,

The Comparison of Ridge Regression and Lasso in High-dimensional Data with Non-sparse Model

Nargiz Ahmadova

[a]*MSc in Economics, The University of Bonn,*

Abstract

This paper analyzes the performance of two known regularization techniques, ridge regression and lasso, in high dimensional data, where all features have an impact on the outcome. By "performance" the paper considers mean squared error (MSE), which is calculated on test data. Data generating process (DGP) is mainly based on the paper by Sirimongkolkasem, T., and Drikvandi, R. (2019) with some modifications made. The sample size in each simulation is taken as 150, which according to the based paper, is in accordance with the common applications of high dimensional data. Firstly, the paper gives an introduction to the issue of high dimensional data, then introduces the two shrinkage techniques, ridge regression and lasso. The rest of the paper then presents simulations and discusses the results. Lastly, a brief conclusion is provided containing the findings of the paper and motivation for the future research.

Keywords: Ridge, Lasso, MSE, High-dimension, Multicollinearity

1. Notation

In this paper, a covariate matrix is denoted by X, which has n rows and p columns, where n represents the sample size and p represents the number of covariates/features in the model. That means each row contains the value of predictors for each individual or each column contains the value of one predictor for all individuals. More explicitly:

$$X = \begin{bmatrix} X_{11} & X_{12} & . & . & . & X_{1p} \\ X_{21} & X_{22} & . & . & . & X_{2p} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ X_{n1} & X_{n2} & . & . & . & X_{np} \end{bmatrix} \tag{1}$$

Vectors are column vectors by default; Y is a column vector of length n containing an outcome variable and β is a column vector of length p containing estimated coefficients. There is no special notation for the vectors, so it is explicitly understood through the context. Lastly, individuals are indexed by i and predictors are indexed by j. Underscripts OLS, R and L are used to indicate the estimated coefficients of OLS, ridge regression and lasso, respectively. In case two underscripts are needed to use, R and L are used as subscripts (e.g. $\hat{\beta}_j^R$).

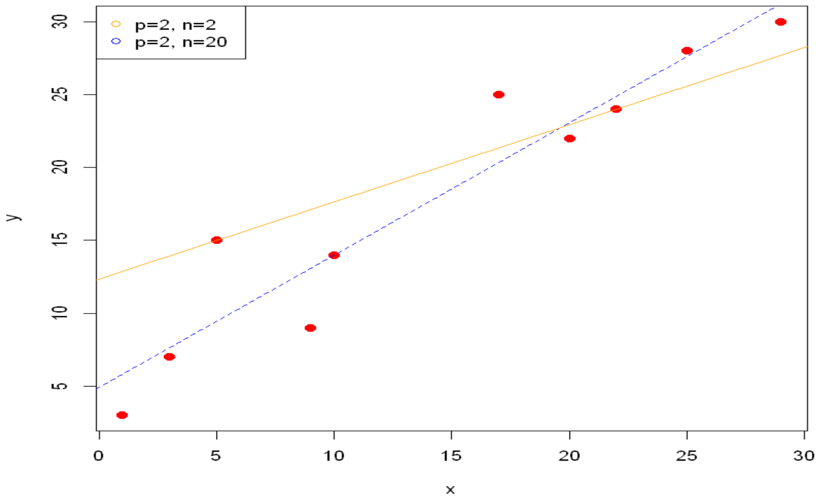
2. Introduction

Thanks to the technological advances that have facilitated generating and storing a mass amount of data, it is now possible to collect a vast amount of features to predict an outcome. One of the prominent example is the field of marketing, where the researcher may have an access to thousands of features to predict a customer behaviour (e.g. online shopping behaviour). Although this development has brought great value in terms of gaining more insight on the behavioral patterns, dealing with high-dimensional data has been a challenging topic. High-dimensional data refers to data sets containing more features (or covariates) than the number of observations ($p > n$). The main challenge with the high dimensional data is that traditional regression analysis, as Ordinary Least Squares (OLS) can not be applied for that kind of data; more explicitly, consider the formula for the OLS coefficients:

$$(X^T X)\hat{\beta}_{OLS} = X^T Y \tag{2}$$

Then the matrix $X^T X$ is of full rank n (assuming no multicollinearity), while $\hat{\beta}_{OLS}$ is a vector of length p. Hence, while there are p unknows, we have n equations, so if $p > n$, the vector $\hat{\beta}_{OLS}$ can not uniquely be determined from this system of equations. If we consider the case where $p = n$, then OLS will give us a solution, (!but) even that case would be problematic in the sense that the regression line will fit the data exactly. That would create an overfitting problem, leading to a high prediction error. We can visualize this from the Figure 1 below.

