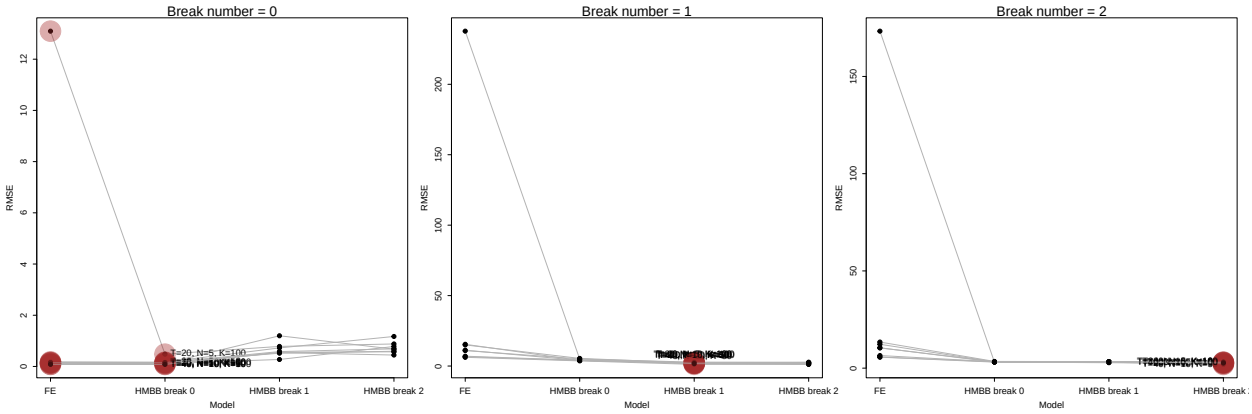
Supplementary Information for Regularization, Change-point Detection, and Variable Selection using Bayesian Inference

Jong Hee Park*
Seoul National University
jongheepark@snu.ac.kr

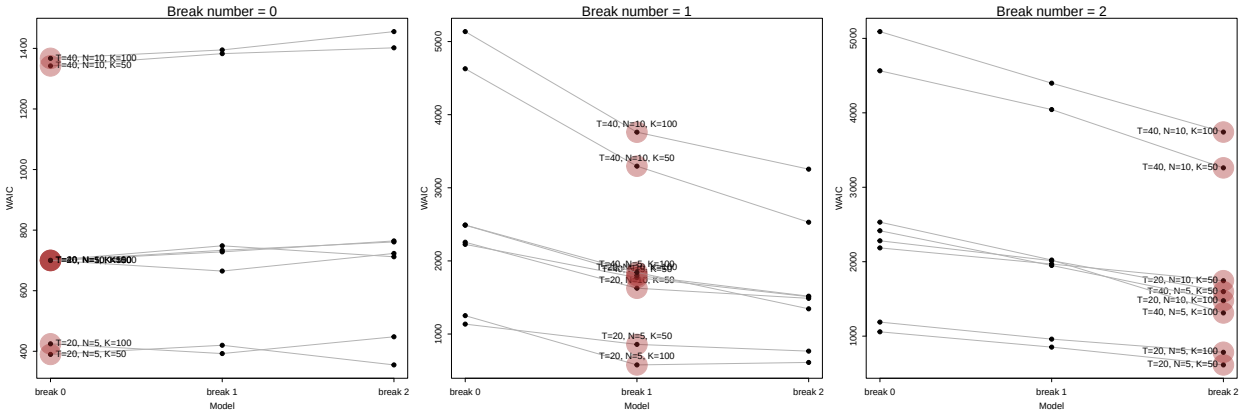
Soichiro Yamauchi†
Harvard University
syamauchi@g.harvard.edu

Contents

*Department of Political Science and International Relations, Director of IR Data Center, Seoul National University
†Department of Government, Harvard University



(a) RMSE: True states are marked by brown circles.



(b) WAIC: True states are marked by brown circles.

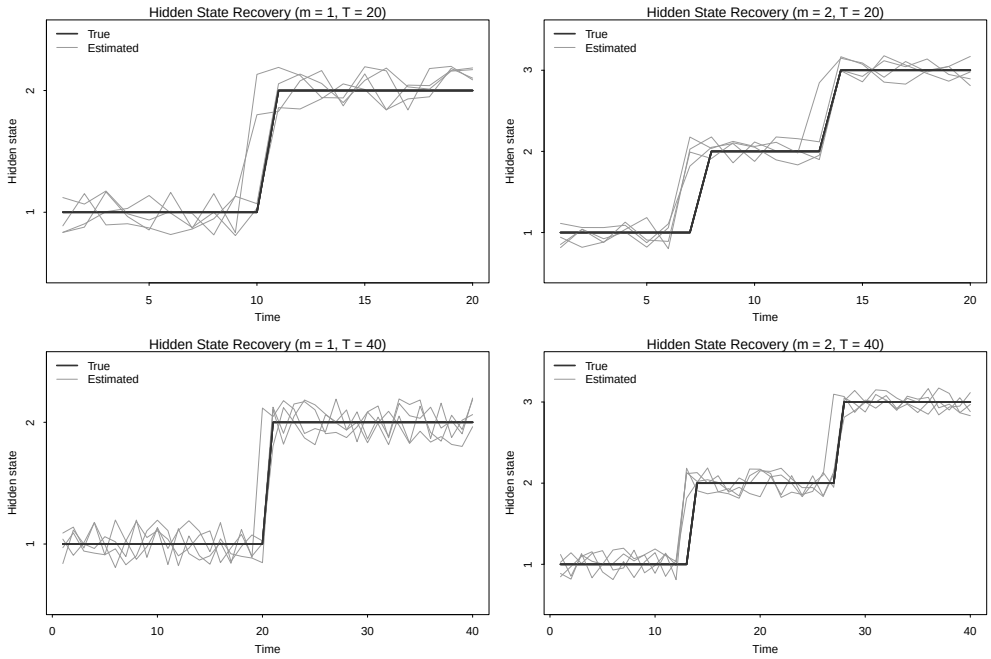
Figure 1: Simulation outcomes from 24 sets of TSCS data. (a) indicates the root mean squared error of time-varying parameters. (b) is the Watanabe-Akaike Information Criterion. A lower score suggests that the model good predictive accuracy.

