BAYESIAN LASSO MODELS - WITH APPLICATION TO SPORTS DATA

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Di Gao

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Statistics

April 2018

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Bayesian Lasso Models – with application to Sports Data

By

Di Gao

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Rhonda Magel
Co-Chair

Dr. Gang Shen
Co-Chair

Dr. Megan Orr

Dr. Changhui Yan

Approved:

4/12/2018
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

Several statistical models were proposed by researchers to fulfill the objective of correctly predicting the winners of sports game, for example, the generalized linear model (Magel & Unruh, 2013) and the probability self-consistent model (Shen et al., 2015). This work studied Bayesian Lasso generalized linear models. A hybrid model estimation approach of full and Empirical Bayesian was proposed. A simple and efficient method in the EM step, which does not require sample mean from the random samples, was also introduced. The expectation step was reduced to derive the theoretical expectation directly from the conditional marginal. The findings of this work suggest that future application will significantly cut down the computation load.

Due to Lasso (Tibshirani, 1996)'s desired geometric property, the Lasso method provides a sharp power in selecting significant explanatory variables and has become very popular in solving big data problem in the last 20 years. This work was constructed with Lasso structure hence can also be a good fit to achieve dimension reduction. Dimension reduction is necessary when the number of observations is less than the number of parameters or when the design matrix is non-full rank.

A simulation study was conducted to test the power of dimension reduction and the accuracy and variation of the estimates. For an application of the Bayesian Lasso Probit Linear Regression to live data, NCAA March Madness (Men's Basketball Division I) was considered. In the end, the predicting bracket was used to compare with the real tournament result, and the model performance was evaluated by bracket scoring system (Shen et al., 2015).

ACKNOWLEDGEMENTS

Firstly, I acknowledge the help and support of my advisors, Dr. Rhonda Magel and Dr. Gang Shen with my immense gratitude. Thank you for giving me continuous guidance and support throughout my stay at NDSU and along this whole process of research. Thank you for providing me the opportunities and letting me present my research topic during the interviews, seminars, and conferences. I gained a lot from the practice, and it is valuable for my future academic career.

Secondly, I would like to thank my other committee members, Dr. Megan Orr, and Dr. Changhui Yan. Thank you for all your advice and valuable time. You both helped me by accommodating my schedule when we needed to set up committee meetings for my proposal oral and final defense.

Thirdly, I would also like to thank all other professors and staff from the Department of Statistics, Dr. Seung Won Hyun, Dr. Ron Degges, Ms. Taryn Chase, Ryan Niemann, and Dawn Halle. I appreciate your advice, help, and support of all kinds.

Next, I would need to thank my fiancée, Yu Wang. She was always there stood by me and support me. She was a good listener and accompanied me through the good times and bad.

Finally, and importantly, I would like to thank my parents, Mengqiu Wang and Xiaojun Gao. Six years ago, my parents made a difficult decision to send me overseas to the United States for Graduate education. They were eager to let me stay with them since I am the only child. However, they still sent me to the airport that day because they wanted me to explore the world. That was back in January 2012 when I started my master's program at University of Missouri. My dear parents are always supporting me and encouraging me with their best efforts and wishes.

This dissertation could not be completed without the efforts of all of you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. BACKGROUND

Sports data often refers to "win or lose" problem. A successful and efficient method to predict the winning team of a competition is to use generalized linear regression model (GLM). GLM is based on the concept of linear regression, which is a statistical approach for modeling the relationship between the response variable and the independent variable. One important aspect for GLM is the choosing of preferable link function, for example, the identity link, log link, etc. This research work was based on GLM and further combined several favorite modern statistical techniques. The application of this work initiated from sports data and would be able to apply to the expanded dataset.

For predicting binary outcomes, several researchers applied NCAA basketball tournament data due to the proper binary setup of "win or lose." Magel & Unruh (2013) introduced the generalized linear model. Logit link was used to fit the model. This generalized linear model with logit link is a natural fit for binary data because of the approaching curve and bounding range. Figure 1 shows the standard logistic cumulative distribution function $F(x)$ and standard normal CDF $\Phi(x)$. Note that $F(x)$ and $\Phi(x) \in (0,1)$ for all $x \in \mathbb{R}$.

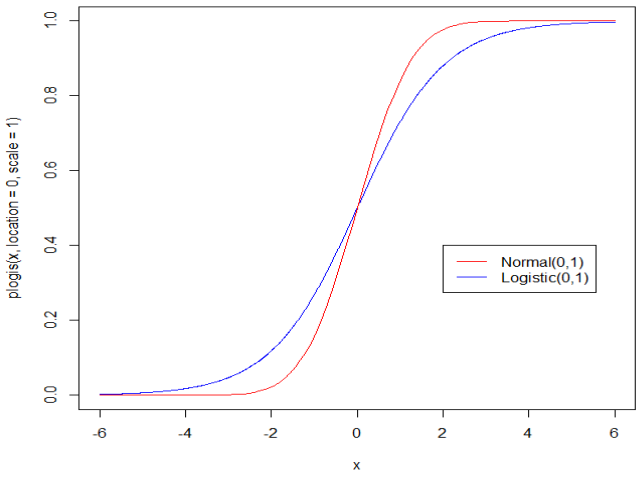

Figure 1. Logistic & Normal CDF

Shen et al. (2015) introduced the probability self-consistency model with the Cauchy link. The probability self-consistency model was first presented by Zhang (2012). The study of Shen et al. (2015) was based on a binomial generalized linear regression with a Cauchy link on the conditional probability of a team winning a game given its rival team.

Hua (2015) proposed a Bayesian inference by introducing logistic likelihood and informative prior. The study of Hua (2015) viewed Frequentist's GLM into a Bayesian approach. The benefit was submitting the prior information on top of the quality generalized linear regression model. The GLM with logit link can be expressed as:

$$log\left(\frac{p_i}{1-p_i}\right) = \eta_i = x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{id}\beta_d,\ i = 1, \ldots, n \quad (1.1)$$

Solving equation (1.1) w.r.t p_i, the result will then be:

$$p_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}$$

Therefore, the likelihood function for given covariates and corresponding coefficients is:

$$L(Y|X, \beta) = \prod_{i=1}^{n} f(y_i|\eta_i) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i} = \prod_{i=1}^{n} \left(\frac{e^{\eta_i}}{1+e^{\eta_i}}\right)^{y_i} \left(\frac{1}{1+e^{\eta_i}}\right)^{1-y_i} \quad (1.2)$$

The prior that Hua (2015) used was the winning probability of team seeding information. Schwertman et al. (1996) first used seed position to bracket NCAA Basketball Division I tournament. It seems reasonable that Hua (2015) involved seed winning probability as the prior because seeds were determined by a group of experts. For example, when seed 1 plays with seed 16. Seed 1's winning probability will be $\frac{16}{1+16} = 0.9412$. By Delta method (Oehlert, 1992), the prior distribution can be fully specified, and it was regarding coefficient β. The posterior was then fully defined, and the sampling technique used was Metropolis algorithm. The bracketing probability matrix was developed after fitting the model, which represented the prediction result.

As an extension of Shen et al. (2015) and Hua (2015), Shen et al. (2016) developed a Bayesian model using one other informative prior and generalized linear regression with logit link as well. The improvement from this study was from the prior part. The prior used in this study was historical information (competing results). Hua (2015) used winning probability generated from seed information. However, this information was not from the real counts. This prior was the information solely based on seed assignment. Shen et al. (2016) developed a new method of collecting previous game results. The actual counts of those results based on seeding setup were gathered, and the winning probability between seeds was defined based on all past season data on file. For example, we collected 15 years of data, and there was a total of 24 games played by seed 1 versus seed 2. We count a total of 10 wins by seed 1. Hence the probability for seed 1 winning the game over seed 2 is $\frac{10}{24} = 0.417$. By Delta method (Oehlert, 1992), the prior information in winning probability was transferred into coefficient β**.** Sampling Importance Resampling (SIR) algorithm was used to draw samples.

Hua (2015) and Shen et al. (2016) both used Bayesian inference with informative prior. These two studies worked well in NCAA's basketball data and turned out to have good prediction accuracy. However, there is still improvement can be made on top of these studies. For live data like NCAA basketball tournament, team statistics, which are the covariates, have large dimension. A dimension reduction is necessary because of the limitation of games played (small sample size). With the help of dimension reduction, valuable information can also be provided to team coaches. The coaches can then be offered scientific knowledge that which team statistics are important. They can then emphasize on these techniques in their future training. To fulfill the need for dimension reduction, Lasso was proposed.

The Lasso of Tibshirani (1996) estimates linear regression coefficients through L_1-constrained least squares. Similar to ordinary linear regression, Lasso is usually used to estimate the regression parameters $\beta = (\beta_1, \dots, \beta_d)'$ in the model:

$$y = \mu \mathbb{1}_n + X\beta + \varepsilon \tag{1.3}$$

where y is the $n \times 1$ vector of responses, μ is the overall mean, X is the $n \times d$ design matrix, and ε is the $n \times 1$ vector of errors. The error terms are independent and identically distributed, and they follow the normal distribution with mean 0 and unknown variance σ^2. The Lasso estimates are on top of the constraint of L_1 norm. For convenience, Tibshirani (1996) viewed this as L_1-penalized least squares estimates. They achieve

$$min \left\{ (\tilde{y} - x\beta)'(\tilde{y} - x\beta) + \lambda \sum_{j=1}^{d} |\beta_j| \right\} \tag{1.4}$$

for some $\lambda \geq 0$, where $\tilde{y} = y - \bar{y}\mathbb{1}_n$.

The penalty term in equation (1.4) is the critical part for Lasso. Tibshirani (1996) suggested that Lasso estimates can be interpreted as posterior mode estimates when the regression parameters have independent and identical Laplace (i.e., double-exponential) priors, which lead to the Bayes view of Lasso.

Park & Casella (2008) then proposed a fully Bayesian analysis using a conditional Laplace prior specification of the form:

$$\pi(\beta|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{\frac{-\lambda|\beta_j|}{\sqrt{\sigma^2}}} \tag{1.5}$$

and the noninformative scale-invariant marginal prior $\pi(\sigma^2)$ is proportion to $\frac{1}{\sigma^2}$. Park & Casella (2008) also found out that the Bayesian Lasso estimates appear to be a compromise between the Lasso and ridge regression estimates (Hoerl & Kennard, 1970). Like Park & Casella (2008) mentioned, Bayesian lasso's paths are smooth, like ridge regression, but are more similar in shape

to the Lasso paths, particularly when the L_1 norm is relatively small. They also provide a hierarchical model which is ready for Gibbs sampler. Gibbs sampler, a sampling technique, was first described by Stuart & Geman (1984). Andrews & Mallows (1974) first developed the scale mixtures of Normal Distributions. X, double exponential, may be generated as the ratio Z/V where Z and V are independent, and Z has a standard normal distribution when $\frac{1}{2}V^2$ is exponential. In other words, the Laplace prior can be expressed as a zero-mean Gaussian prior with an independent exponentially distributed variance.

Based on the representation of the Laplace distribution as a scale mixture of normal and exponential density:

$$\frac{a}{2}e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}} e^{-\frac{z^2}{2s}} \frac{a^2}{2} e^{-\frac{a^2 s}{2}} ds,\ a > 0 \tag{1.6}$$

Park & Casella (2008) suggests the following hierarchical representation of the full model:

$$y|\mu, X, \beta, \sigma^2 \sim N_n(\mu\, \mathbb{1}_n + X\beta\,, \sigma^2 \mathrm{I}_n), \mu\ given\ independent\ flat\ prior$$

$$\beta|\sigma^2, {\tau_1}^2, \dots, {\tau_d}^2 \sim N_d(O_d, \sigma^2 D_\tau), \qquad \sigma^2, {\tau_1}^2, \dots, {\tau_d}^2 > 0$$

$$D_\tau = diag({\tau_1}^2, \dots, {\tau_d}^2)$$

$$\sigma^2, {\tau_1}^2, \dots, {\tau_d}^2 \sim \pi(\sigma^2) \prod_{j=1}^{d} \frac{\lambda^2}{2} e^{\frac{-\lambda^2 {\tau_j}^2}{2}}$$

When calculating the Bayesian Lasso parameter (λ), Park & Casella (2008) proposed the empirical Bayes by marginal maximum likelihood. Casella (2001) proposed a Monte Carlo EM algorithm (Dempster et al., 1977) that complements a Gibbs sampler and provides marginal maximum likelihood estimates of Hyperparameters. Each iteration of the algorithm involves running the Gibbs sampler using a λ value estimated (E-M algorithm) from the sample of the

previous iteration. Casella (2001) produced the estimated posterior distribution and provided the convergence:

$$\hat{\pi}\left(\theta|X,\hat{\psi}\right) = \frac{1}{M}\sum_{j=1}^{M}\pi\left(\theta|X,\hat{\psi},\lambda^{(j)}\right) \tag{1.7}$$

For any measurable set A, we have for each $i = 1, 2, \cdots, d$,

$$\int_A \left|\frac{1}{M}\sum_{j=1}^{M}\pi\left(\theta_i|X,\hat{\psi},\lambda^{(j)}\right) - \pi(\theta_i|X,\psi)\right| \mathrm{d}\theta_i \to 0 \tag{1.8}$$

$$as\ M, d \to \infty$$

Bae & Mallick (2004) provided a process of applying the Bayesian Lasso to Probit model for Gene selection problem. The data is about gene expression level, hence the dimension of covariates (different genes) is very high.

$$X = \begin{pmatrix} X_{11} & \cdots & X_{1d} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nd} \end{pmatrix}, \quad a\ (n \times d)\ matrix$$

The response is a binary setup with "normal or cancer." The probit model was assigned a Laplace prior for β to promote sparsity, so that irrelevant parameters were set exactly to zero. Bae & Mallick (2004) then expressed the Laplace prior distribution as a scale mixture of normal priors, which is equivalent to a two-level hierarchical Bayesian model:

$$\pi(\beta_i|\sigma^2) = \int_0^\infty \pi(\beta_i|\lambda_i)\pi(\lambda_i|\gamma)\, d\lambda_i \sim Laplace(0, \gamma^{-\frac{1}{2}}) \tag{1.9}$$

Bae & Mallick (2004) assign an exponential distribution for the prior distribution of λ_i, which is equivalent to assigning a Laplace prior for β. Hence, their prior is as follows (The prior distribution of β):

$$\beta|\Lambda \sim N(O, \Lambda)$$

where $O = (0, \ldots, 0)', \Lambda = diag(\lambda_1, \ldots, \lambda_d)$ and λ_i is the variance of β_i.

$$\Lambda \sim \prod_{i=1}^{d} Exp(\gamma)$$

The process is similar to the Lasso model but has added flexibility due to the choices of multiple λs against one choice in the Lasso method.

