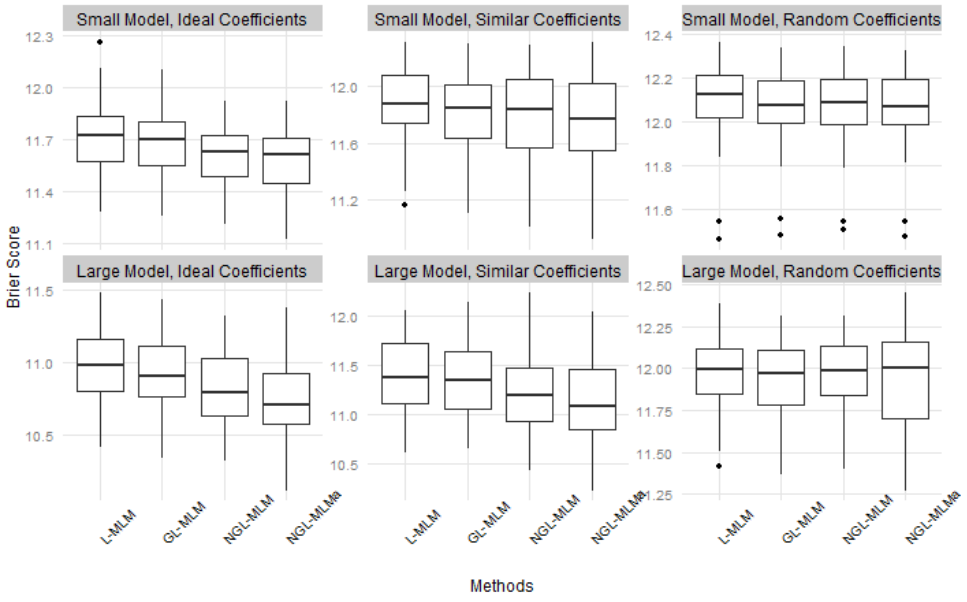

Figure 1.3. Brier scores for small and large models with ideal, similar and random coefficients under ideal structure information.

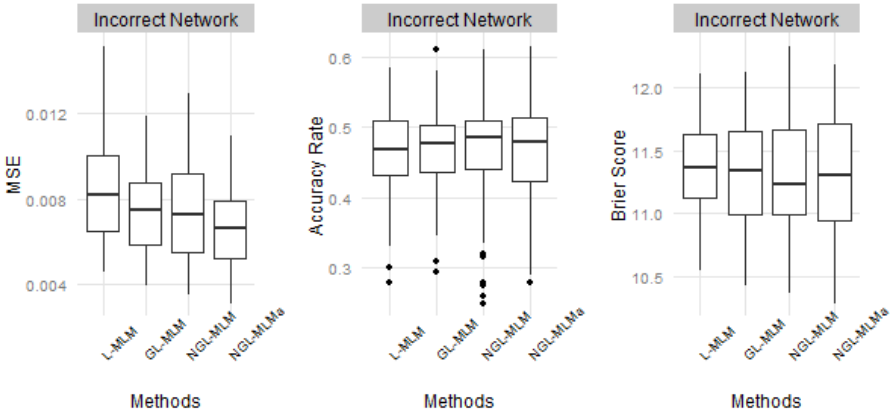

Figure 1.4. Comparison of four candidate methods under incorrect structure information in terms of MSE, accuracy rate and Brier score.

rable to traditional models when the network knowledge is incorrect. This is because that the GL-MLM is a special case of NGL-MLM with $\lambda_2 = 0$. Cross-validation tends to select $\lambda_2 = 0$ when the prior assumption is not correct.

1.5 Application to the GBM Data Set

One important application of our method is cancer subtype prediction and relevant subnetwork identification on large-scale gene expression data. We apply all four candidate methods, L-MLM, GL-MLM, NGL-MLM and NGL-MLMa, to a large-scale TCGA Glioblastoma Multiforme (GBM) subtype prediction problem, which contains the expression of 11,861 genes across 202 samples. The network was built from a variety of sources, including Reactome, KEGG, as well as the inferred gene-interaction from protein interactions, gene co-expression, protein domain interaction and text-mined interaction. The outcome is one of the four subtypes of Glioblastoma Multiforme [26]. The data set, the network information as well as the subtype information were downloaded from http://bioen-compbio.bioen.illinois.edu/NCIS/.

Since the number of genes in the GBM dataset is much larger than the number of samples, which may lead to computation instability, one common practice is to filter irrelevant genes before model building. Starting with 11,861 genes, we conduct gene screening based on the prior weights calculated by NCIS algorithm [26], and include the top 599 genes afterward. To construct the network constraint for model building, the original network is tailored to contain only the screened genes. Then the Laplacian matrix is constructed based on the tailored subnetwork.

To compare the prediction performance of the four methods, 202 samples are divided into two subsets, a training set with 150 samples and a test set with 52 samples. Feature selection and parameter estimation in model building are

	prediction accuracy (mean/sd)	model size
L-MLM	0.824 / 0.043	52.76
GL-MLM	0.858 / 0.044	43.18
NGL-MLM	0.859 / 0.053	37.54
NGL-MLMa	0.907 / 0.040	34.62

Table 1.1. Average prediction accuracy and average number of predictors in each model (model size) for the GBM dataset.

done strictly on the training set. Then the fitted models are tested on test set. The whole procedure of random division, model building and testing are repeated 50 times to assess variability, and results are summarized in table 1.1.

The tuning parameter of the network-constraint controls the impact of the prior structure knowledge on model building. The network information will have no effect if the tuning parameter is set to zero. Among the 50 models built by NGL-MLM, the network tuning parameter is chosen as zero in 28 models, where NGL-MLM is reduced to GL-MLM in this specific case. In contrast, 48 NGL-MLMa models choose non-zero tuning parameter for the network constraint, which indicates that the structure knowledge is useful for prediction, explaining the higher prediction accuracy rate for NGL-MLMa.

Next, we apply NGL-MLMa, the best model in both simulation and the application to GBM subtype analysis, on the whole sample set of GBM gene expression data and investigate the selected subnetwork. It selects 35 predictors, among which 21 are non-singletons and form a subnetwork, shown in Figure 1.5. The selected genes make great biological sense. For example, the most connected gene 'AKT1' plays an important role in the pathogenicity of

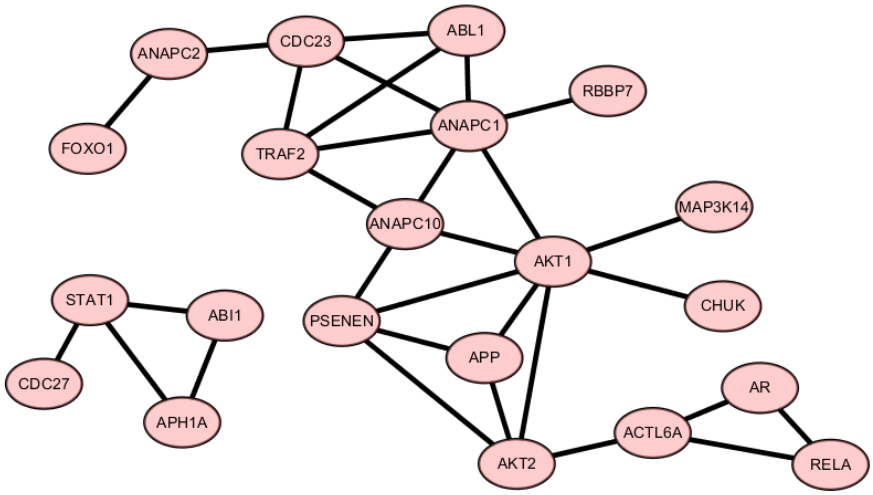

Figure 1.5. The subnetwork selected by NGL-MLMa on GBM gene expression data.

GBM. AKT1 is a downstream serine/theroine kinase in the RTK/PTEN/PI3K pathway and large scale genomic analysis of GBM has demonstrated that this pathway is mutated in many but not all GBMs [27]. Therefore, the AKT1 can be potentially used to define GBM subtypes.

1.6 Conclusion and Discussion

Cancer subtype prediction is crucial in the understanding, diagnosis and treatment of cancer. We introduced a classification model on the basis of multinomial logit regression to identify cancer subtypes from high-throughput gene expression data. The model incorporates a group LASSO penalty and a network-constraint. The group LASSO penalizes all coefficients linked to a predictor at a group level so that it facilitates variable selection at the group level. In addition, the network constraint improves the smoothness of coefficients with respect to the prior structure information and results in more interpretable identification of genes and subnetworks.

The proposed model and its adaptive extension are compared to LASSO and group LASSO multinomial logit model without involving network constraint. From the results of simulation and the application to GBM gene expression data, the proposed model is superior given correct prior network information and are comparable to traditional models given incorrect network information.

A key challenge to the future study is to correctly specify the networks. In the application to real data, we may include too many misspecified edges on the network due to incomplete knowledge of pathways. One possible solution is to use problem-specific network for a particular type of cancer, rather than using a general molecular interaction network. Also, it is important to identify the proper pathway databases to use. For example, KEGG is more accurate because the entries are entered manually rather than discerned automatically

from publications.

The proposed method can be extended by using a non-convex sparsity penalty to reduce estimation bias. SCAD (Smoothly Clipped Absolute Deviation) and MCP (Minimax concave penalty) are two potential alternatives [28–30].

Chapter 2

Bayesian Sparse Models with Probabilistic Programming

T. Park and G. Casella proposed that linear LASSO can be interpreted as a Bayesian posterior mode estimate when the regression parameters have independent Laplace priors [31]. Motived by their work, researchers have discovered even more connections between penalized feature selection models and Bayesian hierarchical structures [32, 33]. Converting a frequentist model into Bayesian models has plenty of benefits. First of all, a probabilistic problem can always be solved by sampling schemes, at the cost of heavy computation though, no matter how complex the penalty terms are. On the other hand, to fit a regularized linear regression by Gradient descent algorithm or its varieties is not promising because it may arrive at a local optimum in case of the non-convexity in loss functions. Another advantage of the Bayesian model lies in its powerful capabilities in inferences. Rather than using point estimates and p values, Bayesian inferences deduce more informative statistics from posterior draws, such as credible intervals and density functions. Furthermore, Bayesian structure is intuitive to construct and to modify, which makes it convenient

to be generalized into more complex ones, such as from linear regression to logistic regression and survival analysis.

In the Bayesian linear regression, a proper prior distribution will enroll shrinkage in parameter estimation. Besides Laplace, some other distributions are also proposed to have good properties in certain specific situations. In this chapter, we will implement several Bayesian models to compare the performance of priors in a variety of scenarios. The comparison will be carried out in feature selection, estimation and prediction with simulated data sets.

2.1 Bayesian Sparse Models

2.1.1 Bayesian Data Analysis

Bayesian data analysis is typically making inferences from data using probability models for quantities we observe and parameters we wish to learn [34]. In Bayesian data analysis, the posterior distribution of a parameter provides us with interval estimates, poterior predictive distribution and some other inference schemes. It also sheds light on feature selection problem. Different from frequentist methods, there are two separate steps to achieve feature selection in Bayesian models - prior selection and feature subset selection.

2.1.2 Priors of Coefficients in Sparse Models

Penalized regression models for simultaneous variable selection and coefficient estimation, such as LASSO and its variations, have received a great deal of attention in the past decades. In Bayesian data analysis, most versions of

LASSO models can be developed into hierarchical Bayesian models by specifying certain priors to the coefficient parameters [31, 32].

Laplace Prior

LASSO algorithm estimates the coefficients of linear regression through L_1-norm penalized least squares:

$$\hat{\beta} = \arg\min_{\beta} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right)^T \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right) + \lambda \sum_{j=1}^{p} \left|\beta_j\right|. \tag{2.1}$$

Tibshirani suggested that LASSO estimates can be interpreted as posterior estimates when the regression parameters have independent and identical double-exponential (Laplace) priors [35].

The probability density function of Laplace distribution with location μ and scale τ is given by

$$\mathit{Laplace}\left(x; \mu, b\right) = \frac{1}{2\tau} \exp\left(-\frac{|x-\mu|}{\tau}\right). \tag{2.2}$$

This prior can better accommodate large regression coefficients due to its heavier tails than standard normal.

According to (2.2), the penalty term in (2.1), $\lambda \sum_{j=1}^{p} |\beta_j|$, can be considered as the summation of the absolute log-likelihood of the parameters $\beta_j, j = 1, \ldots, p$, which follow independent and identical Laplace priors with zero mean. Thus linear LASSO is equivalent to a Bayesian model with a

conditional Laplace prior for the coefficients, that is,

$$f(\beta_j) = \frac{1}{2\tau} \exp\left(-\frac{|\beta_j|}{\tau}\right) \tag{2.3}$$

with $\tau = 1/\lambda$.

Motivated by this connection, Park and Casella proposed the Bayesian LASSO using a conditional Laplace prior specification and the non-informative scale-invariant marginal prior [31].

$$\begin{aligned}
\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2 &\sim N_n\left(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n\right) \\
\boldsymbol{\beta}|\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim N_p\left(0_p, \sigma^2 \boldsymbol{D}_\tau\right) \\
\boldsymbol{D}_\tau &= diag\left(\tau_1^2, \ldots, \tau_p^2\right) \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \pi\left(\sigma^2\right) d\sigma^2 \prod_{j=1}^{p} \frac{\lambda^2}{2} \exp\left(-\frac{1}{2}\lambda^2 \tau_j^2\right) d\tau_j^2 \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &> 0
\end{aligned} \tag{2.4}$$

After integrating out $\tau_1^2, \ldots, \tau_p^2$, the conditional prior on $\boldsymbol{\beta}$ has the desired form (2.3). Note that the prior distribution in model (2.4) is equivalent to Laplace priors (2.3) due to the scale mixture representation of the Laplace

distribution with normal and exponential density [31], that is,

$$\beta_j|\alpha \sim Laplace\left(0, \alpha^{-1/2}\right)$$

is equivalent to

$$\begin{aligned} \beta_j &\sim Normal\left(0, \tau_j^2\right) \\ \tau_j^2 &\sim exponential\left(\frac{\alpha}{2}\right). \end{aligned} \tag{2.5}$$

In the following, we verify the equivalence (2.5). In this process, we will utilize the equation

$$\frac{\sqrt{\alpha}}{2}\exp\left(-\sqrt{\alpha}\,|z|\right) = \int_0^\infty \frac{1}{\sqrt{2\pi s}}\exp\left(-\frac{z^2}{2s}\right)\frac{\alpha}{2}\exp\left(-\frac{\alpha s}{2}\right)ds,$$

as well as substitute the reciprocal of the precision of β_j, denoted as $\frac{1}{t_j}$, for τ_j^2.

$$\begin{aligned}
P\left(\beta_j|\alpha\right) &= \int P\left(\beta_j|\tau_j^2\right) P\left(\tau_j^2|\alpha\right) d\tau_j^2 \\
&= \int \frac{1}{\sqrt{2\pi\tau_j^2}} \exp\left(-\frac{\beta_j^2}{2\tau_j^2}\right) \frac{\alpha}{2} \exp\left(-\frac{\alpha\tau_j^2}{2}\right) d\tau_j^2 \\
&= \int \sqrt{\frac{t_j}{2\pi}} \exp\left(-\frac{\beta_j^2}{2} t_j\right) \frac{\alpha}{2} \exp\left(-\frac{\alpha}{2}\frac{1}{t_j}\right) d\frac{1}{t_j} \\
&= \frac{\alpha}{2} \int \sqrt{\frac{1}{2\pi t_j^3}} \exp\left(-\frac{\beta_j^2}{2t_j}\left(t_j^2 + \frac{\alpha}{\beta_j^2}\right)\right) dt_j \\
&= \frac{\alpha}{2} \int \sqrt{\frac{1}{2\pi t_j^3}} \exp\left(-\frac{\beta_j^2}{2t_j}\left(\left(t_j - \sqrt{\frac{\alpha}{\beta_j^2}}\right)^2 + 2t_j\sqrt{\frac{\alpha}{\beta_j^2}}\right)\right) dt_j \\
&= \frac{\alpha}{2} \int \sqrt{\frac{1}{2\pi t_j^3}} \exp\left(-\frac{\beta_j^2}{2t_j}\left(t_j - \sqrt{\frac{\alpha}{\beta_j^2}}\right)^2\right) \exp\left(-\sqrt{\alpha\beta_j^2}\right) dt_j \\
&= \frac{\sqrt{\alpha}}{2} \exp\left(-\sqrt{\alpha}\,|\beta_j|\right) \\
&= \textit{Laplace}\left(0, \alpha^{-1/2}\right).
\end{aligned}$$

More connections between LASSO-family penalized regressions and fully Bayesian formulation were built by Kyung et. al. [32]. For instance, the conditional prior for group LASSO can be written as

$$\pi\left(\beta|\sigma^2\right) \propto \exp\left(-\frac{\lambda}{\sigma}\sum_{k=1}^{K}\|\beta_{G_k}\|\right),$$

where K is the number of groups, β_{G_k} is the vector of β in group k. Similarly,

the conditional prior for the fused LASSO is

$$\pi\left(\beta|\sigma^2\right) \propto \exp\left(-\frac{\lambda_1}{\sigma}\sum_{j=1}^{p}|\beta_j| - \frac{\lambda_2}{\sigma}\sum_{j=1}^{p-1}|\beta_{j+1}-\beta_j|\right),$$

and that for elastic net is given by

$$\pi\left(\beta|\sigma^2\right) \propto \exp\left(-\frac{\lambda_1}{\sigma}\sum_{j=1}^{p}|\beta_j| - \frac{\lambda_2}{2\sigma}\sum_{j=1}^{p}\beta_j^2\right).$$

Compared with penalized linear regression models, the advantage of the hierarchical Bayesian formulations are huge. In addition to the usual ease-of-interpretation of hierarchical models, the Bayesian formulation produces valid standard errors, and is based on a geometrically ergodic Markov chain. In addition, the results from the Bayesian LASSO are confirmed to be similar to those from the ordinary LASSO [31, 32].