Figure 1



The orange line is OLS line for the case where the sample size (n=2) is equal to the number of covariates (p=2) and as seen the line is a perfect fit for the two data points. That means when we have many features in the model, simple least squares regression line is too flexible and hence overfits the data (James, G., et al., 2021). All of these suggest that traditional OLS method can not be applied if we have many features, particularly relative to the sample size. Hence, huge efforts have been made to develop sufficient statistical approaches in the high-dimensional statistics including some kind of regularization, dimension reduction and/or screening (Bergersen, L. C. (2013). In this paper, I am going to focus on 2 most famous regularization techniques, ridge regression and lasso and compare their performances in the high-dimensional setting through the simulation studies. Basically, the paper aims to analyze which shrinkage techniques deal with high-dimensionality the best in terms of MSE under various contexts. Firstly, the author introduces two methods and explain how they deal with high-dimensional data, then simulation studies are conducted and the results are discussed. Finally, the paper concludes by highlighting the key findings and giving motivation for future studies.

3. Ridge Regression

Ridge regression is another method of obtaining coefficients for the linear regression. It is very similar to the least squares, except that in this method coefficients are found through minimizing a slightly different objective function:

$$S(b) = \sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij})^2 + \lambda \sum_{j=1}^{p} b_j^2 \tag{3}$$

$$\hat{\beta}_R = argminS(b) \tag{4}$$

where $\sum_{j=1}^{p} b_j^2$ is called ***l2 normalization*** of the coefficients. We can also express ridge estimator in a matrix notation:

$$S(b) = (Y - Xb)^T(Y - Xb) + \lambda b^T b \tag{5}$$

$$\hat{\beta}_R = argminS(b) \tag{6}$$

$$\hat{\beta}_R = (X^T X + \lambda I)^{-1} X^T Y \tag{7}$$

As seen from the formula 3, ridge regression has an additional term to minimize, the sum of squared coefficients. Basically, an objective of this method is not only about fitting the data (or minimizing sum of squared residuals), but also minimizing the coefficients (in absolute term). How much weight is given to this additional term depends on the value of λ which is called as a ***shrinkage penalty***. From the solution 7, we can see that unlike least squares ridge regression adds a positive number to the diagonal elements of the matrix $(X^T X)$. This is how ridge regression deal with high dimensional data; because the quantity $(X^T X + \lambda I)$ is always invertible even in the case when the matrix $X^T X$ is non-singular, ridge regression always gives a unique solution in contrast to OLS (Hoerl, A. E., and Kennard, R. W. (2000)). Applying a penalty term to the coefficients also has an effect of reducing the variance of estimated coefficients. Basically, ridge regression with its penalty term reduces the variance of estimated coefficients. To see this, one may look at the difference in the variances between OLS coefficients and that of ridge regression (derivation assumes homoskedasticity of error terms and taken from the paper by van Wieringen, W. N, 2015):

$$Var(\hat{\beta}_{OLS}) - Var(\hat{\beta}_R) = \sigma^2 (X^T X + \lambda I)^{-1} [2\lambda I + \lambda^2 (X^T X)^{-1}][(X^T X + \lambda I)^{-1})]^T \tag{8}$$

The difference is non-negative definite as each component in the matrix product is non-negative definite (van Wieringen, W. N. (2015)). Hence, we have:

$$Var(\hat{\beta}_{OLS}) \geq Var(\hat{\beta}_R) \tag{9}$$

However, this advantage of ridge regression comes with its cost of having bias, which makes one faces ***bias-variance trade-off*** while choosing between OLS and ridge regression. That means the expected value of ridge coefficients is not equal to the true one, or in other words, distribution of ridge coefficients would not be centered around the true mean. This can easily be seen from the equation 10:

$$\mathbb{E}(\hat{\beta}_R) = [I + \lambda(X^T X)^{-1}]^{-1}\beta \tag{10}$$

Unless lambda is zero (in that case which would be just OLS regression), the expected value of the ridge regression coefficients would be different from the true one, β. To summarize, ridge regression deals with high-dimensional data by its additional term in the objective function, the sum of squared coefficients. In this way, ridge regression can provide a unique solution for the estimated coefficients in high-dimensional data, as well as achieving lower variance. As a trade-off, one should accept having a biased estimator while using ridge regression. The choice of lambda determines how much we trade-off bias with variance and a prediction accuracy of ridge regression (as measured in terms of the mean-squared error) depends on how much variance is decreased in respect to increase in bias.