2. INTRODUCTION

For ordinary linear regression $y = \mu + x'\beta + \varepsilon$, when the number of observations is less than the dimension of parameters β, $\beta = (\beta_1, \beta_2, \ \dots \ , \beta_d)'$, the design matrix X is non-full rank. Estimation of β requires special treatment. A classical approach to solve the problem is the ridge regression operator which minimizes

$$(y - X\beta)'(y - X\beta) + \lambda\|\beta\|_2^2 \tag{2.1}$$

where λ is the tuning parameter, and $\|\beta\|_2^2$ refers to the L_2 norm such that $\|\beta\|_2^2 = \left(\sum_{j=1}^{d} \beta_j^2\right)^{1/2}$. Tibshirani (1996) proposed Least Absolute Shrinkage and Selection Operator (Lasso) which minimizes

$$(y - X\beta)'(y - X\beta) + \lambda\|\beta\|_1 \tag{2.2}$$

where λ is the tuning parameter controlling the power of shrinkage, $\|\beta\|_1$ refers to the L_1 norm such that $\|\beta\|_1 = \sum_{j=1}^{d}|\beta j|$. Thanks to Lasso's desired geometric property, it provides a much sharper power in selecting significant explanatory variables than the classical, alternative approach and has become very popular in dimension reduction in the past 20 years. Figure 2 shows the geometric property of Lasso under the constraint of two dimension L_1 norm, which is $|\beta_1|+ |\beta_2| \leq t$ for some $t > 0$.

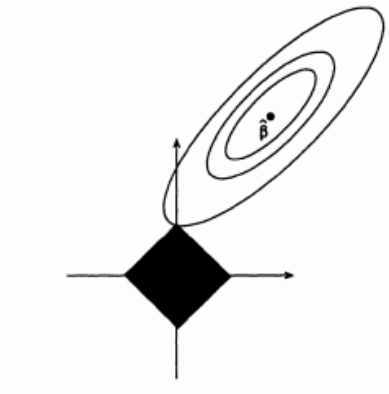

Figure 2. The geometric expression for two dimensions β

The choice of λ is usually via cross-validation. Using a hierarchical Bayesian model and treating λ as a hyper-parameter, Park & Casella (2008) proposed a Bayesian approach for estimation of β in ordinary linear regression above.

Lasso may be easily extended to probit linear regression. Probit linear regression model was first introduced by Bliss (1934). The probit linear regression model uses probit link $\Phi^{-1}(\cdot)$, where $\Phi^{-1}(\cdot)$ is the inverse of the cumulative density function of standard normal. The probit linear regression model expresses as:

$$\Phi^{-1}(p_i) = x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{id}\beta_d, \;\; i = 1, \ldots, n \tag{2.3}$$

Let *y* be a binary response variable with the probability model Bernoulli (*p*). One way to state the model is to assume that there is a latent variable *z* such that:

$$z = x'\beta + \varepsilon, \;\; \varepsilon \sim N(0,1) \tag{2.4}$$

In probit model, we observe that:

$$y_i = \begin{cases} 0, & if\ z_i \leq 0 \\ 1, & if\ z_i > 0 \end{cases} \tag{2.5}$$

Note that $z_i > 0 \;\Rightarrow\; x'\beta + \varepsilon > 0 \Rightarrow\; \varepsilon > -x'\beta$, then

$P(y = 1) = P(z_i > 0) = P(\varepsilon > -x'\beta) = \Phi(x'\beta)$

The likelihood of the probit linear regression is simply:

$$\prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

$$\prod_{i=1}^{n} [\Phi(x'\beta)]^{y_i}[\Phi(-x'\beta)]^{1-y_i} \tag{2.6}$$

In the case where the number of observations is less than the dimension of β, i.e., the design matrix X for the regression is non-full rank, like that in ordinary linear regression, one may apply Lasso for estimation of β, which minimize the negative of the log-likelihood:

$$-\sum_{i=1}^{n} y_i \, log\big(\Phi(X'\beta)\big) - \sum_{i=1}^{n}(1-y_i)\log\big(\Phi(-X'\beta)\big) + \lambda\|\beta\|_1 \qquad (2.7)$$

where λ is again the tuning parameter. In this research work, we considered the Bayesian approach as proposed by Park & Casella (2008) for estimation of β in the probit linear regression above. We applied a hybrid approach of full and Empirical Bayesian like Park & Casella (2008). When reaching to the sampling procedure, we used Gibbs Sampler and EM algorithm to acquire samples. Gibbs Sampler is named after Dr. Josiah Willard Gibbs and was first described by Stuart & Geman (1984). Expectation- maximization (EM) algorithm is used to find maximum likelihood (MLE) or maximum a posteriori (MAP) using an iterative method. EM was first explained by Dempster, Laird & Rubin (1977). Our contribution for this research work was in the EM step of estimating λ. We used the theoretical expectation derived from the conditional marginal directly instead of deriving from the sample mean of the random samples, which greatly cut down the computation load, and made the computation algorithm much more efficient. This more efficient EM computation algorithm did not lose any of the power and accuracy.

A simulation was done to test the power of shrinkage and variation of the estimates. For an application of Bayesian Lasso probit linear regression to live data, we studied NCAA March Madness data as well. In the end, we also presented the previous full Bayesian model to bracket and made a comparison.

3. METHODOLOGY

Based on the concepts from previous works and references, a new Bayesian Lasso model was developed. This new model used probit link of the generalized linear model with a full Bayesian setup. By proper setting up the hierarchical priors, this full Bayesian Model performed similar to the beneficial property of Lasso and automatically omit some covariates with less useful information.

3.1. Bayesian Hierarchical Lasso Model

The same concept as inference was implemented here. Based on the information introduced from background part. We need to perform MLE based on the probit model, which is also the minimization of the negative log-likelihood. Adding up the Lasso shrinkage term $\lambda\|\beta\|_1$, we try to minimize the following function:

$$-\sum_{i=1}^{n} y_i \log\big(\Phi(X'\beta)\big) - \sum_{i=1}^{n}(1-y_i)\log\big(\Phi(-X'\beta)\big) + \lambda\|\beta\|_1 \qquad (3.1)$$

We know that the very last item $\lambda\|\beta\|_1$ geometrically provide the "Lasso" property. To have this specific structure in the posterior function, we proposed Laplace distribution. Because if $\beta \sim \text{Laplace}(0, \frac{1}{\lambda})$, then the probability density function will have the following form: $\frac{\lambda}{2}\exp(-\lambda|\beta|)$. If this term is extended to high dimension, then it can lead to $\prod_{i=1}^{\mathrm{d}} \left\{\frac{\lambda}{2}\exp(-\lambda|\beta|)\right\} = \left(\frac{\lambda}{2}\right)^d \exp\big(-\lambda\sum_{i=1}^{d}|\beta_i|\big)$. This can be rewritten to $\left(\frac{\lambda}{2}\right)^d \exp(-\lambda\|\beta\|_1)$, which happen to be the desired format. Hence, the problem simplified to construct Laplace distribution.

Tibshirani (1996) firstly suggested that the Bayesian approach involve a Laplace prior distribution of β. Genkin (2004) also proposed the similar structure for Lasso probit regression

model. First, β_j need to arise from a normal distribution with mean 0, and variance ${\tau_j}^2$, the distribution is as follows:

$$f(\beta_j \,|{\tau_j}^2) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{{\beta_j}^2}{2{\tau_j}^2}} \tag{3.2}$$

The assumption of mean 0 indicates our belief that β_j could be close to zero. The variance ${\tau_j}^2$ are positive constants. A small value of ${\tau_j}^2$ represents a prior belief that β_j is close to zero. Conversely, a large value of ${\tau_j}^2$ represents a less informative prior belief.

Tibshirani (1996) suggested ${\tau_j}^2$ arises from a Laplace prior (double exponential distribution) with density

$$f({\tau_j}^2 \,|{\lambda_j}^2) = \frac{{\lambda_j}^2}{2}\, e^{-\frac{{\lambda_j}^2{\tau_j}^2}{2}} \tag{3.3}$$

Integrating out ${\tau_j}^2$ can lead us to the distribution of β_j as follows:

$$f(\beta_j \,|\lambda_j) = \frac{\lambda_j}{2}\, e^{-\frac{\lambda_j|\beta_j|}{2}} \tag{3.4}$$

In other words,

$$f(\beta_j|\lambda_j) = \int_0^\infty f(\beta_j \,|{\tau_j}^2)\, f({\tau_j}^2 \,|{\lambda_j}^2) d{\tau_j}^2 \sim Laplace(0, {\lambda_j}^{-1}) \tag{3.5}$$

We need to prove the following result:

If $f(\beta_j \,|{\tau_j}^2) = \frac{1}{\sqrt{2\pi{\tau_j}^2}} \exp\left\{-\frac{1}{2}\frac{{\beta_j}^2}{{\tau_j}^2}\right\}$ and $f({\tau_j}^2 \,|{\lambda_j}^2) = \frac{\lambda^2}{2} \exp\left\{-\frac{\lambda^2}{2}{\tau_j}^2\right\}$,

then $f(\beta_j|\lambda) = \frac{\lambda}{2}\exp\{-\lambda|\beta_j|\}$.

The proof of this result reported in Appendix A. $\frac{\lambda}{2}\exp\{-\lambda|\beta|\}$ is the density of Laplace distribution with location parameter 0 and scale parameter $\frac{1}{\lambda}$. Hence, above process proves the density of double exponential distribution.

Figure 3 shows the plot of Laplace density function together with normal density function. The smoother curve represents normal density, and the sharper curve represents Laplace density.

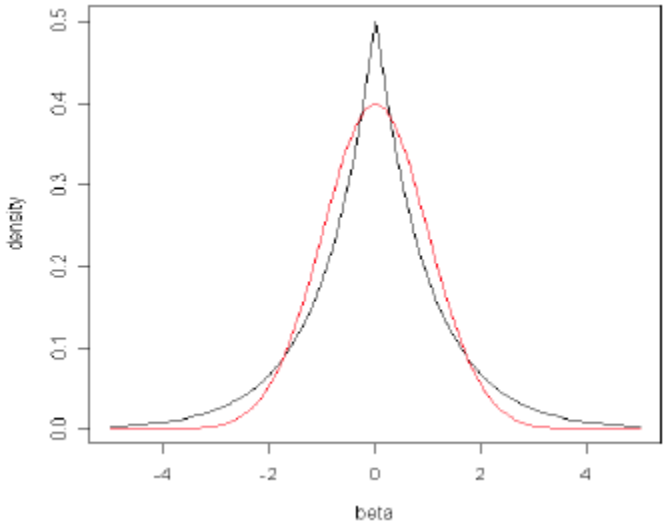

Figure 3. Normal versus Laplace (Double Exponential) density

Noting from the proof of Laplace distribution, we can set up the proper priors to construct the new model. The new hierarchical model was defined to accommodate the probit distribution. Following is the hierarchical representation of the full model (Bayesian Lasso Probit Hierarchical Model):

$$y_i = \mathbb{1}_{(0,+\infty)}(z_i)$$

$$z_i|\beta \sim N(x_i'\beta, 1)$$

$$\beta|{\tau_1}^2, \ \ldots, \ {\tau_d}^2 \sim N_d(O_d, D_\tau), where\ D_\tau = diag({\tau_1}^2, \ \ldots, \ {\tau_d}^2)$$

$${\tau_1}^2, \ \ldots, \ {\tau_d}^2 \sim Exp(\frac{\lambda^2}{2})$$

$$\lambda \sim \pi(\lambda) \propto c$$

$${\tau_1}^2, \ \ldots, \ {\tau_d}^2 > 0$$

Based on probit model's property, a latent variable z is needed to construct the full model. This z variable is the bridge or link between response variable and parameters. That is why we stated the following two lines into the full model:

$$y_i = \mathbb{1}_{(0,+\infty)}(z_i)$$

$$z_i|\beta \sim N(x_i'\beta, 1)$$

Because of the probit link, $z_i|\beta$ will be either larger than 0 or smaller than 0. For those z_i's larger than 0, we will have y_i equals 1; for those z_i's smaller than 0, we will have y_i equals 0. For the rest of the hierarchical lines, λ provides information to τ , τ provides information to β and β decides the z_i's. Figure 4 is the picturized model structure.

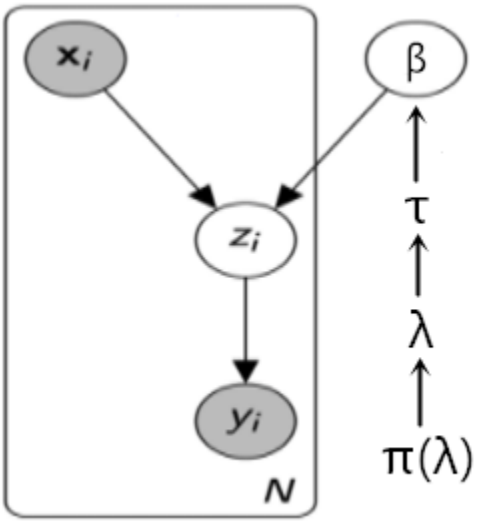

Figure 4. Augmented Binary Probit Regression hierarchical model structure

From the full model, the posterior can be fully specified. Due to the Bernoulli property, $Y_i \sim ind\ Ber(p_i)$. Due to the probit link, $p_i = P(y_i = 1) = \Phi(x_i'\beta)$. The posterior will then be $f(z,\beta,\tau,\lambda|y) \propto f(y|z,\beta,\tau,\lambda) \times f(z,\beta,\tau,\lambda)$. We have conditional independency for term $f(y|z,\beta,\tau,\lambda)$, hence $f(y|z,\beta,\tau,\lambda) = f(y|z)$. Similarly based on the conditional independency, $f(z,\beta,\tau,\lambda) = f(z|\beta) \times f(\beta|\tau) \times f(\tau|\lambda) \times \pi(\lambda)$. The posterior then can be further expressed as

$f(z,\beta,\tau,\lambda|y) \propto f(y|z) \times f(z|\beta) \times f(\beta|\tau) \times f(\tau|\lambda) \times \pi(\lambda)$. The posterior is in detail proportion to the following term:

$$\prod_{i=1}^{N}\{[TN(x_i'\beta,1,0,+\infty)]^{y_i} \times [TN(x_i'\beta,1,-\infty,0)]^{(1-y_i)}\} \times \varphi(\beta;0,D_\tau) \times \prod_{j=1}^{d}\frac{\lambda^2}{2}e^{\frac{-\lambda^2\tau_j{}^2}{2}} \times c$$

where *TN* represents truncated normal density, $\varphi(.)$ represents normal density.

3.2. Computation

The Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method of simulating a random sample from a multivariate posterior. Based on this concept, two sampling techniques were proposed. One is full Bayesian Gibbs Sampler, and the other one is Hybrid Bayesian with EM algorithm.

3.2.1 Full Bayesian Gibbs Sampler

After we discovered the exact format of the posterior, we need to get information from the posterior. That is, in other words, sampling. We have four parameters, so we need to derive four conditional densities.

