BST 235 Project #5: Link Violation in GLMs

Helian Feng, Linglin Huang, Andy Shi

December 15, 2017

Abstract

We show that for generalized linear models, it is possible to proportionally recover the coefficients under link violation using adaptive lasso, for a variety of different scenarios. We show that theoretically, when the predictors follow a multivariate normal distribution, as $n \to \infty$, the adaptive lasso estimates converge to a value proportional to the true coefficients—thus, we can consistently estimate the support of the true coefficients. We verify this result using simulations, and show that this result is fairly robust even when the predictors do not follow a multivariate normal distribution. Finally, we evaluate the prediction errors of adaptive lasso and show that they can achieve comparable performance to predictions made from the true model and those from kernel smoothing.

1 Introduction

In this project, our main interest is to recover the regression coefficients and achieve reasonable predictions for generalized linear regression models under link violation (i.e. when the model is misspecified).

1.1 Problem Description

Suppose we observed n i.i.d. samples, $\{(\boldsymbol{X}_i, Y_i), i = 1, ..., n\}$, where Y_i is a continuous response, and $\boldsymbol{X}_i$ is a $p \times 1$ vector of covariates. Assume these samples were generated from the underlying model:

$$Y_i = \alpha_0 + \Phi(\boldsymbol{\beta_0}^T \boldsymbol{X_i}) + \epsilon_i,$$

with $\Phi(\cdot)$ being the CDF for standard normal distribution and ϵ_i being i.i.d. errors with mean 0 and variance σ^2 (also independent of the covariates $\boldsymbol{X_i}$).

However, we seldom know the true underlying model in reality, and might instead fit a misspecified model with incorrect link function. In this project, we fit the linear model

$$Y_i = \alpha + \boldsymbol{\beta}^T \boldsymbol{X_i} + \epsilon_i$$

with adaptive lasso, and obtained an estimator $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$. In other words,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||Y - \boldsymbol{X\beta}||^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| / |\hat{\beta}_j^{OLS}|^\gamma,$$

where $\gamma > 0$ is a tuning parameter (in our experiments we set $\gamma = 1$).

Our goal is to investigate how trustworthy this misspecified model is: to what extent could $\hat{\boldsymbol{\beta}}$ recover the true coefficients $\boldsymbol{\beta_0}$, and how good is its prediction performance on out-of-sample validation data?

1.2 Project Overview

First, we proved that theoretically, if the covariates $\boldsymbol{X_i}$ follow a multivariate normal distribution, the estimator $\hat{\boldsymbol{\beta}}$ from misspecified adaptive lasso converges to a limit $\bar{\boldsymbol{\beta}}$ that is proportional to the true coefficients $\boldsymbol{\beta_0}$. Thus, we can consistently estimate the support of $\boldsymbol{\beta_0}$.

Next, we performed simulation studies to support this theoretical result. We also investigated the robustness of this result by simulating $\boldsymbol{X_i}$'s that did not satisfy the theoretical requirements (data generated from non-elliptical distributions).

Finally, we compared the prediction performances of the true model, misspecified adaptive lasso, and nonparametric calibration on top of misspecified adaptive lasso. We also dived into the comparisons and explored when and why each of these models performed well or performed poorly.

2 Theoretical Results

In this section, we proved that if the covariates $\boldsymbol{X_i}$ follows a multivariate normal distribution, the estimator $\hat{\boldsymbol{\beta}}$ from misspecified adaptive lasso converges to a limit $\bar{\boldsymbol{\beta}}$, that is proportional to the true coefficients $\boldsymbol{\beta_0}$. Thus, we can consistently estimate the support of $\boldsymbol{\beta_0}$.

As an overview of our proof, we first showed that $\hat{\boldsymbol{\beta}}$ converges to $\bar{\boldsymbol{\beta}}$ by consistency and oracle property of adaptive lasso in Zou (2006) [1]. Next, we showed that $\bar{\boldsymbol{\beta}}$ is proportional to $\boldsymbol{\beta_0}$ by Theorem 2.2 in Li & Duan (1989) [2].

2.1 Review of Li & Duan's Result

In Li & Duan (1989) [2], they defined the objective function for regression models as:

$$R(\alpha, \boldsymbol{\beta}) = \mathbb{E}L(\alpha + \boldsymbol{\beta}^T\boldsymbol{x}, y),$$

where $L(\theta, y)$ is any minimization criterion (loss function). Each objective function $R(\alpha, \boldsymbol{\beta})$ is associated with a set

$$\Omega = \{(\alpha, \boldsymbol{\beta}) : R(\alpha, \boldsymbol{\beta}) \text{ is well-defined and is finite}\}.$$

Thus, regression model fitting can be represented as an optimization problem:

$$\text{minimize} \quad R(\alpha, \boldsymbol{\beta}) \quad \text{over} \quad (\alpha, \boldsymbol{\beta}) \in \Omega$$

They established two theorems that are closely related to our project:

Theorem 2.1 For a model of the form

$$y = g(\alpha + \boldsymbol{\beta}^T\boldsymbol{x}, \epsilon), \qquad \epsilon \sim F(\epsilon),$$

where $g(\cdot, \cdot)$ is a given link function, if

- the criterion function $L(\theta, y)$ is convex in θ with probability 1,
- the conditional expectation $\mathbb{E}[\boldsymbol{\beta}^T\boldsymbol{x}|\boldsymbol{\beta_0}^T\boldsymbol{x}]$ exists and is linear in $\boldsymbol{\beta_0}\boldsymbol{x}$ for all $\boldsymbol{\beta} \in \mathbb{R}^p$,

- Ω is a nonempty convex set in $\mathbb{R}^{p+1}$,
- $R(\alpha, \boldsymbol{\beta})$ has a proper minimizer $(\alpha^*, \boldsymbol{\beta}^*) \in \Omega$,

then $\boldsymbol{\beta}^*$ is proportional to $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^* = \eta \boldsymbol{\beta}$$

for some scalar η.

Theorem 2.2 Theorem 2.1 still holds if the convexity constraint on $L(\theta, y)$ and the conditional mean constraint on $\mathbb{E}[\boldsymbol{\beta}^T \boldsymbol{x} | \boldsymbol{\beta}_0^T \boldsymbol{x}]$ were replaced by the condition that $\boldsymbol{x}$ follows a normal distribution.

2.2 Proof of Our Result

First, by Zou (2006) [1], $\hat{\boldsymbol{\beta}} \to \bar{\boldsymbol{\beta}}$ if we pick a proper λ_n such that $\lambda_n / \sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$, where $\bar{\boldsymbol{\beta}}$ is the vector of the "true coefficients" in the misspecified model. In other words, $(\bar{\alpha}, \bar{\boldsymbol{\beta}})$ minimizes

$$R(\alpha, \boldsymbol{\beta}) = \mathbb{E}(y - \alpha - \boldsymbol{\beta} \boldsymbol{x})^2, \tag{1}$$

where $\bar{\alpha}$ is the estimated intercept corresponding to $\bar{\boldsymbol{\beta}}$ in the misspecified model.
Second, we applied Theorem 2.2 in Li & Duan (1989) [2]:

- Ω is a nonempty convex set in $\mathbb{R}^{p+1}$ given the form of $R(\alpha, \boldsymbol{\beta})$ in (1),
- $\boldsymbol{X_i}$ follows a multivariate normal distribution,
- $(\bar{\alpha}, \bar{\boldsymbol{\beta}})$ minimizes R over Ω,

therefore $\bar{\boldsymbol{\beta}} = \eta \boldsymbol{\beta_0}$ for some scalar η.
Thus, we conclude that

$$\hat{\boldsymbol{\beta}} \to \bar{\boldsymbol{\beta}} = \eta \boldsymbol{\beta_0}$$

and we can consistently estimate the support of $\boldsymbol{\beta_0}$.

3 Simulation Description

For each iteration of our simulation study, we generate training data $(\boldsymbol{X}_i, Y_i)$ for $i = 1, 2, \ldots, n$, where $Y_i = \alpha_0 + \Phi(\boldsymbol{\beta_0^T X_i}) + \epsilon_i$, and ϵ_i is independent of $\boldsymbol{X}_i$. We used varying training data sample sizes, $n \in \{500, 1000, 5000\}$. We used $\alpha_0 = 3$, $\boldsymbol{\beta_0} = (0, 0, 0, 0, 0, -0.3, -0.1, 0.1, 0.3)^T$, and $\epsilon_i \sim N(0, \sigma^2)$, where $\sigma \in \{0.01, 0.3\}$, signifying small and big errors. Also, each $\boldsymbol{x}_i$ (row of $\boldsymbol{X}_i$) was generated one of three ways:

1. Normal: $\boldsymbol{x_i} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ (elliptical);
2. Uniform: $\boldsymbol{x_i} \sim \text{Unif}(-1, 1, \boldsymbol{\Sigma})$;
3. Distorted normal: $\boldsymbol{x_i} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$, except we augment $x_{i8}^* = x_{i8} + 0.5(x_{i7}^2 - 1)$ (non-elliptical);

and where for $\rho_1, \rho_2, \rho_{12} \in \{0, 0.3\}$, $\boldsymbol{\Sigma}$ is in the form in Figure 1. For each simulation iteration, we obtained an estimate $\hat{\beta}$ of the coefficients using adaptive lasso, with the initial estimate chosen using OLS, and the penalty parameter λ selected through 5-fold cross-validation.

