Data Simulation and Evaluation of Variable Selection Methods using Prediction Performance and Estimation Metrics

Gaynanova, Irina
Texas A & M University
irinag@stat.tamu.edu

Kang, Zengxiaoran
Texas A & M University
zk2487@tamu.edu

Adams, Benjamin
Texas A & M University
benadams@tamu.edu

Hernandez, Mayra
Texas A & M University
mhdz@tamu.edu

Lawson, Jada
Texas A & M University
jdlawson@tamu.edu

August 15th, 2021

1 Introduction

We ran a simulation study with the goal of assessing the performance of four regression methods: forward selection, ridge regression, lasso regression, and principal component regression. We chose to assess the quality of these methods using prediction, estimation, and variable selection metrics. We generated the data under the four standard linear regression assumptions, as well as in four additional situations in which each assumption was violated. We also generated the data under a variety of parameters in order to compare how the methods perform in different situations. We focused on 3 main cases for analysis: 1) The difference between the standard and violated assumptions when the sample size and number of covariates are held constant. 2) The difference between 4 combinations of sample size and number of covariates when the other parameters are held constant and the assumptions are not violated. 3) The different combinations of pairwise covariate correlation and size of coefficients when the sample size and number of covariates was held constant and the assumptions were not violated.

2 Methodology

2.1 Review of ordinary linear regression

In linear regression[8], we assume that we observe response $Y \in \mathbb{R}^n$ and corresponding matrix of covariates $X \in \mathbb{R}^{n \times p}$, which follow linear model

$$Y = \beta_0 + X\beta + \varepsilon, \tag{1}$$

where $\beta_0 \in \mathbb{R}$ is an intercept, $\beta \in \mathbb{R}^p$ is a vector of coefficients with elements β_k, $k = 1, \dots p$, and $\varepsilon \in \mathbb{R}^n$ is a vector of errors, with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ independently for all $i = 1, \dots, n$.

For simplicity, we can drop β_0 if Y is centered and X has centered columns. More specifically, let $\overline{Y} = \frac{1}{n}\sum_{i=1}^n y_i \in \mathbb{R}$ be the sample mean of Y, and let $\overline{x} = \frac{1}{n}\sum_{i=1}^n x_i \in \mathbb{R}^p$ be the vector of columns sample means of X. Let new Y and X be the centered original Y and X. Then we can find $\widehat{\beta}$ from centered Y and X (with no intercept), and then find original $\widehat{\beta}_0$ from $\widehat{\beta}$ as

$$\widehat{\beta}_0 = \overline{Y} - \overline{x}^\top \widehat{\beta}. \tag{2}$$

A standard approach for estimation of β is to solve ordinary least squares problem (for centered Y and X as described above)

$$\widehat{\beta}_{LS} = \arg\min_\beta \|Y - X\beta\|_2^2,$$

with the solution having closed form

$$\widehat{\beta}_{LS} = (X^\top X)^{-1} X^\top Y.$$

Given $\widehat{\beta}_{LS}$ and $\widehat{\beta}_0$ constructed from (2), for a new sample with covariate vector $x \in \mathbb{R}^p$, the predicted response is constructed as

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_{LS}^\top x. \tag{3}$$

2.2 Performance metrics

Prediction performance metrics.
To have good prediction performance, a method must produce estimated responses that are close to the observed responses. Giving and denoting a set of training data with feature matrix X_{train} and response matrix Y_{train}, we can measure the performance of the estimated output matrix $\widehat{Y}_{train}$ with formula provided below[11]:

$$MSE^{\text{train}} = \frac{1}{n_1}\sum_{i=1}^{n_1}(y_i^{train} - \widehat{y_i^{train}})^2,$$

where n_1 is the number of observations in the training data.

Similarly, giving and denoting a set of testing data with feature matrix X_{test} and response matrix Y_{test}, we can measure the performance of the estimated output matrix $\widehat{Y}_{test}$ with

formula provided below:

$$MSE^{\text{test}} = \frac{1}{n_2}\sum_{i=1}^{n_2}(y_i^{test} - \widehat{y_i^{test}})^2,$$

where n_2 is the number of observations in the testing data.

