Comparing the Performance of OLS, Ridge Regression and Lasso with Big Data

Submitted to - Prof. Dr. Lena Janys

Raunak Mehrotra

23/08/2022

Contents

1 Introduction –

With the advancement of technology, high-dimensional datasets have become more prevalent than ever. The number of predictors (p) for the variable of interest denotes the dimensionality of the data, and so high-dimensional data is characterized wherein the size of p exceeds the number of observations (n), i.e., $p \gg n$. However, in such large and complex datasets generally, only a subset of predictor variables is significantly related to the response variable. By identifying and removing these irrelevant variables, we can generate a model that doesn't overfit the data and is more easily interpretable (James G et al,2021). Further, many of these predictors might be correlated to each other, that is, the dataset might suffer from multicollinearity. Multicollinearity generates unreliable estimates due to the large variance of the coefficients. The interdependence of the variables also makes it harder to identify the marginal impact of the predictors on the response variable. Hence, the precision of the model decreases which implies that it performs poorly on the given data (Chan J et al, 2022). Hence, to conduct credible data analysis, it is important to select variables that are relevant to the response variable and generate a model that improves the prediction of the same.

Multiple linear regression is one of the most commonly used statistical techniques to predict the performance of a model. Due to their assumptions and simplicity to interpret the results, they are widely used in low-dimensional settings. However, in a high-dimensional setting, a linear regression model tends to perform poorly. To elaborate on the same, consider a high-dimensional linear regression model

$$y = X\beta + \varepsilon, p \gg n$$

Where $y = (y_1 + y_2 + \cdots + y_n)\epsilon R^n$ is the response vector, $X = (x_1 + x_2 + \cdots + x_n)\epsilon R^{(nXp)}$ is the covariate (predictor) matrix, $\beta\epsilon R^p$ is the vector containing unknown regression coefficients and $\varepsilon\epsilon R^n$ is the random noise vector. Also, without loss of generality, assume that the predictors and response variable are mean-centered and so there is no intercept term. In such a scenario, we know that the ordinary least square (OLS) estimate for β is given by $\hat{\beta}^{OLS} = (X^TX)^{-1}X^Ty$. In a high-dimensional dataset, as the full rank assumption of the OLS is violated, we obtain infinitely many solutions, that is, we overfit the data (Sirimongkolkasem T. and Drikvandi R., 2019). This is further accentuated by the fact that while OLS overfits in the training data and the mean square error (MSE) decreases with an increase in the features (predictors); the same model on an independent test dataset performs extremely poorly as shown by the increase in MSE as predictors increase (James G et al,2021). Therefore, when $(p \gg n)$ or $(p \approx n)$, a simple least squares regression model is too flexible and hence overfits the data. This inability of the least squares regression to generate reliable results in a high-dimensional dataset motivates us to look for alternatives.

'Shrinkage' or 'Regularization' is one of the approaches to reduce the variance and improve the predictability of the model. In this approach, a model is fit on all the predictors and the coefficient estimates shrunken towards zero. Based on the method of shrinkage adopted, a few predictors may be shrunken to exactly zero as well. Therefore, regularization can also help in variable selection. However, there's a trade-off. The fall in variance is often associated with a small rise in bias. Ridge regression, Lasso, and elastic net regression are among the few commonly used shrinkage methods (Altelbany S, 2021).

The paper hereby aims to compare the performance of the regularization methods; Ridge regression, and LASSO, on a high-dimensional dataset against the OLS. The following sections cover a brief theoretical review of the aforementioned types of shrinkage, a simulation study, and an empirical application of real data.

2 Theory -

2.1 Ridge Regression

In the presence of non-orthogonal (correlated) explanatory variables, OLS performs poorly. The estimates tend to have high variance and even might have the wrong sign. Hoerl and Kennard (1970) developed 'Ridge Regression' as an alternative technique to control the general instability associated with the least squares estimates. Taking the linear model developed before, the ridge regression estimates are given by:

$$\hat{\beta}^R = (X^TX + \lambda I)^{-1}X^Ty, \lambda \geq 0$$

In other words, these ridge regression estimates are the values that solve the following Lagrangian form:

$$\hat{\beta}^R = argmin_\beta[\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda(\sum_{j=1}^{P}(\beta_j^2)^{1/2})]$$