4. Lasso

In lasso, or l1-regularized regression, we estimate the parameters for the linear regression by minimizing the following objective function:

$$S(b) = \sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \mid b_j \mid \tag{11}$$

$$\hat{\beta}_L = argminS(b) \tag{12}$$

where $\sum_{j=1}^{p} \mid b_j \mid$ is ***l1 norm*** of the coefficients. The use of l1 norm is special because in this way lasso can get some coefficients exactly to zero, fulfilling the role of subset selection, which is not possible in ridge regression. This gives advantage to lasso over ridge regression in terms of being able to produce the final model which is easier to interpret. In order to understand, why l1 normalization has an ability to select variables in contrast to l2 normalization, we can look at the Langrangian minimization problems above in a more explicit way:

$$minimize \sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij})^2 subjectto \sum_{j=1}^{p} b_j^2 \leq s(ridge) \qquad (13)$$

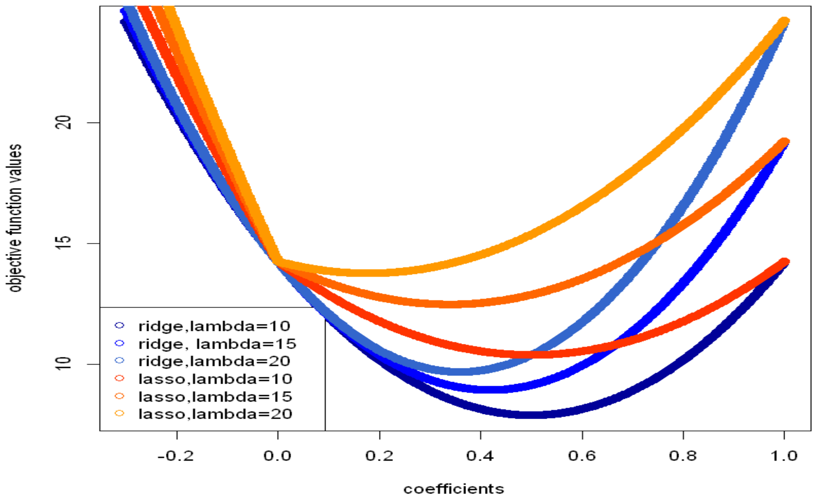
$$minimize \sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij})^2 subjectto \sum_{j=1}^{p} \mid b_j \mid \leq s(lasso) \qquad (14)$$

By Langrangian duality, there is a one-to-one correspondence between the Langrangian form and the constrained problem (Hastie, T., Tibshirani, R., and Wainwright, M. (2015)); that means for each value of s, there is a corresponding value of λ that gives the same coefficients from the Langrangian form. As seen from the equations (13 and 14), for 2 predictors model the constraint region ($\sum_{j=1}^{p} b_j^2 \leq s$) for ridge regression would be the disk, while that ($\sum_{j=1}^{p} \mid b_j \mid \leq s$) would be the diamond for lasso. Because the diamond has corners in contrast to the disk, there is a possibility that contours for the objective function (in 2 predictors model, contours would be the ellipses each representing different combinations of the coefficients having the same RSS) touch or be tangent to the constraint region at the corner (James, et al., 2021). In that case, the lasso would give us only one parameter, setting another one to zero. Although this example is for 2 predictors model, the same logic can be generalized for more than 2 predictors.

In order to understand this issue more, we can use another explanation. [1]. Let us assume that we have a model with only one predictor (no intercept). Then we can create a sequence of coefficients which cover the OLS coefficient as well. Then we can write down the objective function of the ridge ($\sum_{i=1}^{n}(y_i - \hat{\beta}_1 x_i)^2 + \lambda \hat{\beta}_1^2$) and lasso ($\sum_{i=1}^{n}(y_i - \hat{\beta}_1 x_i)^2 + \lambda \mid \hat{\beta}_1 \mid$) and calculate the values of objective functions for each coefficient in the sequence for three lambda values of 10, 15 and 20 (choice of lambda values are arbitrary). Plotting these values against coefficients, we can get the Figure 2 below. As seen, unlike to ridge, an objective function of the lasso has a king at zero, so there is a possibility that lasso shrinks the coefficient to zero. Although this example is made based on just 3 lambda values, it illustrates the idea.