After a prediction method is used to predict values of $\hat{y}_i$, maximal distance gives the largest distance between the observed and predicted value of a response.

$$\|Y^{test} - \widehat{Y}^{test}\|_\infty = max|y_i^{test} - \widehat{y_i^{test}}|.$$

This metric provides information on the worst instance of prediction for a method.

Another performance measure for a prediction method is average absolute value difference. This metric gives the average distance between the predicted and observed value of responses.

$$\|Y^{test} - \widehat{Y}^{test}\|_1 = \frac{1}{n}\sum_{i=1}^{n}|y_i^{test} - \widehat{y_i^{test}}|.$$

Using both maximal distance and average absolute value distance can provide important insight because it is possible for different sets of data to have the same value for one metric but very different values for the other metric.

Estimation metrics.
A good estimation performance means to have unbiased, consistent, and sufficient responding output. The best way to think about Squared Euclidean Norm[2] is by imagining two points plotted in $\mathbb{R}^2$ which represent β_i and $\widehat{\beta}_i$. One way to measure how well $\widehat{\beta}$ is estimating β is by calculating the shortest distance between each of the corresponding β_i and $\widehat{\beta}_i$, squaring the difference and adding them. This can be represented with the following equation:

$$||\beta - \widehat{\beta}||_2^2 = \sum_{i=1}^{p}\left(\beta_i - \widehat{\beta}_i\right)^2$$

where β is the vector containing all true β and $\widehat{\beta}$ is the vector containing all estimated $\widehat{\beta}$.

L1 Norm[3], other known as Manhattan Distance, is the sum of the absolute difference of the components of the vectors previously mentioned. The following equation is a representation of this:

$$||\beta - \widehat{\beta}||_1 = \sum_{i=1}^{p}\left|\beta_i - \widehat{\beta}_i\right|$$

where β is the vector containing all true β and $\widehat{\beta}$ is the vector containing all estimated $\widehat{\beta}$. An analyst will know that $\widehat{\beta}$ is a good estimate of β when the distance is short or the

difference is small between $\widehat{\beta}$ and β.

Variable Selection
The True Positive Rate (TPR)[6] is a measure that takes the values of the true model and intersects them with the values of the estimated model and divides that by the true model. Its formula is given by:

$$TPR = \frac{\text{card}(A \cap B)}{\text{card}(A)}$$

Here $A = \{j : \beta_j \neq 0\}$, $B = \{j : \widehat{\beta}_j \neq 0\}$

The True Negative Rate (TNR)[6] is a measure that takes the values of the true zero betas and intersects them with the values of the estimated model of zero betas and divides it by the true zero betas. Its formula is given by:

$$TNR = \frac{\text{card}(A^C \cap B^C)}{\text{card}(A^C)}$$

Here $A = \{j : \beta_j \neq 0\}$, $B = \{j : \widehat{\beta}_j \neq 0\}$

2.3 Methods for comparison

2.3.1 Forward Step-wise Selection

In forward step-wise selection[4], you start with the null model. Feature variables are then examined one at a time based on a stopping rule that the analyst will determine. Then look at all the variables, find the best one to see if it can go into the model, if it can, put it in. This process is repeated until all the variables are in the model or no variables pass the stopping rule. There are several stopping rules such as, p-value that acts as a hurdle the variables must overcome to be allowed into the model. Although, there are advantages to using forward selection such as being computationally efficient, there are several flaws as well. For example, as feature variables are added, they may overlap and interact in how they explain the variance of the dependent variable. Forward selection is not flexible meaning that it is possible for a variable that entered early to fail the stopping rule later as other variables are added, but because of the rules of forward selection, it is stuck in the model.

2.3.2 Ridge Regression

Ridge regression[5] is very similar to least squares but the coefficients are estimated differently. Ridge regression estimates coefficients by making the residual sum of squares (RSS) value small and the shrinkage penalty small. You have to select a tuning parameter that will serve the purpose of controlling the RSS value and the shrinking penalty value. The tuning parameter is a value between zero and infinity. As it gets closer to infinity the shrinkage penalty has a greater impact and the coefficients if ridge regression will get closer to zero.