Where the first term is the 'residual sum of squares (RSS)' from the OLS and the $(\sum_{j=1}^{P}\beta_j^2)^{1/2} = ||\beta||_2$ is the L_2 norm shrinkage penalty on β and λ is the tuning parameter. When λ equal zero, the ridge estimate is the same as that of OLS, however, as λ increases, the shrinkage penalty grows and the estimate tends to zero. Therefore, λ is the power of the penalty on β (James G et al,2021). Recent studies have shown ridge regression to perform suitably in presence of highly correlated variables and since there's a higher occurrence of multicollinearity in high dimensional data, therefore, the use of ridge regression in big data study is justified (Hoerl and Kennard,1970; Algamal Z.Y et al,2015). However, it is important to note that the ridge regression does not perform variable selection. It takes all the predictors as relevant to the response variable and shrinks them towards zero but not exactly to zero such that the MSE of the estimation improves. Therefore, in a model with a high number of features, it can be difficult to interpret the ridge estimates (Tibshirani,1996).

2.2 LASSO

Glaring over the fact that ridge regression is a continuous process that generates a stable model which however can be difficult to interpret. This is because the ridge regression does not set any coefficients to zero. The 'Least Absolute Shrinkage and Selection Operator (LASSO)' is a relatively new technique proposed by Robert Tibshirani (1996). Lasso like ridge regression shrinks some of the coefficients towards zero and sets others to zero. Hence, the lasso can be said to try to perform a variable selection for easy interpretation of the model and generate a stable model like ridge regression at the same time (Tibshirani,1996). The lasso estimate $\hat{\beta}^L$ is defined by

$$\hat{\beta}^L = argmin_\beta[\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{P}(|\beta_j|)]$$

Similar to the ridge, the first term $\sum_{i=1}^{n}(y_i - x_i\beta)^2$ is the squared error loss from the OLS. However, the lasso uses L_1 norm shrinkage penalty on β rather than L_2 as given by $\sum_{j=1}^{P}|\beta_j| = ||\beta||_1$. λ like before is the tuning parameter, but it also has an additional role in reducing certain coefficients to zero. Since lasso generates sparse models (that is, the number of non-zero coefficient estimates in the model is less than the sample size n) this can be problematic. In presence of correlated predictor variables, lasso tends to randomly select only a few of them (Sirimongkolkasem T. and Drikvandi R., 2019). Further, in high-dimensional data, as a consequence of the generation of sparse models, although there might be more than n non-zero predictors, the predictors involved in the model are however restricted to $\leq n$ (Zhou, 2013).

2.3 Cross-Validation

In this research, I use cross-validation to determine the optimal lambda value. In the empirical application section, I also use the cross-validation method to infer the performance of the predicted model. Hereby, I present a general description of cross-validation.

Consider a dataset with n observations. In K-fold cross-validation, the entire dataset is randomly divided into K subsamples of equal size. Let the k^{th} subset be the validation set, then the remaining $k-1$ subsets are the training datasets used for developing a prediction model. The fitted model is then used on the validation set, the k^{th} set, to obtain the prediction error of that set. This process is repeated K times, with each of the k^{th} subset being used for validation and the rest to fit the model. At the end of the process, we obtain prediction errors for each of the K validation sets. By taking the average of the same, we derive the cross-validation estimate of the prediction error (Ranstam J. and Cook J.A.,2018). Since MSE is commonly used as a measure of prediction performance, therefore, the cross-validation prediction error is

$$CV = \frac{1}{K}\sum_{k=1}^{K}MSE(k)$$

Where K is the number of subsets and $MSE(k)$ is the mean-squared error of the k^{th} validation set. MSE can be defined as

$$MSE(k) = \frac{1}{n_k} \sum_{i=1}^{n_k} (y_i - x_i \beta)^2$$

The decision of what K to use should be based on bias-variance trade-off. While, in the case of Leave-one-out cross-validation (LOOCV) where $K = n$, there is a low bias and high variance because of overlapping in training sets due to a high number of folds; for K-fold cross-validation, the variance decreases as the number of folds is less but for a slightly higher bias (James G et al,2021). We use 10-fold cross-validation from the recommended fivefold and tenfold cross-validation for application purposes (Tibshirani,1996).