Horseshoe Prior

Besides Laplace prior, horseshoe prior is another shrinkage or sparsity-promoting priors for regression coefficients [36], defined as

$$\begin{aligned}
&\boldsymbol{\beta} \sim N_p\left(0_p, \sigma^2\boldsymbol{D}_\tau\right), \text{ where } \boldsymbol{D}_\tau = diag\left(\tau_1^2, \ldots, \tau_p^2\right)\\
&\tau_j \sim Half - Cauchy\left(0, 1\right), \quad \text{for all } j.
\end{aligned}$$

Half-Cauchy distribution is a special case of half-t distribution when the degree of freedom is one. While half-t distribution is also a special case of fold-noncentral-t distributions with zero mean. Particularly, the fold-noncentral-t

distributions are derived from the non-central Student's t-distribution by taking the values greater than the location parameter, whose probability density function with ν degrees of freedom is given by

$$f\left(x|\mu,\sigma,\nu\right)=\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi\sigma^{2}}}\left\{\left[1+\frac{1}{\nu}\frac{\left(x-\mu\right)^{2}}{\sigma^{2}}\right]^{-\frac{\nu+1}{2}}+\left[1+\frac{1}{\nu}\frac{\left(x+\mu\right)^{2}}{\sigma^{2}}\right]^{-\frac{\nu+1}{2}}\right\}\quad(\text{for } x\geq 0),$$

where μ is the location parameter [3].

From the conditionally-conjugate point of view, the half-t distributions can be considered as the absolute value of a normal random variable, divided by the square root of a gamma random variable [3, 37].

The folded non-central t distribution is not commonly used in statistics, but it has some appealing properties. By restricting the prior mean to zero, so that the folded non-central t distribution becomes simply a half-t distribution. We can parameterize this in terms of scale A and degrees of freedom ν

$$p\left(\tau\right)\propto\left(1+\frac{1}{\nu}\left(\frac{\tau}{A}\right)^{2}\right)^{-\frac{\nu+1}{2}}.$$

This family includes, as special case, the improper uniform density (when $\nu=-1$) and the the proper half-Cauchy, $p\left(\tau\right)\propto\left(\tau^{2}+\sigma^{2}\right)^{-1}$ (when $\nu=1$) [37, 38].

A study comparing the Laplace and horseshoe prior pointed out that the Laplace prior may overshrink large coefficients in a sparse situation, while the horseshoe prior is more robust [36,39]. Also, SL Van Der Pas et. al. concluded that the posterior distribution under the horseshoe prior may be more infor-

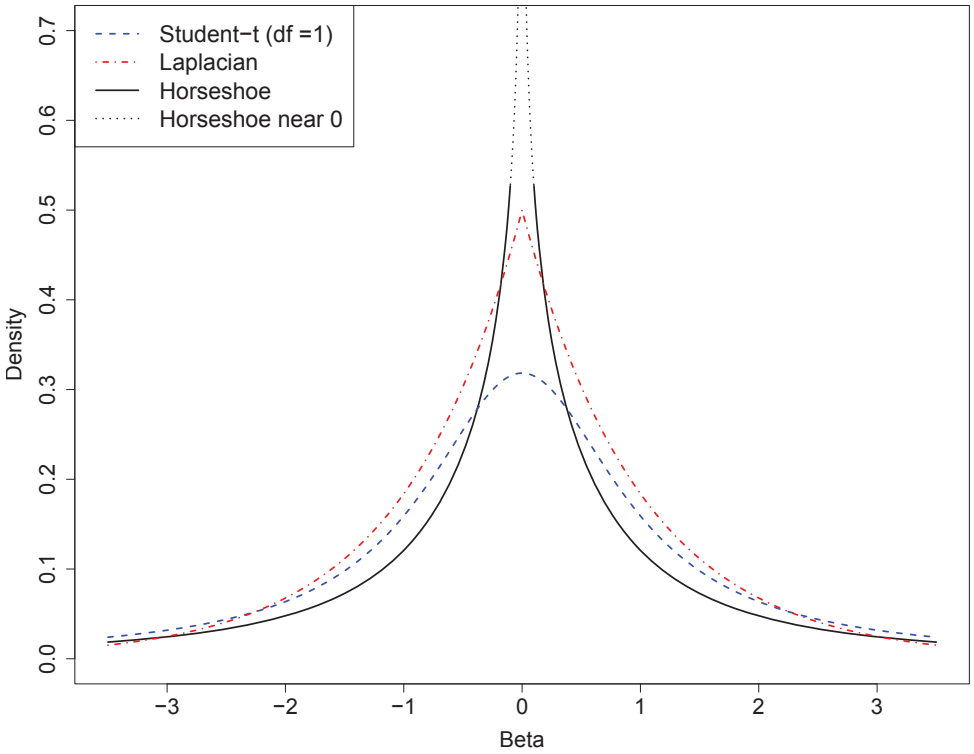

Figure 2.1. The probability density functions of horseshoe prior and two close cousins: Laplace and Cauchy (Student-t df=1).

mative than under the Laplace prior in a sparse normal means problem [40]. In this Chapter, we will compare their abbilities in picking up signals in multiple scenarios.

The density functions of horseshoe, Laplace and Cauchy are displayed in Figure 2.1 [36].

Gaussian Prior

As known, ridge regression is equivalent to a hierarchical model with normal prior for coefficients. According to the notation of (2.4), the posterior distri-

bution of β can be written as

$$\begin{aligned} p\left(\beta|y, \boldsymbol{X}, \sigma^2, \tau\right) &\propto Normal\left(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2\right) Normal\left(\beta|\sigma^2, \tau\right) \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^2\right\} \exp\left\{-\frac{\beta^2}{2\sigma^2\tau^2}\right\}. \end{aligned}$$

Thus the maximum a posteriori (MAP) estimate of β has a ridge format

$$\begin{aligned} \hat{\beta} &= \arg\max_{\beta} p\left(\beta|y, \boldsymbol{X}, \sigma^2, \tau\right) \\ &= \arg\min_{\beta} \left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^2 + \lambda\boldsymbol{\beta}^2, \end{aligned}$$

where $\lambda \propto \frac{1}{\tau^2}$ and τ is a constant. Normal prior will be used as a baseline to compare with in our study.

2.1.3 Estimation of Tuning Parameters

In the implementation of LASSO, cross validation is mostly applied to search for a proper value for the coefficients of penalty terms in loss functions, that is λ in (2.1). This parameter controls the size of the final model, known as the tuning parameter.

In Bayesian analysis, there are two options for the estimation of the tuning parameter λ. One is to estimate the marginal MLE using an Gibbs sampler. Another is to introduce a diffuse hyperprior on λ^2, adopted by Park et. al. [31, 41],

$$\pi\left(\lambda^2\right) = \frac{\delta^r}{\Gamma\left(r\right)}\left(\lambda^2\right)^{r-1} \exp\left(-\delta\lambda^2\right).$$

Gamma hyperprior will result in an easy extension of the Gibbs sampler due

to conjugacy. The choose of the shape parameter and the scale parameter in the Gamma distribution should ensure that the prior density approach zero sufficiently fast as $\lambda^2 \to \infty$ and should be relatively flat and place high probability near the maximum likelihood estimate. Since the n-fold cross validation is usually unstable, the Bayesian estimation of λ often leads to a more stable estimates.

2.2 Implementation of Bayesian Models

The Bayesian framework can be naturally illustrated with hierarchical probabilistic models. Unfortunately, computations in Bayesian framework are intractable even for very simple cases. Most approximation techniques fall into two categories: Markov chain Monte Carlo (MCMC) methods, such as Gibbs sampling, as well as large sample methods, such as Laplace approximation [42]. MCMC algorithms have facilitated an explosion of interest in Bayesian methods by achieving exact results. It is an incredibly useful and important tool but typically requires huge computational resources when used to estimate complex posteriors applied to large data sets, so as to become inefficient for complex models in high data dimensions. Whereas large sample methods are tractable but typically make a rough approximation by building the posteriors over all parameters as Normal. Moreover, they require the computation of the Hessian matrix, which is computationally expensive [43]. In this study, all the Bayesian models are implemented in R package *rstan* with Hamiltonian Monte Carlo.

2.2.1 Algorithms of Bayesian Models

Markov chain Monte Carlo

In 1907, A. A. Markov proposed an important new type of stochastic process. This process is memoryless, meaning that the future states of the process based solely on its present state. This type of process is called a Markov chain [44,45].

Markov chain Monte Carlo (MCMC) methods were developed for cases in which direct sampling is difficult [46]. It samples from a probability distribution based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. After certain "warm-up" steps, the state of the chain can be considered as a sample of the desired distribution. The Markov chains generated by Stan and other MCMC samplers are both ergodic, meaning that any collection of random samples from a process must represent the average statistical properties of the entire process, and stationary, meaning that the transition probabilities do not change at different positions in the chain.

Gibbs sampling is a widely used MCMC algorithm. The point is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. It is a randomized algorithm, and is commonly used as a means of Bayesian inference. BUGS (Bayesian inference Using Gibbs Sampling) and JAGS (Just Another Gibbs Sampling) are both programs for analyzing Bayesian graphical models via Gibbs sampling [47–49].

Hamiltonian Monte Carlo

Despite its advatages in convenient implementation and easy interpretation, MCMC also suffers from several drawbacks. For instance, Gibbs sampling produces highly correlated posteriors, which cannot be addressed even with an efficient and scalable implementation [50]. Stan applies Hamiltonian Monte Carlo (HMC) algorithm, which reduces the correlation between successive sampled states via a Hamiltonian evolution between states and additionally by targeting states with a higher acceptance criteria. But the Hamiltonian dynamics simulation requires the gradient of the log posterior which is unlikely to implement by programs. In order to compute the analytic gradient automatically, reverse-mode algorithmic differentiation is adopted, which allows the computation of the log posterior in only a few multiples of the cost to evaluate the log probability function itself [51]. Here is a simple explanation of reverse-mode automatic differentiation.

- Forward-Prop.

Any differentiable algorithm can be translated into a sequence of assignments of basic operations.

$$x_i \leftarrow f_i\left(\mathbf{x}_{\pi(i)}\right), i = n+1, n+2, ..., N$$

Here, each function f_i is some very basic operation (e.g. addition, multiplication, a logarithm) and $\pi(i)$ denotes the set of "parents" of x_i. So, for example, if $\pi(7) = (2, 5)$ and $f_7 = \text{add}$, then $x_7 = x_2 + x_5$.

Given an algorithm in the previous format, it is easy to compute its derivatives. The essential point here is just the application of the chain rule.

$$\frac{dx_N}{dx_i} = \sum_{j:i\in\pi(j)} \frac{dx_N}{dx_j}\frac{\partial x_j}{\partial x_i}$$

- Back-Prop.

$$\frac{dx_N}{dx_N} \leftarrow 1$$
$$\frac{dx_N}{dx_i} \leftarrow \sum_{j:i\in\pi(j)} \frac{dx_N}{dx_j}\frac{\partial f_j}{\partial x_i}, i = N-1, N-2, ..., 1$$

By creating an expression graph representation of the algorithm, all the derivatives can be computed in reverse order.

2.2.2 Irregular Priors

As discussed in Section 2.1.2, several prior distributions can be assigned to the coefficient parameters to acchieve sparseness. The priors of other parameters, such as scale parameters or hyperparameters, will be discussed in this section.

Improper Uniform Priors

According to Bayes's theorem,

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)},$$

the posterior distribution $P(A_i|B)$ does not change if all prior probabilities $P(A_i)$ were multiplied by a constant, which means, the posterior probabilities

will sum (or integrate) to one even if the prior does not. Taking this idea further, the prior may not even need to be finite to get sensible values for the posterior probabilities. This type of priors is called improper priors.

By default, Stan provides uniform prior on parameters over their legal values. If a parameter is not constrained, a uniform prior on $(-\infty, \infty)$ or $(0, \infty)$ is given, which is an improper uniform prior. Both of these priors are improper in the sense that there is no way to formulate a density function for them that integrates to 1 over its support. Stan allows models to be formulated with improper priors, but in order for sampling or optimization to work, the data provided must ensure a proper posterior. An imporper prior is useful as a starting point for inference and as a baseline for sensitivity analysis [50].

Truncated Priors

If a variable is declared with a lower bound of zero, then assigning it a normal prior in a Stan model produces the same effect as providing a properly truncated half-normal prior. The truncation at zero need not be specified because Stan only requires the density up to a proportion. So declaring the limit of a parameter

```
real<lower=0> sigma;
```

along with a normal prior

```
sigma ~ normal(0, 1000);
```

leads to a half-normal prior, technically

$$p(\sigma) = \frac{Normal(\sigma|0, 1000)}{1 - NormalCDF(0|0, 1000)} \propto Normal(\sigma|0, 1000).$$

Weakly Informative Priors

We characterize a prior distribution as weakly informative if it is proper but is set up so that the information it does provide is intentionally weaker than whatever actual prior knowledge is available [37]. Typically any problem has some natural constraints that would allow a weakly-informative model. An example could be to estimate the mean population height. On the basis of common sense, it should be a value within one to three meter range, that gives us information around which to form a weakly informative prior [50].

Weakly informative priors are recommended due to their abilities to control inference statistically and computationally. Statistically, a weakly informative, or widely spreading, prior is more sensible, because it allows the majority of the prior probability mass fall outside the the expactation region, which can overwhelm the inferences from a small data set. Computationally, a prior increases the curvature around the volume where the solution is expected to lie, which in turn guides Monte Carlo sampling by restricting them within the local [50].

2.2.3 Convergence of MCMC

Samples in a Markov chain only represent the underlying distribution after the chain has converged to its equilibrium. In the implementation of MCMC,

a constant problem is to decide when it is safe to terminate sampling and conclude convergence. Most researchers apply diagnostic tools to deal with the convergence problem, which is summarized by Cowles et. al. [52].

The recommended method for Stan is to run multiple Markov chains, initialized randomly with a diffuse set of initial parameter values, discard the warm-up/adaptation samples, then split the remainder of each chain in half and compute the potential scale reduction statistic, $\hat{R}$ [53].

Potential Scale Reduction The potential scale reduction statistic $\hat{R}$ measures the ratio of the average variance of samples within each chain to the variance of the pooled samples across chains; if all chains are at equilibrium, these will be the same and $\hat{R}$ will be one. If the chains have not converged to a common distribution, the $\hat{R}$ statistic will be greater than one.

The definition of $\hat{R}$ statistic is defined for a set of M Markov chains, $\theta_m, m = 1, 2, \ldots, M$, each of which has N samples $\theta_m^{(n)}, n = 1, 2, \ldots, N$. Accordingly, the between-sample B and within-sample W variance estimates are given below.

$$B = \frac{N}{M-1} \sum_{m=1}^{M} \left(\bar{\theta}_m^{(\cdot)} - \bar{\theta}_\cdot^{(\cdot)} \right)^2,$$

$$W = \frac{1}{M} \sum_{m=1}^{M} s_m^2,$$

where

$$\bar{\theta}_m^{(\cdot)} = \frac{1}{N}\sum_{n=1}^{N}\theta_m^{(n)}$$
$$\bar{\theta}_{\cdot}^{(\cdot)} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_m^{(\cdot)}$$
$$s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}\left(\theta_m^{(n)} - \bar{\theta}_m^{(\cdot)}\right)^2.$$

The variance estimator is

$$\hat{var}^{+}(\theta|y) = \frac{N-1}{N}W + \frac{1}{N}B.$$

Finally, the potential scale reduction statistic is defined by

$$\hat{R} = \sqrt{\frac{\hat{var}^{+}(\theta|y)}{W}}.$$

Note that $\hat{R}$ statistic makes very strong assumptions that the related functions are Gaussian or only first two moments are considered. As a result, it may not work for all functions equally well. In this study, we check the distribution of $\hat{R}$ to ensure convergence before actually analyzing the results.

Effective Sample Size Another technical difficulty posed by MCMC methods is the autocorrelation within a chain. It increases the uncertainty of the estimation of posterior quantities of interest, such as means, variances or quantiles; and this uncertainty can be measured by effective sample size (ESS). According to Central Limit Theorem, by estimating the mean of M indepen-

dent draws rather than the raw samples, the estimation error is proportionally reduced to $1/\sqrt{M}$. If the draw are not independent but positive correlative, such as drawing using Markov chain, the error is proportional to $1/\sqrt{N_{eff}}$, where N_{eff} is the effective sample size and $N_{eff} < M$. Thus, it is also standard practice to monitor the ESS till it is large enough for the estimation or inference task [50].

The effective sample size of a sequence is defined in terms of the autocorrelations within the sequence at different lags,

$$N_{eff} = \frac{N}{\sum_{t=-\infty}^{\infty} \rho_t} = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t},$$

where ρ_t is the autocorrelation of the chain at lag $t \geq 0$ and N is the actual sample size.

In order to reduce the error caused by autocorrelation, thinning samples are frequently used in Bayesian sampling method, which means we take a sample every nth value. For instance, there are two ways of generating 1000 samples

1. Generate 1000 samples after convergence and save all of them.

2. Generate 10000 samples after convergence and save every tenth samples.

Even though both produce 1000 samples, the second approach with thinning will produce more effective samples because the autocorrelation of the thinned samples ρ_{10} will be lower than the autocorrelation of ρ_1 so that the effective sample size is higher. Furthermore, saving all 10000 sample without thinning will result in a even higher effective sample size. More analysis on ESS and MCMC has been discussed particularly by C Geyer et. al. [54].

In this studies, we compared the autocorrelation with and without thinning in several model fitting results, only to find that the autocorrelation is negligible even without thinning. As a result, we adopt the default $thin = 1$ in Stan fitting, which means all samples in chain are used for result analysis.

2.2.4 Bayesian Inferences

After obtaining a valid chain of posterior draws, we are capable to estimate the properties of posterior distributions from the random samples. Typically, the point estimates can be the average of all samples; and the interval estimation is achieved by credible intervals.