When the tuning parameter is zero the shrinkage penalty has no impact. Ridge regression produces a different set of coefficients depending on the tuning parameter. As the tuning parameter increases the variance decreases but the bias increases so picking a good tuning parameter is very important. This is done through cross validation mainly but there are other methods that make it possible. This aspect of bias variance trade off is what makes ridge regression better than least squares in certain situations.

2.3.3 Lasso Regression

Lasso regression[10] is a regression method that works in a similar way to least squares regression, but rather than minimizing the residual sum of squares, lasso regression minimizes the residual sum of squares plus a shrinkage penalty. The shrinkage penalty is defined as a tuning parameter multiplied by the sum of the absolute values of the coefficients. The tuning parameter can be chosen by different methods, but is usually done by cross validation. Increasing the tuning parameter too much typically causes the squared bias and MSE to increase and the variance to decrease. Unlike in ridge regression, the lasso regression shrinkage penalty is able to cause some coefficient estimates to be equal to zero, so lasso performs variable selection. Performing variable selection gives lasso regression the advantage of better model interpretability especially when there are many predictors to be considered. When all predictors in a model have substantial coefficients, lasso regression performs very similarly to ridge regression but typically has higher variance and MSE. However, when only a small subset of the predictors are substantial, lasso regression will generally strongly outperform ridge regression in terms of bias and MSE.

2.3.4 Principal Component Regression

To introduce Principal Component Regression (PCR)[9], we first need to understand the concept of the Principle Component Analysis (PCA). The PCA in a simplistic overview is a technique that reduces the dimension of the data matrix while still obtains valuable information from the original set. The PCA selects a ranking set of orthogonal principal component vectors where observations have the largest variance possible if projecting onto those directions. Then each observation is projected onto the principal component vectors and transformed into a principal component coordinate which can be represented in a linear combination of original covariants as well. With this process, the data matrix can obtain the greatest variability when reducing the data matrix to the desired dimension. Typically, the PCA is explained via eigendecomposition of the covariance matrix and the Singular Value Decomposition (SVD) technique is often performed to obtain the PCA. The PCR in short is a prediction method that applies Ordinary Least Square Regression (OLS) onto the PCA matrix. The PCR constructs the linear regression model by using the principal components as the predictors and fits the model using least squares gradient descent. The PCR method can mitigate overfitting quite well; Though, similar to OLS, the model suffers from high variance and low bias as more principal components are introduced.

2.4 Data generation

2.4.1 The assumption of linear model holds

In our simulations, we utilized many different parameters when it came to generating our data. The main pieces of data that is are most relevant are the generation of β, X, ϵ, and Y[1]. In order to generate β, we first need to create a $p \times 1$ vector β which will be a set of true parameter values. Moreover, there are three changing variables that we control: the sparsity level s, the maximum size of large coefficients Θ, and the maximum size of small coefficients θ. s is a non-negative integer in the range of $[0, p]$, and it determines the number of non-zero index in β. The small coefficient will be in the range of $(0, \theta)$ and the large coefficient will be in the range of (θ, Θ). When it comes to generating the covariate data, X, we used the correlation level, ρ. ρ is a scalar that can take any value in the range $[-1, 1]$. ρ represents the correlation between two covariates. A value of 0 means that covariates are not correlated, a value of 1 means that covariates are perfectly positively correlated, and a value of -1 means that covariates are perfectly negatively correlated.