[1] this explanation is inspired by the video produced by StatQuest

Figure 2: Objective functions of Ridge and Lasso



Note: While working with the ridge regression and lasso, we standardize X matrix, so that each column has a mean of zero and standard deviation of 1. We do it because unlike least squares, the value of $X_j\hat{\beta}_j^R$ and $X_j\hat{\beta}_j^L$ depends on the scaling of j^{th} predictor (James, et al., 2021). Thanks to the standardization, the intercept, $\hat{\beta}_0$, can easily be recovered by:

$$\hat{\beta}_0 = \overline{y} = 1/n\sum_{i=1}^{n} y_i \tag{15}$$

Hence, I will ignore the intercept in the rest of the paper.

5. Simulation 1

In this simulation, I am going to compare the performance of ridge regression and lasso in the high dimensional data setting, where all covariates have an equal impact on the outcome and there is no correlation among them. Data generating process is based on the paper Sirimongkolkasem, T., and Drikvandi, R. (2019) and is as following;

- The sample size (n) is equal to **150.**
- A range of different values, 200, 300, 400, 500 and 600 are considered for number of covariates, p.

- Covariate matrix $X \in \mathbb{R}^{150*p}$ is generated from **multivariate normal distribution**; **mean** of all covariates is equal to **zero** while their **variance** is equal to **1**. No correlation among covariates is assumed, so the off-diagonal values of variance-covariance matrix is zero.

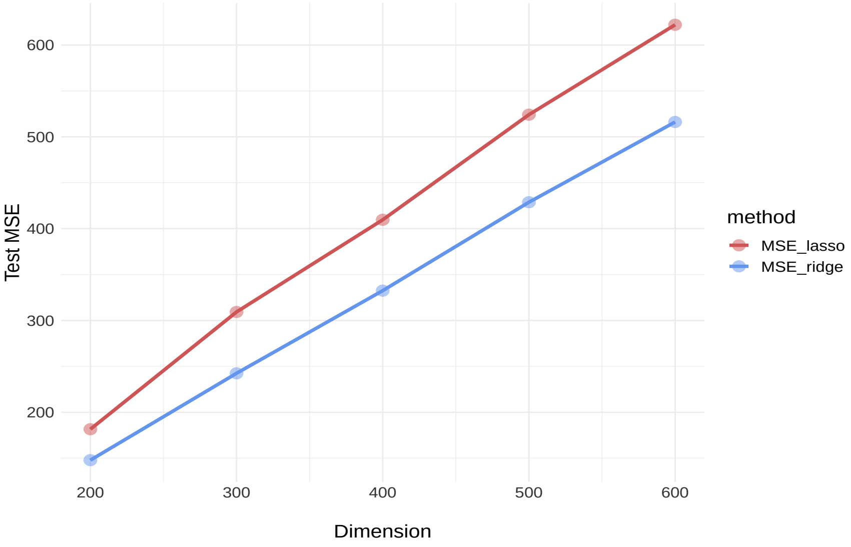
- Error terms is drawn from the **standard normal distribution**; $\varepsilon \sim \mathcal{N}(0,1)$

- The true vector of coefficients, $\beta \in \mathbb{R}^p$ is equal to 1 for all predictors; so, we have a non-sparse model, where each coefficient has an equal impact on the outcome variable.

- Outcome variable, Y has a linear relationship with covariates; specifically, $Y = X\beta + \varepsilon$.