To generate correlated uniform random variates, we used the copula method:

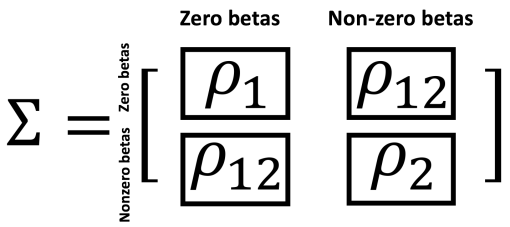

Figure 1: Correlation matrix used. The top-left box corresponds to the coefficients which have a true zero effect, and the bottom-right box corresponds to coefficients which have a true non-zero effect. This matrix has diagonal 1, and ρ_1 correlation between coefficients with truly zero effects, ρ_2 correlation between coefficients with truly non-zero effects, and ρ_{12} cross-correlation between coefficients with truly zero effects and those with truly non-zero effects.

1. Generate $(X_{i1}, \ldots, X_{ip}) \sim N(0, \Sigma)$ where Σ is a *correlation* matrix (1s on diagonal).
2. Then, get $(U_{i1}, \ldots, U_{ip}) = (\Phi(X_{i1}), \ldots, \Phi(X_{1p}))$. Marginally, $U_{ij} \sim \text{Uniform}(0, 1)$.
3. Transform $(b - a)U_{ij} + a$ to get $U_{ij} \sim \text{Uniform}(a, b)$.

Finally, for each configuration of the parameters, we generated 10000 independent test data to measure the prediction performance. Predictions were generated using three different settings:

- True model: $\tilde{m}(\boldsymbol{x}) = \tilde{\alpha} + \Phi(\tilde{\boldsymbol{\beta}}^T \boldsymbol{x})$, where $\tilde{\alpha}$ and $\tilde{\beta}$ are the coefficients obtained by minimizing the squared loss $\sum_{i=1}^{n}(Y_i - \Phi(\alpha + \boldsymbol{\beta}^T \boldsymbol{x_i}))^2$ with respect to $(\alpha, \boldsymbol{\beta})$, which we minimized with BFGS [3] in the R function `optim`.
- Adaptive lasso: $\hat{Y} = \hat{\alpha} + \hat{\boldsymbol{\beta}}^T \boldsymbol{x}$, where $\hat{\alpha}$ and $\hat{\boldsymbol{\beta}}$ are estimates from adaptive lasso, fitted with `glmnet` [4]. The initial estimate of adaptive lasso is the OLS estimate.
- Nonparametric calibration on top of adaptive lasso fit: The nonparametric calibration for a new observation $\boldsymbol{x}^0$ is

$$\hat{m}(\boldsymbol{x}^0) = \frac{\sum_{i=1}^{n} K_h(\hat{\boldsymbol{\beta}}^T \boldsymbol{X_i} - \hat{\boldsymbol{\beta}}^T \boldsymbol{x^0}) Y_i}{\sum_{i=1}^{n} K_h(\hat{\boldsymbol{\beta}}^T \boldsymbol{X_i} - \hat{\boldsymbol{\beta}}^T \boldsymbol{x^0})},$$

  where we selected K to be the standard normal pdf and the bandwidth h was selected using the direct plug-in methodology by Ruppert, Sheather and Wand (1995) [5], as implemented in the `dpill` function in the R package `KernSmooth`. The kernel smoothing was fit using the R function `ksmooth`. We considered using cross-validation to select the bandwidth, but after testing it for $n = 1000$ and $\boldsymbol{x}_i$ normally distributed, we found that there was not a big difference in prediction error, so we decided cross-validation for the bandwidth was not necessary (Figure 14).

4 Simulation Results

In the simulation, we looked at whether we could recover the coefficients proportionally, and how the prediction performance on an independent test dataset varies from 3 different prediction methods.

4.1 Proportional Recovery of Coefficients

The simulated value of the coefficients which were truly zero is shown in Figure 2. When $\sigma = 0.01$, across all of the simulations, the simulated coefficient is 0. For $\sigma = 0.3$, there is some variability around 0, which decreases as the sample size n increases. As shown in Figure 3, as the sample size increases, most of the coefficients that are truly zero are being set to zero. The exact proportion depends on the distribution, with the uniform faring worse than the other two distributions. Across all simulation conditions, at least 85% of simulated coefficients that were truly zero are set to zero (Figure 15), these results show that we are able to recover the truly zero coefficients.

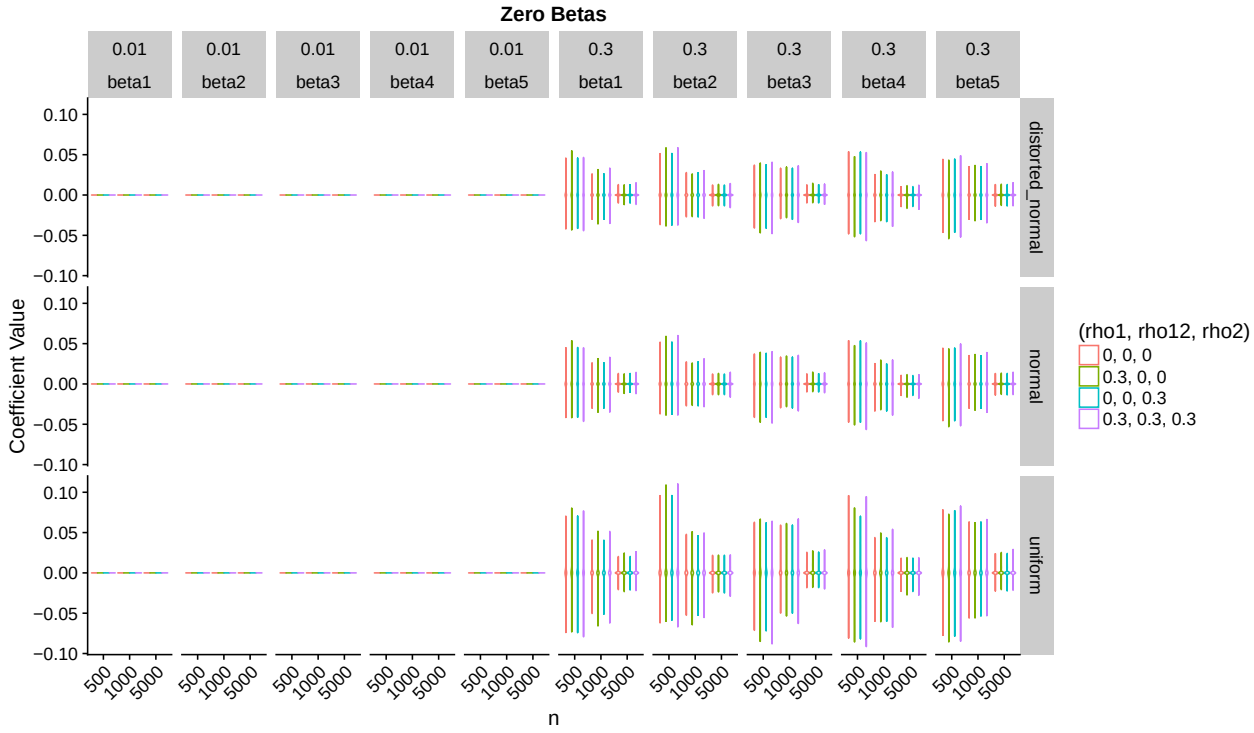

Figure 2: Plot of coefficient values for β that are truly 0. The 5 panels on the left show the case when $\sigma = 0.01$ and the 5 on the right show when $\sigma = 0.3$. The three vertical panels show the three different distributions of the covariates. The colors represent different settings of $(\rho_1, \rho_{12}, \rho_2)$.

The simulated ratio of the simulated coefficient value divided by the true value for the truly nonzero coefficients is shown in Figure 4. When $\sigma = 0.01$, across all of the simulations, the simulated ratio is close to the same value for all of the truly nonzero coefficients, indicating proportional recovery of the coefficients. For $\sigma = 0.03$, there is some variability, which decreases as the sample size n increases. As shown in Figure 5 and Table 1, the exact ratio of the simulated coefficient value divided by the true value varies for each coefficient and the distribution of the covariates. The uniform has more variability, but overall we can conclude that we can approximately recover the truly nonzero coefficients proportionally.

For both the truly zero and truly nonzero case, the correlations $\rho_1, \rho_2, \rho_{12}$ do not seem to have a big effect on the result.

Finally, we were able to recover α proportionally (Figure 18).