(a). $z_i|\,\tau,\beta,\lambda,y_i = [TN(x_i'\beta,1,0,+\infty)]^{y_i} \times [TN(x_i'\beta,1,-\infty,0)]^{(1-y_i)}$ where

$$TN(x_i'\beta,1,0,+\infty) = \frac{exp\left(-\frac{1}{2}(z_i-x_i'\beta)'(z_i-x_i'\beta)\right)}{\sqrt{2\pi}\Phi(x_i'\beta)}\mathbb{1}_{(0,+\infty)}(z_i) \tag{3.6}$$

$$TN(x_i'\beta,1,-\infty,0) = \frac{exp\left(-\frac{1}{2}(z_i-x_i'\beta)'(z_i-x_i'\beta)\right)}{\sqrt{2\pi}\Phi(-x_i'\beta)}\mathbb{1}_{(-\infty,0)}(z_i) \tag{3.7}$$

(b). $\beta|\tau,z,\lambda,y \propto \prod_{i=1}^{N}\varphi(z_i;x_i'\beta,1) \times \varphi(\beta;0,D_\tau)$

$$\beta|\tau,z,\lambda,y \sim N(A^{-1}X'Z,A^{-1}),\ where\ A = D_\tau^{-1} + X'X \tag{3.8}$$

(c). $\tau^2|z,\beta,\lambda,y \propto \varphi(\beta;0,D_\tau) \times \prod_{j=1}^{d}\frac{\lambda^2}{2}e^{\frac{-\lambda^2\tau_j{}^2}{2}}$

$$\tau_j^{-2}\big|z,\beta,\lambda,y \sim IG\left(\mu_j,\lambda^2\right) with\ \mu_j = \lambda\big|\beta_j\big|^{-1} \tag{3.9}$$

(d). $\lambda|\tau,\beta,z,y \propto \prod_{j=1}^{d} \frac{\lambda^2}{2} e^{\frac{-\lambda^2 {\tau_j}^2}{2}} \times$ ($\pi(\lambda)$=c)

$$\lambda|\tau,\beta,z,y \sim Gamma(d+1, \tfrac{1}{2}\textstyle\sum_{j=1}^{d} {\tau_j}^2) \tag{3.10}$$

For deriving $\lambda|\tau,\beta,z,y$, this is still under providing full prior information of λ. If it is impossible or unwilling to specify the prior for λ, E-M algorithm can be implemented for the calculation.

3.2.2 Hybrid Bayesian

This method is the hybrid of both Full and Empirical Bayesian. The difference for this hybrid Bayesian computation process is from the parameter λ. We do not have any prior distribution for λ. The updated posterior will then be treated as the likelihood, and the complete log-likelihood is as follows:

$$\sum_{i=1}^{N}\{y_i log[TN({x_i}'\beta,1,0,+\infty)] + (1-y_i) log[TN({x_i}'\beta,1,-\infty,0)]\} - \tfrac{1}{2}\textstyle\sum_{j=1}^{d}\frac{\beta_j^2}{{\tau_j}^2} -$$

$$\log\left((2\pi)^{\frac{d}{2}}\right) - \textstyle\sum_{j=1}^{d}\log(\tau_j) - dlog2 + dlog(\lambda^2) - \frac{\lambda^2}{2}\textstyle\sum_{j=1}^{d}{\tau_j}^2 \tag{3.11}$$

where

$$TN({x_i}'\beta,1,0,+\infty) = \frac{exp\left(-\frac{1}{2}(z_i - {x_i}'\beta)'(z_i - {x_i}'\beta)\right)}{\sqrt{2\pi}\Phi({x_i}'\beta)} \mathbb{1}_{(0,+\infty)}(z_i)$$

$$TN({x_i}'\beta,1,-\infty,0) = \frac{exp\left(-\frac{1}{2}(z_i - {x_i}'\beta)'(z_i - {x_i}'\beta)\right)}{\sqrt{2\pi}\Phi(-{x_i}'\beta)} \mathbb{1}_{(-\infty,0)}(z_i)$$

Under the condition of all the information from the previous iteration, the E-step only focused on the very last term of the above equation. Since we know the conditional marginal

distribution of $f(\tau_j{}^2)$, we do not need to take the sample mean as the expected value; this will highly reduce the computation load. We have $\frac{1}{\tau_j{}^2}$ follows the Inverse Gaussian distribution.

Note that $E[\tau_j{}^2|y,\ \lambda,\beta] = \frac{1}{\mu_j} + \frac{1}{\lambda^2} = \frac{|\beta_j\lambda|+1}{\lambda^2}$, then

E step: $Q(\lambda|\lambda^{(k-1)}) = d(log\lambda^2) - \frac{\lambda^2}{2}\sum_{j=1}^{d} E[\tau_j{}^2|y,\ \lambda^{(k-1)},\beta] + terms\ not\ involving\ \lambda$

M step: $\lambda^{(k)} = (2d)^{1/2}\left(\sum_{j=1}^{d} \frac{\lambda^{(k-1)}|\beta_j{}^{(k-1)}|+1}{(\lambda^{(k-1)})^2}\right)^{-1/2}$

4. NUMERICAL EXPERIMENTS

Based on the structure of the model, this Bayesian Lasso should perform like ordinary Lasso which have great power in dimension reduction (set some coefficient exact to 0). A simulation study was applied to evaluate the efficiency of the model in dimension reduction.

4.1. Simulation

4.1.1 Data Generation

The NCAA March Madness data will have more than ten years of the result and 15 team statistics as the covariates. Thus, we will randomly generate these team statistics' data with a 16 years setup. The data will then be stored in a 1024 by 15 matrix. To ensure the data is closer to the "real-world," some of the covariates are dependent, and some are independent. To avoid the noise of dimension reduction from data generation, we will check the data and make sure the mean of each independent variable is away from 0. Example data generation of the fifteen covariates are as follows:

$x_1 \sim N(3, 1^2)$ $x_2 \sim N(X_1, 1^2)$ $x_3 \sim N(X_2, 2^2)$ $x_4 \sim Unif(5, 10)$

$x_5 \sim Unif(x_4, x_4 + 3)$ $x_6 \sim N(3.5, 1^2)$ $x_7 \sim N(X_6, 1^2)$ $x_8 \sim x_4 + x_7$

$x_9 \sim Unif(x_8, x_8 + 3)$ $x_{10} \sim Unif(x_9, x_9 + 1)$ $x_{11} \sim N(5, 1^2)$ $x_{12} \sim N(X_{11}, 1^2)$

$x_{13} \sim N(X_{12}, 2^2)$ $x_{14} \sim Unif(5, 10)$ $x_{15} \sim Unif(x_{14}, x_{14} + 3)$

It is essential to set some coefficients to zero before modeling so that we could verify the power of this model in dimension reduction. We expected, after model fit, those zero coefficients could be detected. In other words, the dimension reduction efficiency could be verified. Based on this designed setup, Table 1 shows the coefficients' true values:

Table 1. Coefficient true values (β)

β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}	β_{13}	β_{14}	β_{15}
4	-4	5	7	-6	0	0	0	0	0	0	0	0	0	0

We have $\beta' = (4, -4, 5, 7, -6, \ 0, 0, \dots, 0)$; there are a total of 10 0s assigned to covariates from the 6th to the 15th.

Based on equation $p_i = \Phi(x_i'\beta)$. The probability vector can be obtained. The probability vector should be 1024×1. The response data can then be generated using this weighted probability vector. That is, generate a random binomial distribution of n=1 using the associated probability vector. The response data vector would also be 1024×1, and the values are either 0 or 1.

4.1.2 Results

For the Gibbs Sampler process, we need a reasonable iteration count. For this simulation process, we chose 270,000 as the number of iterations due to the large lag to converge, and 20,000 as the number of burn-in. We need to burn first 20,000 samples to remove the unstable ones. After the first 20,000 burned, we choose 250 as the slicing range to remove the correlation. That will leave us $\frac{270{,}000-20{,}000}{250} = 1000$ samples. Table 2 is the summary information regarding λ.

Table 2. Summary table for posterior distribution of λ (Gibbs Sampler)

Min	Q1	Median	Mean	Q3	Max	S.D.
0.1960	0.5401	0.6623	0.6787	0.7921	1.6522	0.1952

The estimate of λ under Gibbs Sampler procedure is 0.6787. The standard deviation of the λ samples is 0.1957, which is relatively small. That means theses samples are stable and vary

around the sample mean 0.6787. The simulation study from Park & Casella (2008) also suggested that λs maintain in this range. Table 3 provides other summary for the coefficients:

Table 3. Summary table for posterior distribution of coefficients β (Gibbs Sampler)

Posterior Summaries						
Parameter	N	True value	Mean	Std.Dev	2.5%tile	97.5%tile
β_1	1000	4	3.7035	0.8325	2.1188	5.2545
β_2	1000	-4	-3.8061	0.8095	-5.3352	-2.2797
β_3	1000	5	5.0722	1.0386	3.1024	7.0238
β_4	1000	7	5.8721	1.7886	2.5043	9.2839
β_5	1000	-6	-5.8790	1.2655	-8.2550	-3.5115
β_6	1000	0	-0.1001	0.2698	-0.6623	0.4210
β_7	1000	0	-0.7518	1.2195	-3.7927	1.1280
β_8	1000	0	1.1584	1.2727	-0.8094	4.2870
β_9	1000	0	-0.0454	0.5098	-1.1142	0.9897
β_{10}	1000	0	-0.2427	0.4809	-1.2415	0.7298
β_{11}	1000	0	0.1223	0.2381	-0.3398	0.5933
β_{12}	1000	0	0.0983	0.2012	-0.2875	0.5162
β_{13}	1000	0	0.0297	0.1029	-0.1714	0.2339
β_{14}	1000	0	-0.0714	0.2485	-0.5534	0.3998
β_{15}	1000	0	-0.0336	0.2270	-0.5060	0.3935

The estimated D_τ (Gibbs Sampler) is as follows:

$$\hat{D_\tau} = diag(10.50,11.25,13.10,14.46,14.11,3.93,5.42,5.82,4.36,4.13,3.82,3.46,3.67,3.91,4.04)$$

For using EM (Hybrid Bayesian), the setup part remains the same. The difference is that instead of sampling λ with a flat prior, each iteration we input the theoretical value from expected maximization. Table 4 provides the summary for posterior distribution of λ and Table 5 provides

the summary information for coefficients β. For the samples of λ, they vary around 0.5580 with standard deviation 0.0965. This still complies with the simulation study from Park & Casella (2008). The computation speed was noticeably faster when using an Empirical Bayesian MAP when compared with the previous Gibbs Sampler technique.

Table 4. Summary table for posterior distribution of λ (EM)

Min	Q1	Median	Mean	Q3	Max	S.D.
0.3578	0.4845	0.5460	0.5580	0.6211	0.9617	0.0965

Table 5. Summary table for posterior distribution of coefficients β (EM)

Posterior Summaries						
Parameter	N	True value	Mean	Std.Dev	2.5%tile	97.5%tile
β_1	1000	4	3.5771	0.6616	2.4011	4.8946
β_2	1000	-4	-3.6826	0.6325	-4.9252	-2.5460
β_3	1000	5	4.9124	0.8097	3.5103	6.5340
β_4	1000	7	5.6344	1.5386	2.4827	8.7454
β_5	1000	-6	-5.6971	1.0071	-7.7677	-3.9931
β_6	1000	0	-0.0862	0.2560	-0.5988	0.4163
β_7	1000	0	-0.7867	1.1927	-3.6387	1.1440
β_8	1000	0	1.1897	1.2264	-0.7098	4.1629
β_9	1000	0	-0.0664	0.5069	-1.0596	1.0518
β_{10}	1000	0	-0.2191	0.4795	-1.2069	0.6961
β_{11}	1000	0	0.1175	0.1571	-0.1817	0.4352
β_{12}	1000	0	0.0807	0.1444	-0.2068	0.3494
β_{13}	1000	0	0.0368	0.1015	-0.1680	0.2418
β_{14}	1000	0	-0.0677	0.2441	-0.5897	0.3984
β_{15}	1000	0	-0.0352	0.2284	-0.5080	0.4242

The estimated D_τ (EM) is as follows:

$$\hat{D_\tau} = diag(10.33,10.47,12.54,14.61,14.60,3.76,5.33,6.09,4.12,4.17,3.95,3.70,3.67,3.82,3.79)$$

From the simulation results above, this computation method tends to have smaller variance compare to Gibbs Sampler regarding estimation.

This Bayesian Lasso Probit model successfully made the dimension reduction and set all those "zero coefficients" to 0 no matter which computation method to use. Furthermore, a model consistency check was proposed by repeatedly drawing samples and calculate the variance of the estimates.

4.1.3 Consistency and Variation

This simulation was done for 30 times to verify the stability of the results. The coefficients estimation results were stored in Table 6. The standard deviation for all coefficients was very small hence proved of the consistency and concluded the simulation has small variation. Boxplots for these 15 coefficients were also provided in Figure 5. These boxplots present a visual way to check the consistency.

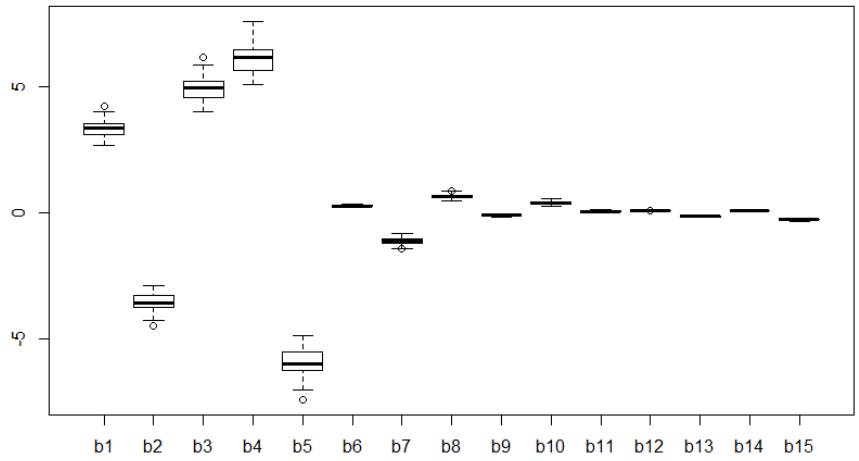

Figure 5. Boxplots for 30 simulations of β

Table 6. Simulation consistency result

	Iteration							Stability	
Para	1	2	3	…	28	29	30	Mean	Var
β_1	3.45(0.61)	3.27(0.72)	3.09(0.64)	…	2.67(0.68)	4.23(0.90)	3.81(0.76)	3.3280	0.1454
β_2	-3.68(0.61)	-3.50(0.73)	-3.31(0.64)	…	-2.87(0.67)	-4.50(0.91)	-4.06(0.77)	-3.5586	0.1587
β_3	5.09(0.69)	4.80(0.86)	4.58(0.74)	…	4.02(0.79)	6.16(1.08)	5.57(0.93)	4.9078	0.2799
β_4	6.22(1.78)	5.92(1.94)	5.67(1.57)	…	5.08(1.56)	7.59(2.13)	6.89(1.82)	6.1036	0.3955
β_5	-6.14(0.87)	-5.81(1.07)	-5.54(0.90)	…	-4.87(0.91)	-7.42(1.24)	-6.73(1.04)	-5.9367	0.3948
β_6	0.26(0.26)	0.26(0.26)	0.25(0.31)	…	0.21(0.29)	0.35(0.37)	0.30(0.31)	0.2640	0.0013
β_7	-1.22(1.27)	-1.13(1.29)	-1.07(1.21)	…	-0.85(1.15)	-1.45(1.51)	-1.30(1.27)	-1.1170	0.0226
β_8	0.73(1.29)	0.68(1.30)	0.64(1.25)	…	0.49(1.20)	0.84(1.59)	0.77(1.34)	0.6573	0.0087
β_9	-0.13(0.70)	-0.06(0.66)	-0.06(0.75)	…	-0.04(0.73)	-0.18(1.06)	-0.14(0.81)	-0.1042	0.0017
β_{10}	0.42(0.68)	0.34(0.64)	0.33(0.68)	…	0.26(0.69)	0.54(0.97)	0.46(0.76)	0.3831	0.0048
β_{11}	0.06(0.26)	0.05(0.25)	0.05(0.24)	…	0.04(0.24)	0.11(0.30)	0.08(0.25)	0.0585	0.0005
β_{12}	0.09(0.23)	0.08(0.24)	0.08(0.26)	…	0.08(0.25)	0.09(0.35)	0.09(0.26)	0.0874	0.0001
β_{13}	-0.16(0.11)	-0.14(0.10)	-0.14(0.12)	…	-0.12(0.12)	-0.20(0.15)	-0.17(0.12)	-0.1480	0.0005
β_{14}	0.08(0.23)	0.10(0.23)	0.08(0.24)	…	0.08(0.23)	0.11(0.28)	0.10(0.25)	0.0916	0.0002
β_{15}	-0.27(0.20)	-0.27(0.20)	0.25(0.20)	…	-0.22(0.20)	-0.34(0.25)	-0.30(0.22)	-0.2683	0.0007

4.2. Live Data

NCAA's "March Madness" refers to the Division I Men's Basketball tournament with single-elimination on each game. March Madness is an American tradition that sends millions of fans into a synchronized frenzy each year. It is this chaos that gives the tournament the nickname of "March Madness." Based on American Gaming Association (AGA)'s report (2015), about 40 million people filled out 70 million March Madness brackets (Moyer, 2015). The bracketing hence

draws high attention every year and is indeed an excellent real-world case for this entire statistical modeling procedure.