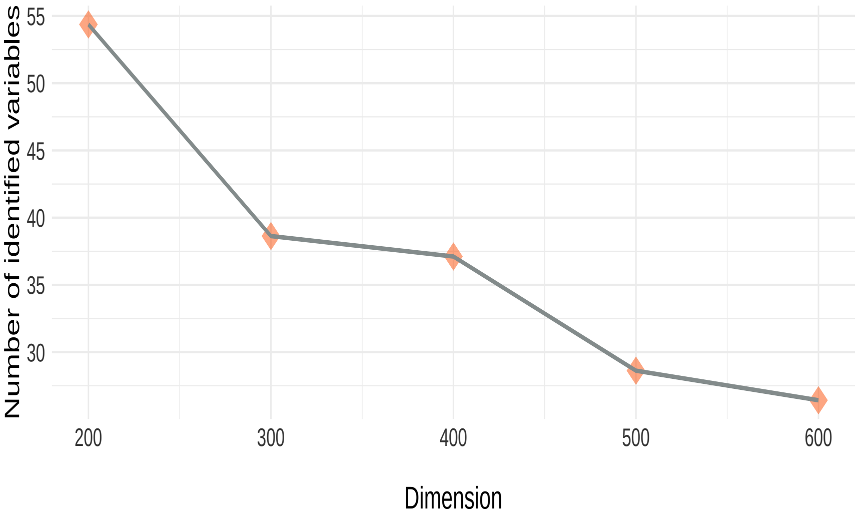
I considered the value of λ for which error is smallest using *cv.glmnet* function and then used this value of λ in training data set. The comparison between ridge and lasso is then done by MSE calculated on the test data. The results of simulation study are summarized in the Figure 3 below:

Figure 3



Apparently, for all dimensions, ridge regression outperforms lasso in terms of having lower test MSE, and the difference is marked for higher dimensions as 400, 500 or 600. This finding is consistent with the findings of the paper by Sirimongkolkasem, T., and Drikvandi, R. (2019), which states that under the setting with non-sparse model and high dimensional data, lasso performed worse in prediction compared to ridge regression. This result is quite intuitive because lasso's shrinkage ability to zero may deviate from the true model more in the non-sparse model, where all covariates are different from zero. Since all covariates are important in explaining outcome variable, their shrinkage to zero becomes costly in terms of prediction accuracy (as measured by MSE). This should particularly hold for much higher dimensions, since the variables correctly identified (meaning not shrinked to zero) by lasso should be smaller in higher dimensions. As a matter of fact, from the Figure 4 below we can see how the number of identified variables by lasso decrease as dimension increases.

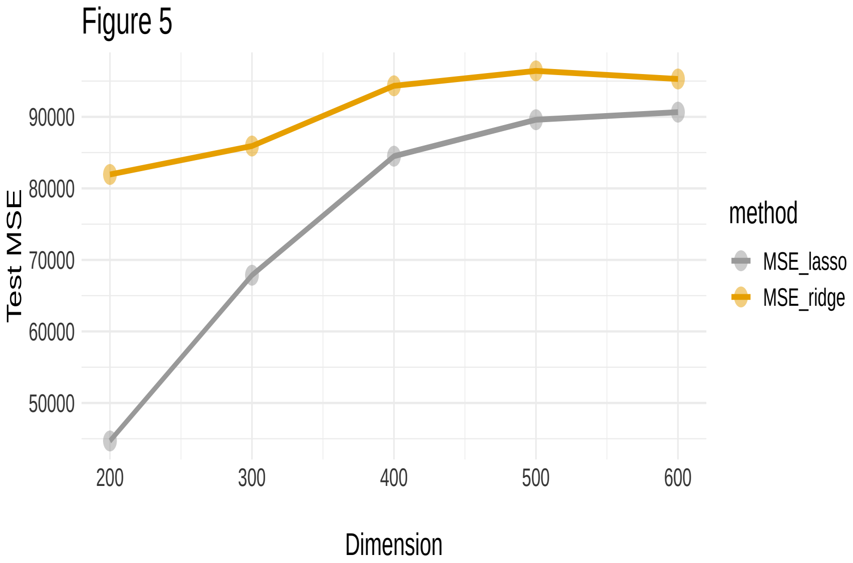
Figure 4



In addition, this result can be explained by the claim that ridge performs well when there are **many predictors** to explain the output and each of the coefficients have approximately **the same size** (Ridge Regression and Lasso, (2014)). Hence, in our DGP, where there are many predictors and all of the coefficients are the same, it is expected that ridge regression performs better than lasso. The claim that having many coefficients being of equal size gives an advantage to ridge regression can be associated to the difference in the way that ridge and lasso shrink the coefficients. Motivated from the simulation 1, in the second simulation I am going to explore more how the size of coefficients affect the relative performance of ridge over lasso in high dimensional data

6. Simulation 2.1

In this simulation, the set-up is the same with the simulation 1 except coefficient sizes. The first 50 entries of the true beta vector are set with high coefficients and the rest with close-to-zero coefficients. More explicitly, the first 50 entries of beta vector are assigned the values from 20 to 69, sequentially increasing by 1 and the rest of the coefficients are assigned a number between 0 and 0.01. The simulation result is presented by the Figure 5 below:



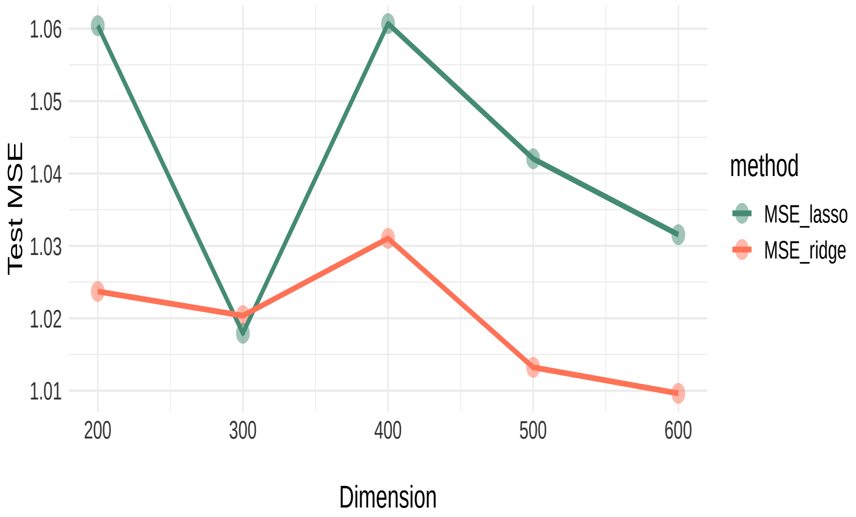
As the Figure 5 shows, in this simulation lasso outperformed ridge regression for all dimensions and particularly the difference is marked for smaller dimensions as 200 or 300. As a matter of fact, this finding is consistent with that of Kurumathur, S. (2022), who claims that lasso performs well in a setting where there are small number of predictors that have substantially high coefficients and others have negligible coefficients. This shows that indeed the magnitude of the coefficients play an important role in the relative performance of ridge over lasso. As mentioned above, this can be linked to different shrinkage methods of two techniques. Namely, ridge regression shrinks proportionally, so it can be the case with proportional shrinkage ridge can preserve true relationship more when all covariates are the same (or similar).

The second observation from the Figure 5 is that outperformance of lasso over ridge decreases as dimension increases. Firstly, this can be because in this simulation the first 50 entries of beta vector are set to be high for all dimensions and the rest are set closer to each other between 0 and 0.01. Hence, as the dimension increases, the proportion of coefficients that are similar to each other increases as well (e.g. out of 600 covariates, having 50 coefficients to be different from each other means 91 percent of them are similar to each other). In addition, in general when the number of predictors is much higher than the number of observations, lasso does not perform well ((Ridge Regression and Lasso, (2014))) and this can be the second explanation for why we can see that the performance difference between lasso and ridge gets closer as dimension increases.

7. Simulation 2.2

As a next part of this simulation, I am going to set all coefficients close to zero; basically, this time all coefficients were set between 0 and 0.01. It is expected that ridge regression performs better due to having similar coefficients, but this time coefficients are very close to zero as opposed to being 1 in the simulation 1, so this set-up may give some advantage to lasso, as well. The Figure 6 present the results:

Figure 6



As expected, this time we obtained a bit mixed results, but dominantly ridge ourperformed lasso except for the dimension size of 300. All in all, from the simulation 1 and 2, we can conclude that under high-dimensional setting, where there is no collinearity, in general ridge regression outperforms lasso for higher dimensions. Ridge regression performs even better if the coefficients are of equal size. However, if there are small number of predictors that have substantially high coefficients and others have negligible coefficients, then lasso performs better. In addition, if the number of variables are much higher than the sample size, then lasso's performance decreases; from all the simulations, it is shown that for very high dimensions (in our case 500 or 600) lasso's prediction accuracy declines. In addition, the simulation 2.2 show that if all the coefficients are very close to zero, then under the specified DGP ridge regression performs better for most of the dimensions that paper looked at. In the next simulation, I am going to look at another important factor, multicollinearity. However, before going to the simulations directly, the next section briefly explains the notion of multicollinearity and problems associated with it to create a motivation for the simulation study.