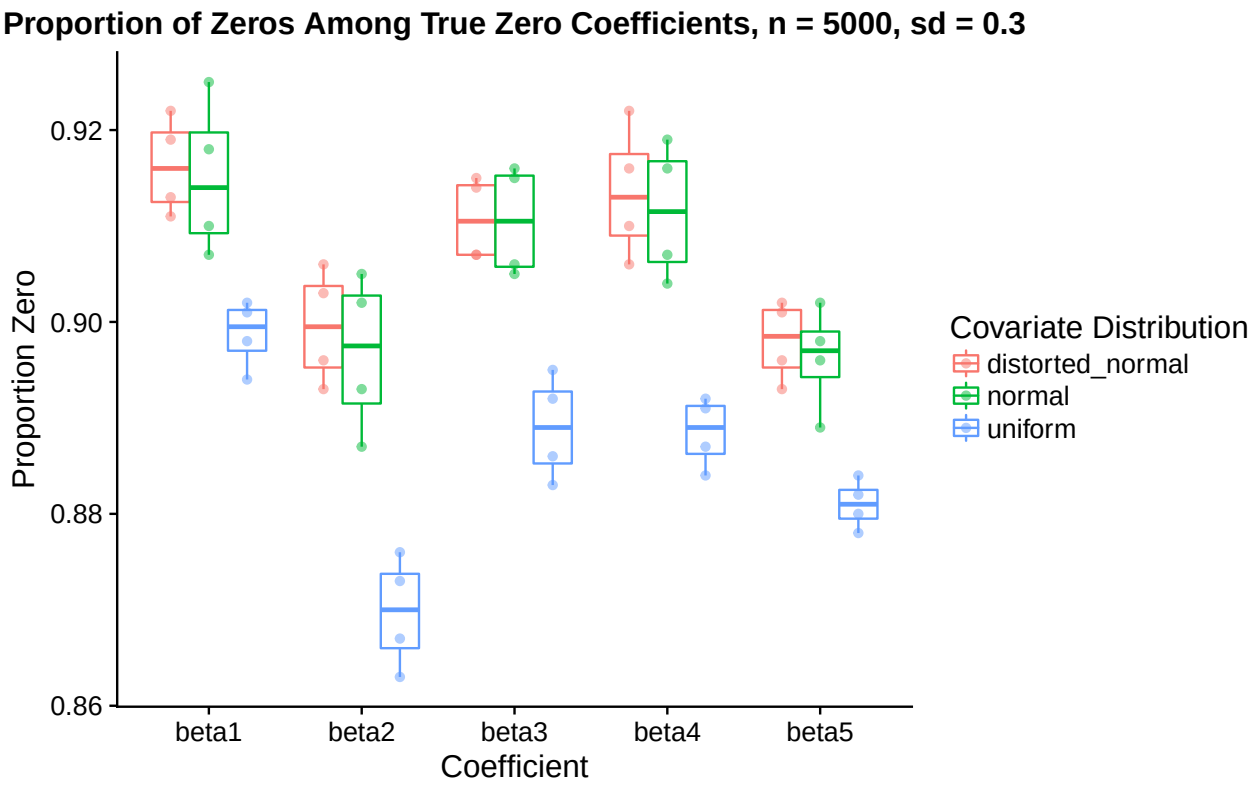

Figure 3: Plot of the proportion of zeros among truly zero coefficients, when $n = 5000$ and $\sigma = 0.3$. The colors represent different covariate distributions. Each boxplot contains coefficients from all different configurations of $(\rho_1, \rho_{12}, \rho_2)$.

	X Distribution	Coefficient	Mean Value
1	normal	beta6	0.373
2	uniform	beta6	0.388
3	distorted_normal	beta6	0.372
4	normal	beta7	0.343
5	uniform	beta7	0.338
6	distorted_normal	beta7	0.341
7	normal	beta8	0.344
8	uniform	beta8	0.340
9	distorted_normal	beta8	0.347
10	normal	beta9	0.366
11	uniform	beta9	0.382
12	distorted_normal	beta9	0.365

Table 1: Table of mean ratio of simulated coefficients divided by true coefficient value for the truly non-zero coefficients. This table corresponds to the data shown in Figure 5.

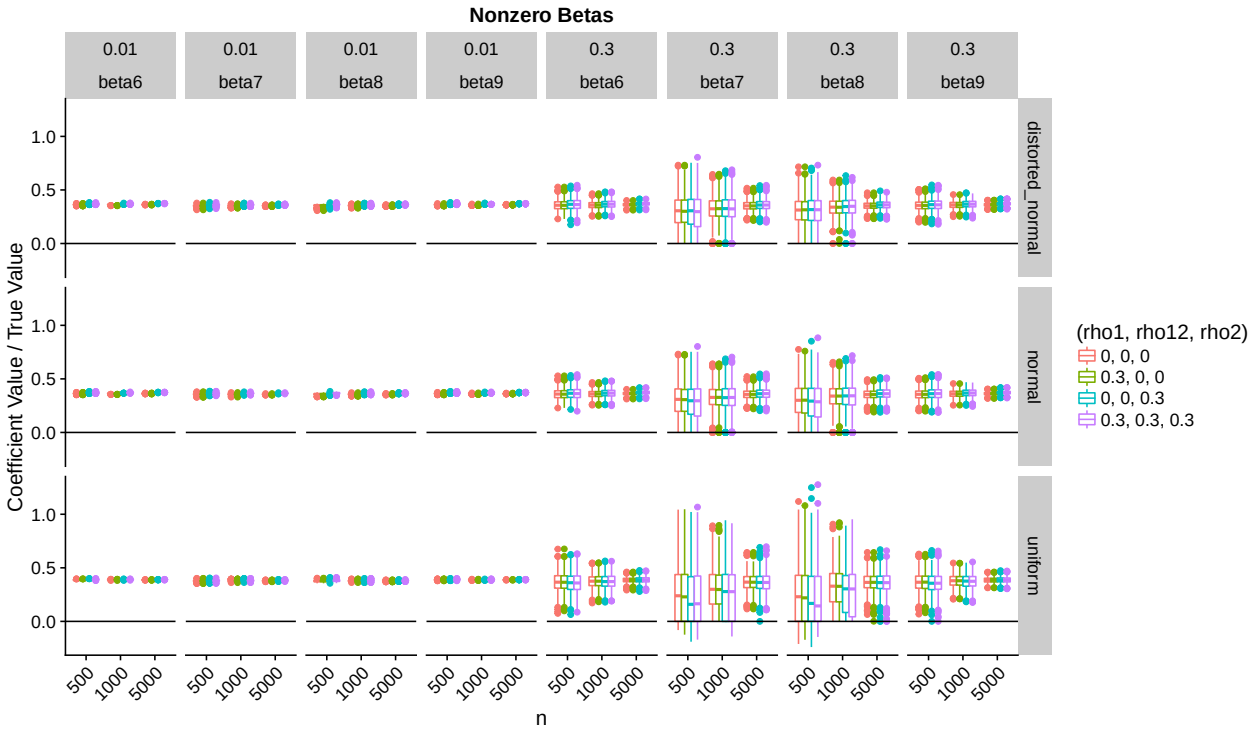

Figure 4: Plot of coefficient values for β that are truly nonzero. The 5 panels on the left show the case when $\sigma = 0.01$ and the 5 on the right show when $\sigma = 0.3$. The three vertical panels show the three different distributions of the covariates.The colors represent different settings of $(\rho_1, \rho_{12}, \rho_2)$.

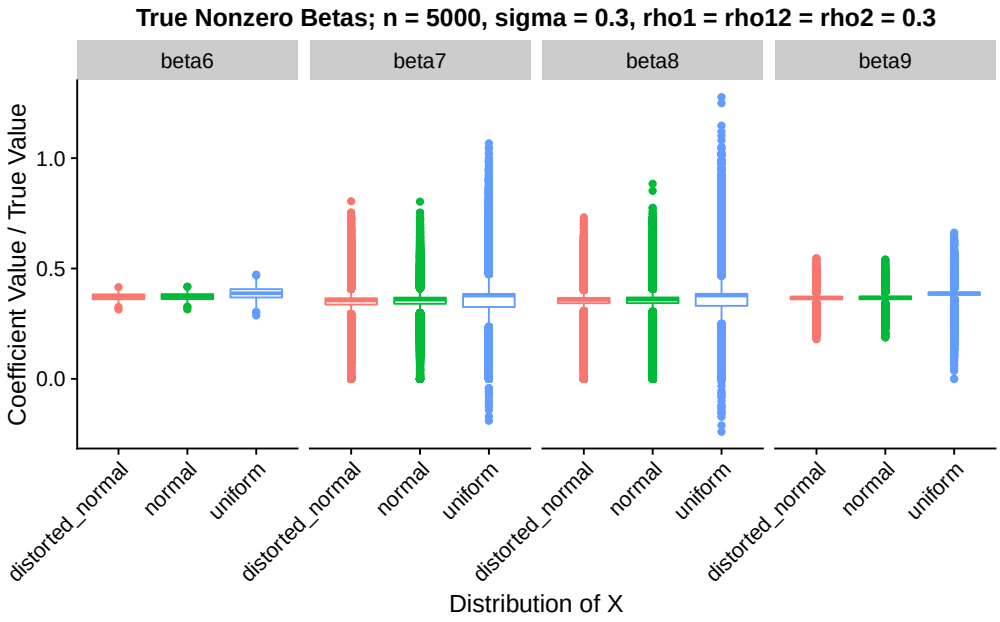

Figure 5: Plot of true nonzero betas for the case when $n = 5000, \sigma = 0.3, \rho_1 = \rho_{12} = \rho_2$. The different colors represent different covariate distributions.

4.2 Prediction Performance

Next, we compared the prediction performance on independent test data. The MSE from different models and data generated with small noise ($\sigma = 0.01$) are reported in Figure 6, and the MSE from different models and data generated with big noise ($\sigma = 0.3$) are reported in Figure 7. For the small errors in Figure 6, the adaptive lasso linear model tends to perform worse than kernel smoothing and using the true model, especially as sample size increases. In some cases, however, kernel smoothing produces predictions with high variance.