1 Simulation Results under a Large Predictor Size

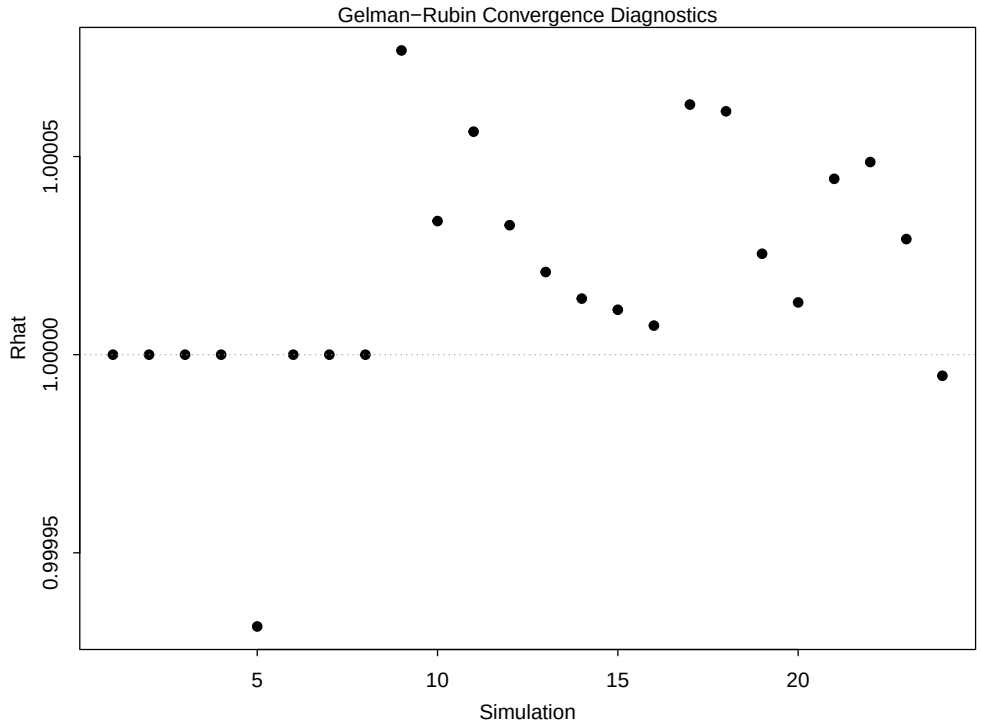
We construct 24 sets of high dimensional TSCS data with varied group sizes ($n = (5, 10)$), time lengths ($t = (20, 40)$), predictor sizes ($k = (50, 100)$), and break numbers ($m = (0, 1, 2)$) to evaluate the validity of our proposed method under a more challenging setting.

Figure **??** shows the results of RMSE and WAIC computation using simulation outputs. Overall, despite the large predictor size, WAIC and RMSE of the "correct" models show low scores compared to "incorrect" models though WAIC still shows a sign of over-detection when the planted break number is one.

Panel (a) of Figure 2 shows that HMBB successfully recovered hidden state changes in high dimensional panel data settings. Panel (b) of Figure 2 shows that HMBB converged very well in all simulation tests.



(a) Hidden state recovery: Bright lines are estimates.



(b) Convergence diagnostics: stabilized Gelman-Rubin statistics

Figure 2: Simulation outcomes from 24 sets of TSCS data. (a) shows recovered hidden states (grey) over true states (black). (b) is a stabilized Gelman-Rubin statistics (Vats and Knudson, 2021). Values close to 1 indicate good convergence.

2 BridgeChange Software

We explain how to use BridgeChange using a synthetic data set. First, download BridgeChange from a public repository.

```
require(devtools)
install_github("jongheepark/BridgeChange")
require(BridgeChange)
```

```
set.seed(11199)
K <- 80
n <- 100
X <- matrix(rnorm(n*K), n, K)
sig2 <- 4
beta.true <- matrix(NA, 2, K)

beta.true[1,] <- matrix(rnorm(K, 1, 1), K, 1)*2
beta.true[2,] <- matrix(rnorm(K, -1, 1), K, 1)
mu1 <- X[1:(n/2), ]%*%beta.true[1,]
mu2 <- X[((n/2)+1):n, ]%*%beta.true[2,]
Y  <- c(rnorm(n/2, mu1, sqrt(sig2)), rnorm(n/2, mu2, sqrt(sig2)))
formula <- Y ~ X
fit.cp0 <- BridgeChangeReg(formula, mcmc=G, burn=G, n.break = 0, Waic = TRUE)
fit.cp1 <- BridgeChangeReg(formula, mcmc=G, burn=G, n.break = 1, Waic = TRUE)
fit.cp2 <- BridgeChangeReg(formula, mcmc=G, burn=G, n.break = 2, Waic = TRUE)
```

Users must specify a formula, data, and the number of break (n.break) to fit BridgeChangeReg.

After fitting HMBB, users can do various types of posterior analysis. For example, in order to detect the number of break points and regime-specific parameters, we can compute WAIC, compare parameter estimates with true values, and draw time-varying parameter movements. Figure 3 shows each of the results.

```
par(mfrow=c(1,4))
## panel 1 waic
waic <- WaicCompare(list(fit.cp0, fit.cp1, fit.cp2), print = TRUE)
plotWaic(waic)

## pull out beta estimates
beta.est <-  matrix(apply(fit.cp1[, grep("beta", colnames(fit.cp1))], 2, mean),
                                2, , byrow=TRUE)

## panel 2 regime 1 parameter estimates vs true regime 1
plot(beta.true[1,], beta.est[1,],  xlab="TRUE", ylab="EST", type = "n",
        main="Regime 1",
```