4.2.1. The Playing Rule and Structure

The March Madness is currently featuring 68 college basketball teams. After the games of first four, a 64 teams' data structure would be pulled. The 64 teams are divided into four regions, which are East Region, South Region, Midwest Region, and West Region. Each region has an equal number of 16 teams. For those 16 teams in each region, every team is provided a seeding position ranked from 1 to 16 by a professional committee. The tournament setting is always seed 1 versus seed 16, seed 2 versus seed 15, seed 3 versus seed 14 and continuous to seed 8 versus seed 9. It is noted that the seeds of the first round add up to 17. After a total of six rounds, namely, Round64 (Rd64), Round32 (R32), Sweet16, Elite8, Final4 and the championship, the national title will be awarded to the team that wins six games in a row. It is easy to discover that there are a total of 63 games played each year (NCAA Basketball Championship, 2018). In Rd64, 64 teams play 32 games. In Rd32, 32 teams play 16 games. 16 teams play 8 games in Sweet16, 8 teams play 4 games in Elite8, 4 teams play 2 games in Final4, and 2 teams fight for the Championship. The calculation will then be $\frac{64}{2}+\frac{32}{2}+\frac{16}{2}+\frac{8}{2}+\frac{4}{2}+\frac{2}{2}=32+16+8+4+2+1=63$. Figure 6 shows the NCAA Men's Division I Basketball Tournament bracket and complete tournament results in the 2017-2018 season.

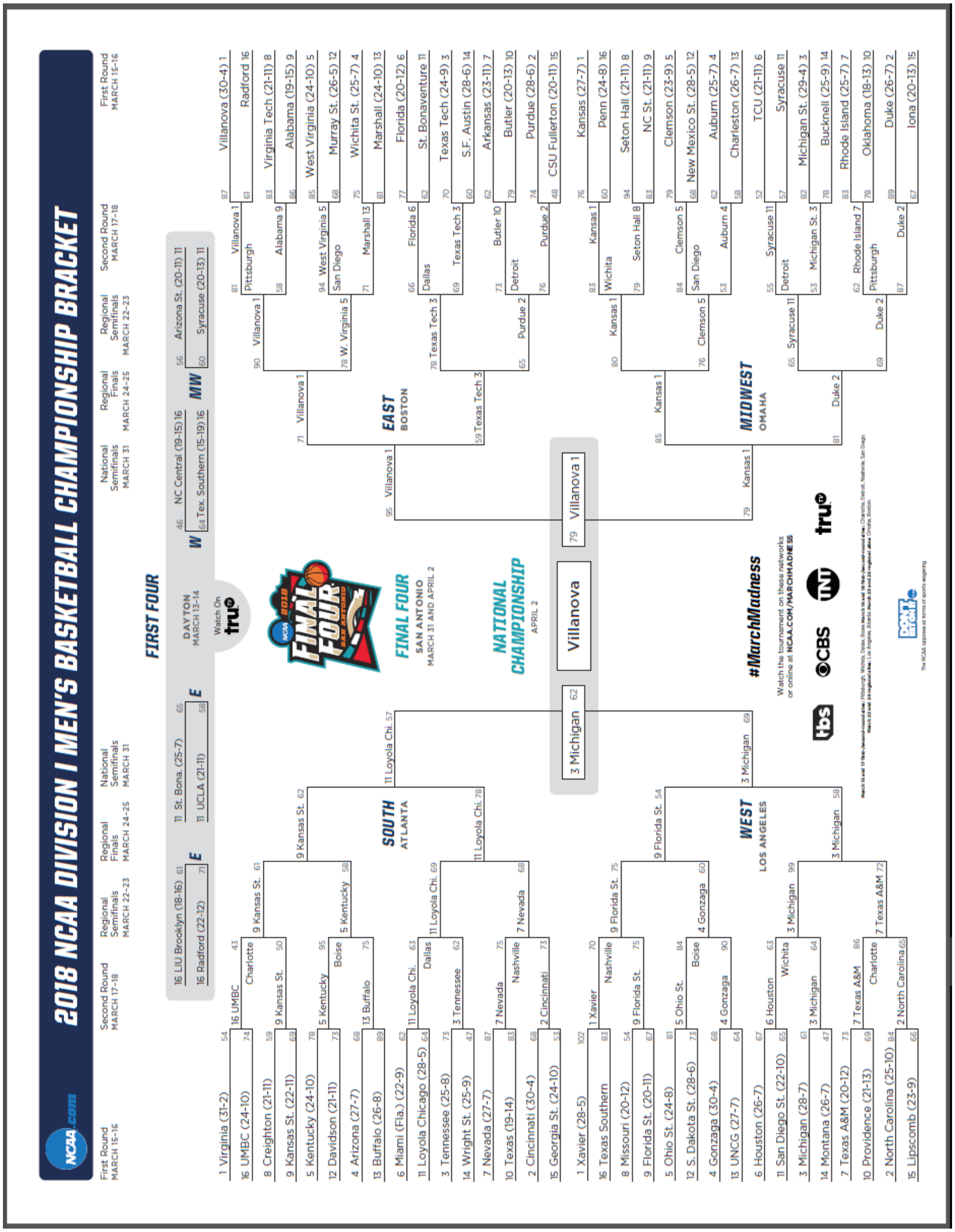

Figure 6. NCAA 2018 March Madness bracket with complete tournament results (This template is downloaded from http://i.turner.ncaa.com/sites/default/files/external/printable-bracket/2018/bracket-ncaa.pdf)

4.2.2. Qualifying Procedure

There are more than three hundred eligible Division I basketball teams. However, only 68 teams can make it into the March Madness. Teams that receive a bid to NCAA tournament are broken into two categories: Automatic bids, and at-large bids (2017 to 2018 Season NCAA Basketball Tournament, 2018). In Division I, there are 32 conferences. Hence, 32 teams qualify for automatic bids granted to the winner of the conference tournament championship. Ivy League used to be an exception (no conference tournament) but conducted its first postseason tournament in 2017. Table 7 shows the Automatic qualifiers in 2018 March Madness.

Table 7. Automatic qualifiers for the 2018 NCAA March Madness

Conference	Team	Conference	Team
American East	UMBC (24-10)	Mid-American	Buffalo (26-8)
AAC	Cincinnati (30-4)	MEAC	NC-Central (19-15)
Atlantic 10	Davidson (21-11)	Missouri Valley	Loyola-Chicago (28-5)
ACC	Virginia (31-2)	Mountain West	San Diego State (22-10)
Atlantic Sun	Lipscomb (23-9)	Northeast	LIU-Brooklyn (18-16)
Big 12	Kansas (27-7)	Ohio Valley	Murray State (26-5)
Big East	Villanova (30-4)	Pac-12	Arizona (27-7)
Big Sky	Montana (26-7)	Patriot	Bucknell (25-9)
Big South	Radford (22-12)	SEC	Kentucky (24-10)
Big Ten	Michigan (28-7)	Southern	UNCG (26-7)
Big West	CS-Fullerton (20-11)	Southland	Stephen F. Austin (28-6)
Colonial Athletic	Charleston (26-7)	SWAC	Texas Southern (15-19)
Conference USA	Marshall (24-10)	Summit League	SDSU (28-6)
Horizon League	Wright State (25-9)	Sun Belt	Georgia State (24-10)
Ivy League	Penn (24-8)	West Coast	Gonzaga (30-4)
MAAC	Iona (20-13)	WAC	New Mexico St. (28-5)

The remaining 36 bids are at-large bids granted by the NCAA Selection Committee. The Committee will select what they feel are the best 36 teams that did not receive automatic bids. Even though each conference receives only one automatic bid, the selection committee may select any number of at-large teams from each conference (NCAA basketball selection process, 2018). The at-large teams come from the following conferences: Atlantic Coast Conference (ACC), Southeastern Conference (SEC), Big 12 Conference, Big East Conference, Big Ten Conference, American Athletic Conference (AAC), Pacific 12 conference (Pac-12), Atlantic 10 Conference (A-10) and Mountain West Conference. Table 8 is at-large qualifiers in 2018 March Madness (2018 NCAA Basketball Tournament, 2018).

Before the final 64 teams bracket filled out, eight teams need to play first four. The winners of these games advanced to the Rd64 and seeded as 11 or 16. The First Four games played in 2018 March Madness are shown in Table 9 (2018 NCAA Basketball Tournament, 2018).

4.2.3. Bracket Scoring System

Two types of scoring systems will be considered to evaluate the performance of using Bayesian Lasso model. One is doubling scoring system, and the other is the simple scoring system (Shen et al. 2015). Based on the brackets structure, there are six rounds in the tournament. Under the simple scoring system, each correct pick will be granted one point. For doubling scoring system, the right pick for the first round will be awarded one point, two points for the second round and continue doubling to the final round with 32 points. Table 10 is the summary of the scoring system.

Table 8. At-large qualifiers for the 2018 NCAA March Madness

Conference	Team	Conference	Team
ACC	N. Carolina	Big 12	Kansas St.
ACC	Duke	Big 12	Texas
ACC	Clemson	Big 12	Oklahoma
ACC	Miami (Fla.)	Big East	Xavier
ACC	Va. Tech	Big East	Creighton
ACC	NC State	Big East	Seton Hall
ACC	Florida St.	Big East	Butler
ACC	Syracuse	Big East	Providence
SEC	Tennessee	Big Ten	Purdue
SEC	Auburn	Big Ten	Michigan St.
SEC	Florida	Big Ten	Ohio St.
SEC	Arkansas	AAC	Wichita St.
SEC	Texas A&M	AAC	Houston
SEC	Missouri	Pac-12	UCLA
SEC	Alabama	Pac-12	Arizona St.
Big 12	Texas Tech	A-10	Rhode Island
Big 12	W. Virginia	A-10	St. Bona.
Big 12	TCU	Mountain West	Nevada

Table 9. First Four games in 2018 NCAA March Madness

At-large	Automatic	At-large	Automatic
East Region (11)	*East Region (16)*	*Midwest Region (11)*	*West Region (16)*
St. Bona.	LIU Brooklyn	Arizona St.	NC Central
UCLA	Radford	Syracuse	Texas So.

Table 10. Scoring System

	Rd64	Rd32	Sweet16	Elite8	Final4	Championship	Total
Number	32	16	8	4	2	1	63
Simple	1	1	1	1	1	1	63
Doubling	1	2	4	8	16	32	192

4.2.4. Bracketing

The simulation construction was based on the Live Data (NCAA March Madness). We have a total of 16 years of data, and the prediction was made with the previous 16 years of data information. Hua (2015) suggested using the following 16 covariates in Table 11. For real application of our Bayesian Lasso Model, we used the same covariates.

Table 11. Covariates used in the model

FGM	Field Goals Made Per Game in Regular Season
3PM	3-Point Field Goals Made Per Game in Regular Season
FTA	Free Throws Made Per Game in Regular Season
ORPG	Offensive Rebounds Per Game in Regular Season
DRPG	Defensive Rebounds Per Game in Regular Season
APG	Assists Per Game in Regular Season
PFPG	Personal Fouls Per Game in Regular Season
SEED	Seed Number
ASM	Average Scoring Margin
SAGSOS	Sagarin Proxy for Strength of schedule (Sagarin ratings)
ATRATIO	Assist to Turnover Ratio in Regular Season
Pyth	Pythagorean Winning Percentage (Pomeroy ratings)
AdjO	Adjusted Offensive Efficiency (Pomeroy ratings)
AdjD	Adjusted Defensive Efficiency (Pomeroy ratings)

We used the first 16 years of data to fit the model and then used our advanced sampling technique to draw samples efficiently. This process includes Gibbs Sampler and EM algorithm. Time series plot and ACF plot for selected coefficient parameters were provided. Figure 7 refers to parameters of coefficient β for FGM and Pyth. Figure 8 provides the parameter τ_1& τ_{15}.