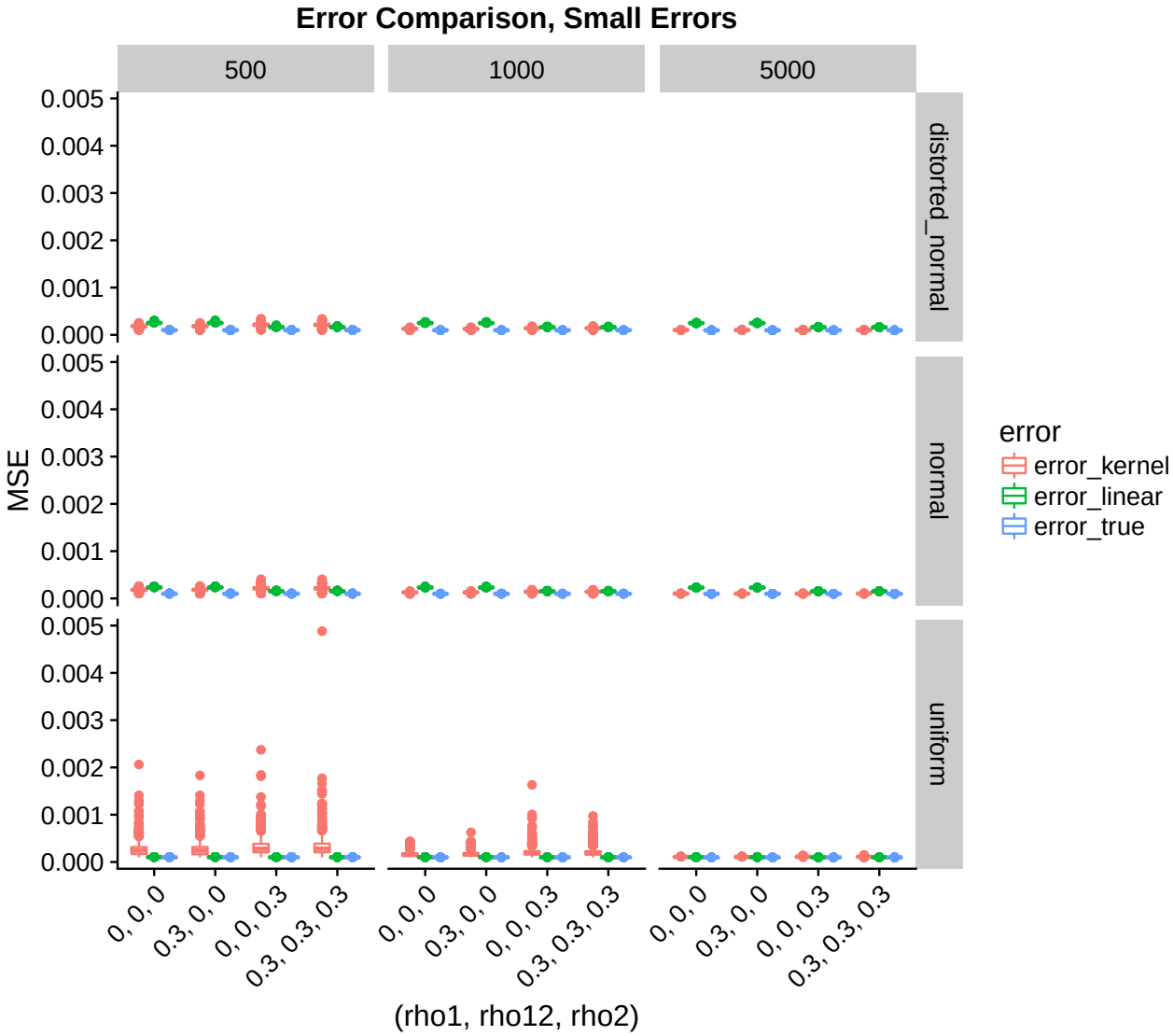

Figure 6: Plot of prediction errors when $\sigma = 0.01$. In red, prediction errors from kernel smoothing (nonparametric calibration); in green, prediction errors from adaptive lasso; in blue, prediction errors from the true model. The numbers on the horizontal axis represent different configurations of $(\rho_1, \rho_{12}, \rho_2)$, the three vertical facets represent different distributions of the covariates, and the three horizontal facets represent different sample sizes n.

For the large errors in Figure 7, the adaptive lasso linear model tends to perform better than the other two methods, with kernel smoothing tending to perform the worst. For another view of the

errors, see Figure 19.

We did not notice a big difference in the result for different correlations $\rho_1, \rho_2, \rho_{12}$.

Overall, the prediction performance from adaptive lasso is comparable to the other two methods, especially in the more realistic case with bigger noise.

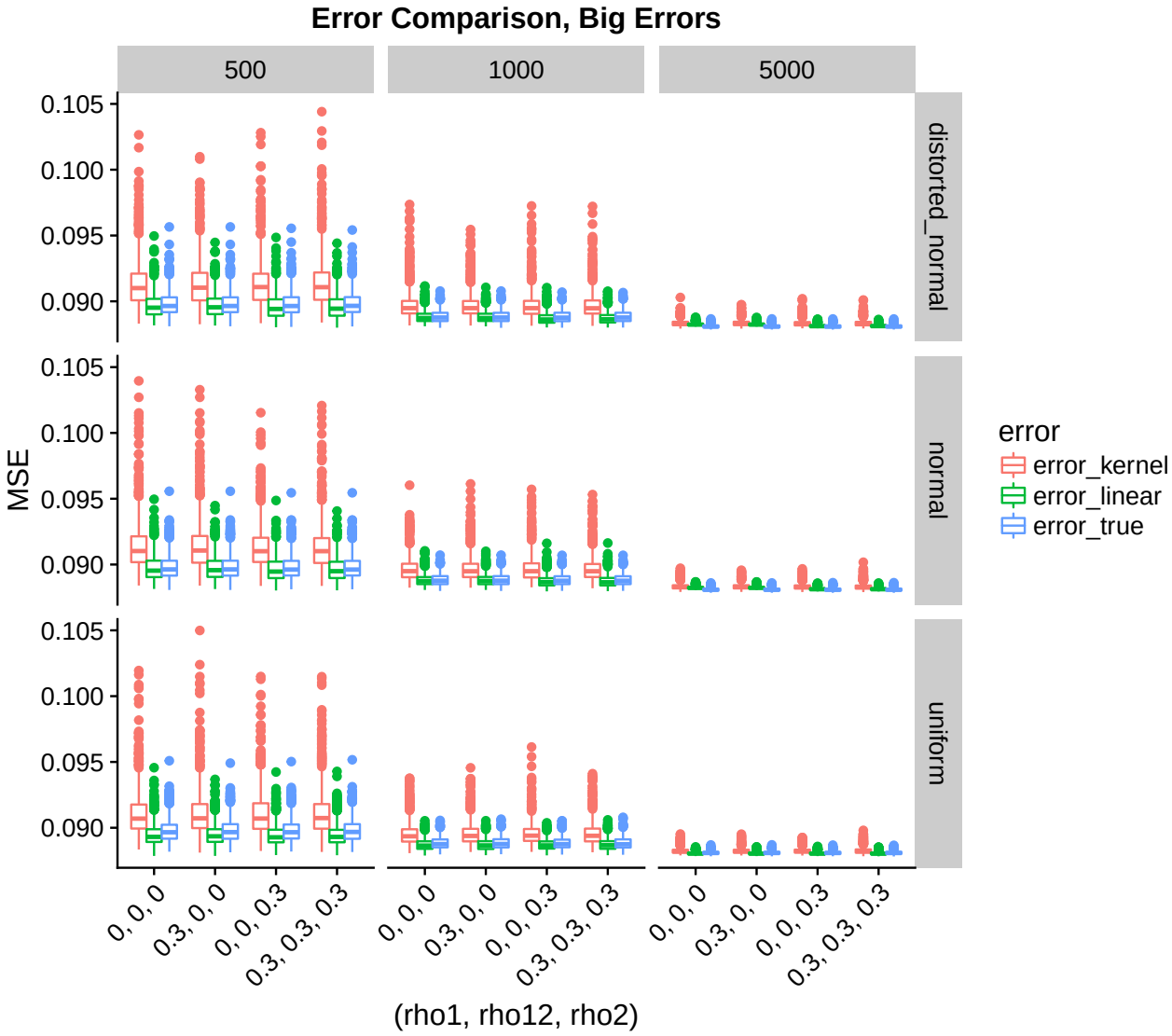

Figure 7: Plot of prediction errors when $\sigma = 0.3$. In red, prediction errors from kernel smoothing (nonparametric calibration); in green, prediction errors from adaptive lasso; in blue, prediction errors from the true model. The numbers on the horizontal axis represent different configurations of $(\rho_1, \rho_{12}, \rho_2)$, the three vertical facets represent different distributions of the covariates, and the three horizontal facets represent different sample sizes n.

4.3 Prediction Error Exploration

We wanted to understand why the different methods for prediction had results contrary to our intuition, especially when errors are large (Figure 7). We start by examining the model performance in the ideal scenario where we have very small random noise ($\sigma = 0.01$). As we can see from

Figure 6, the true model consistently does a good job with little noise, and the adaptive lasso model has a slightly bigger MSE than the true model. With predictors generated from multivariate normal distribution, the kernel smoothing models have similar MSE compared to the adaptive lasso models when sample size is small ($n = 500$), and managed to decrease the MSE as sample size increased to 1000 and 5000. We can see this through the plot of the prediction values as well (Figure 8, top row). The fitted value of true model and kernel smoothing model fit the true Y almost perfectly when error is small, while the adaptive lasso model with linear link fits well for the Ys around the mean value, but fail to capture the nonlinear trend in the tails. We can see this in the distribution of Y and fitted values also (Figure 9). The distribution of the fitted value from the true model seems very similar to the true distribution, the distribution of fitted value of adaptive lasso model is more disperse than the true distribution, and the kernel smoothing helps pull the values on the tails of the adaptive lasso fit towards the center, which makes it closer to the true distribution.