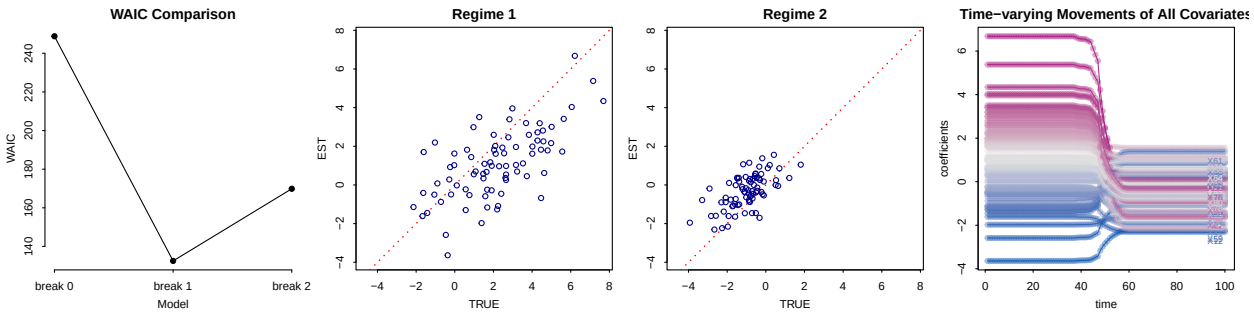


Figure 3: Change-point Analysis using HMBB: $T = 100$ (the number of observations) and $p = 80$ (the number of predictors). A single break is planted at $t = 50$.

```
               xlim = range(beta.est), ylim = range(beta.true), asp = 1)
abline(a=0, b=1, col="red", lty = 3, lwd = 1.5)
points(beta.true[1,], beta.est[1,], col="darkblue")

## panel 3 regime 2 parameter estimates vs true regime 2
plot(beta.true[2,], beta.est[2,],  xlab="TRUE", ylab="EST", type = "n",
       main="Regime 2",
              xlim = range(beta.est), ylim = range(beta.true), asp = 1)
abline(a=0, b=1, col="red", lty = 3, lwd = 1.5)
points(beta.true[2,], beta.est[2,], col="darkblue")

## panel 4 time varying parameter movements
dotplotRegime(fit.cp1, hybrid=FALSE, location.bar=12, x.location="default",
             text.cex=0.8, main="Time-varying Movements of All Covariates")
```

We can compare hidden state estimates of a single break HMBB (left) and a two-break HMBB (right). It is clear that the single break HMBB correctly identifies the planted break point in the middle, while the two-break HMBB has a redundant state (state 2) in the middle.

```
par (mar=c(3,3,2,1), mgp=c(2,.7,0), tck=-.01);
par(mfrow=c(1, 2))
MCMCpack::plotState(fit.cp1, legend.control =c(60, 0.85), main="One break")
MCMCpack::plotState(fit.cp2, legend.control =c(60, 0.85), main="Two breaks")
```

3 Model Diagnostics using Marginal Likelihood

We can check the model uncertainty of HMBB using Chib (1995)'s candidate estimator of log marginal likelihoods. Chib (1995)'s candidate estimator of log marginal likelihoods is applicable when all parameters are sampled by Gibbs sampling methods. Although the sampling algorithm of HMBB is based on Gibbs sampling methods, it is difficult to evaluate the posterior densities of $\boldsymbol{\beta}$, $\boldsymbol{\Lambda}$, and α. We approximate the log marginal likelihood of HMBB

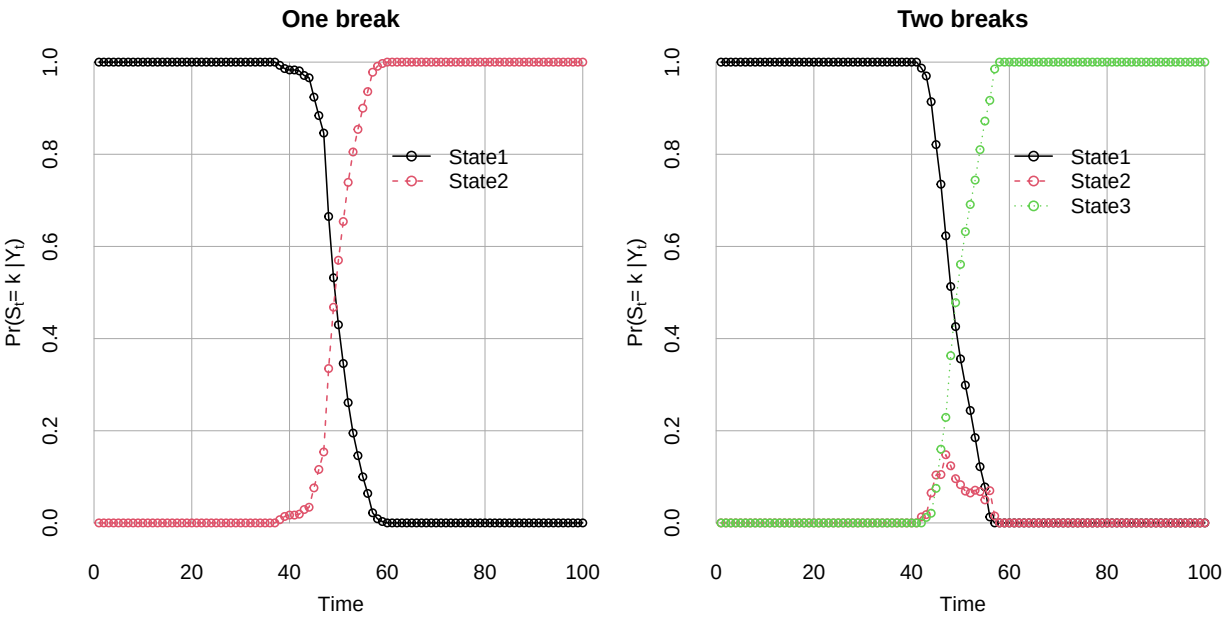


Figure 4: Hidden States Comparison of a single break HMBB (left) and a two-break HMBB (right)

by treating them as latent variables. These parameters will be averaged out in the log marginal likelihood computation. The resulting candidate estimator will be obtained from the following formula:

$$\log p(\boldsymbol{y}|\mathcal{M}_M) \approx \underbrace{\log p(\boldsymbol{y}|\sigma^{2*},\tau^*,\mathbf{P}^*,\mathcal{M}_M)}_{\text{the likelihood ordinate}} + \underbrace{\log p(\sigma^{2*},\tau^*,\mathbf{P}^*)}_{\text{the log prior ordinate}} - \underbrace{\log p(\sigma^{2*},\tau^*,\mathbf{P}^*|\boldsymbol{y})}_{\text{the log posterior ordinate}}$$

where * indicates the posterior mean. The log posterior ordinate will be evaluated from the following reduced Gibbs updates:

$$\begin{aligned}
p(\sigma^{2*},\tau^*,\mathbf{P}^*|\boldsymbol{y}) &= p(\tau^*|\boldsymbol{y})p(\sigma^{2*}|\boldsymbol{y},\tau^*)p(\mathbf{P}^*|\boldsymbol{y},\sigma^{2*},\tau^*) \\
p(\tau^*|\boldsymbol{y}) &\approx \int p(\tau^*|\boldsymbol{y},\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\sigma^2,\mathbf{P},\mathbf{S})dp(\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\sigma^2,\mathbf{P},\mathbf{S}|\boldsymbol{y}) \\
p(\sigma^{2*}|\boldsymbol{y},\tau^*) &\approx \int p(\sigma^{2*}|\boldsymbol{y},\tau^*,\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\mathbf{P},\mathbf{S})dp(\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\mathbf{P},\mathbf{S}|\boldsymbol{y}) \\
p(\mathbf{P}^*|\boldsymbol{y},\sigma^{2*},\tau^*) &\approx \int p(\tau^*|\boldsymbol{y},\sigma^{2*},\tau^*,\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\mathbf{S})dp(\boldsymbol{\beta},\boldsymbol{\Lambda},\alpha,\mathbf{S}|\boldsymbol{y}).
\end{aligned}$$