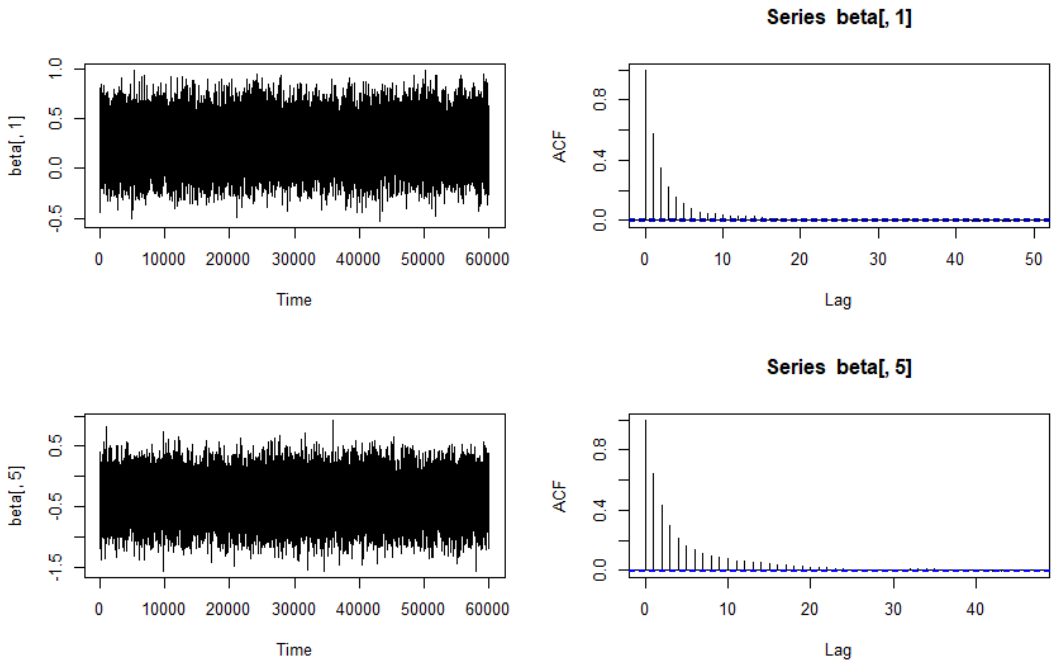

Figure 7. Diagnostic plots for FGM and Pyth

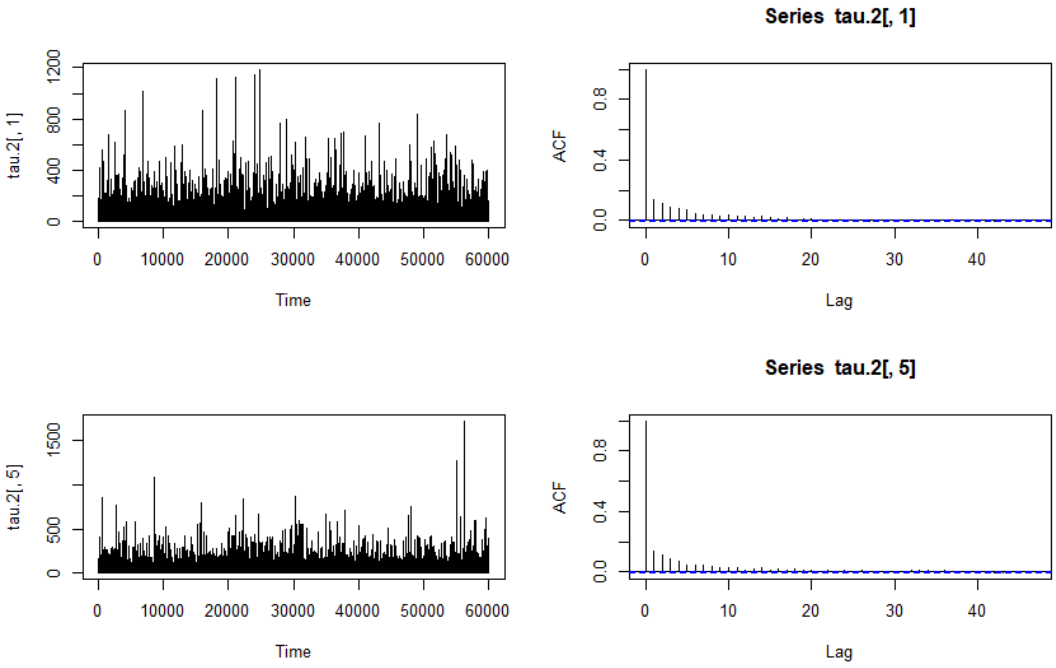

Figure 8. Diagnostic plots for τ_1 & τ_{15}

Based on the information from the time series plot and ACF plot. We decided to burn in 10,000 and slice every 50. That leaves us $\frac{60{,}000-10{,}000}{50} = 1000$ samples. Our estimates were based on these 1000 effective samples. The summary estimation of λ was provided in Table 12.

Table 12. Posterior summaries for λ (March Madness 2018 with Gibbs)

Min	Q1	Median	Mean	Q3	Max	S.D.
0.0904	0.1832	0.2230	0.2336	0.2734	0.9537	0.0750

The samples of λ range from 0.0904 to 0.9537 with a very small standard deviation 0.0750. We concluded the λ sampling was stable. Figure 9 presented an overall view of the samples of tuning parament λ.

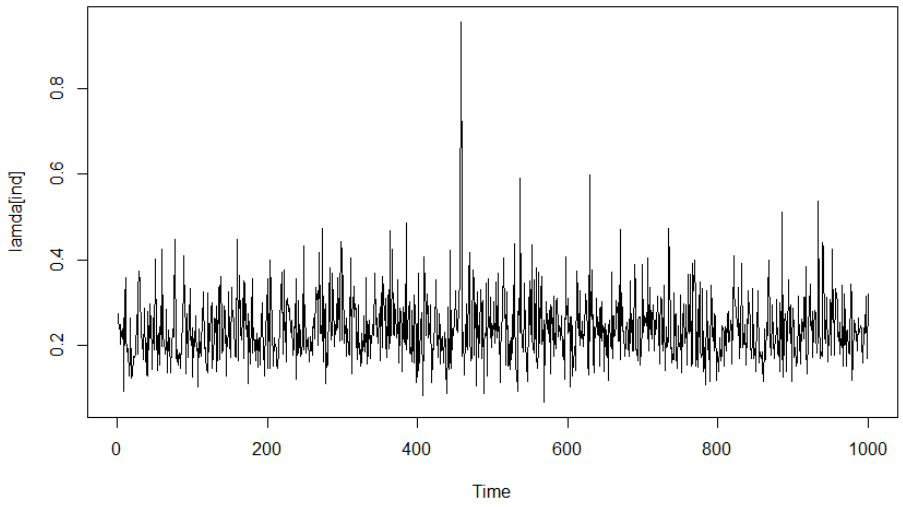

Figure 9. λ after burn-in and slicing (March Madness 2018 with Gibbs)

The posterior summary (coefficients β estimation) were given in Table 13:

Table 13. Posterior summaries for β (March Madness 2018 with Gibbs)

Parameter	N	Mean	Std.Dev	2.5%tile	97.5%tile
SEED	1000	0.2562	0.1762	-0.0827	0.6136
FGM	1000	3.5207	1.5580	0.7602	6.6113
AdjO	1000	26.7509	5.4913	15.9663	36.9251
AdjD	1000	-24.6345	5.2714	-34.6332	-13.5878
ASM	1000	-0.3753	0.2830	-0.9522	0.1794
SAGSOS	1000	-6.2021	2.9522	-12.2208	-0.3990
Pyth	1000	1.8987	1.9892	-1.4140	6.0931
3PM	1000	0.0152	0.4152	-0.7861	0.7772
FTA	1000	-0.2929	0.6641	-1.5763	0.9413
ORPG	1000	0.3613	0.5338	-0.6239	1.3955
DRPG	1000	-1.2840	1.1468	-3.5003	0.8541
APG	1000	-1.5345	0.8822	-3.2936	0.1424
PFPG	1000	-0.2752	0.7775	-1.7579	1.2768
ATRATIO	1000	-0.1587	0.6563	-1.4813	1.1106

From the result in Table 12, quite a few coefficients were set to 0. That is also to indicate some useless information from the covariates. For example, for covariate 3PM, the estimated value was 0.0152 and the 95% credible interval was from -0.7861 to 0.7772, which covers 0. This 3PM was reduced. Because of Bayesian inference, the estimate was all from samples. Hence, we do not have coefficient set precisely to 0 as LASSO. But for those coefficients needed to be deduced, the estimate should be close to 0. The same sample was also drawn using EM algorithm. There was no noticeable difference since this was only the difference between computation technique. Summaries were stored in Table 14, Figure 10 and Table 15.

Table 14. Posterior summaries for λ (March Madness 2018 with EM)

Min	Q1	Median	Mean	Q3	Max	S.D.
0.1483	0.1802	0.1918	0.1939	0.2063	0.2852	0.0201

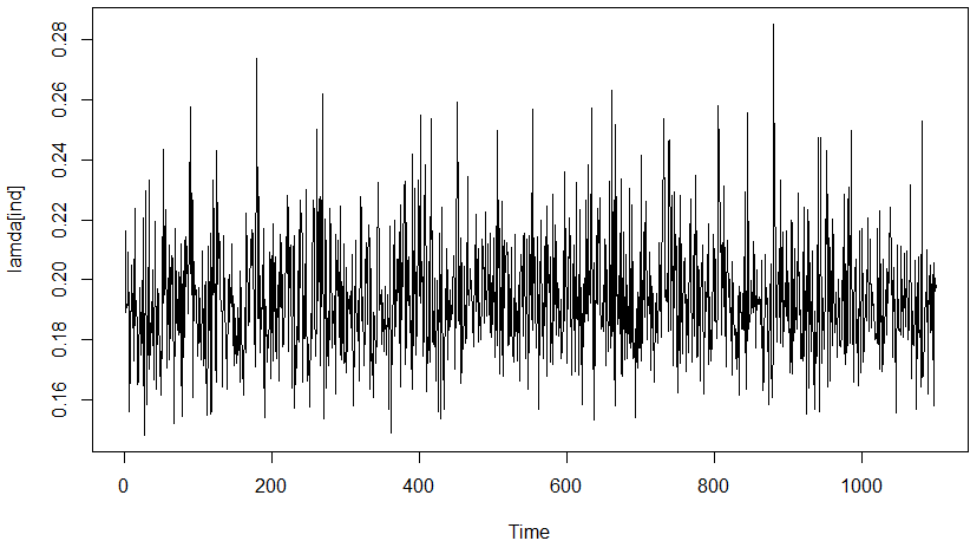

Figure 10. λ after burn-in and slicing (March Madness 2018 with EM)

Table 15. Posterior summaries for β (March Madness 2018 with EM)

Parameter	N	Mean	Std.Dev	2.5%tile	97.5%tile
SEED	1000	0.2735	0.1855	-0.0551	0.6526
FGM	1000	3.6697	1.5776	0.6847	6.6294
AdjO	1000	28.1478	4.9242	18.4729	37.4487
AdjD	1000	-26.1025	4.6936	-35.3224	-16.7845
ASM	1000	-0.4312	0.2981	-0.9910	0.1711
SAGSOS	1000	-6.8253	2.9309	-12.6829	-1.1044
Pyth	1000	1.7056	1.8917	-1.6937	5.8784
3PM	1000	0.0178	0.4037	-0.8128	0.8132
FTA	1000	-0.2973	0.6957	-1.7113	1.0527
ORPG	1000	0.3488	0.5428	-0.7113	1.3987
DRPG	1000	-1.3940	1.1817	-3.7738	0.8126
APG	1000	-1.5776	0.9385	-3.4280	0.2591
PFPG	1000	-0.3021	0.7828	-1.8316	1.2140
ATRATIO	1000	-0.1939	0.6635	-1.4914	1.1072

Based on the design matrix (data information) of the latest year (NCAA Men's Basketball 2017 to 2018 Season)'s data information, the bracketing was provided. Table 16 is the Probability Matrix for South Region.

Table 16. Probability matrix for 2018 NCAA March Madness South Region Bracketing

Round / Seed	R64	R32	S16	E8	F4	Champ
1	0.9807	0.8484	0.6924	4.9523e-01	3.5966e-01	2.4198e-01
16	0.0192	0.0024	0.0002	1.4706e-05	9.0253e-07	4.3333e-08
8	0.5355	0.0847	0.0372	1.2211e-02	4.0043e-03	1.1075e-03
9	0.4644	0.0643	0.0260	7.7244e-03	2.2897e-03	5.6644e-04
5	0.6500	0.3861	0.1094	4.7669e-02	2.0981e-02	8.0021e-03
12	0.3499	0.1548	0.0280	8.4311e-03	2.5292e-03	6.3360e-04
4	0.6717	0.3454	0.0893	3.5876e-02	1.4497e-02	5.0535e-03
13	0.3282	0.1135	0.0172	4.3821e-03	1.1061e-03	2.3091e-04
6	0.5226	0.2418	0.0759	1.9020e-02	6.5729e-03	1.9272e-03
11	0.4773	0.2098	0.0620	1.4421e-02	4.6895e-03	1.2824e-03
3	0.8800	0.5277	0.2059	6.5323e-02	2.7963e-02	1.0385e-02
14	0.1199	0.0206	0.0018	1.2855e-04	1.2614e-05	9.8976e-07
7	0.5138	0.1538	0.0756	2.0390e-02	7.5499e-03	2.3825e-03
10	0.4861	0.1371	0.0657	1.6812e-02	5.9735e-03	1.7871e-03
2	0.8900	0.6776	0.5040	2.5119e-01	1.5854e-01	9.0855e-02
15	0.1099	0.0314	0.0087	1.1638e-03	2.2016e-04	3.3921e-

Based on the probability matrix, the brackets were filled. Between two teams, we took the higher probability as the winner. That was the prediction based on Bayesian Lasso Probit Model. Figure 11 provided a detail prediction bracket. Based on the previous study, we have two different scoring systems (Shen et al. 2015). Table 17 was the scores using both scoring systems.

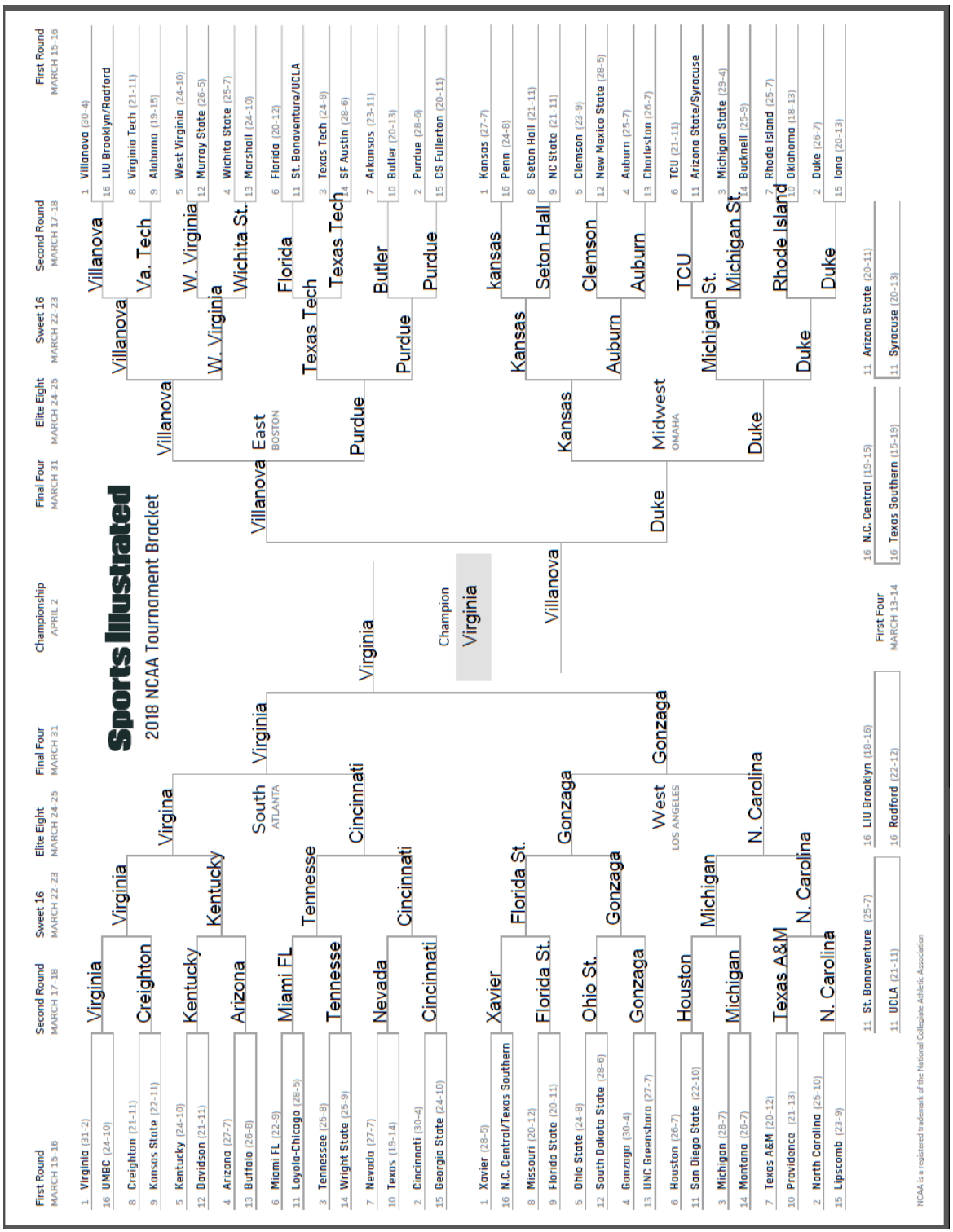

Figure 11. 2018 NCAA March Madness Brackets (Bayesian Lasso) (This template is downloaded from https://cdn-s3.si.com/s3fs-public/download/2018-ncaa-tournament-printable-bracket-march-madness.pdf)

Table 17. Scoring 2018 March Madness Bracket (Bayesian Lasso)

	Rd64	Rd32	Sweet16	Elite8	Final4	Championship	Total
Number	32	16	8	4	2	1	63
Simple	25	9	3	1	1	0	39/63
Doubling	25	18	12	8	16	0	79/192

We had about 62% accuracy when evaluated using single scoring system, about 41% accuracy when evaluated using the double scoring system. This result was reasonable but not super outstanding. That is due to one big upset that one seed 16 team won number 1 seed in the first round. The South region was the region that has the lowest correct prediction. The region top four turned out to be Kansas State, Kentucky, Loyola Chicago, and Nevada. The seeding was 9, 5, 11 and 7. No top four seeds entered the second round (Rd32). If we omit the South Region and only evaluate the other three regions, then the prediction accuracy was about 73% for the single scoring system and about 47% for the double scoring system.