The results of model fit with data with big error ($\sigma = 0.3$) seems counter-intuitive at first glance. As we can see from Figure 7, the true model has a slightly bigger MSE than the adaptive lasso model and the kernel smoothing models have an even bigger MSE than the adaptive lasso model. Examining the fitted values from those models (Figure 8, bottom row; Figure 10), the fitted values of true model and adaptive model seem to have a much smaller range than the true values. This is because the probit function $\Phi(\cdot)$ is bounded by zero and one, so even if we estimate all the coefficients correctly, the fitted values would be bounded in $[\alpha, \alpha + 1]$, while the true Ys have a much wider range. Kernel smoothing model seems been able to extended the range of fitted value, but it has bigger bias for values around mean.

If we remove the random noise and plot the fitted values against the true conditional mean $\mathbb{E}(Y|\boldsymbol{X})$ (Figure 11), we see that the true model noise data still does a good job in capturing the conditional mean of the underlying data. The adaptive lasso model gives a good linear approximation of underlying data with some departure at the tails, while the kernel smoothing model has some curves which dose not reflect the conditional mean. .

Why does kernel smoothing help when error is small, but makes things worse when the noise is big? From the residual plot of adaptive lasso model with small error (Figure 12, top row), we can see there's a clear nonlinear trend of Y in $\boldsymbol{X}^T\boldsymbol{\beta}$ which is not captured by the adaptive lasso model with linear link. Adding a kernel smoothing using $\boldsymbol{X}^T\boldsymbol{\beta}$ as predictor calibrated the fitted value towards to the true Y. However, in the residual plot of adaptive lasso model with big error (Figure 12, bottom row), the nonlinear trend of Y in $\boldsymbol{X}^T\boldsymbol{\beta}$ is completely masked by the random noise. The added kernel smoothing using $\boldsymbol{X}^T\boldsymbol{\beta}$ as predictor is then misguided by the random noise. This explains the curly patterns in Figure 11.

5 Discussion

In theory, adaptive lasso can recover the true coefficients proportionally, thus it provide a consistent estimates of the support of the coefficients, even under link violation. In simulation, adaptive lasso successfully estimates the support of true coefficients, and seems robust when the elliptical distribution assumption is violated. However, the recovery rate the support of coefficients are lower when the elliptical distribution assumption was violated. When the noise is small, adaptive lasso has bigger MSE than the true model, and smoothing over the adaptive lasso fit helps lower the MSE by capturing the nonlinear trend of Y in the fitted value from adaptive lasso. When the noise is big, the predictions are poor for all three models. Even though the true model and adaptive lasso approximate the conditional mean well, they both have big MSE due to the large random noise. The true model suffers from constrained prediction space and adaptive lasso works slightly

better. Kernel smoothing on top of adaptive lasso fit doesn't help in the case of large error because the nonlinear relationship between the outcome and adaptive lasso fit is masked by the noise. Additionally, the correlation structure between the predictors does not seem to matter much, either in estimating the coefficients or with regards to prediction error.

Our study was limited by the number of simulation scenarios we could run, and thus may not generalize to any distribution for the predictors $\boldsymbol{X}$.

Unfortunately, we were unable to find a simulation condition that clearly showed that proportional recovery of the true coefficients $\boldsymbol{\beta}_0$ is not possible. In Li and Duan Remark 6.4 [2], the authors state "empirical study by Brillinger and others suggests that quite often the bias may be negligible even for a moderate violation of the design condition." Also, in Section 6.4, they used $\boldsymbol{X_i}$ from a 2D uniform grid, and designed an antisymmetric link function to break the symmetry condition $\mathbb{E}[\boldsymbol{\beta}^T\boldsymbol{x}|\boldsymbol{\beta}_0^T\boldsymbol{x}]$. Our link function was given to be the probit function, which is symmetric, which may explain why it is hard for us to find a case where we could not proportionally recover the coefficients.

6 Code

Code to reproduce the simulation results and all the tables and figures is available online at https://github.com/shiandy/bst235project.

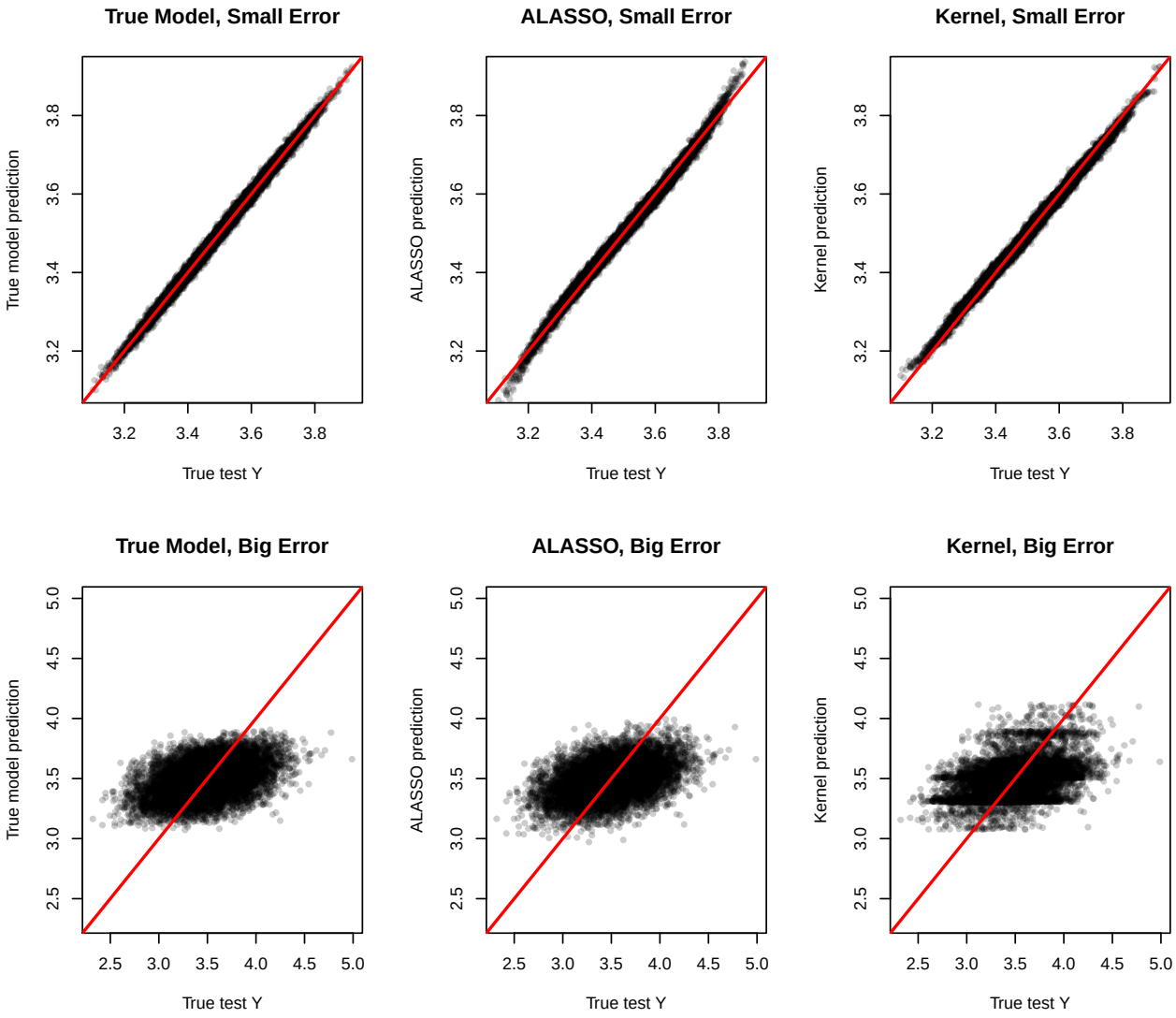

Figure 8: This figure shows plots of predicted values against the true value in the test dataset. The top row is for the case $\sigma = 0.01$, and bottom row for $\sigma = 0.3$. The left two plots show the predictions from the true model on the y-axis, the middle two plots show the predictions from adaptive lasso on the y-axis, and the right two plots show predictions from kernel smoothing on the y-axis.

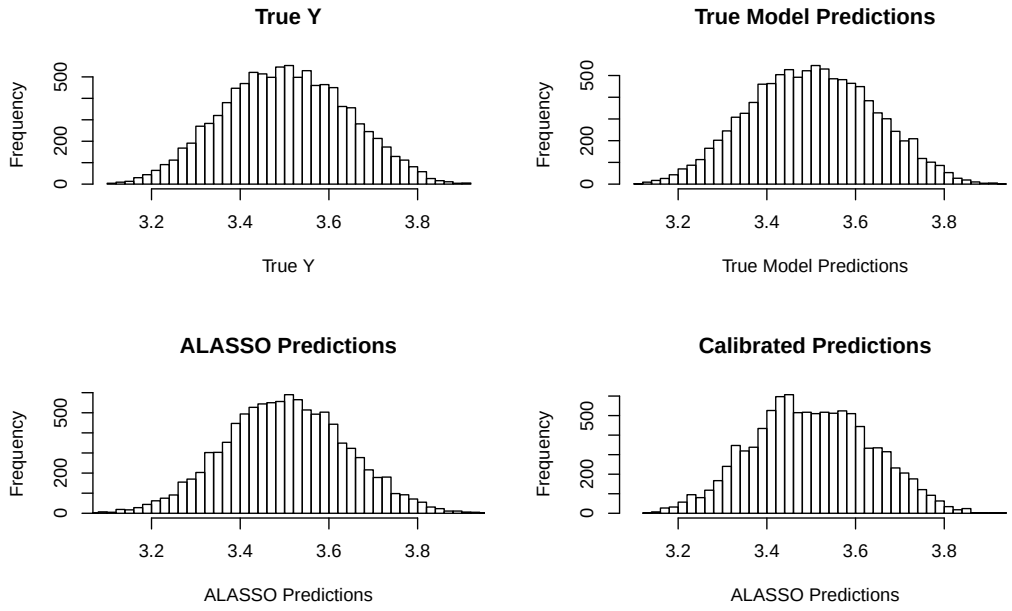

Figure 9: Histograms of the true outcome (top left), predictions from the true model (top right), predictions from adaptive lasso (bottom left) and predictions from nonparametric calibration (bottom right) for the case $\sigma = 0.01$.