In our implementation, all covariates have have the exact same pairwise correlations which means that X is then generated as an n-by-p-dimensional matrix from a multivariate normal distribution with all means as 0 and the covaraince matrix created from ρ. Furthermore, generating epsilon, ϵ is determined by three parameters which are the sample size n, the mean μ, and standard deviation σ. It is important to note that we will be using a function which generates a vector of normally distributed random numbers and that μ will be 0 for every generation of ϵ.[8] The parameter σ represents the noise for each individual X found in our model. The larger the standard deviation, the larger ϵ will be therefore resulting in a worse prediction value for Y while a σ of 0 means that there is no noise. Lastly, Y is an n-dimensional vector that is generated by β, X, β_0, and ϵ. The value for β_0, which is the value of Y when X and ϵ are equal to zero, will not be substantly large so that it does not have an overwhelming impact on Y. Additionally, the sample size for our train data will be bigger than the sample size for our test data but both will be large enough so we can get adequate information from the data to generate Y. Note that the values for the parameters n, p, σ, β_0, β are 50 or 100, 10 or 50, 1, 0, will be same as p, respectfully. For our other parameters which are s, ρ, large size, and small size, the values we will be using are 10, 0 or 0.8, 2 or 4, 0.5 or 1, respectfully.

2.4.2 The assumptions of linear model are violated

We also generated the data under the situation in which each of the four assumptions of linear regression were broken[7]. When hen the linearity assumption was violated we generated Y from a quadratic model rather than a linear one. When the normal error assumption was violated we generated ϵ from a t-distribution with 3 degrees of freedom. When the identical error assumption was violated we generated a vector of σ values from a uniform distribution and then generated ϵ from that unequal σ vector. When the independent error assumption was violated we created a new vector of shifted errors and added that vector to the original ϵ vector to introduce dependence. We then adjusted ϵ to keep the variance intact.

3 Results

In our controlled settings, we conducted a set of data simulations and observed the responses when adjusting varied factors n, p, rho, and signal strength. There are a few universal truth that match our hypothetical expectation and have constant performance as illustrated in figures and tables. For the Prediction Metrics, MSE test always have more error than MSE train, and Maximal Distance Error has the largest error among all four metrics except for special case n=50 p=50; For the Estimation Metrics, L1 norm always have more error than L2 norm; For the Variable Selection Metrics, TNR is 0 and TPR is 1 whenever s is equaled to p.

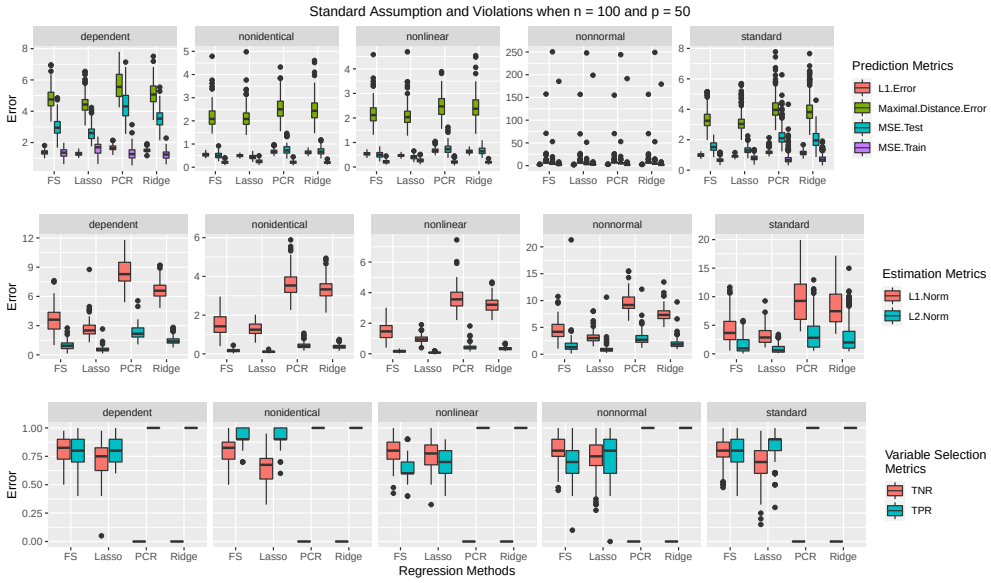

Figure 1: Figures comparing performance between standard and four violated cases .