4.3. Comparison

Hua (2015) and Shen et al. (2016) both used Bayesian inference with informative prior. Both studies need to include all covariates. Bayesian Lasso hence has an advantage over the previous method because the model reduced about half of the covariates. Bayesian Lasso promoted the calculation efficiency and further combined E-M algorithm, which makes the model even more efficient. The likelihood function in Shen et al. (2016) is the same as Hua (2015). By delta method, the prior in p can be transferred into β:

$$\beta \sim MVN((\mathrm{X}'\mathrm{X})^{-1}X'\log\left(\frac{\hat{p}}{1-\hat{p}}\right), (\mathrm{X}'\mathrm{X})^{-1}X'n\hat{p}(1-\hat{p})X(\mathrm{X}'\mathrm{X})') \quad (4.1)$$

Hence,

$$f(\beta|\text{data}) \propto \text{MVN} \times \prod_{i=1}^{n} \left(\frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right)^{y_i} \left(\frac{1}{1+\exp(\eta_i)}\right)^{1-y_i} \quad (4.2)$$

Based on the information (probability matrix) provided from this model including informative prior. The estimation of coefficients β can be obtained in table 18. The bracketing scoring was also presented in table 19 using the model introduced in Shen et al. (2016). The bracketing was also offered in Figure 12.

Table 18. Posterior summaries for β (March Madness 2018 with Full Bayesian)

Parameter	N	Mean	Std.Dev	2.5%tile	97.5%tile
SEED	N/A	N/A	N/A	N/A	N/A
FGM	10,000	1.8442	0.4176	1.0272	2.6722
AdjO	10,000	18.0413	1.1314	15.8001	20.2207
AdjD	10,000	-17.8434	1.0489	-19.8955	-15.7825
ASM	10,000	0.2187	0.0756	0.0725	0.3685
SAGSOS	10,000	5.8576	0.6657	4.5564	7.1584
Pyth	10,000	-1.5674	0.3691	-2.2840	-0.8475
3PM	10,000	0.0558	0.1125	-0.1656	0.2738
FTA	10,000	0.7584	0.2025	0.3660	1.1534
ORPG	10,000	-0.1591	0.1570	-0.4709	0.1498
DRPG	10,000	0.6835	0.3467	-0.0010	1.3557
APG	10,000	-1.0315	0.2674	-1.5494	-0.5149
PFPG	10,000	-0.7960	0.2427	-1.2638	-0.3249
ATRATIO	10,000	-0.4340	0.1884	0.0639	0.8059

If we compare this estimation with the Bayesian Lasso, these full Bayesian estimations do not have many credible intervals cover 0, which is expected and do not have the power of

dimension reduction. The Bayesian Lasso model has beneficial sparsity property and hence can reduce the dimension. Bayesian Lasso reduced the dimension to four valid covariates.

Table 19. Scoring 2018 March Madness Bracket (Full Bayesian)

	Rd64	Rd32	Sweet16	Elite8	Final4	Championship	Total
Number	32	16	8	4	2	1	63
Simple	24	8	3	1	1	0	38/63
Doubling	24	16	12	8	16	0	78/192

Using fewer covariates, the Bayesian Lasso performed better than the full Bayesian and the enhanced the computation method also guaranteed the efficiency.

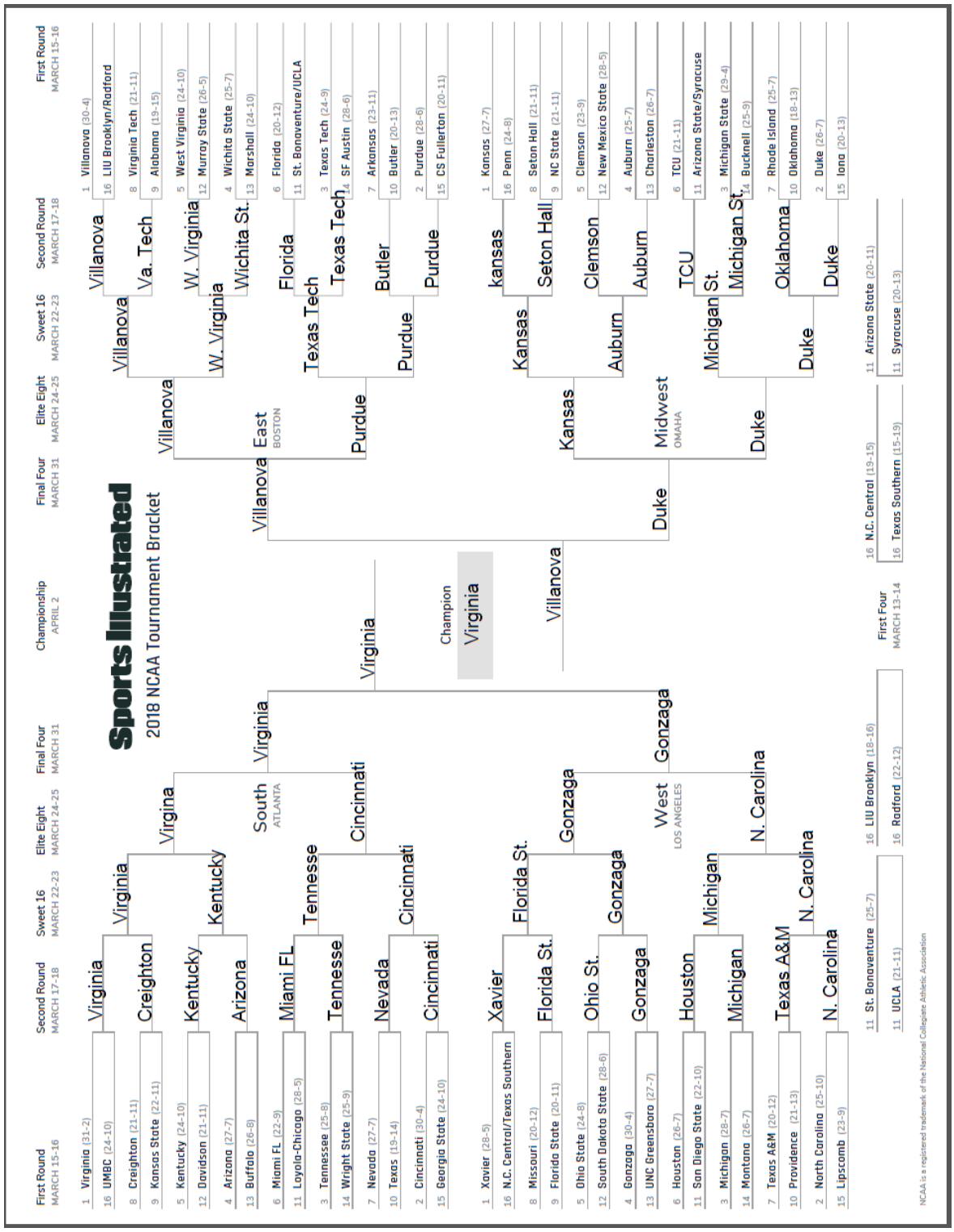

Figure 12. 2018 NCAA March Madness Brackets (Full Bayesian) (This template is downloaded from https://cdn-s3.si.com/s3fs-public/download/2018-ncaa-tournament-printable-bracket-march-madness.pdf)

5. DISCUSSION

One significant contribution of this study was the computation efficiency by introducing EM algorithm and further using theoretical value for the expectation step. This EM algorithm can be extended to other conditional marginals. For example, Introduce E-M algorithm to the probit classifier. In this work, we sampled z from truncated normal distribution. We can also involve E-M algorithm to do a soft clustering instead of sampling directly. We expect more efficient by introducing E-M to other conditional marginals.

For sampling the β parameter, we derived the β parameter follows a multivariate normal distribution which the calculation of matrix was required. Because of the large dimension, calculation of matrix was time-consuming and lack of efficiency. However, the transformation of the matrix dimension can be performed to make the computation faster. When n<<p and by Woodbury-Sherman-Morrison matrix, we can finish the calculation by transforming matrix to the smaller dimension $\boldsymbol{n}$ instead of the original dimensions of the covariates p.

In this research work, we studied Lasso with a probit Bayesian model. Lasso may also be extended to logistic linear regression. The logistic linear regression model uses logit link, and the model can be expressed as:

$$g(p_i) = logit(p_i) = \log\frac{pi}{1-p_i} = \eta_i = x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{id}\beta_d, i = 1, \dots, n$$

That leads to $p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}$. The likelihood of the probit linear regression is simply:

$$\prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

$$\prod_{i=1}^{n} \left[\frac{e^{\eta_i}}{1+e^{\eta_i}}\right]^{y_i} \left[\frac{1}{1+e^{\eta_i}}\right]^{1-y_i} \tag{5.1}$$

One may apply LASSO for estimation of β, which minimize the negative of the log-likelihood:

$$-\sum_{i=1}^{n} y_i \, log\left(\frac{e^{\eta_i}}{1+e^{\eta_i}}\right) - \sum_{i=1}^{n}(1-y_i)\log\left(\frac{1}{1+e^{\eta_i}}\right) + \lambda\|\beta\|_1 \quad (5.2)$$

We can then apply the Gibbs Sampler or Hybrid Bayesian with EM to sample. Due to logistic format, we will not have normal marginal distribution. The marginal will not have a specific distribution. However, by introducing SIR algorithm, the conditional marginals can still be sampled and further finish the computation of Gibbs Sampler process.

Bae & Mallick (2004) provided another application of the Bayesian Lasso Probit model, which was the Gene selection problem. The data was about gene expression level hence with large dimension. Bae & Mallick (2004) successfully reduced the dimension, and that was a real example of Bayesian Lasso solving big data problem. For big data, where $n<<p$, although cross-validation can fulfill the dimension reduction, it cannot successfully make a decision when the optimal dimension is happening to be in between of *n* and *p*.

REFERENCES

[1] D. F. Andrews and C. L. Mallows (1974). Scale Mixtures of Normal Distribution. *Journal of the Royal Statistical Society*. Series B (Methodological). Vol. 36, No. 1 (1974), pp. 99-102

[2] Dempster, A. P., Laird, N., Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. Series B (Methodological). Vol. 39, No. 1 (1974), pp. 1-38

[3] Bae, K., Mallick, Bani K., (2004). Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*. Vol. 20 no. 18 2004, pages 3423-3430

[4] C. I. Bliss (1934). The Method of Probits. *Science.* New Series, Vol. 79, No. 2037 (Jan. 12, 1934), pp. 38-39

[5] Casella, G., (2001). Empirical Bayes Gibbs Sampling. *Biostatistics* (2001), 2, 4, pp. 485-500

[6] Hoerl, A.E., Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*. Vol. 12, No. 1 (Feb., 1970), pp. 55-67

[7] Hua, S. (2015). *Comparing Several Modeling Methods on NCAA March Madness.* Unpublished Thesis Paper, North Dakota State University, Fargo, ND.

[8] Magel, R., Unruh S. (2013). Determining Factors Influencing the Outcome of College Basketball Fames. *Open Journal of Statistics*, Vol. 3 No. 4, 2013, p.225-230

[9] Moyer, C. (2015, March 12). *Americans To Bet $2 Billion on 70 Million March Madness Brackets This Year, Says New Research*. *American Gaming Association*. Retrieved from http://www.americangaming.org/newsroom/press-releases/americans-to-bet-2-billion-on-70-million-march-madness-brackets-this-year

[10] Oehlert, G. W. (1992), A Note on the Delta Method, *The American Statistician*, Vol. 46, No. 1, p. 27-29

[11] Park, T., Casella, G., (2008). The Bayesian Lasso. *Journal of the American Statistical Association.* Vol. 103, 2008, pp. 681-686

[12] Schwertman, C. N., Schenk, L. K., Holbrook, C. B. (1996). More Probability Models for the NCAA Regional Basketball Tournaments. *The American Statistician*, Vol. 50, No. 1., p. 34-38

[13] Shen, G., Gao, D., Wen, Q., Magel, R., (2016). Predicting Results of March Madness Using Three Different Methods. *Journal of Sports Research*. 2016, Vol. 3, issue 1, pp. 10-17

[14] Shen, G., Hua, S., Zhang, X., Mu, Y., Magel, R. (2015). Prediction Results of March Madness Using the Probability Self-Consistent Method. *International Journal of Sports Science*, 5(4), p.139-144

[15] Geman, S., Geman D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, November 1984

[16] Tibshirani, R., (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*. Series B (Methodological), Vol. 58, No. 1 (1996), pp. 267-288

[17] Zhang, X. (2012). *Bracketing NCAA Men's Division I Basketball Tournament.* Unpublished Thesis Paper, North Dakota State University, Fargo, ND.

APPENDIX A. PROOF

1. Prove if $f(\beta_j \,|{\tau_j}^2) = \frac{1}{\sqrt{2\pi{\tau_j}^2}} \exp\left\{-\frac{1}{2}\frac{{\beta_j}^2}{{\tau_j}^2}\right\}$ and $f({\tau_j}^2 \,|{\lambda_j}^2) = \frac{\lambda^2}{2} \exp\left\{-\frac{\lambda^2}{2}{\tau_j}^2\right\}$,

then $f(\beta_j|\lambda) = \frac{\lambda}{2}\exp\{-\lambda|\beta_j|\}$.