Credible Interval In Bayesian statistics, a credible interval is an interval in the domain of a posterior probability distribution [55]. Credible intervals are analogous to confidence intervals in frequentist statistics, although they differ on a philosophical basis [56]; Bayesian intervals treat their bounds as fixed and the estimated parameter as a random variable, whereas frequentist confidence intervals treat their bounds as random variables and the parameter as a fixed value. In particular, a 95% confidence interval for parameter β covers the true but unknown value of β with 95% of chance; and a 95% credible interval means the probability of β lying within that interval is 0.95.

Credible Intervals can be created based on the posterior draws of a parameter. For example, the interval limited by 2.5% quantile and 97.5% quantile constitutes an equal-tailed credible interval when the sample size is big enough. T. Park and G. Casella used Bayesian credible interval to guide variable selec-

tion [31]. If the credible interval for a parameter covers 0, there is not enough evidence to conclude this parameter deviates from 0, so that this corresponding variable can be considered insignificant. Moreover, they confirmed its decent performance by finding all of the Lasso estimates are well within the credible intervals. In this Chapter, we will implement feature selection with credible intervals for all Bayesian models. Further discussion on feature selection in Bayesian models will take place in the next Chapter.

2.3 Prior Distribution Comparison in Simulation

2.3.1 Data Simulation

We performed simulation studies to compare the performance of Bayesian models with different prior distributions for coefficients in terms of variable selection, coefficients estimation and prediction.

We simulate a linear relationship between dependent variable y and predictors $\mathbf{x}$ based on the generative model

$$\begin{aligned} x &\sim N\left(0, R\right), R \in \mathbb{R}^{p \times p} \\ y|x &\sim N\left(x\beta, \sigma^2\right). \end{aligned}$$

To compute the variance of y, we use Law of total variance,

$$\begin{aligned} var\left(y\right) &= E\left[var\left(y|x\right)\right] + var\left(E\left[y|x\right]\right) \\ &= \sigma^2 + var\left(x\beta\right). \end{aligned} \tag{2.6}$$

The simulation is designed with group information — variables in the same group are correlated and have equal effect size [57]. The details of the settings are described below.

To begin with, there will be two scenarios and the only difference between them is the sample size, where $N_1 = 100$ and $N_2 = 400$. Next in each scenario, the number of variables is $P = 100$. As mentioned above, the variables are divided into 20 groups, 5 variables in each group. Each variable x_j follows a normal distribution marginally with a zero mean and unit variance, and is correlated with other variables in the same group with coefficient ρ but uncorrelated with variables in the other groups, that is, the correlation matrix is block diagonal. Among the 20 groups of variables, three are generative groups, which means the dependent variable y is derived solely based on them. To set the 15 variables as relevant, we specify them with non-zero coefficients $(\beta^{1:5}, \beta^{6:10}, \beta^{11:15}) = (\zeta, 0.5\zeta, 0.25\zeta)$ whereas the rest have zero weight. At last, the constant ζ is determined adjusting for the signal-to-noise ratio (SNR) of the data. To get comparable results for different levels of correlation ρ, we set ζ so that $\sigma^2/var\,(y) = 0.3$, where the noise variance $\sigma^2 = 1$. According to (2.6), the SNR is

$$\begin{aligned}\frac{\sigma^2_{signal}}{\sigma^2_{noise}} &= \frac{\sigma^2_{signal} + \sigma^2_{noise}}{\sigma^2_{noise}} - 1 \\ &= \frac{var\,(y)}{\sigma^2_{noise}} - 1 \\ &= \frac{1}{0.3} - 1 = 7/3.\end{aligned}$$

For $\rho = 0, 0.5, 0.9$, this is satisfied by setting approximately $\zeta = 0.59, 0.34, 0.28$

Model 1	Double exponential (Laplace) prior
Model 2	Gaussian prior
Model 3	Gaussian-exponential hierarchical structure
Model 4	Gaussian-exponential hierarchical structure, λ^2 follows a Gamma distribution
Model 5	Half-t distribution
Model 6	Half-Cauchy distribution, which is the special case of half-t distribution with one degree of freedom

Table 2.1. The settings of prior distributions for the coefficient parameters in the candidate models for this simulation study.

respectively. The trial is carried out 50 times to adjust for the randomness in simulation and Monte Carlo sampling.

2.3.2 Candidate Priors

In this section, we will discuss the priors for the coefficient parameters β in each model being compared in this simulation study (see table 2.1) and how they are implemented in Stan. A stan program is organized into a sequence of named blocks [50]. Some selected transformed parameters blocks and model blocks will be shown is this document.

- Double Exponential Prior

The coefficients $\beta_1, \ldots, \beta_P$ follow a single double-exponential prior distribution with location parameter μ and scale parameter γ, both of which have improper uniform priors. The model is described in the model block in a stan file as following.

```
model {
```

```
y ~ normal(x * beta, sigma);
sigma ~ normal(0,1);
beta ~ double_exponential(mu, gamma);
for (k in 1:K)
target += - lambda * N * fabs(beta[k]);
}
```

"lambda" controls the magnitude of the penalty term, which follows a improper uniform prior; N is the number of samples. "target" is an embedded variable, representing the log probability accumulator.

In fact, the basic purpose of a Stan program is to compute a log probability function and its derivatives. The log probability function in a Stan model outputs the log density on the unconstrained scale. The variables are first transformed from unconstrained to constrained, and the log Jacobian determinant added to the log probability accumulator. Then the model block is executed on the constrained parameters, with each sampling statement (~). At the end of the model block execution, the value of the log probability accumulator is the log probability value returned by the build-in function target() in Stan program.

- Gaussian prior

The coefficient parameters $\beta_1, \ldots, \beta_P$ follow a single Gaussian prior distribution with mean μ and standard deviation γ, both of which have improper uniform priors. The model is described in the model block in a stan file as following.

```
model {
y ~ normal(x *beta, sigma);
beta ~ normal(mu, gamma);
for (k in 1:K)
target += - lambda * N * fabs(beta[k]);
}
```

- Gaussian-exponential prior

In this model, each β_j has a Gaussian prior with mean μ and standard deviation $\sigma\sqrt{\tau_j^2}$. Both mean μ and the global variance follows improper uniform priors. While the local variance parameter follows an exponential distribution, making a Gaussian-exponential hierarchy for β. The model block is coded as below for this model.

```
model {
y ~ normal(x *beta, sigma);
for (j in 1:K)
beta[j] ~ normal(mu, sigma*sqrt(tau_sq[j]));
tau_sq ~ exponential(lambda_sq * N^2 / 8);
for (k in 1:K)
target += - lambda * N * fabs(beta[k]);
}
```

- Gaussian-exponential prior for coefficients β and Gamma prior for λ^2

This difference between this model and the precious lies in the prior of λ^2. It was an improper uniform prior in model 3 but is a Gamma distribution in Model 4. The model code is given below. Note that the logarithm of the Jacobian determinants of non-linear transformation should be added to target specially.

```
model {
y ~ normal(x *beta, sigma);
for (j in 1:K)
beta[j] ~ normal(mu, sigma*tau[j]);
tau_sq ~ exponential(lambda_sq * N^2 / 8);
lambda_sq ~ gamma(2,50);
for (k in 1:K)
target += - lambda * N * fabs(beta[k]);
target += sum(log(tau));
target += log(lambda);
}
```

- Half-t prior

In this model, the beta is constrained by a global scale parameter τ and a local scale parameter λ, both of which follow half-Cauchy distributions [58]. From the discussion of section 2.1.2, we learnt that half-Cauchy can be considered as the result of the absolute value of a Gaussian variable divided by a Gamma variable, that is the product of the absolute value of a Gaussian variable and a inverse-Gamma variable. This is how these two scale parameters are defined

in the stan model.

```
transformed parameters {
real<lower=0> tau;
vector<lower=0>[K] lambda;
vector[K] beta;


tau = r1_global * sqrt(r2_global);
lambda = r1_local .* sqrt(r2_local);
beta = z .* lambda*tau;
}
model {
y ~ normal(x*beta, sigma);
z ~ normal(0, 1);
r1_local ~ normal(0.0, 1.0);
r2_local ~ inv_gamma(0.5*nu, 0.5*nu);
r1_global ~ normal(0.0, 1.0);
r2_global ~ inv_gamma(0.5, 0.5);
}
```

- Half-Cauchy prior

As discussed in section 2.1.2, half-Cauchy prior is a special case of half-t distribution. Therefore, this model is exactly the same as the previous one except for prespecifying one degree of freedom.

2.3.3 Simulation Results Analysis

Feature Selection

In this study, features are selected based on credible intervals, introduced in Section 2.2.4. Three credible levels are applied, 0.8, 0.95 and 0.99. Of course, in each setting the 0.8 credible level leads to a larger model whereas the 0.99 leads to a smaller model.

Statistical Measures Feature selection accuracy are measured by positive predictive value (PPV), negative predictive value (NPV), true positive rate or sensitivity (SEN), true negative rate or specificity (SPC) and overall accuracy rate (ACC). Those terms are defined by the following formula:

$$\begin{aligned}
\text{PPV} &= \frac{\text{TP}}{\text{TP+FP}} \\
\text{NPV} &= \frac{\text{TN}}{\text{TN+FN}} \\
\text{SEN} &= \frac{\text{TP}}{\text{TP+FN}} \\
\text{SPC} &= \frac{\text{TN}}{\text{TN+FP}} \\
\text{ACC} &= \frac{\text{TN+TP}}{\text{TN+TP+FN+FP}}.
\end{aligned}$$

The terms in the above formula are defined in the table below.

	prediction positive	prediction negative
condition positive	True Positive (TP)	False Negative (FN)
condition negative	False Positive (FP)	True Negative (TN)

In each setting, we repeat the trial for 50 times and the five criteria, including PPV, NPV, SEN, SPC and ACC, are calculated for all trials. The mean and standard error across the 50 trials are given in table 2.2 - 2.7. Some conclusion can be drawn from these tables.

First of all, the low PPV for Model 3 and 4 suggests that they tend to selected more features, especially at the credible level of 0.8. Comparing with Model 1, Model 3 and 4 use one more hierarchy to construct the Laplace distribution, which undoubtedly brings in more uncertainty and thus larger intervals at a certain credible level. Moreover, Model 3 and 4 obtain higher ACC at higher credible levels, which confirms the wide spreading of the posteriors. The hierarchical structure is supposed to accommodate more heteroskedasticity coefficients.

Second, t-distribution-based models, Model 5 and 6, behave most accurately when the variables are independent, especially at the credible level of 0.8. That is to say, they capture weak signals well enough even without borrowing information from other signals. This could be a result of using local variance in the model design, in 2.3.2, which allows parameters estimated separately when they do not behave alike.

Further, Gaussian prior outperforms others when large within-group correlation ρ applies. The high correlation within group leads to a lower overall variance of the data, and thus the lower variance in posterior distributions. Therefore, it is less likely to see unexpected large posterior draws comparing to the case of independent variables. Referring to Figure 2.1, the heavy tail distributions become less appropriate in this situation.

At last, all models behave more alike in case of large sample size $N = 400$, that makes sense because a large amount of samples overwhelm the characteristics in the prior distribution.

AUC Besides the specific measures, we calculate AUC for over all performance of feature selection. Area Under Curve (AUC) refers to the area under the receiver operating characteristic (ROC) curve, which is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the sensitivity against one minus specificity at various threshold settings. The value of AUC is equal to the probability that a classifier, or a selective system in this case, will rank a randomly chosen positive instance higher than a randomly chosen negative one.

AUC values are also compared across all settings. The results are given in table 2.8. As shown, the Laplace prior (Model 1) behaves the best in all settings, but the advantage diminishes as the sample size increases.

As a reference, the ROC curves for all settings can be found in the Appendix in Figure A.1 - A.6.

Parameter Estimation

The parameter estimated is evaluated by the root-sum-square error (RSSE), which is the square root of the L_2- norm of the deviation between the true

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.6(0.15)	0.9(0.13)	0.96(0.09)
	Model2	0.56(0.14)	0.87(0.16)	0.95(0.11)
	Model3	0.32(0.09)	0.47(0.17)	0.62(0.2)
	Model4	0.32(0.09)	0.5(0.19)	0.63(0.19)
	Model5	0.84(0.12)	0.98(0.07)	1(0.02)
	Model6	0.95(0.08)	1(0.02)	1(0)
NPV	Model1	0.94(0.02)	0.91(0.01)	0.89(0.01)
	Model2	0.94(0.02)	0.91(0.02)	0.88(0.02)
	Model3	0.94(0.03)	0.93(0.02)	0.92(0.02)
	Model4	0.94(0.03)	0.93(0.02)	0.92(0.02)
	Model5	0.91(0.01)	0.89(0.01)	0.88(0.01)
	Model6	0.91(0.01)	0.89(0.01)	0.88(0.01)
SEN	Model1	0.68(0.12)	0.44(0.1)	0.29(0.1)
	Model2	0.66(0.12)	0.43(0.12)	0.26(0.13)
	Model3	0.77(0.11)	0.63(0.12)	0.53(0.12)
	Model4	0.76(0.11)	0.63(0.11)	0.51(0.13)
	Model5	0.47(0.1)	0.33(0.08)	0.25(0.09)
	Model6	0.41(0.09)	0.31(0.07)	0.24(0.08)
SPC	Model1	0.91(0.06)	0.99(0.02)	1(0.01)
	Model2	0.9(0.06)	0.98(0.02)	1(0.01)
	Model3	0.67(0.14)	0.83(0.13)	0.91(0.11)
	Model4	0.69(0.14)	0.85(0.13)	0.92(0.11)
	Model5	0.98(0.02)	1(0.01)	1(0)
	Model6	1(0.01)	1(0)	1(0)
ACC	Model1	0.87(0.05)	0.91(0.02)	0.89(0.01)
	Model2	0.86(0.05)	0.9(0.02)	0.89(0.02)
	Model3	0.69(0.12)	0.8(0.11)	0.86(0.09)
	Model4	0.7(0.11)	0.81(0.11)	0.86(0.08)
	Model5	0.9(0.02)	0.9(0.01)	0.89(0.01)
	Model6	0.91(0.01)	0.9(0.01)	0.89(0.01)

Table 2.2. The performance of six models in feature selection when $N = 100$ and $\rho = 0$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.81(0.13)	0.98(0.08)	1(0.03)
	Model2	0.7(0.14)	0.96(0.08)	0.99(0.05)
	Model3	0.32(0.09)	0.49(0.17)	0.71(0.21)
	Model4	0.35(0.1)	0.53(0.16)	0.76(0.2)
	Model5	0.95(0.09)	0.99(0.04)	1(0)
	Model6	0.98(0.06)	0.99(0.04)	1(0)
NPV	Model1	0.92(0.02)	0.89(0.01)	0.87(0.01)
	Model2	0.93(0.01)	0.9(0.01)	0.88(0.01)
	Model3	0.92(0.02)	0.9(0.01)	0.89(0.01)
	Model4	0.92(0.02)	0.9(0.01)	0.89(0.01)
	Model5	0.9(0.01)	0.87(0.01)	0.86(0.01)
	Model6	0.89(0.01)	0.87(0.01)	0.86(0.01)
SEN	Model1	0.53(0.1)	0.3(0.09)	0.14(0.07)
	Model2	0.58(0.08)	0.38(0.08)	0.22(0.08)
	Model3	0.61(0.1)	0.41(0.11)	0.3(0.07)
	Model4	0.61(0.1)	0.4(0.09)	0.28(0.07)
	Model5	0.36(0.08)	0.16(0.06)	0.07(0.05)
	Model6	0.26(0.07)	0.13(0.06)	0.06(0.06)
SPC	Model1	0.97(0.02)	1(0.01)	1(0)
	Model2	0.95(0.03)	1(0.01)	1(0)
	Model3	0.75(0.1)	0.9(0.08)	0.97(0.05)
	Model4	0.78(0.09)	0.92(0.06)	0.98(0.03)
	Model5	1(0.01)	1(0)	1(0)
	Model6	1(0.01)	1(0)	1(0)
ACC	Model1	0.91(0.02)	0.89(0.01)	0.87(0.01)
	Model2	0.9(0.03)	0.9(0.01)	0.88(0.01)
	Model3	0.73(0.08)	0.83(0.06)	0.87(0.04)
	Model4	0.75(0.08)	0.84(0.05)	0.87(0.03)
	Model5	0.9(0.01)	0.87(0.01)	0.86(0.01)
	Model6	0.89(0.01)	0.87(0.01)	0.86(0.01)

Table 2.3. The performance of six models in feature selection when $N = 100$ and $\rho = 0.5$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.99(0.05)	1(0)	1(0)
	Model2	0.98(0.06)	1(0)	1(0)
	Model3	0.48(0.21)	0.75(0.28)	0.85(0.29)
	Model4	0.58(0.23)	0.83(0.25)	0.89(0.28)
	Model5	1(0)	1(0)	1(0)
	Model6	1(0)	1(0)	1(0)
NPV	Model1	0.89(0.01)	0.86(0)	0.85(0)
	Model2	0.91(0.01)	0.87(0.01)	0.85(0)
	Model3	0.89(0.01)	0.87(0.01)	0.86(0.01)
	Model4	0.89(0.01)	0.87(0.01)	0.86(0.01)
	Model5	0.86(0.01)	0.85(0)	0.85(0)
	Model6	0.86(0.01)	0.85(0)	0.85(0)
SEN	Model1	0.3(0.09)	0.05(0.04)	0.01(0.02)
	Model2	0.41(0.06)	0.19(0.06)	0.03(0.04)
	Model3	0.34(0.09)	0.14(0.06)	0.06(0.06)
	Model4	0.34(0.08)	0.13(0.05)	0.06(0.06)
	Model5	0.09(0.05)	0.02(0.03)	0.01(0.02)
	Model6	0.06(0.05)	0.02(0.03)	0.01(0.02)
SPC	Model1	1(0)	1(0)	1(0)
	Model2	1(0)	1(0)	1(0)
	Model3	0.9(0.09)	0.98(0.05)	0.99(0.02)
	Model4	0.93(0.07)	0.99(0.03)	1(0.01)
	Model5	1(0)	1(0)	1(0)
	Model6	1(0)	1(0)	1(0)
ACC	Model1	0.89(0.01)	0.86(0.01)	0.85(0)
	Model2	0.91(0.01)	0.88(0.01)	0.85(0.01)
	Model3	0.82(0.07)	0.85(0.04)	0.85(0.01)
	Model4	0.84(0.06)	0.86(0.02)	0.86(0.01)
	Model5	0.86(0.01)	0.85(0)	0.85(0)
	Model6	0.86(0.01)	0.85(0.01)	0.85(0)