4 Simulation of HMBB

In this section we report details of simulation tests not reported in the main text. First, we test our method using high dimensional uncorrelated time series data with no change-point. Second, we test our method using high dimensional correlated time series data with no change-point. Then, we test our method using high dimensional correlated time series data

with a change-point. The level of correlation is set at 0.7 and 0.3.

4.1 Design

Following Donoho (2005) and Donoho and Stodden (2006), simulated data vary by two dimensions: the level of underdeterminedness ($\delta = n/p$) and the level of sparsity ($\rho = k/n$) where n is the number of observations and k is the number of non-sparse predictors. To make interpretation simple, we fix the number of predictors (p) at 200 and vary n from 10 to 200, and k from 1 to 200 so that both the level of underdeterminedness ($\delta = n/p$) and the sparsity level ($\rho = k/n$) take 10 equidistance points on the interval $[0.1, 1]$.

Then, we use an underlying model of $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, x_{ij} \sim N(0,1), \boldsymbol{\epsilon} \sim \text{Normal}(\mathbf{0}, 4^2\mathbf{I}_n)$ by varying δ and ρ. The change point is set at the mid point, $\lfloor n/2 \rfloor$, and coefficients are drawn independently for each regime. Based on the value of k, regression coefficients are set as $\boldsymbol{\beta}_{1:k} \sim \text{N}(0,3)$ and $\boldsymbol{\beta}_{k+1:p} = \mathbf{0}$.[1] We create 25 unique pairs of (δ, ρ) and for each pair (δ, ρ) and simulate 20 datasets from the same underlying model. In total, the number of simulated data sets for each test is $25 \times 20 = 500$. The entire test results are reported both in a numerical summary table and in the format of "phase diagrams" used by Donoho (2005) and Donoho and Stodden (2006).

4.2 Estimating Procedure for Benchmark Estimates

Table 1: Simulation Performance Criteria: I^c is an indicator matrix for the test set in 3 fold cross-validation test.

Metric	Formula	Property
Prediction Loss	$\mathcal{L}_{\text{pred}}(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^\star) = \frac{1}{n}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^\star\|^2$	in-sample model fit
Normalized Estimation Loss	$\mathcal{L}_2(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^\star) = \frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|^2}{\|\boldsymbol{\beta}^\star\|^2}$	parameter consistency.
Cross-validation Loss	$\mathcal{L}_{\text{CV}}(\widehat{\boldsymbol{y}}; \boldsymbol{y}^\star) = \frac{1}{\lvert\mathcal{I}^c\rvert}\sum_{t\in\mathcal{I}^c}(y_t - \mathbf{X}_t^\top\widehat{\boldsymbol{\beta}})^2$	out-of-sample predictive accuracy

We evaluate performance of different regularization methods using the criteria summarized in Table 1. First, Prediction Loss is related with the persistency or risk consistency (e.g., see Greenshtein and Ritov, 2004) – one of the oracle properties that high-dimensional regression estimator wishes to satisfy. Second, Normalized Estimation Loss captures parameter consistency. Achieving high performance on Normalized Estimation Loss usually requires stronger assumptions than those for the prediction loss. Last, Cross-validation Loss checks out-of-sample predictive accuracy. We conduct a 2-fold cross-validation prediction to compute the cross-validation loss.

For **`BridgeChange`** estimates, we set the correct number of break, but the location of the break point is determined by the HMBB. The point of comparison is to see (1) whether HMBB

[1] We also consider cases of a correlated design matrix. Results are reported in the supplementary information.

Table 2: Hybrid Lasso Methods and HMBB for Change-point Simulation Test

Method	Break Point	Algorithm
HMM Lasso	Unknown	Two step estimation 1. Identify the break point by the HMM analysis of the lasso residuals. ($\epsilon_t = y_t - \sum_{j=1}^{p} \mathbf{x}_{t,j}\hat{\beta}_j^{\text{Lasso}}$) 2. Apply the lasso method to subset data for regime-specific regularization.
Oracle Lasso	Known	Two step estimation 1. Subset the data based on the true break point. 2. Apply the lasso method to subset data for regime-specific regularization.
HMBB	Unknown	One step estimation

with an unknown break point outperforms a two-step approach of `Lasso-Estimate` and (2) how closely HMBB perform against `Lasso-Oracle` that uses the ground truth knowledge about a break point.

Case 1: No Change-point and Uncorrelated Data

We first report simulation results from the no change-point and uncorrelated data. We compare the performance of the Bayesian bridge estimator with that of three popular non-Bayesian regularization methods: lasso, elastic net, and ridge. We use the 2-fold cross-validation to obtain estimates from these popular methods using `cv.glmnet` in **`glmnet`** (Friedman et al., 2010).

Table 3: Summary of the No Change-point Case with 0 Correlation: The reported numbers are averaged from 500 simulated data sets. Data has no break. MCMC simulation for HMBB is 100 and burn-in is 100. Bayes Bridge indicates a HMBB with no break.

	Prediction Loss		Normalized Estimation Loss		Cross-validation Loss	
Method	Mean	SD	Mean	SD	Mean	SD
Bayes Bridge	7.683	1.983	0.677	0.276	234.590	165.139
ElasticNet	21.720	29.051	0.515	0.291	216.603	229.540
Lasso	27.810	32.055	0.513	0.301	228.641	241.369
Ridge	214.005	195.36	0.882	0.054	396.486	341.182

Table 3 reports the numerical summary of the test. Bayes Bridge has the lowest prediction loss among the four regularization techniques, and the difference is significant. In terms of cross-validation loss and normalized estimation loss, Bayes Bridge performs better than Ridge, but somewhat worse than Lasso and ElasticNet, but the difference is small.