Proof:

$$f(\beta_j|\lambda) = \int_0^\infty \frac{1}{\sqrt{2\pi{\tau_j}^2}} \exp\left\{-\frac{1}{2}\frac{{\beta_j}^2}{{\tau_j}^2}\right\} \frac{\lambda^2}{2} \exp\left\{-\frac{\lambda^2}{2}{\tau_j}^2\right\} \mathrm{d}{\tau_j}^2$$

$$= \frac{\lambda^2}{2\sqrt{\pi}} \int_0^\infty \frac{1}{\sqrt{2{\tau_j}^2}} \exp\left\{-(\frac{\lambda^2}{2}{\tau_j}^2 + \frac{{\beta_j}^2}{2{\tau_j}^2})\right\} \mathrm{d}{\tau_j}^2$$

$Let\ s^2 = \frac{\lambda}{2}{\tau_j}^2\ and\ a = \frac{1}{2}\lambda|\beta_j|, then\ f(\beta|\lambda) = \frac{\lambda}{\sqrt{\pi}} e^{-2a} \int_0^\infty \exp\left\{-\left(s - \frac{a}{s}\right)^2\right\} ds.$

$$Note\ that\ \int_0^\infty \exp\left\{-\left(s - \frac{a}{s}\right)^2\right\} ds = \int_0^1 \exp\left\{-\left(s - \frac{a}{s}\right)^2\right\} ds + \int_1^\infty \exp\left\{-\left(s - \frac{a}{s}\right)^2\right\} ds$$

$$and\ let\ s = \frac{a}{u}, then\ ds = \frac{-a}{u^2} du, then$$

$$\int_0^\infty \exp\left\{-\left(s - \frac{a}{s}\right)^2\right\} ds = \int_\infty^1 \exp\left\{-\left(\frac{a}{u} - u\right)^2\right\} (\frac{-a}{u^2} du) + \int_1^\infty \exp\left\{-\left(\frac{a}{u} - u\right)^2\right\} du$$

$$= \int_1^\infty \exp\left\{-\left(\frac{a}{u} - u\right)^2\right\} (\frac{a}{u^2} du) + \int_1^\infty \exp\left\{-\left(\frac{a}{u} - u\right)^2\right\} du$$

$$= \int_1^\infty \exp\left\{-\left(u - \frac{a}{u}\right)^2\right\} d\left(u - \frac{a}{u}\right) = \int_0^\infty e^{-t^2}\, dt = \frac{\sqrt{\pi}}{2}$$

It follows $f(\beta_j|\lambda) = \frac{\lambda}{\sqrt{\pi}} e^{-2a} \frac{\sqrt{\pi}}{2} = \frac{\lambda}{2}\exp\{-\lambda|\beta|\}$.

2. $\beta|\tau,z,\lambda,y \propto \prod_{i=1}^{N}\varphi(z_i;x_i'\beta,1)\times\varphi(\beta;0,D_\tau)$

Proof:

Only the above two terms contain parameter β. We discovered the formula above is the product of normal prior with normal model.

Note $f(z|\beta)=(2\pi)^{-1/2}\exp\{-(z_i-x_i'\beta)^2/2\}\ and\ f(\beta)\propto\exp\{-\beta' D_\tau^{-1}\beta/2\}$, then

$$\prod_{i=1}^{N} f(z_i;x_i'\beta)\times f(\beta;D_\tau)\propto\ \exp\left\{-\frac{(\beta-A^{-1}X'Z)'A(\beta-A^{-1}X'Z)}{2}\right\}, where\ A=D_\tau^{-1}+X'X$$

Therefore, $\beta|\tau,z,\lambda,y\sim N(A^{-1}X'Z,A^{-1})$

3. $\tau^2|z,\beta,\lambda,y\propto\varphi(\beta;0,D_\tau)\times\prod_{j=1}^{d}\frac{\lambda^2}{2}e^{\frac{-\lambda^2{\tau_j}^2}{2}}$

Proof:

The two terms contain parameter τ^2 would be $\varphi(\beta;0,D_\tau)\times\prod_{j=1}^{d}\frac{\lambda^2}{2}e^{\frac{-\lambda^2{\tau_j}^2}{2}}$. For This is ${\tau_j}^2$, we cannot write down any specific distribution. However, for $\frac{1}{{\tau_j}^2}$, we have an explicit format because:

$$f(\tau^2|z,\beta,\lambda,y)\propto\prod_{j=1}^{d}\frac{1}{\sqrt{{\tau_j}^2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{d}\frac{\beta_j^2}{{\tau_j}^2}\right)\exp\left(-\frac{\lambda^2}{2}{\tau_j}^2\right)$$

Let $s_j=\tau_j^{-2}, then\ d{\tau_j}^2=s_j^{-2}ds_j. So\ f(s|z,\beta,\lambda,y)\propto\prod_{j=1}^{d}{s_j}^{-\frac{3}{2}}\exp\left\{-\frac{\lambda^2(s_j-\mu_j)^2}{2\mu_j{}^2 s_j}\right\}$.

So $s_j|z,\beta,\lambda,y\ \sim\ IG(\mu_j,\lambda^2)\ with\ \mu_j=\lambda|\beta_j|^{-1}$.

4. $\lambda|\tau,\beta,z,y \propto \prod_{j=1}^{d} \frac{\lambda^2}{2} e^{\frac{-\lambda^2 {\tau_j}^2}{2}} \times (\ \pi(\lambda)=\text{c}\ )$

Proof:

We have λ treated as flat prior. From the whole posterior, the condition density for $\boldsymbol{\lambda}$ will then be the product of exponential kernel. This is a gamma distribution. After transforming this exponential kernel into gamma format, we discovered that this density follows:

$$f(\lambda|\tau,\beta,z,y) \propto \prod_{j=1}^{d} \frac{\lambda^2}{2} e^{\frac{-\lambda^2 {\tau_j}^2}{2}} \propto (\lambda^2)^d \exp\left\{-\frac{\lambda^2}{2}\sum_{j=1}^{d} {\tau_j}^2\right\}$$

$$\lambda|\tau,\beta,z,y \sim Gamma(d+1, \frac{1}{2}\sum_{j=1}^{d} {\tau_j}^2)$$

APPENDIX B. R CODE FOR SIMULATION

```
##############################
########  Gibbs Sampler  ########
##############################
library(mvtnorm)
library(MASS)
library(truncnorm) ## rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
library(statmod) ## rinvgauss(10,1,1)
library(VGAM) ## rinv.gaussian(10,1,1)
library(LearnBayes)
## Prepare Simulation Data ##
set.seed(100)
n=1024
d=15
x1<-rnorm(n,3,1)
x2<-rnorm(n,x1,1)
x3<-rnorm(n,x2,2)
x4<-runif(n,5,10)
x5<-runif(n,x4,x4+3)
x6<-rnorm(n,3.5,1)
x7<-rnorm(n,x6,1)
x8<-x4+x7+1
x9<-runif(n,x8,x8+3)
x10<-runif(n,x9,x9+1)
x11<-rnorm(n,5,1)
```

```
x12<-rnorm(n,x11,1)
x13<-rnorm(n,x12,2)
x14<-runif(n,5,10)
x15<-runif(n,x14,x14+3)
X<-cbind(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15)
X[1:5,]
summary(x8)
summary(x15)
apply(X,2,mean)
beta<-c(4,-4,5,7,-6,rep(0,10));beta
length(beta)
Xb<-X%*%beta
Xb
# Obtain the vector with probabilities of success p using the probit link
p<-pnorm(Xb)
p
y.data<-rbinom(1024,1,p)
y.data
length(y.data)
## find some initial betas
fit<-glm(y.data~0+X,family=binomial(link=logit))
summary(fit)
## Z
z<-rep(NA,n)
N.sim<-270000
```

```
N1<-sum(y.data);N1
N0<-n-N1;N0
## beta
beta<-matrix(NA,nrow=N.sim+1,ncol=d)
beta[1,]<-c(4.3,-2.9,5.4,12.3,-10.5,0.3,-0.2,0,1.0,-0.5,0.4,-0.4,-0.09,-0.6,0.4)
beta
## tau
tau.2<-matrix(NA,nrow=N.sim+1,ncol=d)
tau.2[1,]<-c(rep(1,d))
tau.2
## lamda
lamda<-rep(NA,n)
lamda[1]<-2
lamda
## sampling
for (j in 1:N.sim){
  mu.z<-X%*%beta[j,]
  z[y.data==0]<-rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
  z[y.data==1]<-rtruncnorm(N1,mean=mu.z[y.data==1],sd=1,a=0,b=Inf)
  tau<-diag(tau.2[j,])
  E<-solve(solve(tau)+t(X)%*%X)
  beta[j+1,]<-rmvnorm(1,E%*%t(X)%*%z,E)
  tau.2[j+1,]<-1/rinv.gaussian(d,(lamda[j]^2/beta[j+1,]^2)^0.5,lamda[j]^2)
  lamda[j+1]<-(rgamma(1,d+1,0.5*sum(tau.2[j+1,])))^0.5
}
```

```
## estimation
summary(lamda)
plot(lamda)## very stable
z
beta[,1]
tau.2[,1]
ind<-seq(from=20000,to=270000,by=125)
summary(lamda[ind])##estimate lambda
sd(lamda[ind])
ts.plot(lamda[ind]);acf(lamda[ind])
colMeans(beta[ind,])##estimate beta
colMeans(tau.2[ind,])##estimate tau
newb<-beta[ind,]
sd.beta<-rep(NA,15)
for (k in 1:15){
  sd.beta[k]<-sd(newb[,k])
  }
sd.beta##sd of beta estimates
quantile(beta[ind,1],c(0.025,0.5,0.975))
quantile(beta[ind,2],c(0.025,0.5,0.975))
quantile(beta[ind,3],c(0.025,0.5,0.975))
quantile(beta[ind,4],c(0.025,0.5,0.975))
quantile(beta[ind,5],c(0.025,0.5,0.975))
quantile(beta[ind,6],c(0.025,0.5,0.975))
quantile(beta[ind,7],c(0.025,0.5,0.975))
```

```
quantile(beta[ind,8],c(0.025,0.5,0.975))
quantile(beta[ind,9],c(0.025,0.5,0.975))
quantile(beta[ind,10],c(0.025,0.5,0.975))
quantile(beta[ind,11],c(0.025,0.5,0.975))
quantile(beta[ind,12],c(0.025,0.5,0.975))
quantile(beta[ind,13],c(0.025,0.5,0.975))
quantile(beta[ind,14],c(0.025,0.5,0.975))
quantile(beta[ind,15],c(0.025,0.5,0.975))
## plots check
par(mfrow=c(2,2))
ts.plot(lamda);acf(lamda,lag=5000)
ts.plot(beta[,1]);acf(beta[,1],lag=500)
ts.plot(beta[,5]);acf(beta[,5])
ts.plot(tau.2[,1]);acf(tau.2[,1])
ts.plot(tau.2[,5]);acf(tau.2[,5])
## after burn in and slicing plots
ts.plot(beta[ind,1]);acf(beta[ind,1],lag=100)

##################################
############  E-M  ###############
##################################
library(mvtnorm)
library(MASS)
library(truncnorm) ## rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
library(statmod) ## rinvgauss(10,1,1)
```

```
library(VGAM) ## rinv.gaussian(10,1,1)
library(LearnBayes)
## Prepare Simulation Data ##
set.seed(100)
n=1024
d=15
x1<-rnorm(n,3,1)
x2<-rnorm(n,x1,1)
x3<-rnorm(n,x2,2)
x4<-runif(n,5,10)
x5<-runif(n,x4,x4+3)
x6<-rnorm(n,3.5,1)
x7<-rnorm(n,x6,1)
x8<-x4+x7+1
x9<-runif(n,x8,x8+3)
x10<-runif(n,x9,x9+1)
x11<-rnorm(n,5,1)
x12<-rnorm(n,x11,1)
x13<-rnorm(n,x12,2)
x14<-runif(n,5,10)
x15<-runif(n,x14,x14+3)
X<-cbind(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15)
X[1:5,]
apply(X,2,mean)
beta<-c(4,-4,5,7,-6,rep(0,10));beta
```

```
length(beta)
Xb<-X%*%beta
Xb
# Obtain the vector with probabilities of success p using the probit link
p<-pnorm(Xb)
p
y.data<-rbinom(1024,1,p)
y.data
length(y.data)
## z
z<-rep(NA,n)
N.sim<-270000
N1<-sum(y.data);N1
N0<-n-N1;N0
## beta, initail beta now from Giibbs Sampler
beta<-matrix(NA,nrow=N.sim+1,ncol=d)
beta[1,]<-c(4.3,-2.9,5.4,12.3,-10.5,0.3,-0.2,0,1.0,-0.5,0.4,-0.4,-0.09,-0.6,0.4)
beta
## tau
tau.2<-matrix(NA,nrow=N.sim+1,ncol=d)
tau.2[1,]<-c(rep(1,15))
tau.2
## lamda
lamda<-rep(NA,n)
lamda[1]<-2
```

```
lamda
for (j in 1:N.sim){
  mu.z<-X%*%beta[j,]
  z[y.data==0]<-rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
  z[y.data==1]<-rtruncnorm(N1,mean=mu.z[y.data==1],sd=1,a=0,b=Inf)
  tau<-diag(tau.2[j,])
  E<-solve(solve(tau)+t(X)%*%X)
  beta[j+1,]<-rmvnorm(1,E%*%t(X)%*%z,E)
  tau.2[j+1,]<-1/rinv.gaussian(d,(lamda[j]^2/beta[j+1,]^2)^0.5,lamda[j]^2)
  lamda[j+1]<-(2*d/(sum((abs(lamda[j]*beta[j+1,])+1)/lamda[j]^2)))^0.5
}
##estimation
lamda
summary(lamda)
z
beta[,1]
tau.2[,1]
ind<-seq(from=20000,to=270000,by=250)
summary(lamda[ind])##estimate lambda
sd(lamda[ind])
plot(lamda[ind])
colMeans(beta[ind,])##estimate beta
colMeans(tau.2[ind,])##estimate tau
newb<-beta[ind,]
sd.beta<-rep(NA,15)
```

```
for (k in 1:15){
  sd.beta[k]<-sd(newb[,k])
}
sd.beta##sd of beta estimates
quantile(beta[ind,1],c(0.025,0.5,0.975))
quantile(beta[ind,2],c(0.025,0.5,0.975))
quantile(beta[ind,3],c(0.025,0.5,0.975))
quantile(beta[ind,4],c(0.025,0.5,0.975))
quantile(beta[ind,5],c(0.025,0.5,0.975))
quantile(beta[ind,6],c(0.025,0.5,0.975))
quantile(beta[ind,7],c(0.025,0.5,0.975))
quantile(beta[ind,8],c(0.025,0.5,0.975))
quantile(beta[ind,9],c(0.025,0.5,0.975))
quantile(beta[ind,10],c(0.025,0.5,0.975))
quantile(beta[ind,11],c(0.025,0.5,0.975))
quantile(beta[ind,12],c(0.025,0.5,0.975))
quantile(beta[ind,13],c(0.025,0.5,0.975))
quantile(beta[ind,14],c(0.025,0.5,0.975))
quantile(beta[ind,15],c(0.025,0.5,0.975))
## plots check
par(mfrow=c(2,2))
ts.plot(lamda);acf(lamda,lag=5000)
ts.plot(beta[,1]);acf(beta[,1],lag=500)
ts.plot(beta[,5]);acf(beta[,5])
ts.plot(tau.2[,1]);acf(tau.2[,1])
```

```
ts.plot(tau.2[,5]);acf(tau.2[,5])
## after burn in and slicing plots
ts.plot(beta[ind,1]);acf(beta[ind,1],lag=100)


############################
######   Consistency check    ###
############################
############################
############  E-M  ##########
############################
library(mvtnorm)
library(MASS)
library(truncnorm) ## rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
library(statmod) ## rinvgauss(10,1,1)
library(VGAM) ## rinv.gaussian(10,1,1)
library(LearnBayes)
## Prepare Simulation Data ##
n=1024
d=15
x1<-rnorm(n,3,1)
x2<-rnorm(n,x1,1)
x3<-rnorm(n,x2,2)
x4<-runif(n,5,10)
x5<-runif(n,x4,x4+3)
x6<-rnorm(n,3.5,1)
```

```
x7<-rnorm(n,x6,1)
x8<-x4+x7+1
x9<-runif(n,x8,x8+3)
x10<-runif(n,x9,x9+1)
x11<-rnorm(n,5,1)
x12<-rnorm(n,x11,1)
x13<-rnorm(n,x12,2)
x14<-runif(n,5,10)
x15<-runif(n,x14,x14+3)
X<-cbind(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15)
X[1:5,]
apply(X,2,mean)
beta<-c(4,-4,5,7,-6,rep(0,10));beta
length(beta)
Xb<-X%*%beta
Xb
# Obtain the vector with probabilities of success p using the probit link
p<-pnorm(Xb)
p
y.data<-rbinom(1024,1,p)
y.data
length(y.data)
## z
z<-rep(NA,n)
N.sim<-60000
```

```
N1<-sum(y.data);N1
N0<-n-N1;N0
## beta, initail beta now from Giibbs Sampler
beta<-matrix(NA,nrow=N.sim+1,ncol=d)
beta[1,]<-c(4.3,-2.9,5.4,12.3,-10.5,0.3,-0.2,0,1.0,-0.5,0.4,-0.4,-0.09,-0.6,0.4)
beta
## tau
tau.2<-matrix(NA,nrow=N.sim+1,ncol=d)
tau.2[1,]<-c(rep(1,15))
tau.2
## lamda
lamda<-rep(NA,n)
lamda[1]<-2
lamda
replicate(
  30,
{for (j in 1:N.sim){
  mu.z<-X%*%beta[j,]
  z[y.data==0]<-rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
  z[y.data==1]<-rtruncnorm(N1,mean=mu.z[y.data==1],sd=1,a=0,b=Inf)
  tau<-diag(tau.2[j,])
  E<-solve(solve(tau)+t(X)%*%X)
  beta[j+1,]<-rmvnorm(1,E%*%t(X)%*%z,E)
  tau.2[j+1,]<-1/rinv.gaussian(d,(lamda[j]^2/beta[j+1,]^2)^0.5,lamda[j]^2)
  lamda[j+1]<-(2*d/(sum((abs(lamda[j]*beta[j+1,])+1)/lamda[j]^2)))^0.5
```

```
}
  ind<-seq(from=10000,to=60000,by=50)
  check<-colMeans(beta[ind,])##estimate beta
  check
}
)
par(mfrow=c(1,1))
box<-read.csv("C:/Users/di.gao/Desktop/simulation 30.csv",header=T)
box
boxplot(box)
```