Table 2.4. The performance of six models in feature selection when $N = 100$ and $\rho = 0.9$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.58(0.09)	0.88(0.08)	0.98(0.04)
	Model2	0.5(0.07)	0.81(0.09)	0.96(0.05)
	Model3	0.54(0.08)	0.82(0.09)	0.95(0.05)
	Model4	0.55(0.07)	0.82(0.09)	0.97(0.05)
	Model5	0.78(0.09)	0.96(0.05)	0.99(0.03)
	Model6	0.9(0.08)	0.98(0.03)	1(0.02)
NPV	Model1	0.99(0.01)	0.98(0.01)	0.97(0.01)
	Model2	0.99(0.01)	0.98(0.01)	0.97(0.01)
	Model3	0.99(0.01)	0.98(0.01)	0.97(0.01)
	Model4	0.99(0.01)	0.98(0.01)	0.97(0.01)
	Model5	0.98(0.01)	0.97(0.01)	0.96(0.01)
	Model6	0.98(0.01)	0.96(0.01)	0.95(0.01)
SEN	Model1	0.96(0.05)	0.9(0.08)	0.8(0.08)
	Model2	0.96(0.05)	0.91(0.07)	0.84(0.08)
	Model3	0.96(0.05)	0.9(0.08)	0.83(0.08)
	Model4	0.96(0.05)	0.91(0.07)	0.83(0.08)
	Model5	0.91(0.07)	0.83(0.08)	0.75(0.07)
	Model6	0.88(0.08)	0.79(0.08)	0.72(0.06)
SPC	Model1	0.87(0.04)	0.98(0.02)	1(0.01)
	Model2	0.82(0.05)	0.96(0.02)	0.99(0.01)
	Model3	0.85(0.04)	0.96(0.02)	0.99(0.01)
	Model4	0.85(0.04)	0.96(0.02)	0.99(0.01)
	Model5	0.95(0.03)	0.99(0.01)	1(0)
	Model6	0.98(0.02)	1(0)	1(0)
ACC	Model1	0.88(0.04)	0.96(0.02)	0.97(0.01)
	Model2	0.85(0.04)	0.95(0.02)	0.97(0.01)
	Model3	0.87(0.04)	0.95(0.02)	0.97(0.01)
	Model4	0.87(0.04)	0.95(0.02)	0.97(0.01)
	Model5	0.94(0.02)	0.97(0.01)	0.96(0.01)
	Model6	0.97(0.02)	0.97(0.01)	0.96(0.01)

Table 2.5. The performance of six models in feature selection when $N = 400$ and $\rho = 0$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.69(0.11)	0.94(0.06)	0.99(0.03)
	Model2	0.56(0.09)	0.86(0.1)	0.98(0.04)
	Model3	0.61(0.11)	0.88(0.08)	0.98(0.04)
	Model4	0.62(0.11)	0.89(0.1)	0.99(0.03)
	Model5	0.87(0.09)	0.99(0.04)	1(0)
	Model6	0.95(0.06)	0.99(0.02)	1(0)
NPV	Model1	0.97(0.01)	0.94(0.01)	0.92(0.01)
	Model2	0.97(0.01)	0.95(0.01)	0.93(0.01)
	Model3	0.97(0.01)	0.94(0.01)	0.93(0.01)
	Model4	0.97(0.01)	0.94(0.01)	0.93(0.01)
	Model5	0.96(0.01)	0.93(0.01)	0.91(0.01)
	Model6	0.95(0.01)	0.92(0.01)	0.91(0.01)
SEN	Model1	0.84(0.07)	0.67(0.08)	0.54(0.07)
	Model2	0.84(0.06)	0.69(0.07)	0.57(0.07)
	Model3	0.83(0.07)	0.67(0.07)	0.55(0.06)
	Model4	0.83(0.07)	0.67(0.07)	0.56(0.07)
	Model5	0.74(0.06)	0.57(0.07)	0.46(0.06)
	Model6	0.68(0.06)	0.53(0.07)	0.41(0.07)
SPC	Model1	0.93(0.03)	0.99(0.01)	1(0)
	Model2	0.88(0.04)	0.98(0.02)	1(0)
	Model3	0.9(0.04)	0.98(0.01)	1(0)
	Model4	0.9(0.04)	0.98(0.02)	1(0)
	Model5	0.98(0.02)	1(0)	1(0)
	Model6	0.99(0.01)	1(0)	1(0)
ACC	Model1	0.92(0.03)	0.94(0.01)	0.93(0.01)
	Model2	0.87(0.04)	0.94(0.02)	0.93(0.01)
	Model3	0.89(0.04)	0.94(0.02)	0.93(0.01)
	Model4	0.89(0.04)	0.94(0.02)	0.93(0.01)
	Model5	0.94(0.02)	0.93(0.01)	0.92(0.01)
	Model6	0.95(0.01)	0.93(0.01)	0.91(0.01)

Table 2.6. The performance of six models in feature selection when $N = 400$ and $\rho = 0.5$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
PPV	Model1	0.87(0.11)	1(0.03)	1(0)
	Model2	0.75(0.14)	0.96(0.08)	1(0.04)
	Model3	0.79(0.15)	0.96(0.08)	1(0.04)
	Model4	0.82(0.14)	0.98(0.06)	1(0.04)
	Model5	0.98(0.05)	0.99(0.04)	1(0)
	Model6	1(0.02)	1(0)	1(0)
NPV	Model1	0.93(0.01)	0.89(0.01)	0.87(0.01)
	Model2	0.94(0.01)	0.91(0.01)	0.89(0.01)
	Model3	0.92(0.01)	0.89(0.01)	0.87(0.01)
	Model4	0.92(0.01)	0.89(0.01)	0.87(0.01)
	Model5	0.9(0.01)	0.87(0.01)	0.86(0.01)
	Model6	0.88(0.01)	0.86(0.01)	0.86(0.01)
SEN	Model1	0.57(0.09)	0.3(0.08)	0.13(0.04)
	Model2	0.62(0.08)	0.42(0.07)	0.27(0.07)
	Model3	0.55(0.08)	0.31(0.08)	0.16(0.06)
	Model4	0.55(0.08)	0.32(0.08)	0.15(0.05)
	Model5	0.35(0.09)	0.14(0.05)	0.06(0.04)
	Model6	0.23(0.05)	0.1(0.05)	0.05(0.05)
SPC	Model1	0.98(0.02)	1(0)	1(0)
	Model2	0.96(0.03)	1(0.01)	1(0)
	Model3	0.97(0.03)	1(0)	1(0)
	Model4	0.98(0.02)	1(0)	1(0)
	Model5	1(0)	1(0)	1(0)
	Model6	1(0)	1(0)	1(0)
ACC	Model1	0.92(0.02)	0.9(0.01)	0.87(0.01)
	Model2	0.91(0.03)	0.91(0.01)	0.89(0.01)
	Model3	0.91(0.03)	0.89(0.01)	0.87(0.01)
	Model4	0.91(0.03)	0.9(0.01)	0.87(0.01)
	Model5	0.9(0.01)	0.87(0.01)	0.86(0.01)
	Model6	0.88(0.01)	0.87(0.01)	0.86(0.01)

Table 2.7. The performance of six models in feature selection when $N = 100$ and $\rho = 0.9$. Each cell displays the mean and standard error across 50 trials in each setting of credible level .

		$\rho = 0$	$\rho = 0.5$	$\rho = 0.9$
$N = 100$	Model1	0.88(0.06)	0.87(0.04)	0.91(0.05)
	Model2	0.86(0.06)	0.84(0.05)	0.86(0.05)
	Model3	0.8(0.08)	0.74(0.08)	0.69(0.09)
	Model4	0.8(0.08)	0.75(0.07)	0.72(0.08)
	Model5	0.85(0.05)	0.84(0.05)	0.88(0.05)
	Model6	0.85(0.05)	0.85(0.05)	0.84(0.07)
$N = 400$	Model1	0.98(0.02)	0.95(0.03)	0.92(0.04)
	Model2	0.98(0.02)	0.93(0.03)	0.87(0.04)
	Model3	0.98(0.02)	0.93(0.03)	0.87(0.05)
	Model4	0.98(0.02)	0.94(0.03)	0.88(0.05)
	Model5	0.98(0.02)	0.95(0.03)	0.91(0.05)
	Model6	0.98(0.02)	0.95(0.03)	0.88(0.05)

Table 2.8. The mean and standard error of AUC for the six models in all settings.

value and the estimated value of β, defined below.

$$RSSE = \sqrt{\sum_{j=1}^{P} \left(\beta_j - \hat{\beta}_j\right)^2}.$$

The results are summarized in table 2.9 and 2.10. Generally, the estimation results perform in accordance with feature selection accuracy, so that all conclusions drawn from Section 2.3.3 are verified.

As a reference, RSSE are illustrated in plots for each setting, which locate in the Appendix Figure A.7 - A.12.

Prediction

For each setting, we also simulate a test set with 100 samples $\left\{\left(\tilde{X}_i, \tilde{y}_i\right), i = 1, 2, \ldots, 100\right\}$, on which we compute the prediction root-mean-squared error (PRMSE) with

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
$\rho = 0$	Model1	0.84 (0.12)	0.88 (0.14)	1 (0.18)
	Model2	0.97 (0.1)	0.99 (0.15)	1.13 (0.2)
	Model3	1.27 (0.52)	1.21 (0.53)	1.12 (0.52)
	Model4	1.22 (0.5)	1.15 (0.51)	1.09 (0.49)
	Model5	0.81 (0.13)	0.9 (0.16)	1.04 (0.19)
	Model6	0.81 (0.13)	0.91 (0.15)	1.05 (0.18)
$\rho = 0.5$	Model1	0.52 (0.09)	0.61 (0.1)	0.76 (0.07)
	Model2	0.5 (0.08)	0.5 (0.08)	0.63 (0.1)
	Model3	1.11 (0.36)	1 (0.33)	0.89 (0.28)
	Model4	1 (0.28)	0.91 (0.26)	0.84 (0.21)
	Model5	0.65 (0.11)	0.78 (0.07)	0.84 (0.04)
	Model6	0.75 (0.11)	0.84 (0.07)	0.86 (0.04)
$\rho = 0.9$	Model1	0.52 (0.1)	0.7 (0.04)	0.72 (0.03)
	Model2	0.37 (0.05)	0.55 (0.07)	0.69 (0.03)
	Model3	1.12 (0.61)	0.98 (0.53)	0.89 (0.42)
	Model4	0.94 (0.45)	0.86 (0.35)	0.82 (0.25)
	Model5	0.83 (0.16)	0.81 (0.15)	0.77 (0.13)
	Model6	0.89 (0.19)	0.84 (0.2)	0.79 (0.17)

Table 2.9. The RSSE of parameter estimation in all settings when $N = 100$.

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
$\rho = 0$	Model1	0.37 (0.04)	0.32 (0.05)	0.34 (0.06)
	Model2	0.45 (0.05)	0.37 (0.05)	0.35 (0.05)
	Model3	0.38 (0.04)	0.33 (0.05)	0.33 (0.05)
	Model4	0.38 (0.04)	0.33 (0.05)	0.32 (0.05)
	Model5	0.32 (0.05)	0.31 (0.05)	0.35 (0.05)
	Model6	0.3 (0.05)	0.33 (0.05)	0.37 (0.05)
$\rho = 0.5$	Model1	0.33 (0.05)	0.33 (0.05)	0.37 (0.04)
	Model2	0.38 (0.06)	0.32 (0.06)	0.34 (0.04)
	Model3	0.37 (0.06)	0.35 (0.05)	0.37 (0.03)
	Model4	0.36 (0.06)	0.35 (0.05)	0.37 (0.04)
	Model5	0.34 (0.05)	0.38 (0.04)	0.43 (0.05)
	Model6	0.36 (0.05)	0.42 (0.05)	0.48 (0.07)
$\rho = 0.9$	Model1	0.43 (0.08)	0.53 (0.08)	0.64 (0.04)
	Model2	0.39 (0.09)	0.38 (0.07)	0.47 (0.07)
	Model3	0.49 (0.1)	0.55 (0.08)	0.63 (0.05)
	Model4	0.47 (0.1)	0.54 (0.08)	0.63 (0.05)
	Model5	0.62 (0.1)	0.7 (0.07)	0.73 (0.06)
	Model6	0.74 (0.08)	0.77 (0.07)	0.76 (0.07)

Table 2.10. The RSSE of parameter estimation in all settings when $N = 400$.

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
	Model1	1.06 (1.2)	1.06 (1.23)	1.11 (1.27)
	Model2	1.06 (1.19)	1.09 (1.26)	1.23 (1.39)
$\rho = 0$	Model3	1.62 (1.7)	1.57 (1.73)	1.4 (1.59)
	Model4	1.57 (1.69)	1.54 (1.77)	1.36 (1.55)
	Model5	1.04 (1.21)	1.09 (1.26)	1.34 (1.52)
	Model6	1.08 (1.24)	1.08 (1.24)	1.23 (1.41)
	Model1	1.19 (1.36)	1.2 (1.4)	1.4 (1.67)
	Model2	1.21 (1.36)	1.2 (1.39)	1.28 (1.46)
$\rho = 0.5$	Model3	1.35 (1.58)	1.38 (1.62)	1.3 (1.48)
	Model4	1.32 (1.55)	1.35 (1.56)	1.36 (1.52)
	Model5	1.2 (1.38)	1.39 (1.66)	1.53 (1.74)
	Model6	1.2 (1.39)	1.52 (1.74)	1.52 (1.74)
	Model1	1.2 (1.35)	1.43 (1.78)	1.62 (2.06)
	Model2	1.18 (1.31)	1.22 (1.38)	1.62 (2.06)
$\rho = 0.9$	Model3	1.24 (1.51)	1.54 (1.95)	1.4 (1.74)
	Model4	1.21 (1.47)	1.43 (1.79)	1.39 (1.72)
	Model5	1.36 (1.66)	1.62 (2.06)	1.62 (2.06)
	Model6	1.34 (1.63)	1.62 (2.06)	1.62 (2.06)

Table 2.11. The PRMSE of prediction in all settings when $N = 100$.

the following formula.

$$PRMSE = \sqrt{\sum_{i=1}^{100} \left(\tilde{y}_i - \tilde{X}_i\hat{\beta}\right)^2 /100} \tag{2.7}$$

Here $\hat{\beta}$ is the estimates of β in a randomly one out of the 50 trials in that setting. Results are summarized in table 2.11 and 2.12. Comparing table 2.12 to table 2.12, there is no doubt that larger sample size leads to a more accurate fitting. Moreover, model fitting deteriorates as the correlation among features raises. At the same time, all conclusion drawn from Section 2.3.3 are confirmed from the prediction results.

		Credible Level = 0.8	Credible Level = 0.95	Credible Level = 0.99
	Model1	0.84 (0.89)	0.84 (0.9)	0.84 (0.92)
	Model2	0.84 (0.92)	0.85 (0.92)	0.85 (0.93)
$\rho = 0$	Model3	0.83 (0.89)	0.84 (0.9)	0.86 (0.94)
	Model4	0.83 (0.89)	0.84 (0.91)	0.84 (0.91)
	Model5	0.83 (0.9)	0.83 (0.9)	0.84 (0.91)
	Model6	0.83 (0.9)	0.84 (0.93)	0.84 (0.92)
	Model1	1.09 (1.33)	1.07 (1.32)	1.11 (1.38)
	Model2	1.1 (1.33)	1.07 (1.31)	1.07 (1.31)
$\rho = 0.5$	Model3	1.09 (1.33)	1.07 (1.32)	1.1 (1.38)
	Model4	1.09 (1.33)	1.07 (1.32)	1.1 (1.38)
	Model5	1.05 (1.29)	1.09 (1.37)	1.16 (1.36)
	Model6	1.05 (1.28)	1.1 (1.37)	1.15 (1.36)
	Model1	1.06 (1.23)	1.2 (1.31)	1.42 (1.77)
	Model2	1.08 (1.23)	1.25 (1.35)	1.19 (1.33)
$\rho = 0.9$	Model3	1.08 (1.24)	1.26 (1.34)	1.22 (1.4)
	Model4	1.08 (1.24)	1.2 (1.31)	1.3 (1.56)
	Model5	1.1 (1.22)	1.27 (1.5)	1.62 (2.06)
	Model6	1.25 (1.46)	1.35 (1.66)	1.62 (2.06)

Table 2.12. The PRMSE of prediction in all settings when $N = 400$.

2.4 Discussion

This chapter presented a comparison of the Laplace(Model 1), Gaussian (Model 2) and horseshoe (Model 6) priors on Bayesian linear models and demonstrated an expected difference in their behavior. In particular, the heavy tail distribution, such as horseshoe, is more capable to capture dispersive parameters so as to favor independent predictors. On the other hand, when the variables are highly correlated, their signals are more likely to concentrate at zero and a Gaussian prior outperforms others in this case.

In addition, we compared different implementation of Laplace prior (Model 1, 3, 4) and confirmed the impact of hierarchical structures. In this simulation, the simpler structures outperform complex ones, probably because the data generation model is so straightforward that a simple model is sufficient to seize the majority of signals. Whereas the real-world data, where the data generation models are usually quite complicated or do not even exist, may need a complex model to fit well.