Figure 5 visualizes the performance of the four regularization techniques across two dimensions. The x-axis (δ) increases with the size of data relative to predictors, which is set to 200 in this simulation. The y-axis (ρ) increases with the number of sparse signals relative to data.

The blue color indicates a smaller value of loss and the red color indicates a larger value of loss in each graph. Panel (A) shows the striking performance of Bayes Bridge in minimizing the prediction loss. Panel (B), which shows normalized estimation loss, shows that the performance of Bayes Bridge depends largely on the size of data compared to predictors. When the size of data is as large as that of predictors ($0.9 < \delta \leq 1$), Bayes Bridge performs well regardless of the sparsity level. In contrast, the performance of Lasso and ElasticNet depend on the sparsity level. However, when the size of data is extremely smaller than that of predictors ($0 < \delta \leq 0.1$), we recommend not to use the Bayes bridge estimator. Ridge performs poorly in most cases. Panel (C) demonstrates that Bayes Bridge has an out-of-sample prediction accuracy comparable to that of Lasso and ElasticNet.

One notable fact in Panel (B) of Figure 5 is that the performance of elastic net and lasso is clearly distinguished by the diagonal line, which corresponds to what Donoho and Stodden (2006) call the theoretical threshold of ℓ_1-based methods.[2] That is, only when $n/p > k/n$ (below the diagonal line), elastic net and lasso recover the true coefficient values very successfully.

4.3 No Change-point and Correlated Data

Next, we discuss simulation results from a correlated design matrix setup without change-point. The correlation is set at $\rho = 0.7$.

Table 4 reports the numerical summary of the test. As we have seen in the above, Bayes Bridge has the lowest prediction loss among the four regularization techniques. Also, Bayes Bridge has the lowest cross-validation loss among the four regularization techniques. In terms of normalized estimation loss, Bayes Bridge performs better than Ridge, but somewhat worse than Lasso and ElasticNet. Consequently, Bayes Bridge can be considered a superior alternative when data are correlated, which is common in empirical analysis.

Table 4: Summary of the No Change-point, Correlated Case: The reported numbers are averaged from 500 simulated data sets. Data has no break. MCMC simulation for HMBB is 100 and burn-in is 100. Bayes Bridge indicates a HMBB with no break.

	Prediction Loss		Normalized Estimation Loss		Cross-validation Loss	
Method	Mean	SD	Mean	SD	Mean	SD
Bayes Bridge	6.112	1.482	0.76	0.27	73.228	53.200
ElasticNet	11.196	9.055	0.63	0.25	74.391	70.865
Lasso	11.384	9.256	0.62	0.27	77.052	74.547
Ridge	114.959	122.949	0.97	0.01	132.698	123.946

[2] Donoho and Stodden (2006) wrote "there is a breakdown point for standard model selection schemes, such that model selection only works well below a certain critical complexity level" (Donoho and Stodden, 2006, 1)