3 Simulation Study

3.1 Data

For the simulation study, I use the data available in the paper "Robust linear regression for high-dimensional data: An overview" by Peter Filzmoser and Klaus Nordhausen (2021). The paper looks at alternative regression estimators to least square regression that are robust against the challenges encountered in a high-dimensional setting. The paper looks at the performance of MM estimators, elastic net regression, and partial least squares regression to name a few. The data used is from a production process that for simplicity purposes has been centered and scaled, and also the time series characteristic has been ignored by the authors of the papers. Therefore, I can use the data without diving deep into considerations of autocorrelations or heteroskedasticity for instance. The data contains 648 observations for 468 predictors. Although, it's not high-dimensional data ($p \gg n$), we can expect a large number of predictors ($p \approx n$) to show similar issues in the least squares regression that I might have encountered for the ($p \gg n$) case, like multicollinearity, overfitting, etc.

3.2 Simulation Model

I randomly split the data into training (75% of the observations) and a testing set (25% of the observations). While the former is used to fit the different models, the evaluation is done on the latter, that is, the testing set. For analysis purposes, I fit models of least squares regression (as described in the Introduction section), ridge regression, and lasso regression, and compare their performance based on the MSE of the testing set. I use the glmnet() package in R statistical software for the same.

For the ridge regression and lasso, we require the value of the regularization (tuning) parameter. I derive lambda from 100 values ranging from 10^{10} to 10^{-2}. This is done to prevent the unexceptionally high value of lambda which might lead to a very strict model and make interpretation of it more difficult. Further, the optimal value of lambda is decided through 10-fold cross-validation as described above. The optimal value of lambda is associated with the minimum MSE obtained through cross-validation. I use the built-in cross-validation function, cv.glmnet() for calculating the lambda value.

Finally, this model selection algorithm was repeated 100 times. So, we took the average of MSEs obtained from 100 repetitions, to arrive at the final value.

3.3 Results and Discussion

Before, delving into the analysis of the fitted models, I graph out a Pearson correlation matrix heatmap for variables between $X100$ to $X135$ to visualize the presence of multicollinearity among the predictors. As can be seen from the plot, there exists a high degree of correlation among the 36 predictors. Interestingly, the Pearson coefficient goes as high as 0.99 between variables like $X103$ and $X104$, $X106$ and $X103$ and so on. The Pearson coefficient for $X110$ and $X120$ is 1.00 and for $X120$ and $X122$ is -0.80. Considering that we find such a high degree of correlation between 36 predictors, it is sufficient to extend the presence of such a high degree of correlation between the predictors in the entire dataset. Therefore, we can expect the least squares model to perform poorly and possibly overfit the data. In the simulation study, due to perfect linear relationship between several predictors, we can expect the least square model to not estimate the coefficients. However, for study purpose, I set the value of these coefficients to zero.