To select the subset of variables to be included in the model, we made use of the credible intervals, which is a particular output of probabilistic models. The size of selected model is controlled by the credible levels of these intervals, higher credible levels leading to smaller models. Based on that, we compared the prior distributions at different credible levels, which actually helps to explore the impact of the shape, primarily the tails, on the performance of the candidates. In general, priors with heavier tails benefit from a smaller credible level because it results in a large model. Credible levels was applied to select

features by Park and Casella in 2008 [31] and it is the most intuitive strategy in this field. More approaches will be discussed in the next chapter.

Stan, an efficient, powerful and user-friendly tool for probabilistic programming, is applied to fit Bayesian models in this study. First of all, it utilizes Hamiltonian Monte Carlo in sampling, which reduces the correlation between two consecutive samples in the Markov chain. Thus, the computation time is shortened tremendously due to the fast converge and zero thinning. Hamiltonian Monte Carlo also helps prevent the sampling process from choking at a local optimum point. In terms of computational stability, stan is more practical than BUGS because it can be applied to large data sets. Last but not least, because of the flexibility of the hierarchical structure and the feasibility of Stan scripts, this study can be easily expanded to generalized linear models, such as logistic and survival models, which is part of the future work of this study.

In the future, extensive simulation work will also be performed for reaching a solid conclusion. For instance, the scenario when the number of predictors surpasses the number of samples is of great interest and should be constructed in the simulation.

Chapter 3

Group-wise Projective Bayesian Feature Selection

In Bayesian data analysis, feature subset selection is a separate step from the prior selection. Unlike LASSO, which achieves parameter shrinkage and feature selection simultaneously in model fitting, the estimates of parameters in Bayesian models will not shrink to exact zero because they are random samples of a posterior distribution essentially. We cannot decide which variables should be included in a model based on their point estimates. Therefore, a strategy should be proposed to select a subset of variables for the model construction based on the posterior draws produced by a proper shrinkage prior.

Fortunately, Bayesian inferences is so powerful that it provides a much broader description of parameters, such as the credible intervals we adopted in Chapter 2. Based on this knowledge, numerous methods are proposed for the feature selection in Bayesian models. In 2003, J. Dupuis and C. Robert proposed the Projection method [59]. The idea of this method is to fit an all-encompassing model first and then to search for a submodel which is sparse and sufficiently close to the full model. Posterior distributions are used to

estimate the divergence of two probabilistic models and to evaluate the predictive performance of a candidate model. This strategy is efficient because it requires fitting the full model only, and all submodels will be searched within a certain model space by means of cross-validation.

In this Chapter, we will extend the searching process of the Projective method to the group level, so as to enhance the accuracy of feature selection by incorporating the grouping information and further improve the computation efficiency. To produce shrinkage, we apply the horseshoe prior in the model fitting due to its advantageous performance observed in Section 2.3.

3.1 Review of Feature Selection Methods in Bayesian Models

Numerous techniques have been proposed for feature subset selection in Bayesian models. Some of them will be reviewed in Section 3.1.2. To get started, the measures of the predictive ability of candidate models should be specified.

3.1.1 Predictive Ability Evaluation

The predictive performance of a model is typically defined in terms of a utility function that describes the quality of the predictions. An often used utility function to measure the quality of the predictive distribution of the candidate model M is the logarithmic score

$$u\left(M, \tilde{y}\right) = \log p\left(\tilde{y}|D, \tilde{x}, M\right),$$

where D is the training data set.

Since the future observations $\tilde{y}$ are unknown, the utility function $u(M, \tilde{y})$ cannot be evaluated beforehand. Therefore one usually works with the expected utilities instead

$$\begin{aligned} \bar{u}(M) &= E[\log p(\tilde{y}|D, \tilde{x}, M)] \\ &= \int p_t(\tilde{y}) \log p(\tilde{y}|D, \tilde{x}, M)\, d\tilde{y} \end{aligned} \tag{3.1}$$

where $p_t(\tilde{y})$ denotes the true data generating distribution. This expression will be referred to as the generalization utility or more loosely as the predictive performance of model M. Maximizing (3.1) is equivalent to minimizing the Kullback–Leibler (KL) divergence from the true data generating distribution $p_t(\tilde{y})$ to the predictive distribution of the candidate model M.

Kullback–Leibler divergence In probability theory, Kullback–Leibler divergence (KL divergence) is a measure of the non-symmetric difference between a theory model Q to the true model P. The Kullback–Leibler divergence was originally introduced by Solomon Kullback and Richard Leibler in 1951 as the directed divergence between two distributions [60]. The Kullback–Leibler divergence from Q to P is defined as

$$KL(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)},$$

which has a continuous version

$$KL(P||Q) = \int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)} dx.$$

Although the KL divergence measures the "distance" between two distributions, it is not a distance measure. This is because that the KL divergence is not a metric measure — it is not symmetric that the KL from P to Q is generally not the same as the KL from Q to P. Furthermore, it does not satisfy triangular inequality.

In Bayesian model selection system, we use the predictive KL divergence to compare the reference model M_* and the candidate model M.

KL divergence in normal distribution For normal distribution, the KL divergence of a Gaussian model M_2 from another Gaussian model M_1 is defined as

$$
\begin{aligned}
KL\left(M_1||M_2\right) &= \int \log \frac{Normal\left(y|\mu_1,\sigma_1\right)}{Normal\left(y|\mu_2,\sigma_2\right)} Normal\left(y|\mu_1,\sigma_1\right) dx \\
&= \int \left[-\frac{1}{2}\log(2\pi) - \log(\sigma_1) - \frac{1}{2}\left(\frac{y-\mu_1}{\sigma_1}\right)^2 + \frac{1}{2}\log(2\pi) + \log(\sigma_2) + \frac{1}{2}\left(\frac{y-\mu_2}{\sigma^2}\right)^2 \right] \\
&\times \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{1}{2}\left(\frac{y-\mu_1}{\sigma_1}\right)^2 \right] dx \\
&= \int \left[-\log(\sigma_1) - \frac{1}{2}\log(\left(\frac{y-\mu_1}{\sigma_1}\right)^2 + \log(\sigma_2) + \frac{1}{2}\log\left(\frac{y-\mu_2}{\sigma^2}\right)^2 \right] \\
&\times \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{1}{2}\left(\frac{y-\mu_1}{\sigma_1}\right)^2 \right] dx \\
&= E\left\{ \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2}\left[\left(\frac{y-\mu_2}{\sigma^2}\right)^2 - \left(\frac{y-\mu_1}{\sigma_1}\right)^2 \right] \right\} \\
&= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2\sigma_2^2} E_1\left[(y-\mu_2)^2\right] - \frac{1}{2\sigma_1^2} E_1\left[(y-\mu_1)^2\right] \\
&= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2\sigma_2} E_1\left[(y-\mu_2)^2\right] - \frac{1}{2},
\end{aligned}
$$

where $E_1(\cdot)$ denotes the expectation under model M_1.

In Bayesian model selection system, the two Gaussian model can be designed as follows.

$$M_1 : y = \mu_1 + \epsilon_1, \epsilon_1 \sim N\left(0, \sigma_1^2\right) \tag{3.2}$$

$$M_2 : y = \mu_2 + \epsilon_1 + \Delta\epsilon_2, \Delta\epsilon_2 \sim N\left(0, \sigma_2^2\right), \tag{3.3}$$

where σ_1 and $\Delta\sigma_2$ are independent. Thus, we can define another error term $\epsilon_2 = \epsilon_1 + \Delta\epsilon_2$ and $\epsilon_2 \sim N\left(0, \sigma_1^2 + \sigma_2^2\right)$. The KL divergence of model M_2 from

model M_1 is

$$\begin{aligned} KL\left(M_1||M_2\right) &= \log\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}}{\sigma_1}\right) + \frac{1}{2\left(\sigma_1^2+\sigma_2^2\right)} E_1\left[(y-\mu_2)^2\right] - \frac{1}{2} \\ &= \log\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}}{\sigma_1}\right) + \frac{1}{2\left(\sigma_1^2+\sigma_2^2\right)} E_1\left[(y-\mu_1)^2 + (\mu_1-\mu_2)^2\right] - \frac{1}{2} \\ &= \log\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2 - (\sigma_1^2+\sigma_2^2)}{2\left(\sigma_1^2+\sigma_2^2\right)} \\ &= \log\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}}{\sigma_1}\right) + \frac{(\mu_1-\mu_2)^2 - \sigma_2^2}{2\left(\sigma_1^2+\sigma_2^2\right)} \\ &= \log\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}}{\sigma_1}\right). \end{aligned} \tag{3.4}$$

The last equation holds because $\mu_1 - \mu_2 = \Delta\epsilon_2$ so that the numerator of the second term is 0.

In particular, model M_1 (3.2) is a linear regression and can be written as

$$\begin{aligned} M_1 : y &= \mu_1 + \epsilon_1, \epsilon_1 \sim N\left(0, \sigma_1^2\right) \\ \mu_1 &= \hat{y} = \mathrm{x}\hat{\beta}, \end{aligned}$$

where $\hat{\beta}$ is the least square estimate of the coefficients in M_1. After Selecting a subset of predictors x_p with any method introduced in Section 3.1.2 and 3.2, another linear regression of $\hat{y}$ can be built based on x_p, which is

$$\begin{aligned} M_2 : y &= \mu_2 + \epsilon_1 + \Delta\epsilon_2, \Delta\epsilon_2 \sim N\left(0, \sigma_2^2\right), \\ \hat{y} &= \mu_2 + \Delta\epsilon_2 = \mathrm{x}_p\hat{\beta}_p + \Delta\epsilon_2. \end{aligned}$$

As a result, we can approximate σ_2^2 with $\left(\hat{y} - \mathrm{x}_p\hat{\beta}_p\right)^2$, which updates the KL divergence of M_2 from M_1 (3.4) into

$$\begin{aligned} KL\left(M_1||M_2\right) &= \frac{1}{2}\log\left(\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2}\right) \\ &\approx \frac{1}{2}\log\left(\frac{\sigma_1^2 + \left(\hat{y} - \mathrm{x}_p\hat{\beta}_p\right)^2}{\sigma_1^2}\right). \end{aligned} \tag{3.5}$$

3.1.2 Review of Methodology

Following [57, 61–64], the feature selection strategies naturally fall into three categories, $\mathcal{M}$-closed, $\mathcal{M}$-completed and $\mathcal{M}$-open views, based on the properties of the data generator. The relationship among the three views can be summarized as

$$\mathcal{M} - open \succ \mathcal{M} - complete \succ \mathcal{M} - closed,$$

where $\succ$ represents a decreasing complexity ordering.

$\mathcal{M}$-closed Methods In the $\mathcal{M}$-closed view, it assumes that the true data generating model is one of the models under consideration, i.e., the true model is actually on the model list (at least in the sense that error due to misspecification is negligible compared to any other source of error), under uncertainty about which of the candidate model is the said true model. This class of problems is comparatively simple and well studied.

If the number of alternative models is countable, the actual belief model

of the future observations is constructed as the Bayesian model averaging (BMA) predictive distribution $p\left(\tilde{y}|D, M_*\right) = p_{BMA}\left(\tilde{y}|D\right)$. Bayesian model averaging is a strategy to build a richer model by averaging over a class of simpler parametric models, which is thoroughly discussed by J. Hoeting et. al. [65]. In a situation in which a set of alternative models $\{M_k\}_{k=1}^{K}$ and a corresponding prior $p(M_k)$ on that set have been specified, one can integrate over the models and thereby arrive at the BMA predictive distribution

$$p_{BMA}\left(\tilde{y}|D\right) = \sum_{k=1}^{K} p\left(\tilde{y}|\tilde{x}, D, M_k\right) p\left(M_k|D\right), \tag{3.6}$$

where $p\left(M_k|D\right)$ are the posterior probabilities of the models M_k.

Literally, the $\mathcal{M}$-closed view only applies to the situation when it is known for certain that the true data generating mechanism is among candidates. However, Bayesian model averaging has been shown to have good predictive performance even without the strict interpretation of the $\mathcal{M}$-closed view holds .

From a model selection point of view, one may choose the model maximizing the posterior distribution of models $p(M_k|D)$ ending up with a maximum a posteriori (MAP) model. Assuming the true data generating model belongs to the set of the candidate models, MAP model can be shown to be the optimal choice under the zero-one utility function (utility being one if the true model is found, and zero otherwise). If the models are given equal prior probabilities, $p\left(M\right) \propto 1$, finding the MAP model reduces to maximizing the marginal likelihood.

$\mathcal{M}$-completed Methods The $\mathcal{M}$-completed view abandons the idea of a true model in the $\mathcal{M}$-open view, but still forms a rich enough model M^*, whose predictive distribution $p\left(\tilde{y}|D, M_*\right)$ is considered as the best available description of the uncertainty of future data. There are basically two different but related approaches that fit $\mathcal{M}$-completed view , reference predictive method and projection predictive method. The projection predictive method will be discussed in Section 3.2.

Reference Predictive Method Reference model M_* is built as the best description of our knowledge about the future observations, which is considered as a proxy of the true model. Based on this point of view, the utilities of a candidate model M can be estimated by replacing the true distribution $p_t\left(\tilde{y}\right)$ in (3.1) with the predictive distribution of the reference model $p(\tilde{y}|D, M_*)$. Averaging this over the training inputs $\{x_i\}_{i=1}^n$ gives the reference utility

$$\bar{u}_{ref}\left(M\right) = \frac{1}{n}\sum_{i=1}^{n}\int p_t\left(\tilde{y}|x_i, D, M_*\right)\log p\left(\tilde{y}|x_i, D, M\right)d\tilde{y}. \tag{3.7}$$

As the reference model is in practice different from the true data generating model, the reference utility is a biased estimate of the true generalization utility (3.1).

The maximization of the reference utility is equivalent to minimizing the predictive KL-divergence between the reference model M_* and the candidate

model M at the training inputs

$$\delta\left(M_{*}||M\right)=\frac{1}{n}\sum_{i=1}^{n}KL\left(p_{t}\left(\tilde{y}|x_{i},D,M_{*}\right)||p_{t}\left(\tilde{y}|x_{i},D,M\right)\right). \tag{3.8}$$

Thus, the model can be chosen based on the strict minimization of the discrepancy measure (3.1), or the simplest model that has an acceptable discrepancy (3.8).

The reference predictive approach is inherently a less straightforward approach to model selection than the $\mathcal{M}$-open views, because it requires the construction of the reference model. San Martini et. al. proposed using Bayesian model average (BMA) as the reference [66]. In fact, any other models or priors can be used as long as we believe it reflects our best knowledge of the problem and allows convenient computation.

$\mathcal{M}$-open Methods The $\mathcal{M}$-open class of problems is one step more elusive. $\mathcal{M}$-open problems are those in which the data generation is too complex to admit a true model, such as the nucleotide sequence in a chromosome. In this case, we are only able to compare different predictors without reference to a true model.

$\mathcal{M}$-open view corresponds to avoiding the explicit specification of the predictive posterior $p\left(\tilde{y}|D,M^{*}\right)$ by reusing observations D as proxy for the predictive distribution of the actual belief model. One option is using information criteria [57].

A fully Bayesian criterion is the widely applicable information criterion

(WAIC) [67, 68], defined as

$$WAIC = \frac{1}{n}\sum_{i=1}^{n} \log p\left(y_i|x_i, D, M\right) - \frac{V}{n}$$

where the first term is the training utility and V is the functional variance given by

$$V = \sum_{i=1}^{n} \left\{ E\left[\left(\log p\left(y_i|x_i, \theta, M\right)\right)^2\right] - E\left[\log p\left(y_i|x_i, \theta, M\right)\right]^2 \right\}.$$

Here both of the expectations are taken over the posterior $p\left(\theta|D, M\right)$. WAIC is proved to be asymptotically equal to the Bayesian LOO-CV.

Another still popular way is the deviance information criterion (DIC) proposed by Spiegelhalter et. al. [69]. DIC estimates the generalization performance of the model with parameters fixed to the posterior mean $\bar{\theta} = E\left[\theta|D, M\right]$. DIC can be written as

$$DIC = \frac{1}{n}\sum_{i=1}^{n} \log p\left(y_i|x_i, \bar{\theta}, M\right) - \frac{p_{eff}}{n},$$

where p_{eff} is the effective number of parameters, estimated by

$$p_{eff} = 2\sum_{i=1}^{n} \left(\log p\left(y_i|x_i, \bar{\theta}, M\right) - E\left[\log p\left(y_i|x_i, \bar{\theta}, M\right)\right]\right).$$

Here the expectation is taken over the posterior, which is questionable from a practical point of view especially when the model is singular.

3.2 Feature Subset Selection Strategy

Projection predictive method is an extension of the reference predictive method 3.1.2. The idea is to project the information in the posterior of the reference model M_* onto the candidate model space M so that the predictive distribution of the candidate model remains as close to the reference model as possible [58,63]. Thus the candidate model parameters are determined by the fit of the reference model, not by the data; and the reference model is fitted with the Bayesian models constructed in the previous chapter. In this section, we will discuss the Projection predictive scheme in feature subset selection.