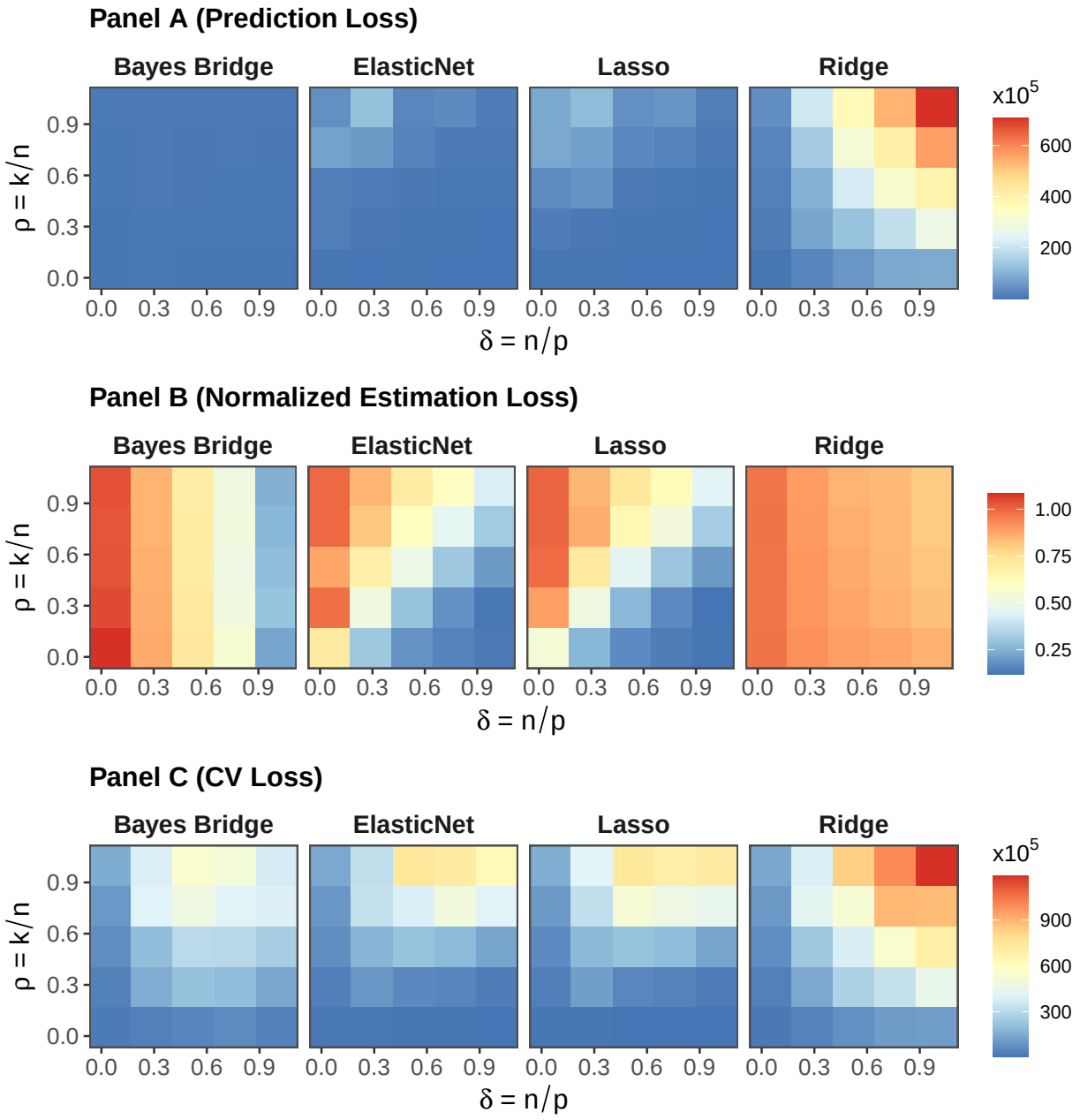


Figure 5: Panel (A): Prediction Loss, $\mathcal{L}_{\text{pred}}(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}})$. **Panel (B)**: Normalized Estimation Loss, $\mathcal{L}_2(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}}) = \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{\text{true}}\|_2 / \|\boldsymbol{\beta}^{\text{true}}\|_2$. **Panel (C)**: Cross-validation Loss, $\mathcal{L}_{\text{CV}}(\widehat{\boldsymbol{y}}^{\text{test}}; \boldsymbol{y}^{\text{test}})$. We fix $p = 200$ and vary α and ρ between 0.1 and 1. Thus, each cell in the graph represents a data with (N, p, k). We simulate 20 data sets from each (N, p, k) and take the median error.

Figure 6 visualizes the performance of the four regularization techniques against correlated data. Panel (A) shows that Bayes Bridge produces the minimum prediction loss across different sparsity levels (vertical) and data sizes (horizontal). Panel (B) shows the same pattern with Figure 5: the performance of Bayes Bridge depends largely on the size of data compared to predictors, having the best performance when $0.9 < \delta \leq 1$). Panel (C) demonstrates that Bayes Bridge has the best out-of-sample prediction accuracy among the four regularization techniques against correlated data.

4.4 A Single Change-point and Correlated Data

Last, we showed the simulation results from correlated data with a single change-point. Table 5 reports the numerical summary of the test. In this test, we compare HMBB with two best-case scenario methods. The first method is HMM Lasso, which is a two-step method that estimates a break point first using the Bayesian change-point residual test introduced by Park (2012), and then applies the lasso estimator (Tibshirani, 1996) to break-partitioned data. The second method is Oracle Lasso, which applies the lasso estimator to correctly partitioned data. The difference of the two methods is whether the break information is obtained from data (HMM Lasso) or known to researchers a priori (Oracle Lasso). In contrast, HMBB assumes no knowledge about break points and incorporate break detection and parameter estimation in one model.

Table 5 reports that in terms of in-sample fit, HMBB is comparable to HMM Lasso, and in terms of out-of-sample prediction accuracy HMBB is comparable to Oracle Lasso. However, HMBB's prediction loss is large compared to the two benchmarks. The HMBB prediction loss is greatest, as shown in the upper-left corner of the left-most Panel (A) in Figure 7. Excluding these cases, HMBB's performance in prediction loss is as good as HMM Lasso. Panel (B) of Figure 7 shows that HMBB's performance in normalized estimation loss is comparable to HMM Lasso. Panel (C) shows that in terms of cross-validation loss, HMBB slightly outperforms HMM Lasso.

Table 5: Summary of the No Change-point, Correlated Case: The reported numbers are averaged from 500 simulated data sets. Data has no break. MCMC simulation for HMBB is 100 and burn-in is 100. Bayes Bridge indicates a HMBB with no break.

	Prediction Loss		Normalized Estimation Loss		Cross-validation Loss	
Method	Mean	SD	Mean	SD	Mean	SD
HMBB	0.419	0.236	0.14	0.02	3.964	1.750
HMM Lasso	0.290	0.185	0.14	0.03	4.178	1.723
Oracle Lasso	0.215	0.091	0.09	0.02	2.751	1.476