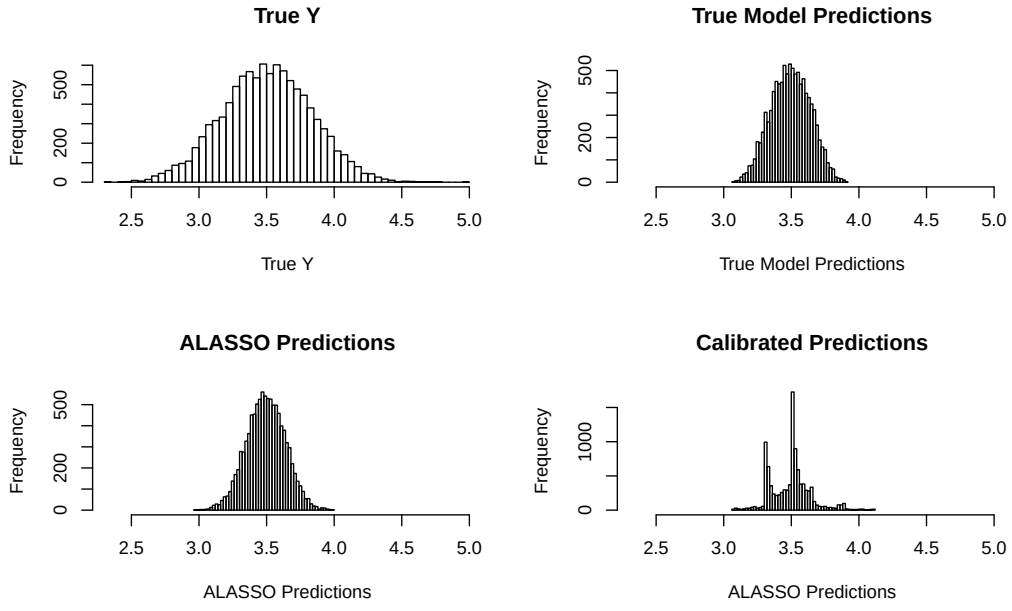

Figure 10: Histograms of the true outcome (top left), predictions from the true model (top right), predictions from adaptive lasso (bottom left) and predictions from nonparametric calibration (bottom right) for the case $\sigma = 0.3$.

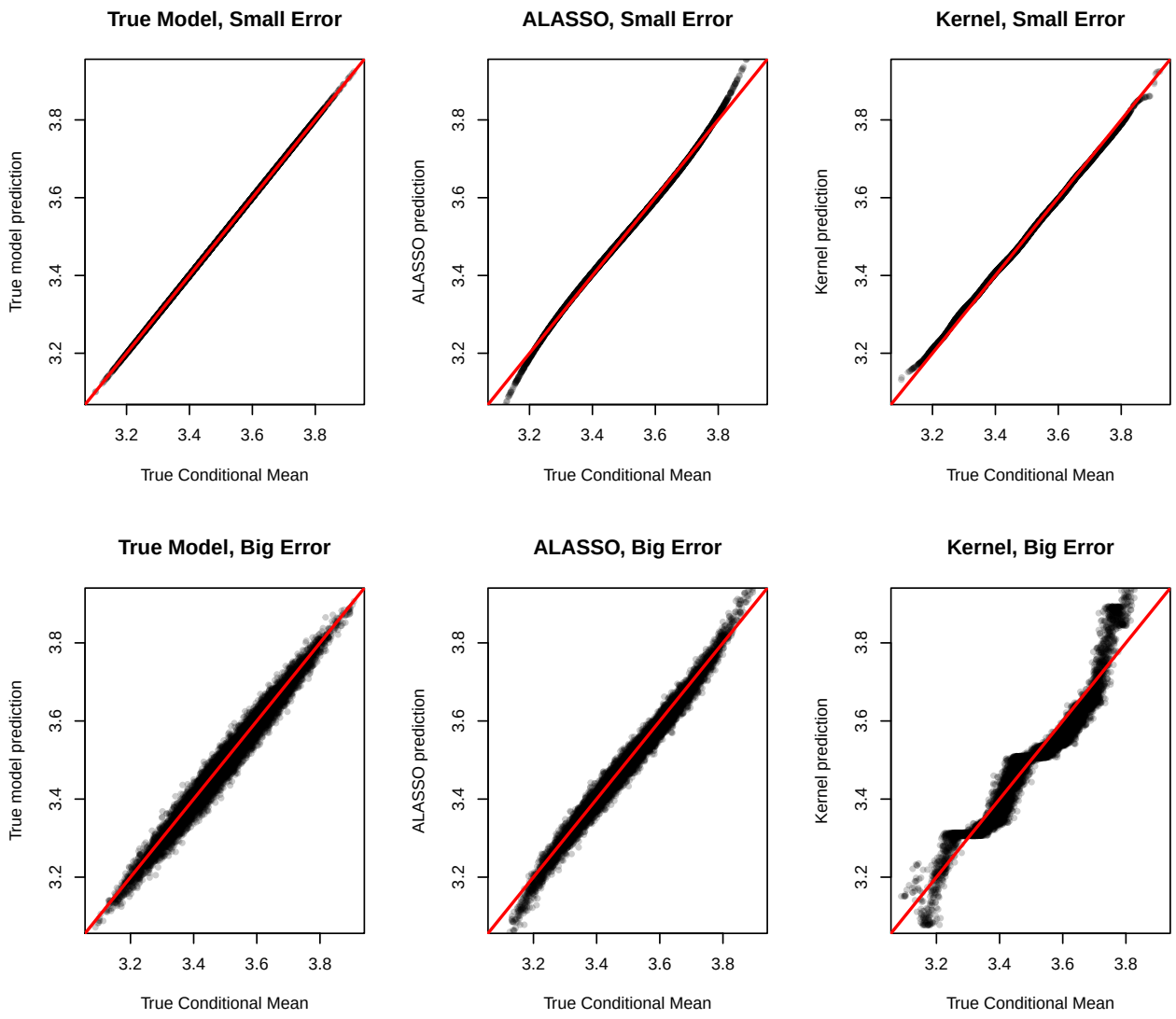

Figure 11: This figure shows plots of predicted values against the true conditional mean $\alpha_0 + \Phi(\boldsymbol{\beta}_0^T \boldsymbol{X}_i)$ in the test dataset. The top row is for the case $\sigma = 0.01$, and bottom row for $\sigma = 0.3$. The left two plots show the predictions from the true model on the y-axis, the middle two plots show the predictions from adaptive lasso on the y-axis, and the right two plots show predictions from kernel smoothing on the y-axis.

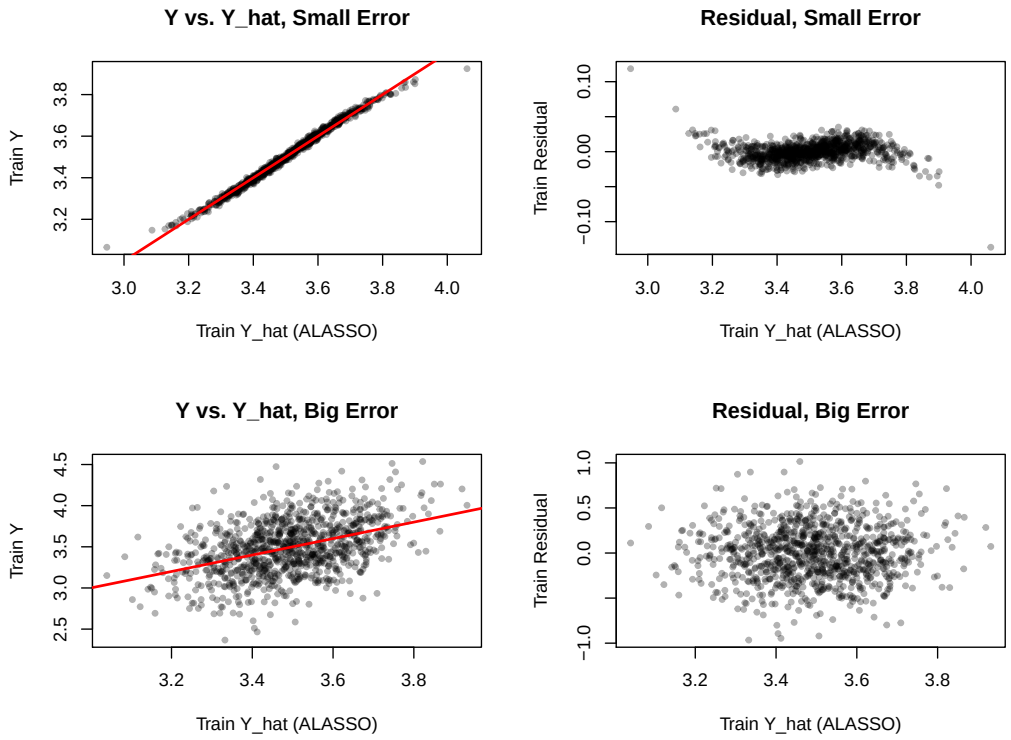

Figure 12: The left two plots show the relationship between the true outcome in the training data (y-axis) against the prediction from adaptive lasso (x-axis), and the right two plots show the residual (y-axis) against the prediction from adaptive lasso (x-axis). The top row shows the case when $\sigma = 0.01$, and the bottom row shows when $\sigma = 0.3$.