3.2.1 Projective Submodels

Simply speaking, the projection of a vector onto a subspace is to reduce the "distance" between the vector and the subspace. Given a reference model parameter θ_*, the projected parameter $\theta^{\perp}$ onto the parameter space of model M is defined via

$$\theta^{\perp} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} KL\left(p\left(\tilde{y}|x_i, \theta_*, M_*\right) || p\left(\tilde{y}|x_i, \theta, M\right)\right). \tag{3.9}$$

The discrepancy between the reference model M_* and the projected candidate model M is then defined to be the expectation of the divergence over the posterior of the reference model

$$\delta\left(M_* || M\right) = \frac{1}{n} \sum_{i=1}^{n} E\left[KL\left(p\left(\tilde{y}|x_i, \theta_*, M_*\right) || p\left(\tilde{y}|x_i, \theta^{\perp}, M\right)\right)\right]. \tag{3.10}$$

Since, the posterior expectation in the discrepancy (3.10) is in general not available analytically, it can be approximated with the average of individual discrepancies between the reference model and each candidate model defined with $\theta_s^{\perp}$, where $\theta_s^{\perp}, s = 1, 2, \ldots .S$ is the projection, based on (3.9), of random samples $\theta_s, s = 1, 2, \ldots .S$ from the posterior of the reference model.

$$\delta\left(M_* || M\right) \approx \frac{1}{nS} \sum_{i=1}^{n} \sum_{s=1}^{S} KL\left(p\left(\tilde{y}|x_i, \theta_*, M_*\right) || p\left(\tilde{y}|x_i, \theta_s^{\perp}, M\right)\right). \quad (3.11)$$

Predictive Performance Evaluation

To evaluate the predictive performance of a model, we can use the logarithm of the predictive density (LPD) at an actual observation. This scoring rule is proper and measures the calibration and sharpness of the predictive distribution simultaneously [70]. Since the predictive densities are usually not available analytically, we estimate the LPD score with randoms samples of the posterior distribution, instead of the posterior distribution itself:

$$LPD\left(M\right) \approx \log \frac{1}{S} \sum_{s=1}^{S} p\left(\tilde{y}|\tilde{x}, \theta_s^{\perp}, M\right), \quad (3.12)$$

where $\theta_s^{\perp}, s = 1, 2, \ldots .S$ is the projection of $\theta_s, s = 1, 2, \ldots .S$, the random draws from the Markov chain Monte Carlo samples of the posterior distribution of the reference model. Given predictions (3.12), the model predictive performance is summarized by the mean LPD (MLPD) over the full set of n data points [58].

For the purpose of model comparison, we introduce q-value. For each

model M, $LPD\left(M\right)$ is a $n \times J$ matrix, where n is the total number of samples and J is the number of posterior draws. To reduce gauge uncertainty, the MLPD difference between model M_a and model M_b is computed with Bayesian bootstrap samples

$$\Delta MLPD^{(j)}\left(M_a, M_b\right) = \sum_{i=1}^{n} \omega_i^{(j)} \left[LPD_i\left(M_a\right) - LPD_i\left(M_b\right)\right], \tag{3.13}$$

where $\omega_i^{(j)}, i = 1, \ldots, n$, are the bootstrap weights for the j-th bootstrap sample generated using the Dirichlet distribution with parameters set to 1. The comparison is summarized as the probability of model M_a performing better than model M_b based on the posterior distribution of parameters, named q-values,

$$q\left(M_a, M_b\right) = \frac{1}{J} \sum_{j=1}^{J} I\left(\Delta MLPD^{(j)}\left(M_a, M_b\right) \geq 0\right). \tag{3.14}$$

3.2.2 Submodel Search

Exhaustive search of the model space is not feasible when the number of candidate covariates is large. The projection approach works in the suboptimal forward selection strategy for its simplicity and its scalability to large covariate sets, Algorithm 1.

In algorithm 1, M_j represents the submodel containing j variables. To start with, the model is a constant model, M_0. At each forward selection step, we add into model M_j with one more variable, which is the one leading to a model M_{j+1} with minimum KL divergence (3.5) among all candidates. These steps terminate till all variables enter the model. That is to say, j grows from

0 to P in this process, P being the total number of variables. This search ends up with a ranking of all variables, which defines a submodel for each model size (from 1 to P), as well as the KL divergence (3.5) of each chosen submodel $M_j, j = 1, \ldots, P$, with regards to the reference model.

Algorithm 1 Submodel search for projection predictive methods

1. Begin with the submodel M_0 (no variables) and set j to 0, where j indicates the size of a candidate model
2. Repeat until the candidate model is full:

   Find the projections for all submodels that are obtainable by adding one new variable to M_j. Choose the one with smallest KL divergence and set it as M_{j+1}

   Set j to $j + 1$

Cross Validation The forward selection scheme has been used in searching for a sequence of submodels. Further more, Peltola et. al. proposed using cross validation outside the searching process to decide the model size [58]. In each fold of the cross validation, the forward selection scheme is executed on the training samples in that fold, and LPD (3.12) of the submodels is computed with the test part of the fold. At last, each pair of submodels will be compared with MLPD (3.13) and q-values (3.14). The number of variables is then decided by summarizing the predictive performance estimates, and a new model with the chosen model size is built with all training data.

3.2.3 Submodel Search at Group Level

Group-wise feature selection schemes have been discussed in the cases where explanatory facts are represented by a group of predict variables [71, 72]. We

will also expand the Projection predictive feature selection to the group level, Algorithm 2.

Algorithm 2 Submodel search for projection predictive methods

1. Begin with the submodel M_0 (no variables) and set g to 0, where g indicates the number of groups of a candidate submodel

2. Repeat until the candidate model is full:

Find the group of projections for all submodels that are obtainable by adding one new group into M_g. Choose the group leading to the smallest KL divergence after being added to the model, resulting in M_{g+1}

Set g to $g + 1$

The difference between Algorithm 1 and 2 lies in the selection steps. In Algorithm 2, $g \in (1, \ldots, G)$ represents the number of groups chosen in model M_g. This algorithm returns the ranking of all feature groups based on the order they are selected in the model and the KL divergence of the submodels.

3.3 Simulation in Feature Subset Selection

We simulate both the training data and test data with the exact same settings as that in Section 2.3.1. The trial is repeated 50 times and the measures of model performance are averaged across the 50 trials.

3.3.1 Variable Selection

In the cross validation of projection algorithm, we make predictions for the validation set of each fold, and further compute the predictive performance estimates for all submodels chosen in that fold. Since the forward selection in each fold might (and usually will) select different sets of variables, the pre-

diction performance does not necessarily compare any certain submodels, but only the performance of the selection procedure. This process mostly resembles tuning the parameter λ with cross validation in the LASSO-like algorithms, where the same value of λ does not ensure the same feature selection results in every fold. However, a proper λ, such as λ_{min} and λ_{1se}, will still be chosen based on the prediction measures of all folds.

Similarly, we will choose the size of the final model by summarizing the prediction measures of all folds. Unfortunately, T. Peltola, the researcher who proposed Projection feature selection [58], did not provide any analytical solution to choosing an appropriate model size. As a result, we adopt these strategies in this study: for individual search, we choose the size leading to highest MLPD value; for group search, we choose the size in which MLPD reaches a local maximum for the first time as the submodels expand, that is, the most sparse model with local peak of MLPD.

We compared the correctly identified groups averaged over 50 trials. The feature selection results are summarized in Table 3.1 and 3.2 for individual search and group search respectively. It contains the means and standard errors (in the parentheses) of the sensitivities (SEN) of each positive group, the positive prediction values (PPV) as well as the AUC for feature selection. From the following tables we can first conclude that group 1 can be selected for sure due to its high effect size. Also, the sensitivities of selecting group 2 and 3 drop dramatically when the within group correlation is as high as 0.9. This agrees with the relationship between correlation and signal-to-noise ratio (SNR), that is, the high correlation among features amplifies the overall SNR,

		SEN 1	SEN 2	SEN 3	PPV	AUC
$N = 100$	$\rho = 0$	1(0)	0.88(0.33)	0.52(0.5)	0.69(0.29)	0.86(0.05)
	$\rho = 0.5$	1(0)	1(0)	0.9(0.3)	0.62(0.29)	0.84(0.05)
	$\rho = 0.9$	1(0)	1(0)	0.88(0.33)	0.63(0.31)	0.85(0.05)
$N = 400$	$\rho = 0$	1(0)	0.88(0.33)	0.64(0.48)	0.83(0.23)	0.99(0.01)
	$\rho = 0.5$	1(0)	1(0)	0.9(0.3)	0.68(0.31)	0.95(0.02)
	$\rho = 0.9$	1(0)	1(0)	0.9(0.3)	0.63(0.3)	0.9(0.02)

Table 3.1. The means and standard errors of the group-wise sensitivities (SEN), positive prediction values (PPV) and AUC for all settings of the variable-wise Projection selection. Note that in the definition of group-wise sensitivity, we consider a group positive if any feature in that group is selected.

so that the group of correlated features explain away a larger portion of the data variance, turning the weak signals less important.

To compare between the individual search and group search, the sensitivities of group 2 and 3 are higher in the individual search. Actually, it is a result of the definition of being positive in group search, which is that we consider a group as positive if any single feature in that group is selected. It is highly likely that only a small portion of the group are selected, because it explains the lower PPV in individual search than that in the group search algorithm. In addition, we observed much higher AUC in group-level algorithm, which suggests that Algorithm 2 is favorable when the grouping information is accurate.

3.3.2 Prediction

For each setting, we simulate a test set with 100 samples, on which we then compute the prediction root-mean-squared error (PRMSE) (2.7) and MLPD (3.12) based on the models selected in 3.3.1. Results are summarized in table

		SEN 1	SEN 2	SEN 3	PPV	AUC
	$\rho = 0$	1(0)	0.76(0.43)	0.36(0.48)	0.91(0.17)	0.95(0.08)
$N = 100$	$\rho = 0.5$	1(0)	0.96(0.2)	0.54(0.5)	0.85(0.21)	0.98(0.05)
	$\rho = 0.9$	1(0)	0.32(0.47)	0.12(0.33)	0.97(0.1)	1(0.01)
	$\rho = 0$	1(0)	0.8(0.4)	0.58(0.5)	0.87(0.17)	1(0)
$N = 400$	$\rho = 0.5$	1(0)	0.96(0.2)	0.52(0.5)	0.87(0.2)	1(0)
	$\rho = 0.9$	1(0)	0.34(0.48)	0.1(0.3)	0.96(0.13)	1(0)

Table 3.2. The means and standard errors of the group-wise sensitivities (SEN), positive prediction values (PPV) and AUC for all settings of the group-wise Projection selection.

		Number of Variables	Number of Groups
	$\rho = 0$	21.36(22.25)	2.46(1.09)
$N = 100$	$\rho = 0.5$	25.32(25.34)	3.3(1.64)
	$\rho = 0.9$	22.24(28.79)	1.58(1.14)
	$\rho = 0$	16.72(13.21)	3.02(1.6)
$N = 400$	$\rho = 0.5$	25.4(22.77)	3.22(1.71)
	$\rho = 0.9$	24.48(25.91)	1.66(1.38)

Table 3.3. The number of selected variables/groups in the two projection algorithms.

		Individual Projection	Group Projection	Credible Level=0.95
$N = 100$	$\rho = 0$	1.22(0.16)	1.17(0.1)	1.08 (1.24)
	$\rho = 0.5$	1.23(0.26)	1.14(0.07)	1.52 (1.74)
	$\rho = 0.9$	1.5(0.39)	1.23(0.09)	1.62 (2.06)
$N = 400$	$\rho = 0$	1.03(0.03)	1.07(0.07)	0.84 (0.93)
	$\rho = 0.5$	1.1(0.2)	1.05(0.05)	1.1 (1.37)
	$\rho = 0.9$	1.24(0.3)	1.22(0.12)	1.35 (1.66)

Table 3.4. The PRMSE of the Projection group selection algorithm for all settings.

		Individual Projection	Group Projection
$N = 100$	$\rho = 0$	-1.63(0.11)	-1.59(0.08)
	$\rho = 0.5$	-1.63(0.2)	-1.55(0.06)
	$\rho = 0.9$	-1.81(0.28)	-1.62(0.08)
$N = 400$	$\rho = 0$	-1.45(0.03)	-1.48(0.07)
	$\rho = 0.5$	-1.5(0.15)	-1.47(0.05)
	$\rho = 0.9$	-1.61(0.22)	-1.61(0.11)

Table 3.5. The comparison on MLPD between the Projection group-wise selection and the Projection variable-wise selection for all settings.

3.4 and 3.5. We compare MLPD in the two proposed algorithms only because it can only be calculated in probabilistic models. For PRMSE, we compare the two proposed model and the credible level method discussed in Section 2.2.4.

Typically, the group-wise Projection selection outperforms the individual-wise model in both MLPD and PRMSE in most settings, even though it constantly selected fewer variables according to Table 3.3. The advantage of group-wise model in correct identification of signal in group structure data set is obvious. In addition, all model deteriorates under the condition of high correlation within group.

3.4 Discussion and Future Work

In this chapter, we have briefly reviewed working schemes for Bayesian model selection and extended the Projection predictive model for group-level search. Projection methods first construct a full encompassing model, which is considered as the reference model, and then search for a sequence of submodels producing similar answers to the full model. The numerical experiments are conducted showing that the group-level Projection methods improved the model predictive performance and feature selection accuracy.

However, the estimated discrepancy between the reference model and a submodel is an unreliable indicator of the predictive performance of the submodel, which means, even if the submodel performs as well as the reference model, it may not be able to predict well. On the other hand, a well-performed reference model also caused some problems. In the searching process, all candidate models are nested to the reference, so that they are unlike to outperform the reference even if they can selected most relevant features. As a result, it brought in difficulties in summarizing the evaluation results in cross validation. Consider the cross validation in fitting a linear LASSO frequentist model. The fitted model starts to deteriorate when the model contains too many features, that is when λ is small. It is a favorable property because it yields a global minimum of MSE, based on which a proper tuning parameter can be selected. Whereas, it does not apply to the Projection algorithms because a larger model usually favors prediction rather than worsens it. Therefore, in most cases, the best prediction occurs in large models, even in the full model, which could not

be used as a decision of model size. In this study, we chose the smallest model among those producing local maximal MLPDs. The choice is based on the pre-information on the sparsity of the model.

Although the proposed algorithm works well when there is accurate group information, it is still unknown how non-perfect information will affect the algorithm. In the future, more simulation should be designed to detect the influence of inaccurate or insufficient knowledge on grouping. Further, the features within a group have the same effect size in the simulation, which may not be the case in reality. Specifically, maybe only part of the group is influential; or some features may yield exact opposite impact on the response. As known, the sparse group LASSO is a solution to this situation. Similarly, we can enroll sparsity inside a group through the credible level method in Chapter 2 to achieve a sparse group selection based on the Projection algorithm.

Finally, this algorithm will be applied to a read-world data set to identify relevant genes and pathways for a particular endpoint.

Bibliography

1. Xinyu Tian, Xuefeng Wang, and Jun Chen. Network-constrained group lasso for high-dimensional multinomial classification with application to cancer subtype prediction. *Cancer informatics*, 13(Suppl 6):25, 2014.

2. Laura J van't Veer and René Bernards. Enabling personalized cancer medicine through analysis of gene-expression patterns. *Nature*, 452(7187):564–570, 2008.

3. Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L Johnson. *Continuous multivariate distributions, models and applications*. John Wiley & Sons, 2004.

4. Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004.

5. Danh V Nguyen and David M Rocke. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18(9):1216–1226, 2002.

6. X Zhou, X Wang, and ER Dougherty. Multi-class cancer classification using multinomial probit regression with bayesian gene selection. *IEE Proceedings-Systems Biology*, 153(2):70–78, 2006.

7. Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

8. Yanni Zhu, Xiaotong Shen, and Wei Pan. Network-based support vector machine for classification of microarray samples. *BMC bioinformatics*, 10(1):S21, 2009.

9. Hui Zou and Ming Yuan. The f-norm support vector machine. *Statistica Sinica*, pages 379–398, 2008.

10. Terrence S Furey, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

11. Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3(1):140, 2007.

12. Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

13. Gavin C Cawley, Nicola LC Talbot, and Mark Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. *Advances in neural information processing systems*, 19:209, 2007.

14. Noah Simon, Jerome Friedman, and Trevor Hastie. A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. *arXiv preprint arXiv:1311.6529*, 2013.

15. Yong Liang, Cheng Liu, Xin-Ze Luan, Kwong-Sak Leung, Tak-Ming Chan, Zong-Ben Xu, and Hai Zhang. Sparse logistic regression with a l 1/2 penalty for gene selection in cancer classification. *BMC bioinformatics*, 14(1):198, 2013.

16. Martin Vincent and Niels Richard Hansen. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, 71:771–786, 2014.

17. Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

18. Gerhard Tutz, Wolfgang Pößnecker, and Lorenz Uhlmann. Variable selection in general multinomial logit models. *Computational Statistics & Data Analysis*, 82:207–222, 2015.

19. Ke-Qin Liu, Zhi-Ping Liu, Jin-Kao Hao, Luonan Chen, and Xing-Ming Zhao. Identifying dysregulated pathways in cancers from pathway interaction networks. *BMC bioinformatics*, 13(1):126, 2012.

20. Salim A Chowdhury, Rod K Nibbe, Mark R Chance, and Mehmet Koyutürk. Subnetwork state functions define dysregulated subnetworks in cancer. *Journal of Computational Biology*, 18(3):263–281, 2011.

21. Marilena V Iorio and Carlo M Croce. Microrna dysregulation in cancer: diagnostics, monitoring and therapeutics. a comprehensive review. *EMBO molecular medicine*, 4(3):143–159, 2012.

22. Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

23. Jun Chen, Frederic D Bushman, James D Lewis, Gary D Wu, and Hongzhe Li. Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*, 14(2):244–258, 2013.

24. Jian Huang, Shuangge Ma, Hongzhe Li, and Cun-Hui Zhang. The sparse laplacian shrinkage estimator for high-dimensional regression. *Annals of statistics*, 39(4):2021, 2011.

25. Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

26. Yiyi Liu, Quanquan Gu, Jack P Hou, Jiawei Han, and Jian Ma. A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression. *BMC bioinformatics*, 15(1):37, 2014.

27. Eric C Holland, Joseph Celestino, Chengkai Dai, Laura Schaefer, Raymond E Sawaya, and Gregory N Fuller. Combined activation of ras and akt in neural progenitors induces glioblastoma formation in mice. *Nature genetics*, 25(1):55–57, 2000.

28. Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

29. Cun-Hui Zhang et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

30. Patrick Breheny and Jian Huang. Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and computing*, 25(2):173–187, 2015.

31. Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

32. Minjung Kyung, Jeff Gill, Malay Ghosh, George Casella, et al. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.

33. Enes Makalic and Daniel F Schmidt. High-dimensional bayesian regularised regression with the bayesreg package. *arXiv preprint arXiv:1611.06649*, 2016.

34. Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

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

36. Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

37. Andrew Gelman et al. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3):515–534, 2006.

38. Nicholas G Polson, James G Scott, et al. On the half-cauchy prior for a global scale parameter. *Bayesian Analysis*, 7(4):887–902, 2012.

39. Nicholas G Polson and James G Scott. Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

40. SL Van Der Pas, BJK Kleijn, AW Van Der Vaart, et al. The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2):2585–2618, 2014.

41. Kyu Ha Lee, Sounak Chakraborty, and Jianguo Sun. Bayesian variable selection in semiparametric proportional hazards model for high dimensional survival data. *The International Journal of Biostatistics*, 7(1):1–32, 2011.

42. David Maxwell Chickering and David Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine learning*, 29(2):181–212, 1997.

43. Jim E Griffin, Philip J Brown, et al. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188, 2010.

44. Nicolaas Godfried Van Kampen and William P Reinhardt. Stochastic processes in physics and chemistry, 1983.

45. Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

46. Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

47. David J Spiegelhalter, Andrew Thomas, Nicky G Best, Wally Gilks, and D Lunn. Bugs: Bayesian inference using gibbs sampling. *Version 0.5,(version ii) http://www. mrc-bsu. cam. ac. uk/bugs*, 19, 1996.

48. David Spiegelhalter, Andrew Thomas, Nicky Best, and Dave Lunn. Winbugs user manual, 2003.

49. Martyn Plummer et al. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, volume 124, page 125. Vienna, 2003.

50. Stan Development Team. Stan modeling language users guide and reference manual. Version 2.14.0, 2016.

51. David M Gay. Semiautomatic differentiation for efficient gradient computations. In *Automatic differentiation: applications, theory, and implementations*, pages 147–158. Springer, 2006.

52. Mary Kathryn Cowles and Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

53. Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

54. Charles Geyer. Introduction to markov chain monte carlo. *Handbook of markov chain monte carlo*, pages 3–48, 2011.

55. Ward Edwards, Harold Lindman, and Leonard J Savage. Bayesian statistical inference for psychological research. *Psychological review*, 70(3):193, 1963.

56. Peter M Lee. *Bayesian statistics: an introduction*. John Wiley & Sons, 2012.

57. Juho Piironen and Aki Vehtari. Comparison of bayesian predictive methods for model selection. *Statistics and Computing*, pages 1–25, 2016.

58. Tomi Peltola, Aki S Havulinna, Veikko Salomaa, and Aki Vehtari. Hierarchical bayesian survival analysis and projective covariate selection in cardiovascular event risk prediction. In *Proceedings of the Eleventh UAI Conference on Bayesian Modeling Applications Workshop-Volume 1218*, pages 79–88. CEUR-WS. org, 2014.

59. Jérome A Dupuis and Christian P Robert. Variable selection in qualitative models via an entropic explanatory power. *Journal of Statistical Planning and Inference*, 111(1):77–94, 2003.

60. S Kullback. Statistics and information theory. *J. Wiley and Sons, New York*, 1959.

61. José M Bernardo and Adrian FM Smith. Bayesian theory. 1994. *John Willey and Sons. Valencia (España)*, 1994.

62. José M Bernardo and Adrian FM Smith. Bayesian theory, 2001.

63. Aki Vehtari, Janne Ojanen, et al. A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

64. Jennifer Lynn Clarke, Bertrand Clarke, Chi-Wai Yu, et al. Prediction in m-complete problems with limited sample size. *Bayesian Analysis*, 8(3):647–690, 2013.

65. Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

66. A San Martini and Fulvio Spezzaferri. A predictive model selection criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 296–303, 1984.

67. Sumio Watanabe. *Algebraic geometry and statistical learning theory*, volume 25. Cambridge University Press, 2009.

68. Sumio Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(Dec):3571–3594, 2010.

69. David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

70. Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

71. Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

72. Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Appendix

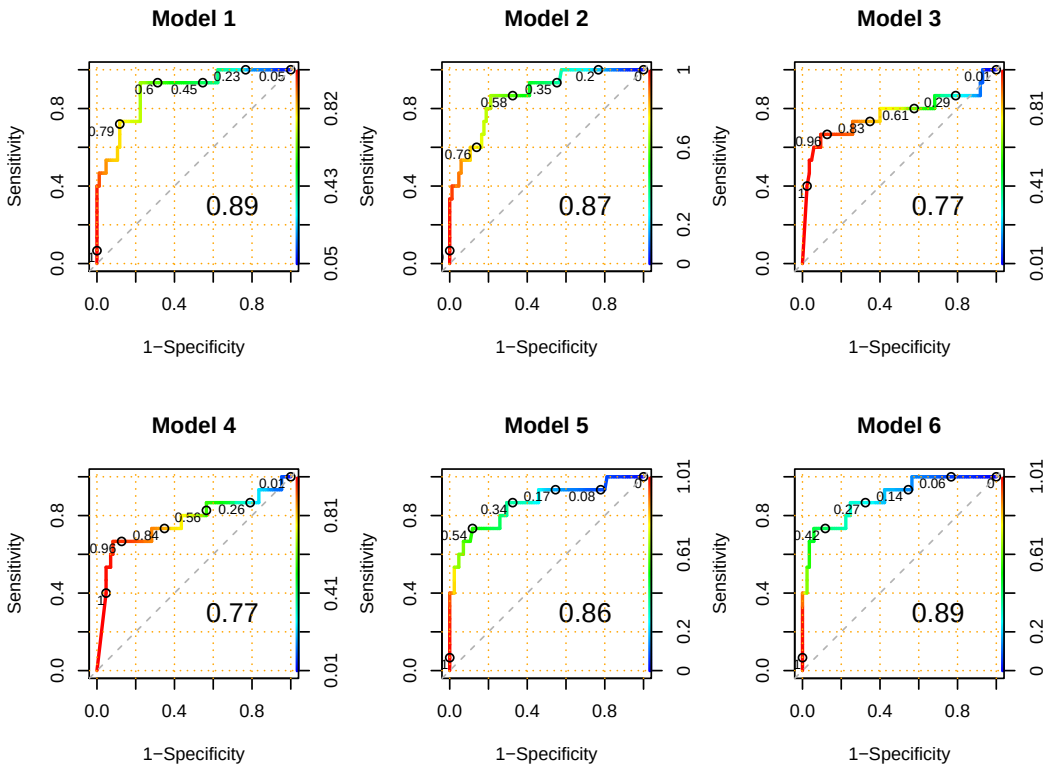

Figure A.1. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 100$ and $\rho = 0$. The decimal under the diagonal is the AUC of each curve.

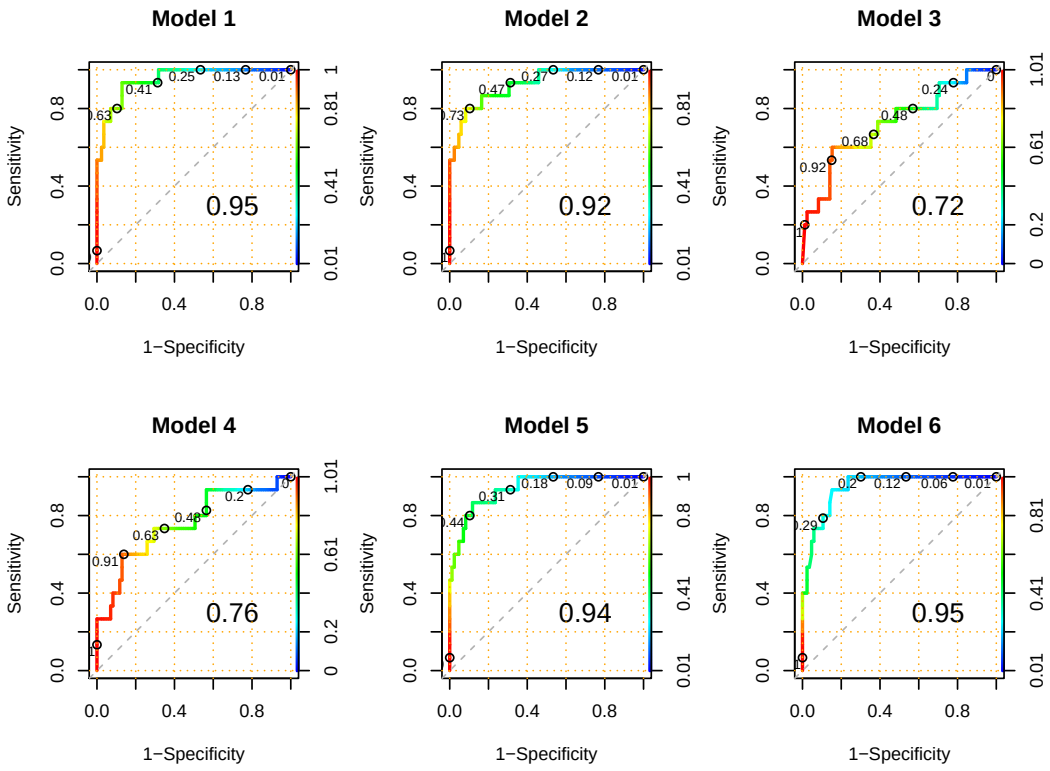

Figure A.2. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 100$ and $\rho = 0.5$. The decimal under the diagonal is the AUC of each curve.

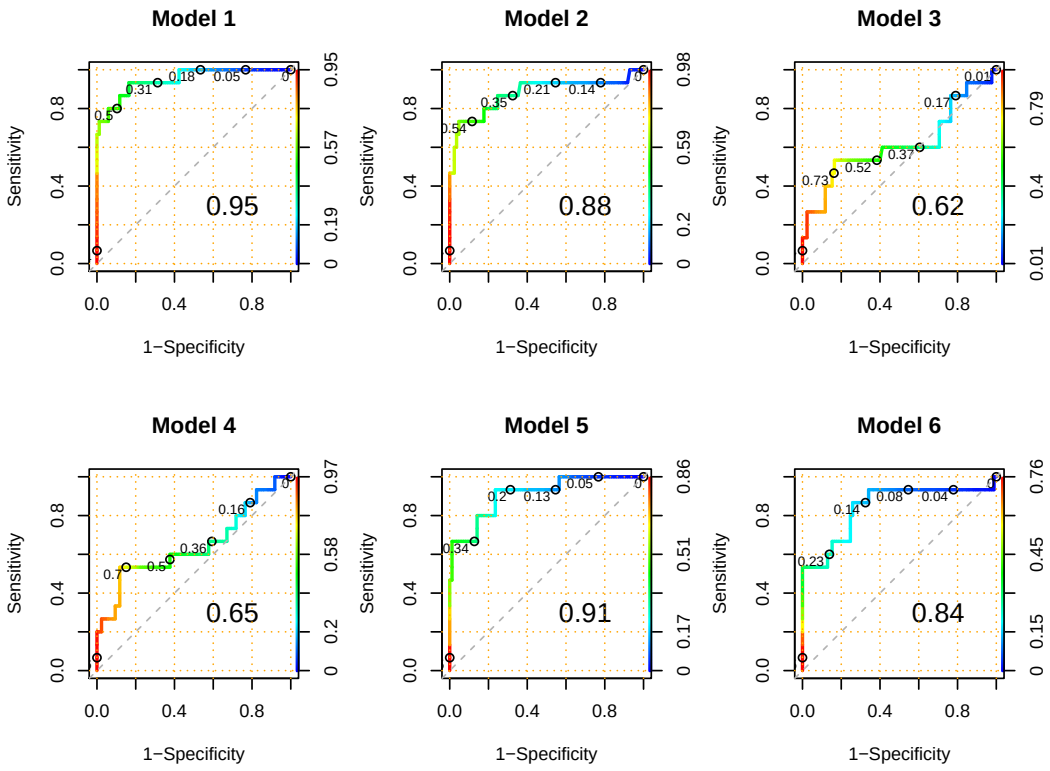

Figure A.3. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 100$ and $\rho = 0.9$. The decimal under the diagonal is the AUC of each curve.

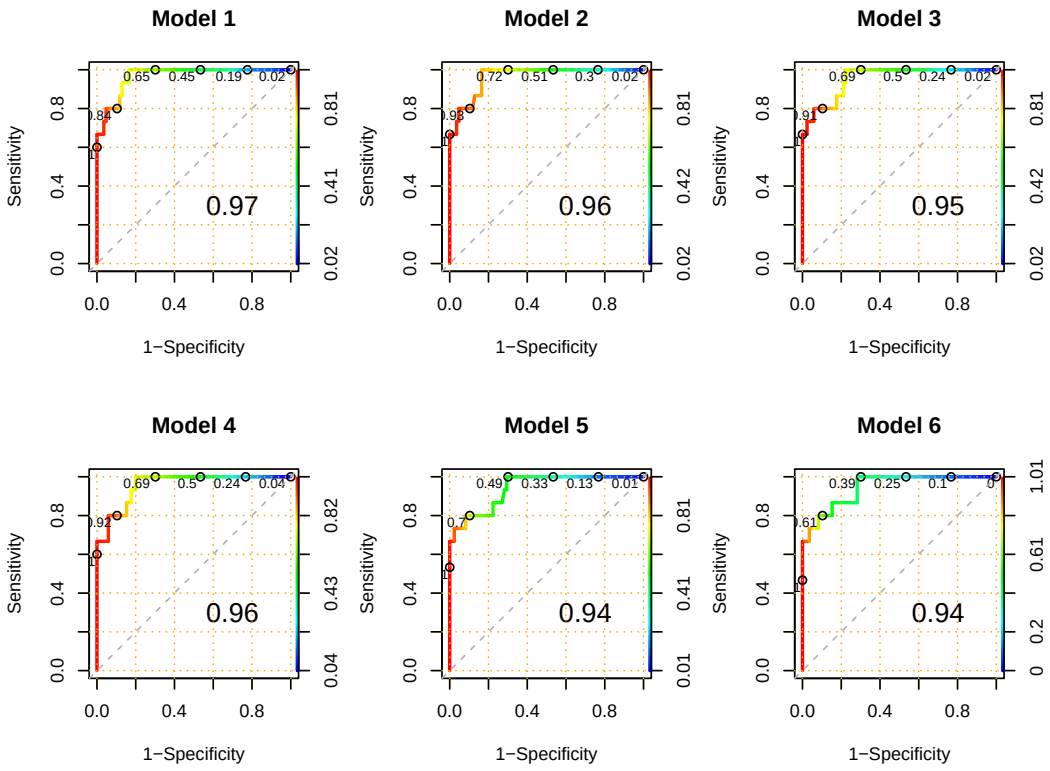

Figure A.4. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 400$ and $\rho = 0$. The decimal under the diagonal is the AUC of each curve.

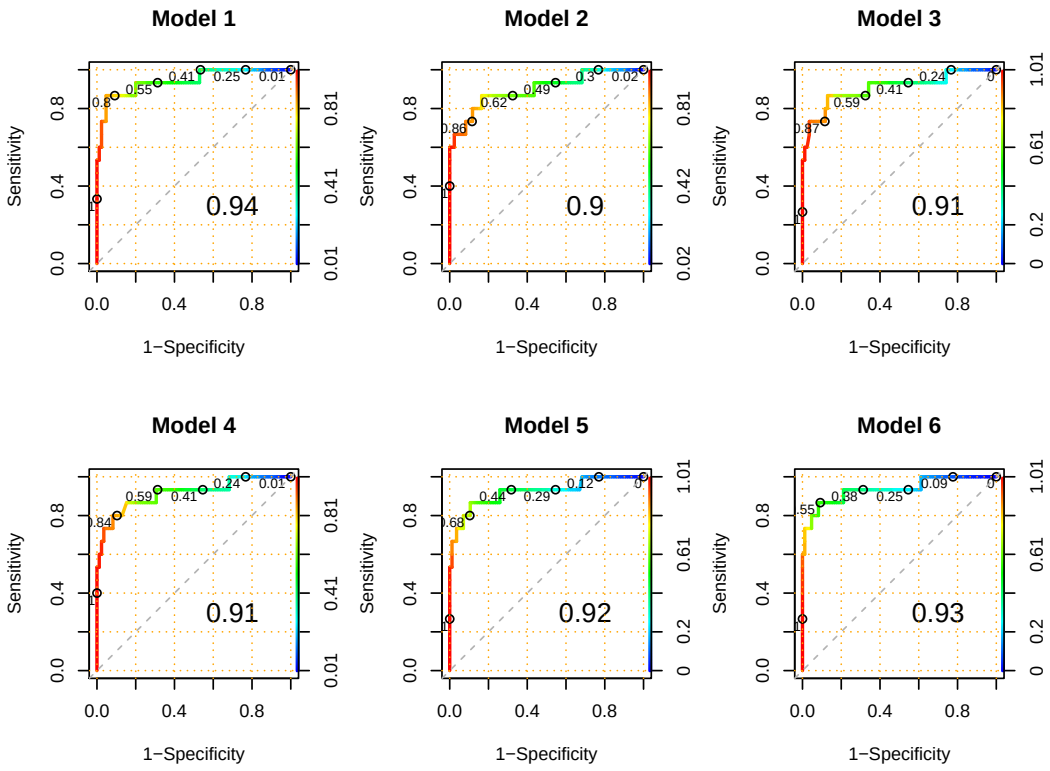

Figure A.5. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 400$ and $\rho = 0.5$. The decimal under the diagonal is the AUC of each curve.