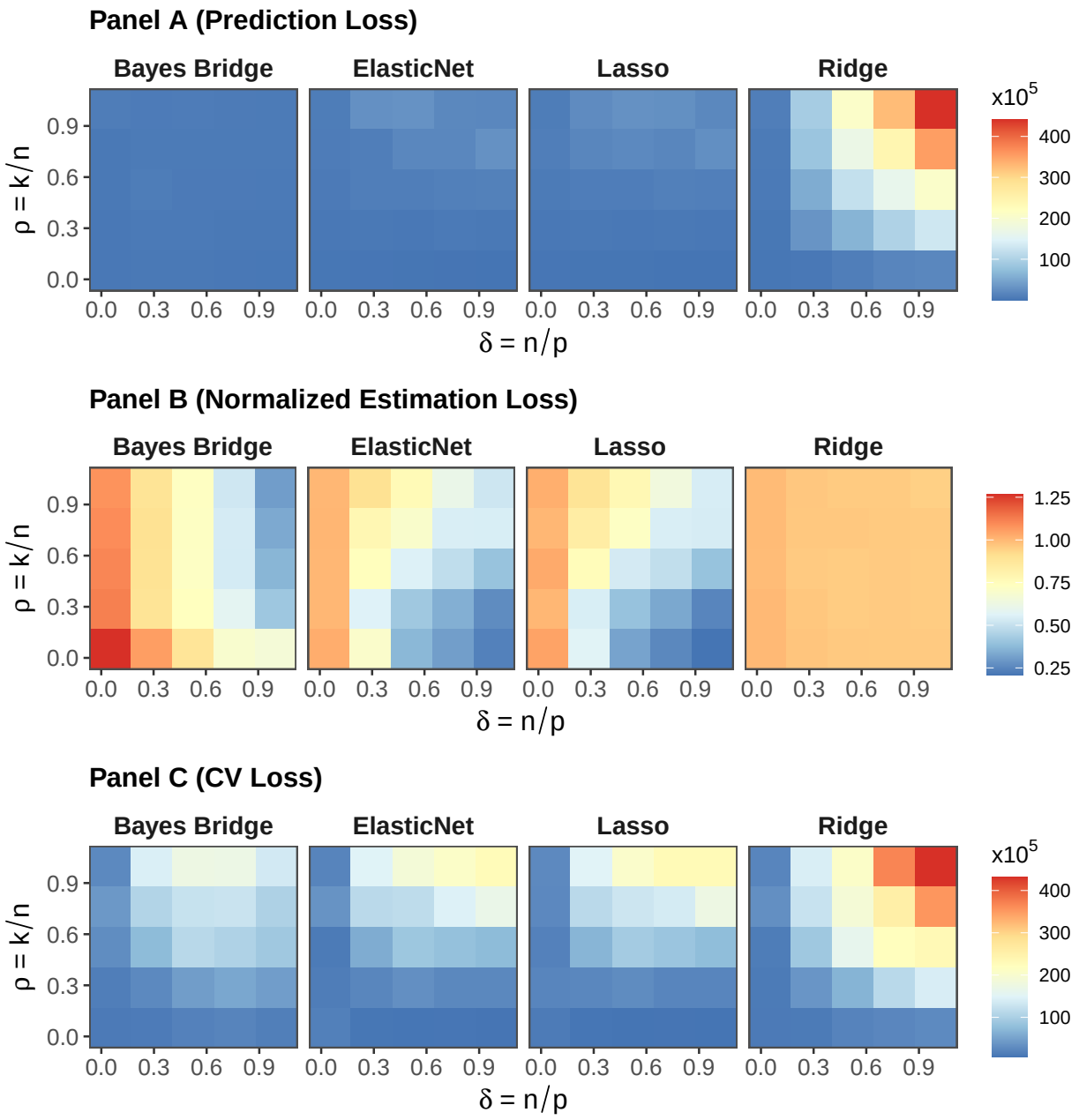


Figure 6: Results of Simulation Studies using Correlated ($\rho = 0.7$) Univariate Time Series Data with No Change-point: **Panel (A)**: Prediction Loss, $\mathcal{L}_{\text{pred}}(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}})$. **Panel (B)**: Normalized Estimation Loss, $\mathcal{L}_2(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}}) = \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{\text{true}}\|_2 / \|\boldsymbol{\beta}^{\text{true}}\|_2$. **Panel (C)**: Cross-validation Loss, $\mathcal{L}_{\text{CV}}(\widehat{\boldsymbol{y}}^{\text{test}}; \boldsymbol{y}^{\text{test}})$. We fix $p = 200$ and vary α and ρ between 0.1 and 1. Thus, each cell in the graph represents a data with (n, p, k). We simulate 20 data sets from each (δ, ρ) and take the median error.

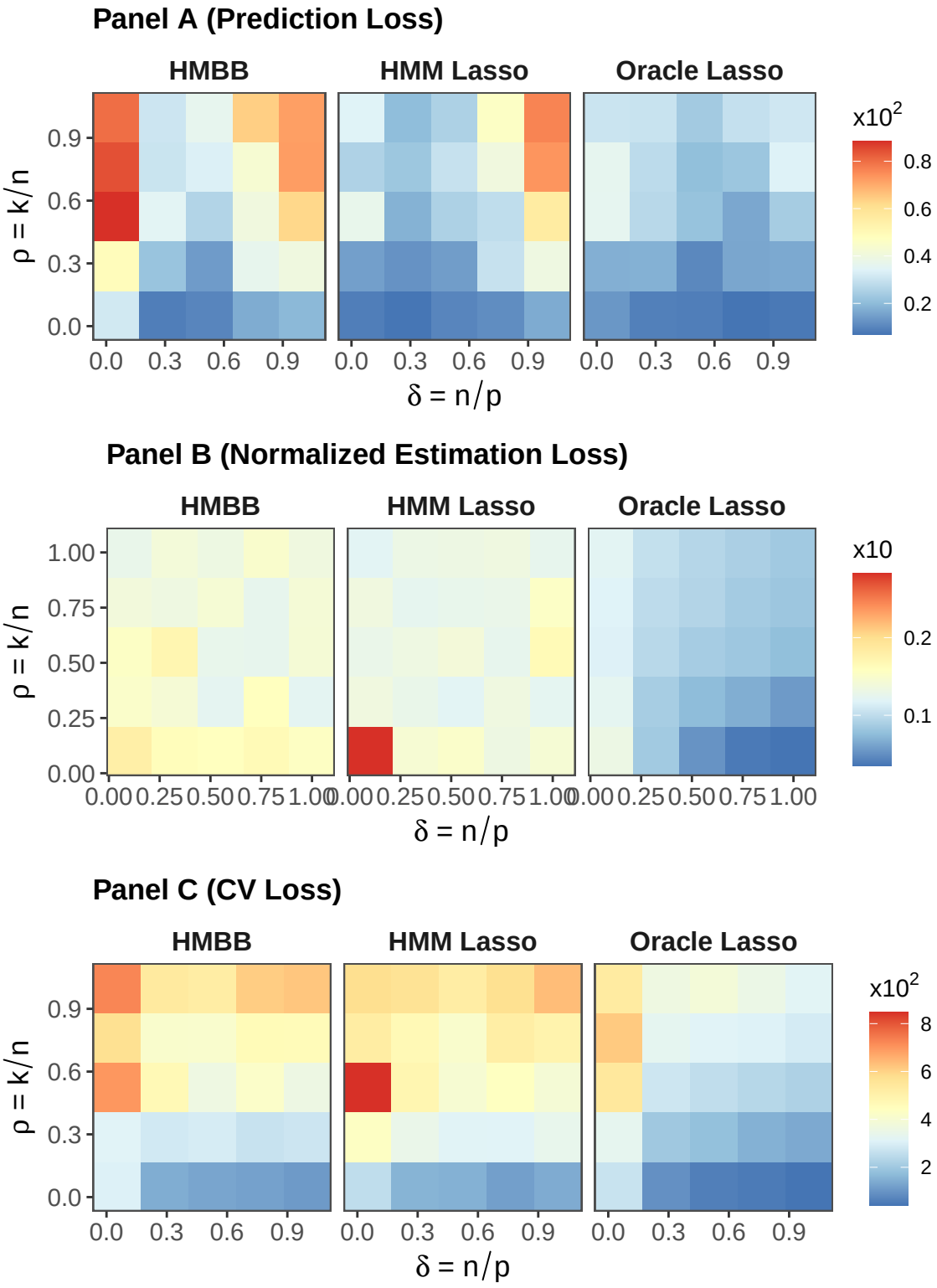


Figure 7: Results of Simulation Studies using Univariate Time Series Data with One Change-point (Cor = 0.7): **Panel (A)**: Prediction Loss, $\mathcal{L}_{\text{pred}}(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}})$. **Panel (B)**: Normalized Estimation Loss, $\mathcal{L}_2(\widehat{\boldsymbol{\beta}}; \boldsymbol{\beta}^{\text{true}}) = \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{\text{true}}\|_2 / \|\boldsymbol{\beta}^{\text{true}}\|_2$. **Panel (C)**: Cross-validation Loss, $\mathcal{L}_{\text{CV}}(\widehat{\boldsymbol{y}}^{\text{test}}; \boldsymbol{y}^{\text{test}})$. We fix $p = 200$ and vary α and ρ between 0.05 and 1. Thus, each cell in the graph represents a data with (n, p, k). We simulate 25 data sets from each (δ, ρ) and take the median error.