status	Metric	FS	Lasso	PCR	Ridge
dependent	MSE.Test	3.04 (0.07)	2.64 (0.05)	4.41 (0.1)	3.55 (0.06)
	MSE.Train	1.33 (0.03)	1.63 (0.04)	1.31 (0.04)	1.23 (0.03)
nonidentical	MSE.Test	0.51 (0.01)	0.44 (0.01)	0.76 (0.02)	0.69 (0.02)
	MSE.Train	0.21 (0.01)	0.25 (0.01)	0.22 (0.01)	0.2 (0)
nonlinear	MSE.Test	0.5 (0.01)	0.41 (0.01)	0.75 (0.02)	0.67 (0.02)
	MSE.Train	0.22 (0.01)	0.27 (0.01)	0.21 (0.01)	0.2 (0)
nonnormal	MSE.Test	7.38 (2.51)	6.62 (2.49)	8.68 (2.44)	7.7 (2.49)
	MSE.Train	3.82 (1.84)	4.45 (1.97)	3.99 (1.9)	3.74 (1.78)
standard	MSE.Test	1.55 (0.03)	1.33 (0.02)	2.27 (0.05)	2.09 (0.05)
	MSE.Train	0.65 (0.01)	0.78 (0.02)	0.66 (0.02)	0.62 (0.01)

Table 1: Measurement of MSE Test and Train between standard and four violated cases.

Table 1 and Figure 1 compares the results of a standard case and four violated assumption cases when other settings remain the same. Accordingly, we conclude that all four methods are not effective when applying to non-normal and dependent violations with the non-normal violation in particular results in big outliers; Lasso has the best overall performance following Forward Selection to be the second-best.

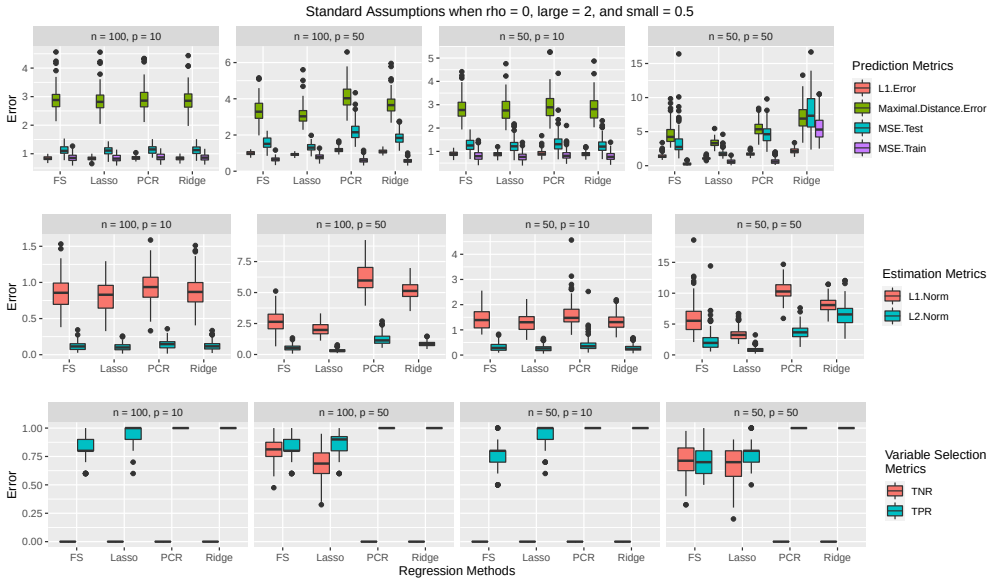

Figure 2: Figures comparing performance when n & p are changing.

n	p	rho	FS_TNR	FS_TPR	Lasso_TNR	Lasso_TPR
100	10	0	0 (0)	0.87 (0.01)	0 (0)	0.97 (0.01)
	10	0.8	0 (0)	0.76 (0.01)	0 (0)	0.93 (0.01)
	50	0	0.8 (0.01)	0.87 (0.01)	0.67 (0.01)	0.9 (0.01)
	50	0.8	0.79 (0.01)	0.76 (0.01)	0.71 (0.01)	0.79 (0.01)
50	10	0	0 (0)	0.8 (0.01)	0 (0)	0.95 (0.01)
	10	0.8	0 (0)	0.67 (0.01)	0 (0)	0.88 (0.02)
	50	0	0.7 (0.01)	0.8 (0.01)	0.66 (0.02)	0.82 (0.01)
	50	0.8	0.7 (0.02)	0.73 (0.01)	0.7 (0.01)	0.76 (0.01)

Table 2: Measurement of TNR and TPR among four methods.