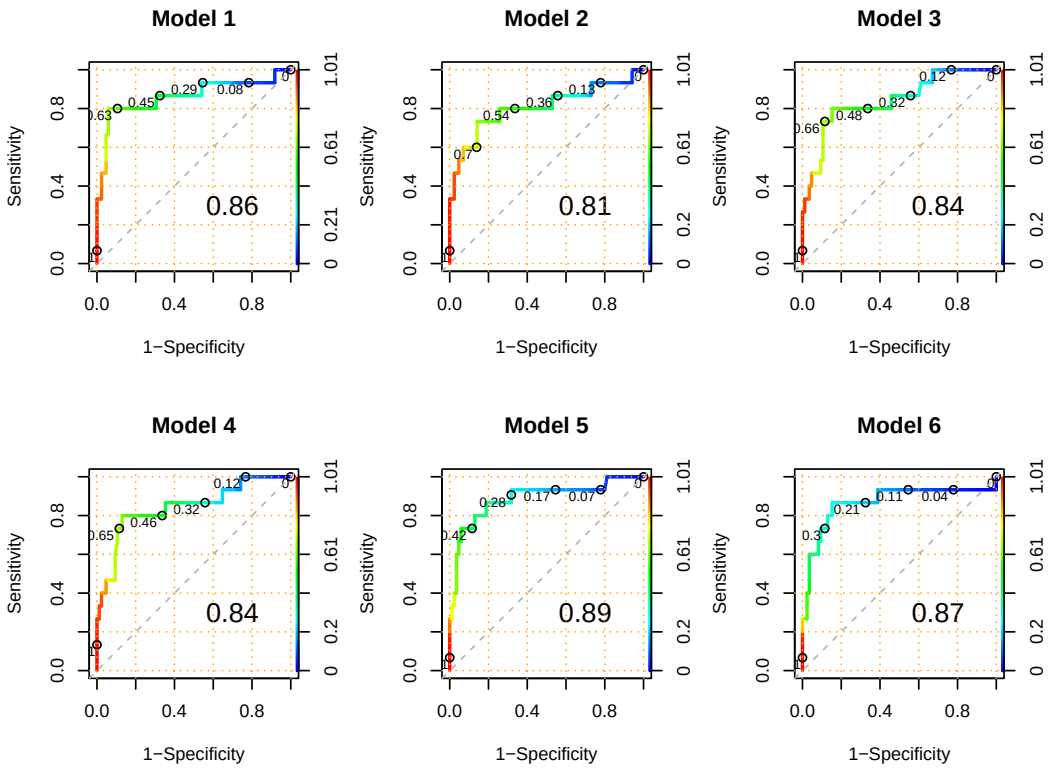

Figure A.6. The ROC curve of a randomly chosen trial of Simulation 1 when $N = 400$ and $\rho = 0.9$. The decimal under the diagonal is the AUC of each curve.

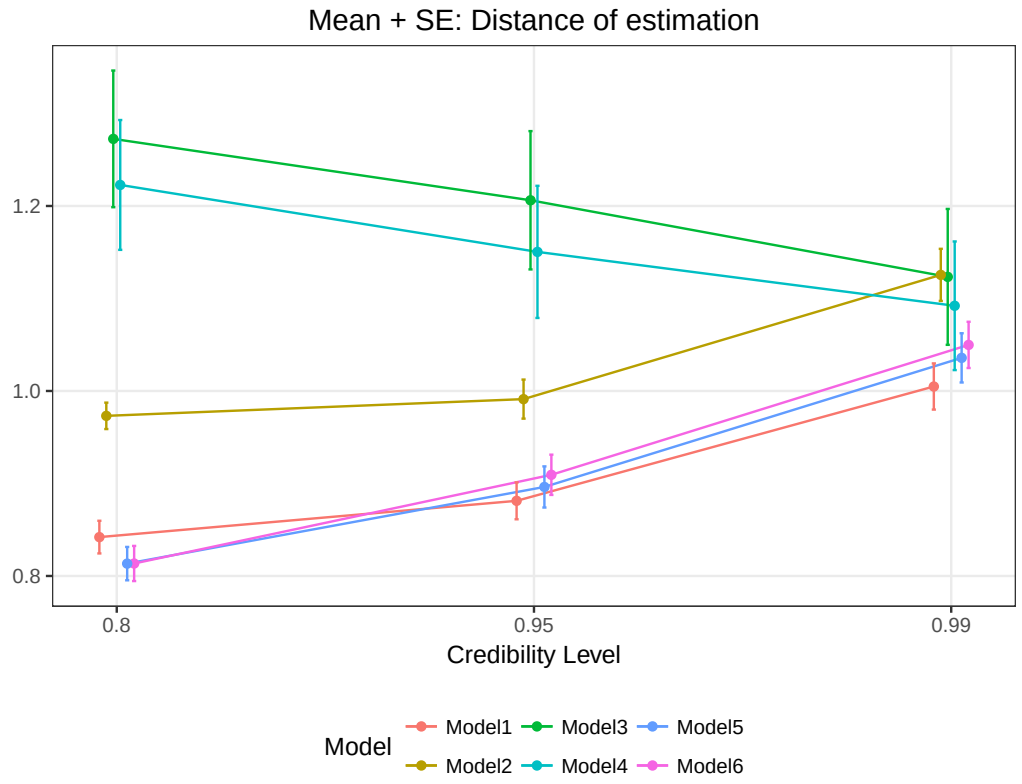

Figure A.7. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 100$ and $\rho = 0$.

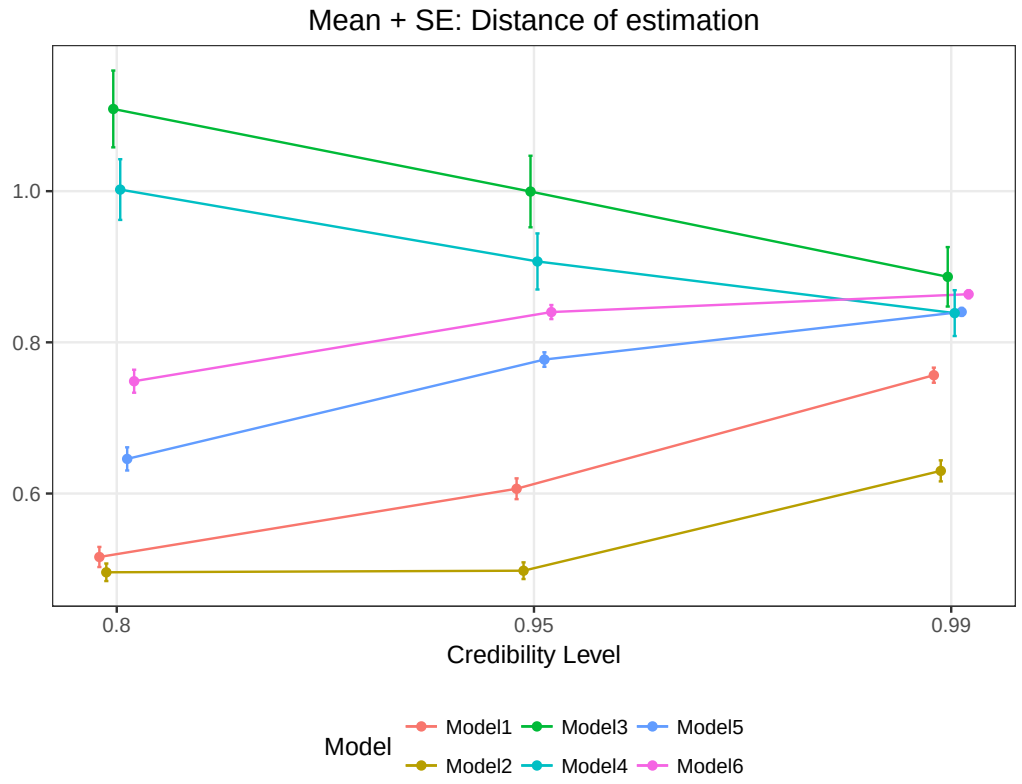

Figure A.8. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 100$ and $\rho = 0.5$.

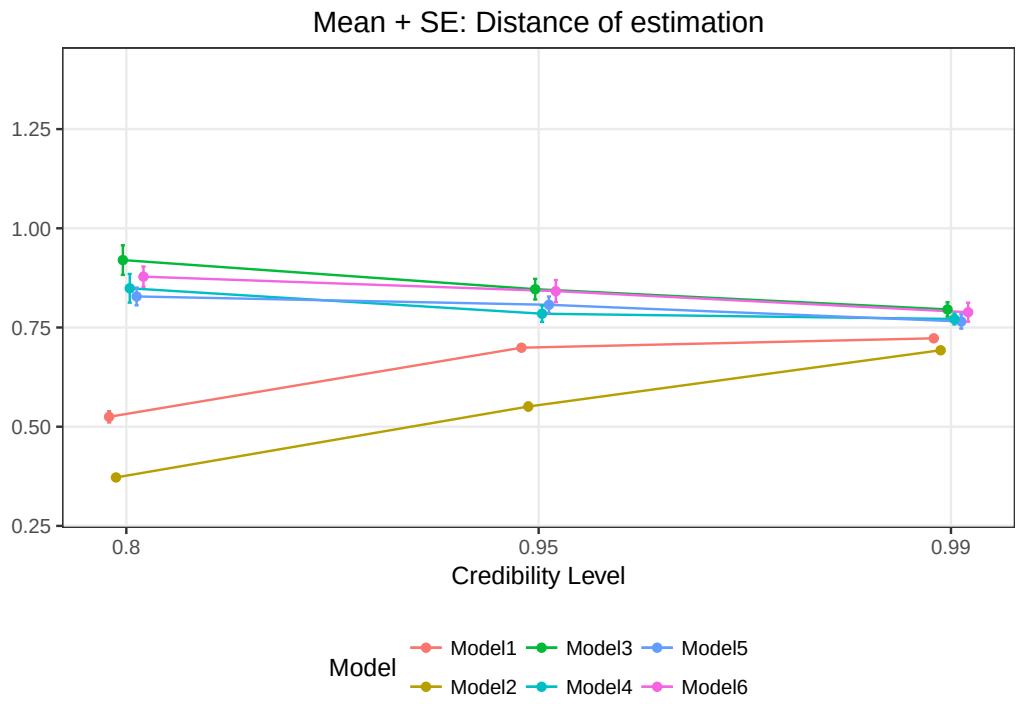

Figure A.9. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 100$ and $\rho = 0.9$.

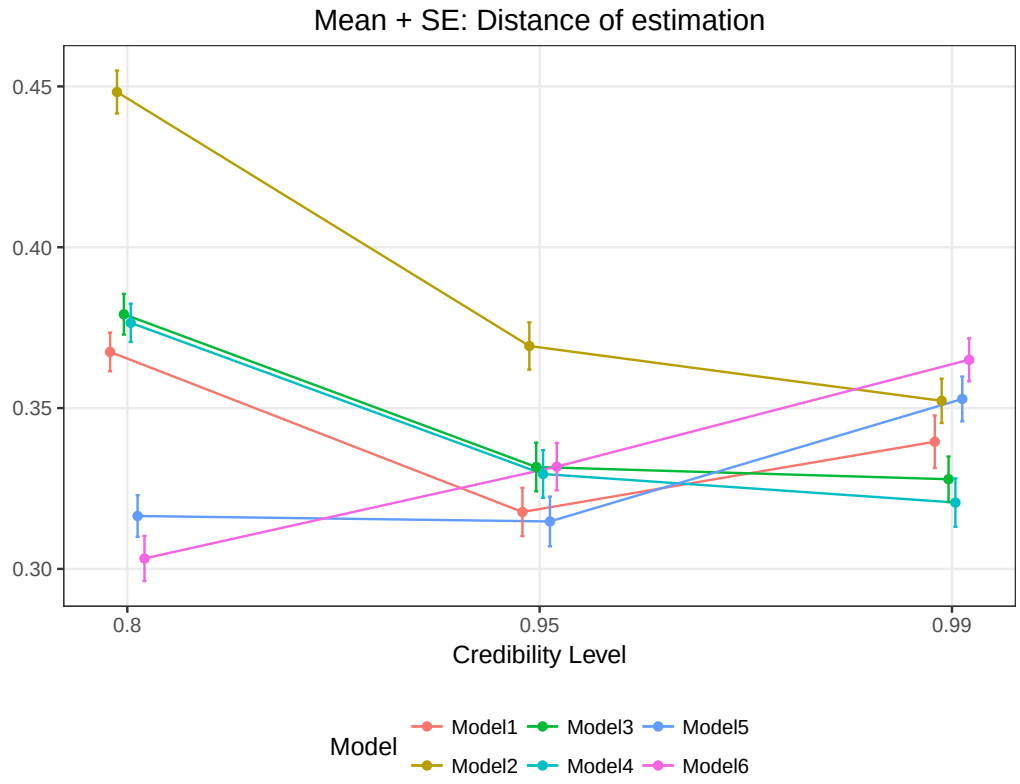

Figure A.10. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 400$ and $\rho = 0$.

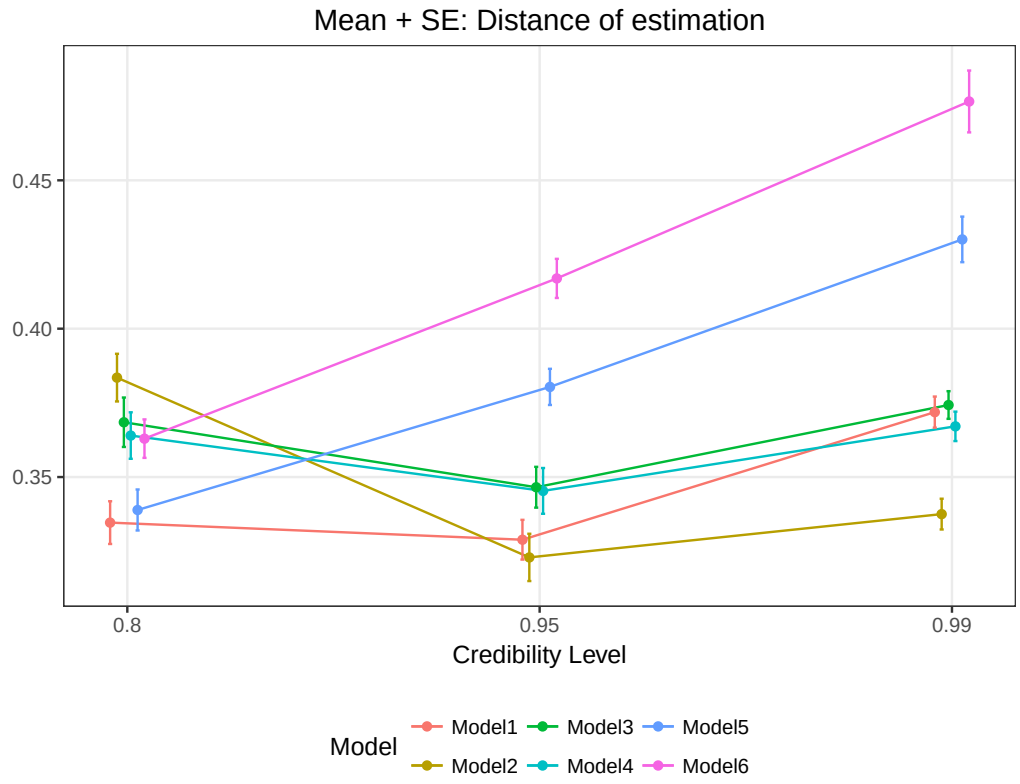

Figure A.11. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 400$ and $\rho = 0.5$.

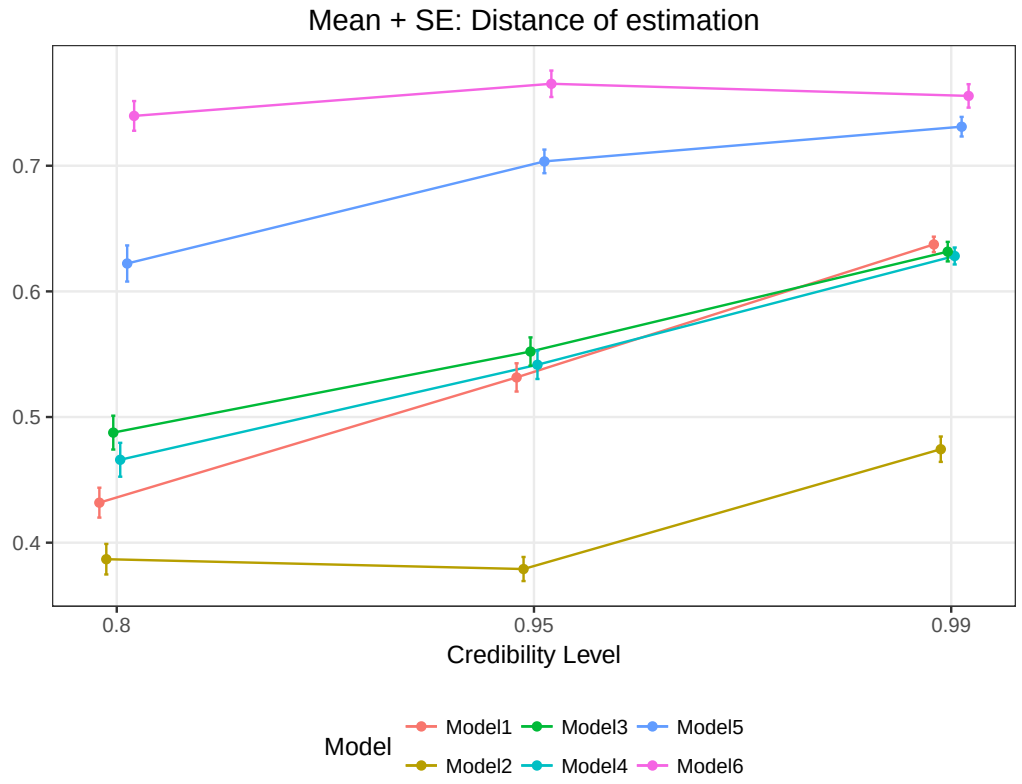

Figure A.12. The L_2- norm of parameter estimation deviance by six models at each credible level in Simulation 1 when $N = 400$ and $\rho = 0.9$.