5 On the Bayesian Bridge Model

A natural candidate for prior distributions of the bridge estimator $\boldsymbol{\beta}$ is a product of independent exponential power priors:

$$p(\boldsymbol{\beta}) \propto \prod_{j=1}^{p} \exp(-|\beta_j/\tau|^{\alpha}), \quad \tau = \nu^{-\frac{1}{\alpha}}.$$

Polson et al. (2014) introduce latent variables of local shrinkage parameter $\Lambda = (\lambda_1, \ldots, \lambda_p)$ to de-link the correlation between $\boldsymbol{\beta}$ and τ (global shrinkage parameter). Then, using Lévy processes and scale mixtures of normal representation discussed in Polson and Scott (2012), a joint prior distribution of regression parameter $\boldsymbol{\beta}$ and local shrinkage parameter Λ are represented as follows:

$$p(\boldsymbol{\beta}, \Lambda) \quad \propto \quad \prod_{j=1}^{p} \exp\left(-\frac{\beta_j^2}{2\tau^2}\lambda_j\right) p(\lambda_j). \tag{1}$$

$p(\lambda_j)$ is the density of $2S_{\alpha/2}$ where S_α is the Lévy alpha-stable distribution. It should be noted that the Lévy process representation is introduced because the Lévy process representation allows us to construct *joint priors for* β, which provides a unifying probabilistic structure for penalized regression and variable selection from both Bayesian and classical viewpoints (Polson and Scott, 2012). According to Polson and Scott (2012), "all totally monotone penalty functions that vanish at zero correspond to priors that can be represented in terms of a subordinator"(Polson and Scott, 2012, 292).

Prior distributions of the other parameters (σ^2, α, τ) are defined as usual.

$$\begin{aligned}
\sigma^2 &\sim \text{Inverse-Gamma}\left(\frac{a_0}{2}, \frac{b_0}{2}\right) \\
\alpha &\sim \text{Uniform}\,(0, 1) \\
\nu &\sim \text{Gamma}\,(c_0, d_0)\,.
\end{aligned}$$

The posterior distribution of the Bayesian bridge linear regression model is

$$\begin{aligned}
p(\boldsymbol{\beta}, \sigma^2, \Lambda, \alpha, \nu | \boldsymbol{y}, \mathbf{X}) \quad &\propto \quad p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}, \Lambda) p(\sigma^2) p(\alpha) p(\nu) \\
&\propto \quad \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})\right\} \prod_{j=1}^{p} \exp\left(-\frac{\beta_j^2}{2\tau^2}\lambda_j\right) p(\lambda_j) \\
&\quad \times \left(\frac{1}{\sigma^2}\right)^{\frac{a_0}{2}+1} \exp\left(-\frac{b_0}{2\sigma^2}\right) \nu^{c_0-1} \exp(-d_0\nu)
\end{aligned} \tag{2}$$

According to Polson et al. (2014), this fully Bayesian approach to bridge estimation has several advantages. First, a point estimate by classical estimates or pseudo-Bayesian approaches cannot effectively summarize a multimodal surface in high dimensional parameter space. Also,

the LASSO estimator approximates the posterior mode of the Bayesian model with Laplacian prior ($\beta_j \sim \frac{\lambda}{2}\exp(-\lambda|\beta_j|)$). However, the posterior mode is inferior to the posterior mean in prediction and estimation. Thus, a fully Bayesian approach is known to outperform classical counterparts. Second, Polson et al. (2014)'s computational algorithm is more efficient than existing Bayesian regularization methods. Existing Bayesian regularization estimators suffer from a slow mixing largely because the number of local shrinkage variables (λ_j), which is not observed, grows proportionally with the number of input variables. The poor mixing problem is particularly worrisome to social scientists because design matrices in social science researches tend to be highly correlated. Polson et al. (2014) recommend the Bayesian bridge estimator using scale mixtures of normals, which we use as the baseline model, for the case of correlated design matrices.

6 Computation Time

BridgeChange is written as a high performance R package using **Rcpp** package. Table 6 shows the running time of **BridgeChange** on the `agl` data set of Alvarez et al. (1991) with $n = 16$, $k = 6$, $t = 15$ and $n \times t = 240$. The testing machine is $x86_64 - w64 - mingw32/x64(64 - bit)underWindows10x64(build19044).MCMCsimulationforparameterestimationis1000andburn - inis1000.HMBBwithnobreaktook3.8secondstorun2000MCMCsimulations.Thecomputationtimegrowsli$

Table 6: Computation time of HMBB on the `agl` data set

model	break number	test	elapsed	relative	user.self	sys.self
1	break 0	1	3.750	1.000	3.670	0.030
2	break 1	1	7.620	2.032	7.540	0.090
3	break 2	1	11.090	2.957	11.050	0.050

References

Alvarez, R. M., Garrett, G., and Lange, P. (1991). Government partisanship, labor organization, and macroeconomic performance. *American Political Science Review*, 85(2):539–556.

Chib, S. (1995). Marginal likelihood from the gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321.

Donoho, D. (2005). High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete and Computational Geometry*, 35(4):617–652.

Donoho, D. and Stodden, V. (2006). Breakdown point of model selection when the number of variables exceeds the number of observations. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1916–1921.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Greenshtein, E. and Ritov, Y. (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988.

Park, J. H. (2012). A unified method for dynamic and cross-sectional heterogeneity: Introducing hidden markov panel models. *American Journal of Political Science*, 56(4):1040–1054.

Polson, N. G. and Scott, J. G. (2012). Local shrinkage rules, lévy processes, and regularized regression. *Journal of the Royal Statistical Society (Series B)*, 74:287–311.

Polson, N. G., Scott, J. G., and Windle, J. (2014). The bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):713 – 733.

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

Vats, D. and Knudson, C. (2021). Revisiting the Gelman–Rubin Diagnostic. *Statistical Science*, 36(4):518 – 529.