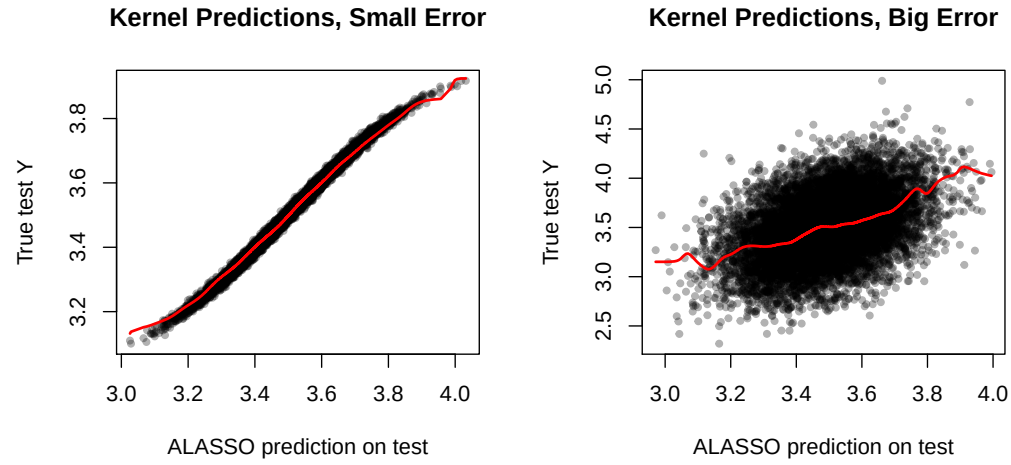

Figure 13: These two plots show the true outcome on the test dataset plotted against the adaptive lasso prediction. The red line indicates the kernel smoothing predictions. The left plot is for $\sigma = 0.01$ and the right is for $\sigma = 0.3$.

References

[1] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[2] Ker-Chau Li and Naihua Duan. Regression analysis under link violation. *The Annals of Statistics*, pages 1009–1052, 1989.

[3] R. Fletcher. *Practical methods of optimization*. Wiley, 1986.

[4] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[5] David Ruppert, Simon J Sheather, and Matthew P Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.

A Supplementary Figures

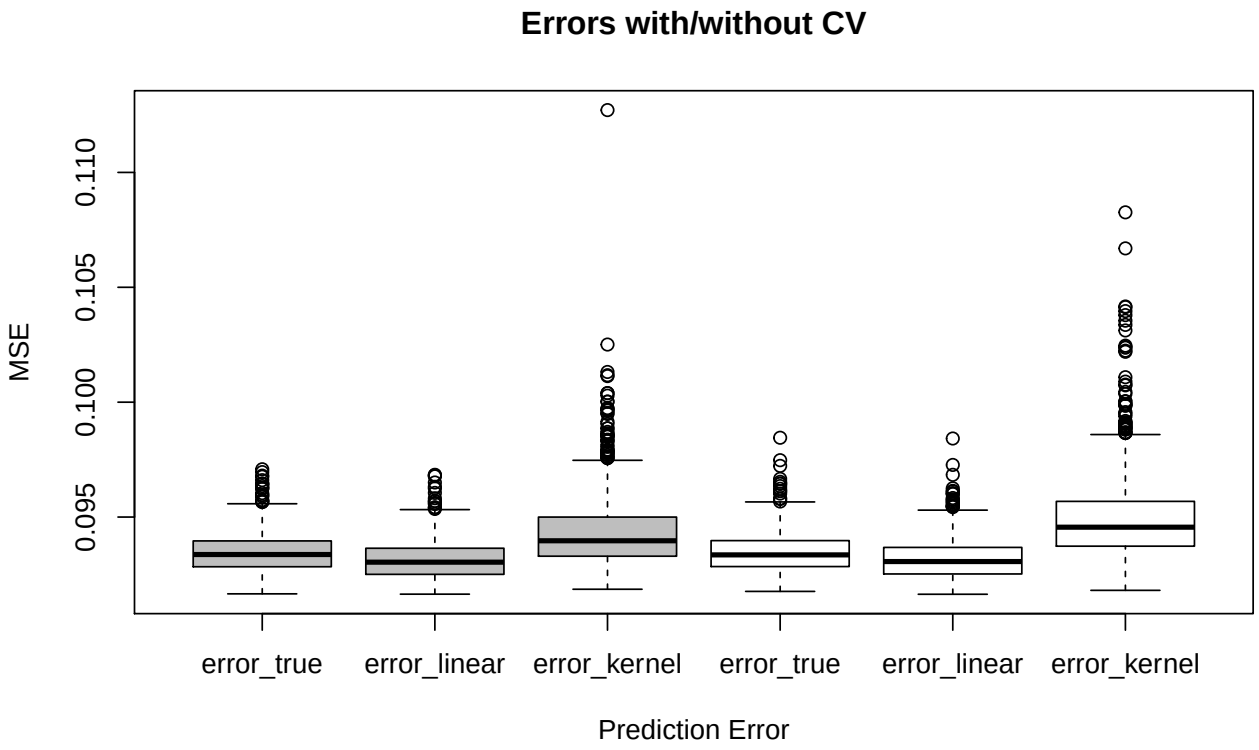

Figure 14: Plot of MSE using $n = 500, \sigma = 0.3, \rho_1 = \rho_{12} = \rho_2 = 0.3$, and predictors distributed multivariate normal. The boxplots in grey selected the optimal bandwidth using cross-validation, and those in white did not use cross-validation. We find no significant difference between selecting the bandwidth with and without cross-validation.

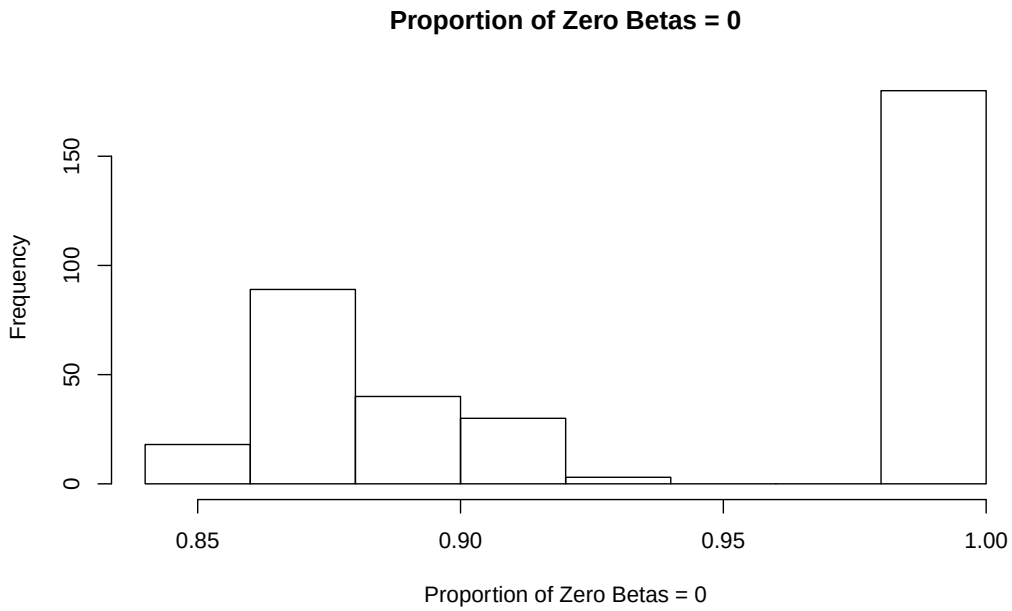

Figure 15: Histogram showing the frequency across different simulation conditions where truly zero coefficients were set to 0.

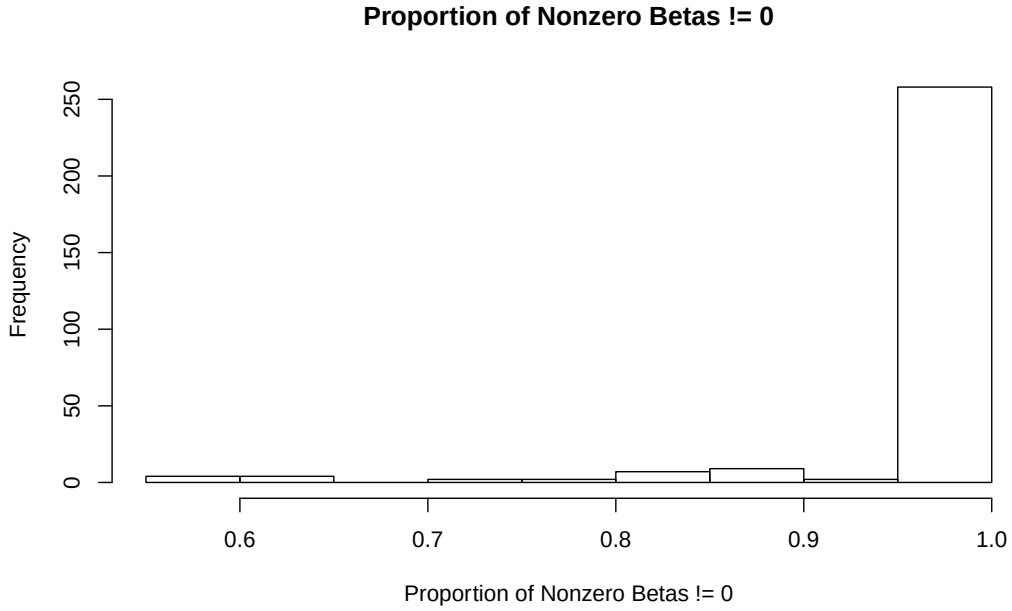

Figure 16: Histogram showing the frequency across different simulation conditions where truly nonzero coefficients were set to nonzero values.