Next, we look at results when n & p are changing but everything else stays the same. We derive two findings from observing Figure 2: 1) larger n smaller error 2) increasing p with same size n results in larger error. From Table 2, we analyze the effect of rho on the Variable Selection Metrics and conclude that in general increasing rho makes TPR and TNR have worse results.

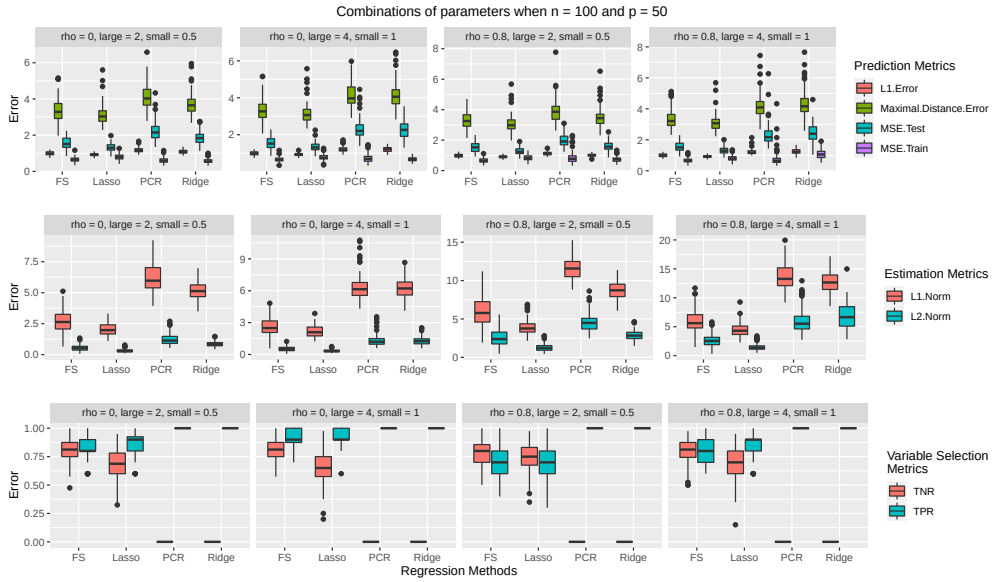

Figure 3: Figures comparing performance when rho, small & large are changing.

n	p	Metric	large	small	FS	Lasso	PCR	Ridge
100	10	L1.Norm	2	0.5	0.87 (0.02)	0.82 (0.02)	0.94 (0.02)	0.89 (0.02)
	10	L1.Norm	4	1	0.85 (0.02)	0.82 (0.02)	1.24 (0.03)	1.21 (0.03)
	10	L2.Norm	2	0.5	0.12 (0.01)	0.11 (0.01)	0.14 (0.01)	0.12 (0.01)
	10	L2.Norm	4	1	0.11 (0.01)	0.11 (0.01)	0.28 (0.01)	0.23 (0.01)
100	50	L1.Norm	2	0.5	2.65 (0.09)	2.04 (0.05)	6.23 (0.12)	5.19 (0.07)
	50	L1.Norm	4	1	2.62 (0.09)	2.17 (0.05)	6.31 (0.12)	6.23 (0.1)
	50	L2.Norm	2	0.5	0.54 (0.02)	0.32 (0.01)	1.24 (0.05)	0.87 (0.02)
	50	L2.Norm	4	1	0.53 (0.02)	0.34 (0.01)	1.31 (0.05)	1.29 (0.04)
50	10	L1.Norm	2	0.5	1.44 (0.04)	1.32 (0.03)	1.59 (0.05)	1.33 (0.03)
	10	L1.Norm	4	1	1.41 (0.04)	1.34 (0.03)	1.95 (0.05)	1.66 (0.04)
	10	L2.Norm	2	0.5	0.33 (0.02)	0.27 (0.01)	0.41 (0.03)	0.28 (0.01)
	10	L2.Norm	4	1	0.32 (0.02)	0.28 (0.01)	0.64 (0.03)	0.44 (0.02)
50	50	L1.Norm	2	0.5	6.08 (0.28)	3.37 (0.11)	10.43 (0.15)	8.09 (0.11)
	50	L1.Norm	4	1	6.18 (0.25)	3.84 (0.11)	17.54 (0.31)	15.96 (0.23)
	50	L2.Norm	2	0.5	2.42 (0.19)	0.87 (0.05)	3.74 (0.11)	6.56 (0.19)
	50	L2.Norm	4	1	2.39 (0.15)	1.06 (0.05)	10.55 (0.4)	26.35 (0.75)