8. Multcollinearity

In general, multicollinearity in the data refers to high correlation between two or more independent variables. Multicollinearity is a matter of degree, so we call it *perfect multicollinearity* if there is an exact linear dependency between independent variables and an *imperfect multicollinearity* if there is a correlation among covariates, but they are not linearly dependent; here I will only consider an imperfect multicollinearity. When independent variables are highly correlated, then variance of the estimated coefficients are inflated, which makes them *instable*. By "instable" one means that for each new draw of data, we would get coefficients that are substantially different from what we got in the previous data. Or in other words, coefficients become very sensitive to the small changes in the model, which decreases credibility of the results obtained for a given data. Hence, this paper considers the issue of multicollinearity by comparing the performance of ridge and lasso in high-dimensional data with multicollinear covariates.

9. Simulation 3.1

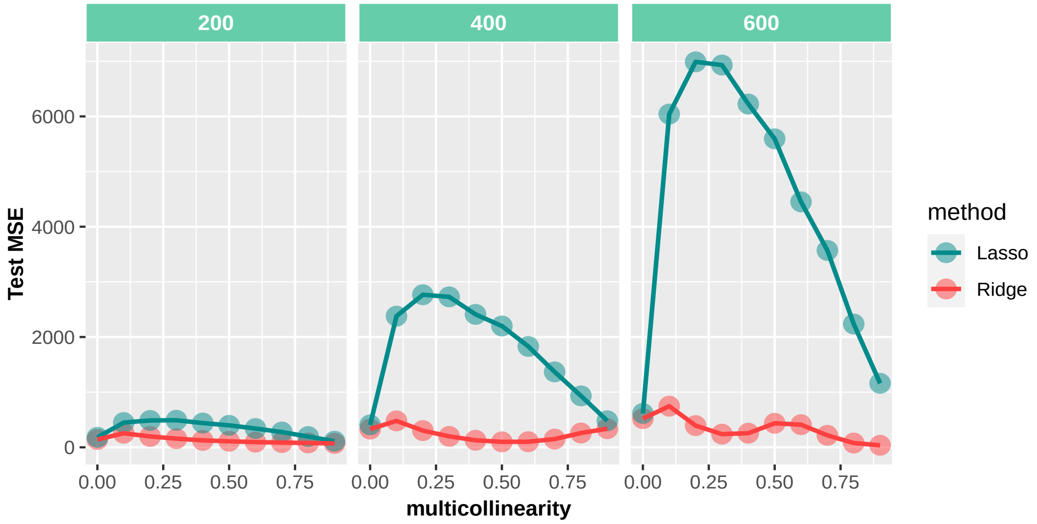
In this simulation, I am going to introduce the parameter "mult" controlling the degree of collinearity among predictors. The simulation will look at a range of values from 0.1 to 0.9 for the level of collinearity. Except for variance-covariance matrix of independent variables, data generating process is the same as in simulation 1, which is again shown below:

- The sample size (n) is equal to 150.
- A range of different values, 200,300,400,500 and 600 are considered for number of covariates, p
- Covariate matrix $X \in \mathbb{R}^{150*p}$ is generated from **multivariate normal distribution**; **mean** of all covariates is equal to **zero** while their variance is equal to 1. There is a correlation among covariates, and correlation is assumed to be the same among all pairs in the data. That means in this simulation off-diagonal values of the matrix will be equal to one number (different from zero).
- Error terms is drawn from the standard normal distribution
- The true vector of coefficients, $\beta \in \mathbb{R}^p$ is equal to 1 for all predictors; so, we have a non-sparse model, where each coefficient has an equal impact on the outcome variable.

- An outcome variable, Y has a linear relationship with covariates; specifically $Y = X * \beta + \varepsilon$

The figure 7 illustrates the results.