APPENDIX C. R CODE FOR BAYESIAN LASSO MODEL BRACKETING

```
##############################
###### Package Installing #######
##############################
library(mvtnorm)
library(MASS)
library(truncnorm) ## rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
library(statmod) ## rinvgauss(10,1,1)
library(VGAM) ## rinv.gaussian(10,1,1)
library(LearnBayes)
###############################
###### Data Cleaning ############
###############################
##setwd("I:/NCAA/NewModel")
BB<-read.csv("C:/Users/di.gao/Desktop/NCAA Data (updated on 3.14.2018).csv",header=T)
BB
m<- 16*64; Cx<- 2:15; Cy<- 16:21
X<- BB[1:m,Cx]; RR<- BB[1:m,Cy]
X
RR
X[1:6,]
RR[1:6,]
#summary(X[,15])
#table(X[,15])
```

```
#1st round#
a<- seq(from=1, to=m-1, by=2)
b<- seq(from=2, to=m, by=2)
Z<- (X[a,]-X[b,])/(abs(X[a,])+abs(X[b,]));
seedinfo<- cbind(X[a,1],X[b,1])
rY<- cbind(RR[a,1],RR[b,1])
rY

#2nd-6th round#
for (k in 2:6) {
  id<- which(RR[,(k-1)]==1)
  s<- 2^(k-1)
  a<- seq(from=1, to=m/s-1, by=2)
  b<- seq(from=2, to=m/s, by=2)
  Z.t<- (X[id[a],]-X[id[b],])/(abs(X[id[a],])+abs(X[id[b],]));
  Y.t<- cbind(RR[id[a],k],RR[id[b],k])
  Z<- rbind(Z,Z.t)
  rY<- rbind(rY,Y.t)
  seedinfo.k<- cbind(X[id[a],1],X[id[b],1])
  seedinfo<- rbind(seedinfo,seedinfo.k)
}

Z<- as.matrix(Z)
rowSums(rY)
Y<- rY[,1]
```

```
Y
seed.c<- seedinfo[,1]*17+seedinfo[,2]

game.data<- data.frame(seed.c,Y,Z)
colnames(game.data)[1]<- c("seed.c")
n<- m/2+m/4+m/8+m/16+m/32+m/64
n

Y
Z[1:6,]
y.data<-Y
X<-Z
y.data
X
n<-length(y.data);n
d<-length(t(X[1,]));d
########  Initial Betas Estimate  ##
fit<-glm(Y~0+X,family=binomial(link=logit))
summary(fit)
################################
########  Gibbs Samplar  ########
################################
z<-rep(NA,n)
N.sim<-60000
N1<-sum(y.data);N1
```

```
N0<-n-N1;N0

beta<-matrix(NA,nrow=N.sim+1,ncol=d)
beta[1,]<-c(0.8,7.7,62.7,-58.8,-1.2,-17.1,1.2,0.1,-0.3,0.3,-3.2,-2.8,-0.6,-0.3)
beta

tau.2<-matrix(NA,nrow=N.sim+1,ncol=d)
tau.2[1,]<-c(rep(1,d))
tau.2

lamda<-rep(NA,n)
lamda[1]<-2
lamda

for (j in 1:N.sim){
  mu.z<-X%*%beta[j,]
  z[y.data==0]<-rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
  z[y.data==1]<-rtruncnorm(N1,mean=mu.z[y.data==1],sd=1,a=0,b=Inf)
  tau<-diag(tau.2[j,])
  E<-solve(solve(tau)+t(X)%*%X)
  beta[j+1,]<-rmvnorm(1,E%*%t(X)%*%z,E)
  tau.2[j+1,]<-1/rinv.gaussian(d,(lamda[j]^2/beta[j+1,]^2)^0.5,lamda[j]^2)
  lamda[j+1]<-(rgamma(1,d+1,0.5*sum(tau.2[j+1,])))^0.5
}
```

```
lamda
par(mfrow=c(1,1))
plot(lamda)## very stable but need to check after burn in and slcing
z
beta
beta[,1]
tau.2
tau.2[,1]
par(mfrow=c(2,2))
ts.plot(lamda);acf(lamda,lag=50)
ts.plot(beta[,1]);acf(beta[,1],lag=50)
ts.plot(beta[,5]);acf(beta[,5])
ts.plot(tau.2[,1]);acf(tau.2[,1])
ts.plot(tau.2[,5]);acf(tau.2[,5])
ind<-seq(from=10001,to=60000,by=50)
ts.plot(lamda[ind])
summary(lamda[ind])##estimate lambda
sd(lamda[ind])
colMeans(tau.2[ind,])##estimate tau squared
colMeans(beta[ind,])##estimate beta
newb<-beta[ind,]
newb
sd.beta<-rep(NA,14)
for (k in 1:14){
  sd.beta[k]<-sd(newb[,k])
```

```
}
sd.beta##sd of beta estimates
quantile(beta[ind,1],c(0.025,0.5,0.975))
quantile(beta[ind,2],c(0.025,0.5,0.975))
quantile(beta[ind,3],c(0.025,0.5,0.975))
quantile(beta[ind,4],c(0.025,0.5,0.975))
quantile(beta[ind,5],c(0.025,0.5,0.975))
quantile(beta[ind,6],c(0.025,0.5,0.975))
quantile(beta[ind,7],c(0.025,0.5,0.975))
quantile(beta[ind,8],c(0.025,0.5,0.975))
quantile(beta[ind,9],c(0.025,0.5,0.975))
quantile(beta[ind,10],c(0.025,0.5,0.975))
quantile(beta[ind,11],c(0.025,0.5,0.975))
quantile(beta[ind,12],c(0.025,0.5,0.975))
quantile(beta[ind,13],c(0.025,0.5,0.975))
quantile(beta[ind,14],c(0.025,0.5,0.975))

##################################
############  E-M  ###############
##################################
z<-rep(NA,n)
N.sim<-60000
N1<-sum(y.data);N1
N0<-n-N1;N0
```

```
beta<-matrix(NA,nrow=N.sim+1,ncol=d)
beta[1,]<-c(0.8,7.7,62.7,-58.8,-1.2,-17.1,1.2,0.1,-0.3,0.3,-3.2,-2.8,-0.6,-0.3)
beta

tau.2<-matrix(NA,nrow=N.sim+1,ncol=d)
tau.2[1,]<-c(rep(1,d))
tau.2

lamda<-rep(NA,n)
lamda[1]<-2
lamda
##lamda[2]<-(2*d/(sum((abs(lamda[1]*beta[1,])+1)/lamda[1]^2)))^0.5
##lamda
for (j in 1:N.sim){
  mu.z<-X%*%beta[j,]
  z[y.data==0]<-rtruncnorm(N0,mean=mu.z[y.data==0],sd=1,a=-Inf,b=0)
  z[y.data==1]<-rtruncnorm(N1,mean=mu.z[y.data==1],sd=1,a=0,b=Inf)
  tau<-diag(tau.2[j,])
  E<-solve(solve(tau)+t(X)%*%X)
  beta[j+1,]<-rmvnorm(1,E%*%t(X)%*%z,E)
  tau.2[j+1,]<-1/rinv.gaussian(d,(lamda[j]^2/beta[j+1,]^2)^0.5,lamda[j]^2)
  lamda[j+1]<-(2*d/(sum((abs(lamda[j]*beta[j+1,])+1)/lamda[j]^2)))^0.5
}

lamda
```

```
summary(lamda)
plot(lamda)## very stable
z
beta
beta[,1]
tau.2
tau.2[,1]
par(mfrow=c(2,1))
ts.plot(lamda);acf(lamda)
ts.plot(beta[,2]);acf(beta[,2])
ts.plot(beta[,7]);acf(beta[,7])
ts.plot(tau.2[,1]);acf(tau.2[,1])
ts.plot(tau.2[,5]);acf(tau.2[,5])
ind<-seq(from=5001,to=60000,by=50)
ts.plot(lamda[ind]) ## plot check lambda stalibility
summary(lamda[ind])##estimate lambda
sd(lamda[ind])
beta[ind,]
colMeans(tau.2[ind,])##estimate tau squared
colMeans(beta[ind,])##estimate beta
newb<-beta[ind,]
newb
sd.beta<-rep(NA,14)
for (k in 1:14){
  sd.beta[k]<-sd(newb[,k])
```

```
}
sd.beta##sd of beta estimates


quantile(beta[ind,1],c(0.025,0.5,0.975))
quantile(beta[ind,2],c(0.025,0.5,0.975))
quantile(beta[ind,3],c(0.025,0.5,0.975))
quantile(beta[ind,4],c(0.025,0.5,0.975))
quantile(beta[ind,5],c(0.025,0.5,0.975))
quantile(beta[ind,6],c(0.025,0.5,0.975))
quantile(beta[ind,7],c(0.025,0.5,0.975))
quantile(beta[ind,8],c(0.025,0.5,0.975))
quantile(beta[ind,9],c(0.025,0.5,0.975))
quantile(beta[ind,10],c(0.025,0.5,0.975))
quantile(beta[ind,11],c(0.025,0.5,0.975))
quantile(beta[ind,12],c(0.025,0.5,0.975))
quantile(beta[ind,13],c(0.025,0.5,0.975))
quantile(beta[ind,14],c(0.025,0.5,0.975))



##################################
########## Bracketing ############
##################################
np<- Q<- matrix(NA,nrow=64,ncol=6)   #Probability matrix#
nX<- BB[(m+1):(m+64),Cx]
```

```
#1st round prediction#
a1<- seq(from=1, to=63, by=2)
b1<- seq(from=2, to=64, by=2)
nZ<- (nX[a1,]-nX[b1,])/(abs(nX[a1,])+abs(nX[b1,]))
nZ<- as.matrix(nZ);nZ

neta.s<- nZ%*%t(beta.rs)
np[a1,1] <- temp<- apply(exp(neta.s)/(1+exp(neta.s)),1,mean)
np[b1,1] <- 1-temp
np

#2nd - 6th round prediction#
for (k in 2:6) {
  nu<- 2^(6-k); ri<-2^(k-1)
  for (t in 1:nu) {
    ini<-(t-1)*2*ri; mi<-ini+ri; ui<-ini+2*ri;

    for (s in (ini+1):mi) {
      psum<- 0
      for (i in 1:ri) {
        mZ<- (nX[s,]-nX[(mi+i),])/(abs(nX[s,])+abs(nX[(mi+i),]))
        mZ<- as.matrix(mZ)
        meta.s<- mZ%*%t(beta.rs)
        pc<- apply(exp(meta.s)/(1+exp(meta.s)),1,mean)
        psum<- psum+np[s,(k-1)]*np[(mi+i),(k-1)]*pc
```

```
    }
    np[s,k]<- psum
   }

   for (s in (mi+1): ui) {
    psum<- 0
    for (i in 1:ri) {
     mZ<- (nX[s,]-nX[(ini+i),])/(abs(nX[s,])+abs(nX[(ini+i),]))
     mZ<- as.matrix(mZ)
     meta.s<- mZ%*%t(beta.rs)
     pc<- apply(exp(meta.s)/(1+exp(meta.s)),1,mean)
     psum<- psum+np[s,(k-1)]*np[(ini+i),(k-1)]*pc
    }
    np[s,k]<- psum
   }
  }

  for (t in 1:nu) {
   ini<-(t-1)*2*ri; mi<-ini+ri; ui<-ini+2*ri;
   cat(sum(np[(ini+1):ui,k]), "\n")
  }
}


for (r in 1:6) {
```

```
  nu<- 2^(6-r); ri<-2^r
  for (t in 1:nu) {
    ini<-(t-1)*ri; mi<-ini+1; ui<-ini+ri;
    idmax<- which.max(np[mi:ui,r])+ini
    for (s in mi:ui) {
      Q[s,r]<- (s==idmax)*1
    }
  }
}
colSums(Q)
np;Q
m

R<- as.matrix(BB[(m+1):(m+64),16:21])

1-(sum(abs(Q-R))/2)/63  #accuracy of bracketing using single scoring system#
ds<- matrix(c(1,2,4,8,16,32),6,1)
1-sum((abs(Q-R)%*%ds)/2)/192  #accuracy of bracketing using double scoring system#
```