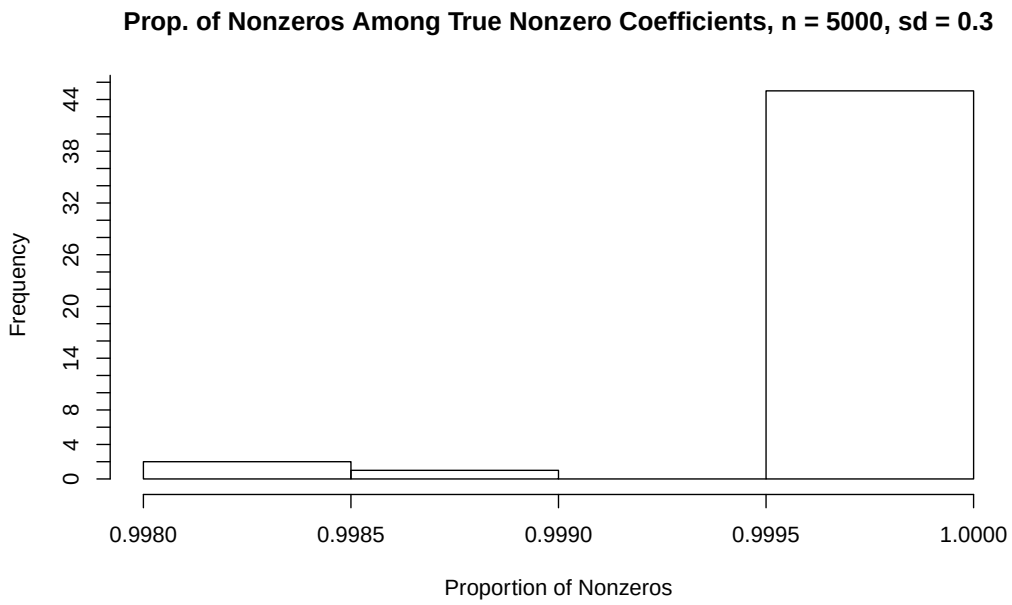

Figure 17: Histogram showing the frequency across different simulation conditions when $n = 5000$ and $\sigma = 0.3$ where truly nonzero coefficients were set to nonzero values. Note the lower bound at 0.998, indicating almost all cases, all the coefficients are set to nonzero values.

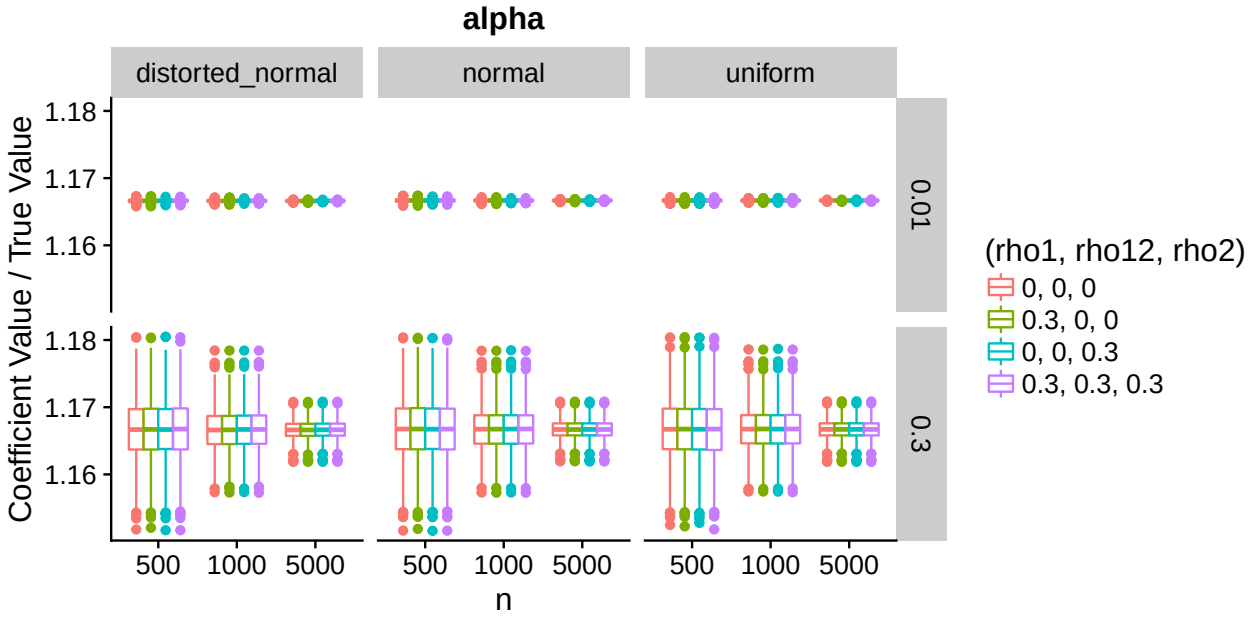

Figure 18: Plot of the ratio of coefficient value divided by true value for α.

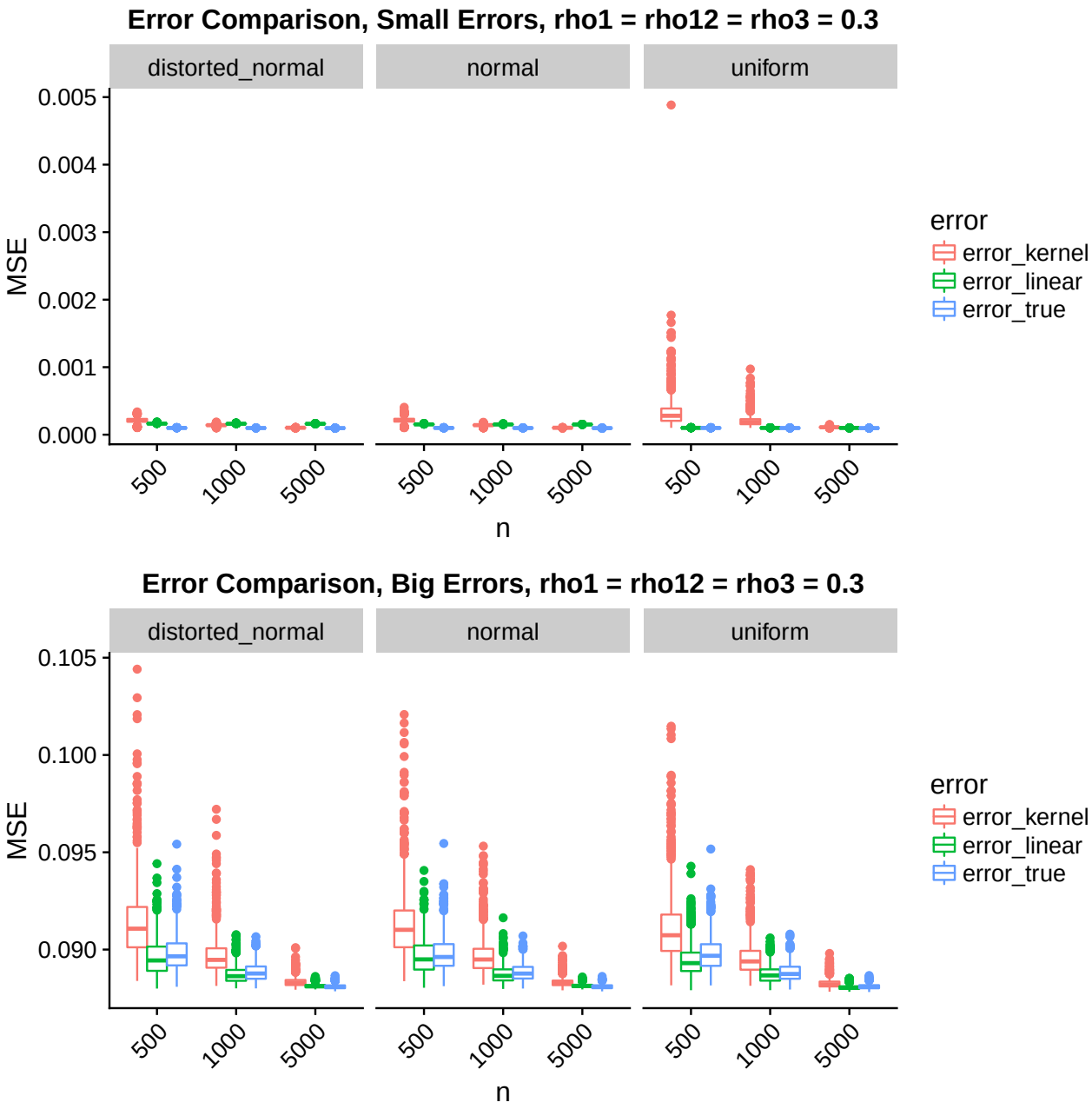

Figure 19: Plot of prediction errors when $\rho_1 = \rho_{12} = \rho_2 = 0.3$. The top plot shows the case $\sigma = 0.01$ and the bottom shows the case $\sigma = 0.3$. In red, prediction errors from kernel smoothing (nonparametric calibration); in green, prediction errors from adaptive lasso; in blue, prediction errors from the true model. The numbers on the horizontal axis represent different sample sizes n, and the three horizontal facets represent different distributions of the predictors.