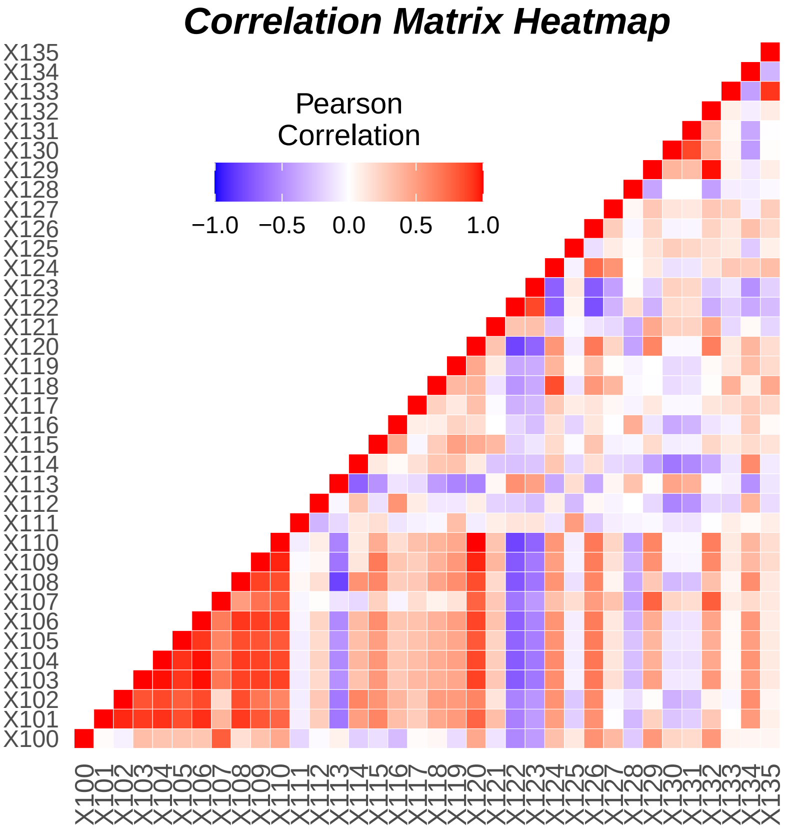


Figure 1: Pearson Correlation Matrix Heatmap

Having established the presence of multicollinearity in the dataset, I proceed with the linear regression model. Based on the MSE calculated for the least square model from the simulation, we can conclude the same. While the MSE for the training set is 0.034, it exorbitantly increases to 65.737 for the testing set. The result is similar to that of Filzmoser and Nordhausen, 2021 and shows overfitting of the training data and a poor prediction performance as suggested by the literature. Further, the presence of outliers (as seen on the test data plot) could have had an impact on the parameter estimation. The left plot shows the fitted values $\hat{y}$ versus the response y on the training set, the right plot is for the test set. Evidently, the least-square model is a poor choice as shown by the stark contrast between the fitted model (using the training set) and the predicted model (using the testing set).

Moving on to the ridge regression and lasso regression. The optimal value of lambda using the 10-fold cross-validation was 6.136 and 0.123 respectively. I used this value of lambda to fit the ridge regression and lasso regression using the R-package glmnet. While the former is achieved by setting the argument alpha=0, for the latter, we fix alpha=1. Finally, I calculate the MSE which is the average of the squared difference between true values and the predicted values of the dataset used to fit the model.

But there is a caveat in the selected optimal value of lambda. Shown below is the cross-validation plot for optimizing lambda for lasso. The upper part shows the number of non-zero coefficients in the model for a given value of lambda. The left vertical dashed line is the value of λ associated with the minimum MSE and the right vertical dashed line is the value of λ selected by the 'one-standard-error' rule, that is, this value of λ is within 1 standard deviation away from the minimum MSE. One can notice the large error bars for