Table 3: Measurement of L1 Norm and L2 Norm among four methods.

Figure 3 and Table 3 both illustrate the effects of signal strength (i.e. large, small). There are two findings we derive from performing pairwise comparison: 1) when signal strength stays the same larger rho results in larger error in both the Prediction and Estimation Metrics. 2) Large signal strength has slightly less error than small signal strength in the Estimation Metrics.

4 Discussion

We wanted to assess the performance of ridge regression, lasso regression, forward selection and principle component regression. We did this using estimation, prediction, and variable selection metrics. With our standard assumptions of n = 100 and p = 50 we found that lasso performed the best when it came to performance metrics. In estimation metrics lasso regression also performed the best. Since PCR and ridge regression to not do variable selection it comes down to forward selection and ridge regression. Lasso performed better for TPR and forward selection performed better for TNR. When we look at the results wen n and p are changing we see that increasing n reduces error, and increasing p increases error. When looking aat the effects of signal strength we see that having a larger signal strength reduces error.

References

[1] Anne-Laure Boulesteix, Rolf Hh Groenwold, Michal Abrahamowicz, Harald Binder, Matthias Briel, Roman Hornung, Tim P Morris, Jörg Rahnenführer, Willi Sauerbrei, STRATOS Simulation Panel, and Simulation Panel. Introduction to statistical simulations in health research. *BMJ Open*, 10(12):e039921, December 2020.

[2] M Emre Celebi, Fatih Celiker, and Hassan A Kingravi. On euclidean norm approximations. August 2010.

[3] Latifa Greche, Maha Jazouli, Najia Es-Sbai, Aicha Majda, and Arsalane Zarghili. Comparison between euclidean and manhattan distance measure for facial expressions classification. In *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*. IEEE, April 2017.

[4] Trevor Hastie, Robert Tibshirani, and Ryan Tibshirani. Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. *Stat. Sci.*, 35(4):579–592, November 2020.

[5] A Hoerl, R Kennard, and K Baldwin. Ridge regression: Some simulations. *Communications in statistics: Simulation and computation*, 4(2):105–123, 1975.

[6] Chong Sun Hong and Tae Gyu Oh. TPR-TNR plot for confusion matrix. *Commun. Stat. Appl. Methods*, 28(2):161–169, March 2021.

[7] Ker-Chau Li and Naihua Duan. Regression analysis under link violation. *Ann. Stat.*, 17(3):1009–1052, September 1989.

[8] Michael A Poole and Patrick N O'Farrell. The assumptions of the linear regression model. *Trans. Inst. Br. Geogr.*, (52):145, March 1971.

[9] Markus Ringnér. What is principal component analysis? *Nat. Biotechnol.*, 26(3):303–304, March 2008.

[10] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.*, 58(1):267–288, January 1996.

[11] D Wallach and B Goffinet. Mean squared error of prediction as a criterion for evaluating and comparing system models. *Ecol. Modell.*, 44(3-4):299–306, January 1989.