Figure 7

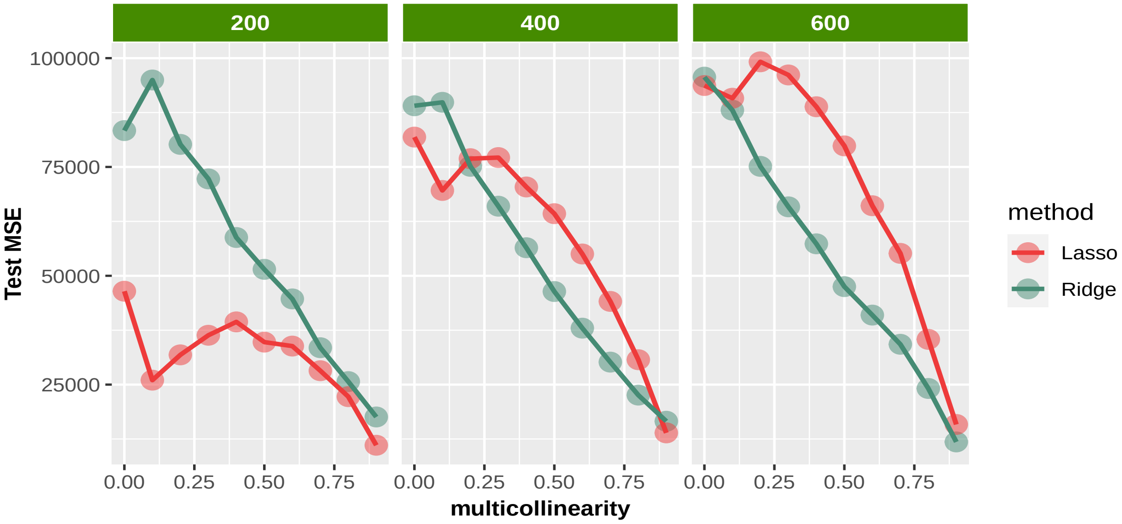


In general, we can see that in all dimensions, ridge outperforms lasso, and the difference becomes larger as dimension size increases. Also, the figure shows that as multicollinearity increases, MSE of lasso increases, but from some value of multicollinearity the trend reverses; this can be because if there is a a very high multicollinearity level between 2 covariates, then the same information can be obtained just using one covariate, so this creates justification for lasso's shrinkage ability to zero. Hence, we can see declining test MSE for lasso in high level of multicollinearity.

10. Simulation 3.2

In this simulation, I am going to set the coefficient sizes as in simulation 2.1 (in favor for lasso) to see if the findings in the previous simulations for the relative performance of ridge over lasso (or vice verse) hold with multicollinear data as well. If that is the case, it is expected that this time lasso will performs better over ridge.

Figure 8



From the Figure 8, we can see that for the dimension size of 200, lasso outperformed ridge for all multicollinearity levels, being consistent with the result of the simulation 2.1. However, for the dimension size of 400, lasso performed better than ridge only for smaller level of multicollinearity (for 0.1 and 0.2) and a very high level of multicollinearity (0.9). Similar statements can be made for the dimension size of 600. The results are interesting because in the simulation 2.1, where the setting is exactly similar except for multicollinearity level, lasso performed better than ridge for all dimensions. That means with multicollinear data lasso lost its outperformance. However, in the simulation 3.1, ridge could preserve its dominance for multicollinear data as well. This show that multicollinearity affects the performance of lasso substantially especially for the medium-level collinearity. Surely, all these results are stated in reference to DGP set in the paper, so it should not be generalized.

11. Conclusion

To conclude, this paper compared two shrinkage techniques so called ridge regression and lasso in high dimensional data setting. Given the data generating process, the paper found that for very high dimensions, lasso decreases its performance, so ridge regression performs better in terms of having lower test MSE. The magnitude of true coefficients substantially affect the performance of two methods; if the coefficients are of equal or similar size, then ridge regression has a better prediction. On the other hand, lasso performs well when a small proportion of coefficients is high and the rest have close-to-zero effects. The paper also looked at how they perform with multicollinear data. Basically, it seems that lasso considerably decreases its performance with multicollinear data, but performs well if the multicollinearity level is very high (close to 1). Basically, relative performance of ridge over lasso depends on data generating process and one should consider it before making decision. Although in real life, we can not observe data generating process, it can be cleverly guessed. This can be motivation for the further research on using real data to analyze performance properties of ridge regression versus lasso.

12. References

Bergersen, L. C. (2013). Guiding the lasso: Regression in high dimensions.

Hoerl, A. E., and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

James, G., Witten, D., Hastie, T. J., and Tibshirani, R. J. (2021). An introduction to statistical learning: With applications in R. Springer.

Kurumathur, S. (n.d.). RPubs. Retrieved August 22, 2022, from https://rpubs.com/kkshalini/8

Ridge Regression and Lasso. Information Systems Research. (2014). Retrieved August 22, 2022, from https://www.is.uni-freiburg.de/resources/seminar-papers

van Wieringen, W. N. (2015). Lecture notes on ridge regression. arXiv preprint arXiv:1509.09169.