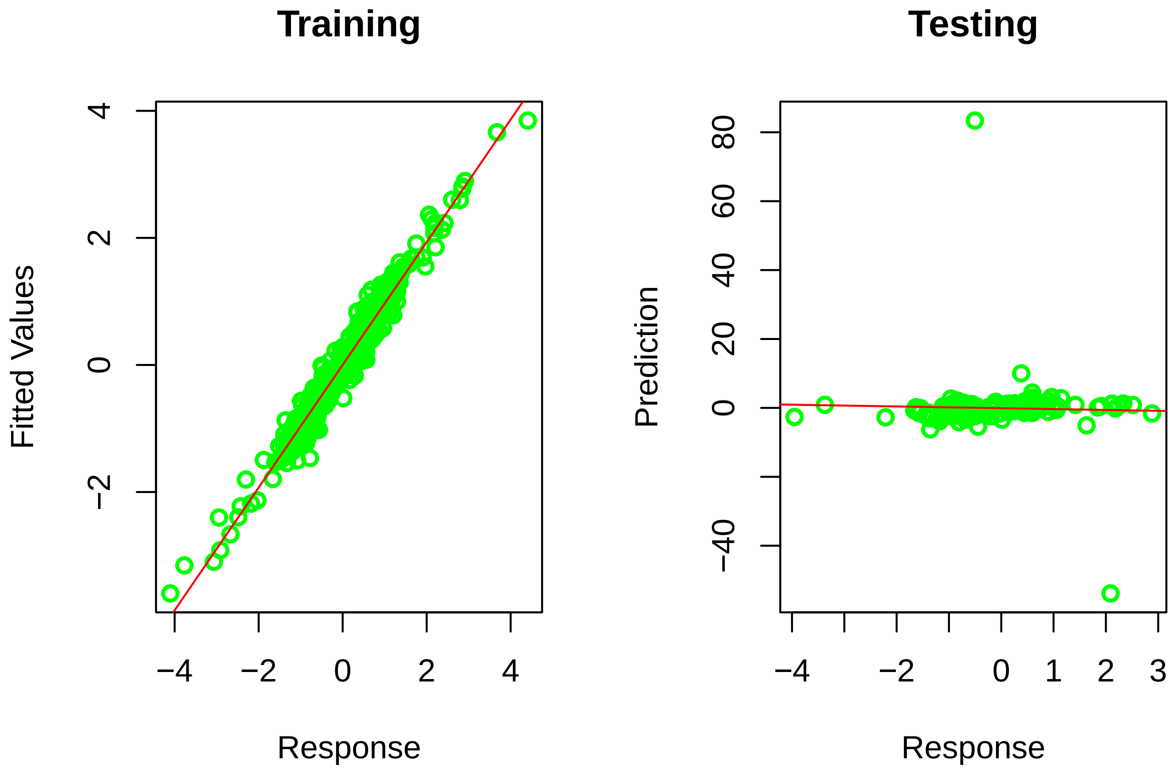


Figure 2: Linear Regression Model

every cross-validated value of λ, even for optimal $\lambda = 0.123$, possibly due to a lack of sufficient observations compared to the number of predictors. Therefore, the lasso model might not generate the true model and there might be a larger number of truly non-zero coefficients. Hence, the lasso may lead to inconsistent variable selection when there are a large number of features (Sirimongkolkasem T. and Drikvandi R., 2019).

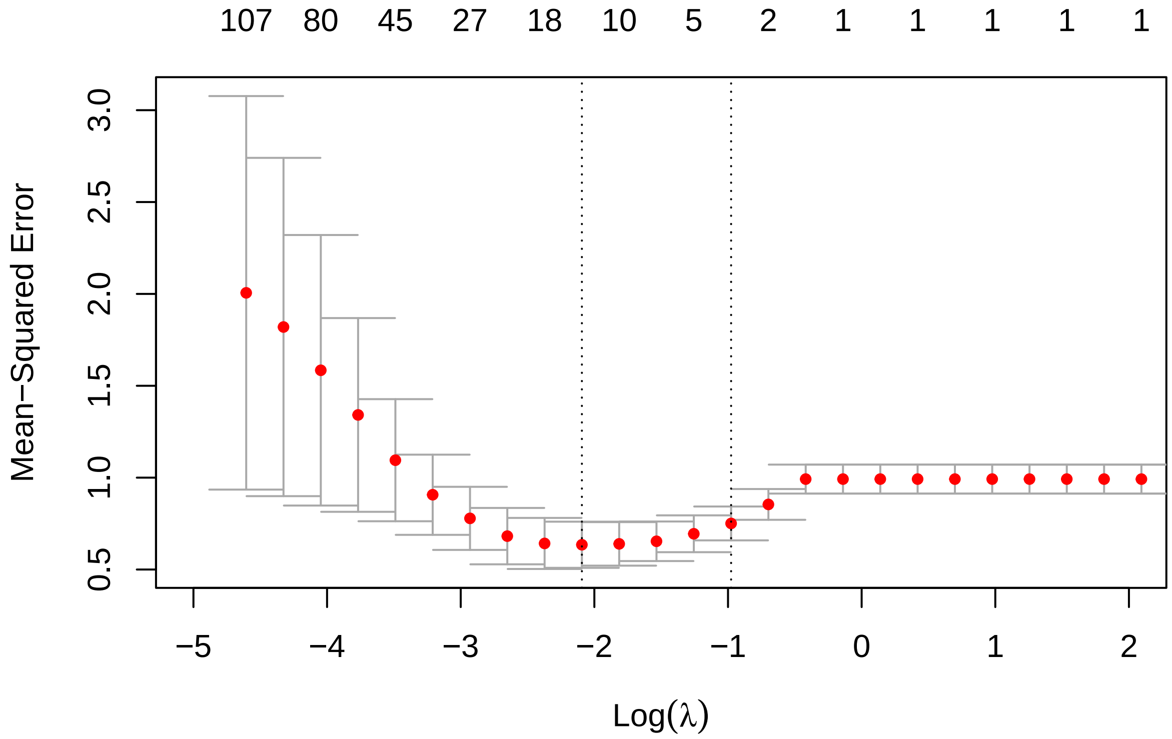
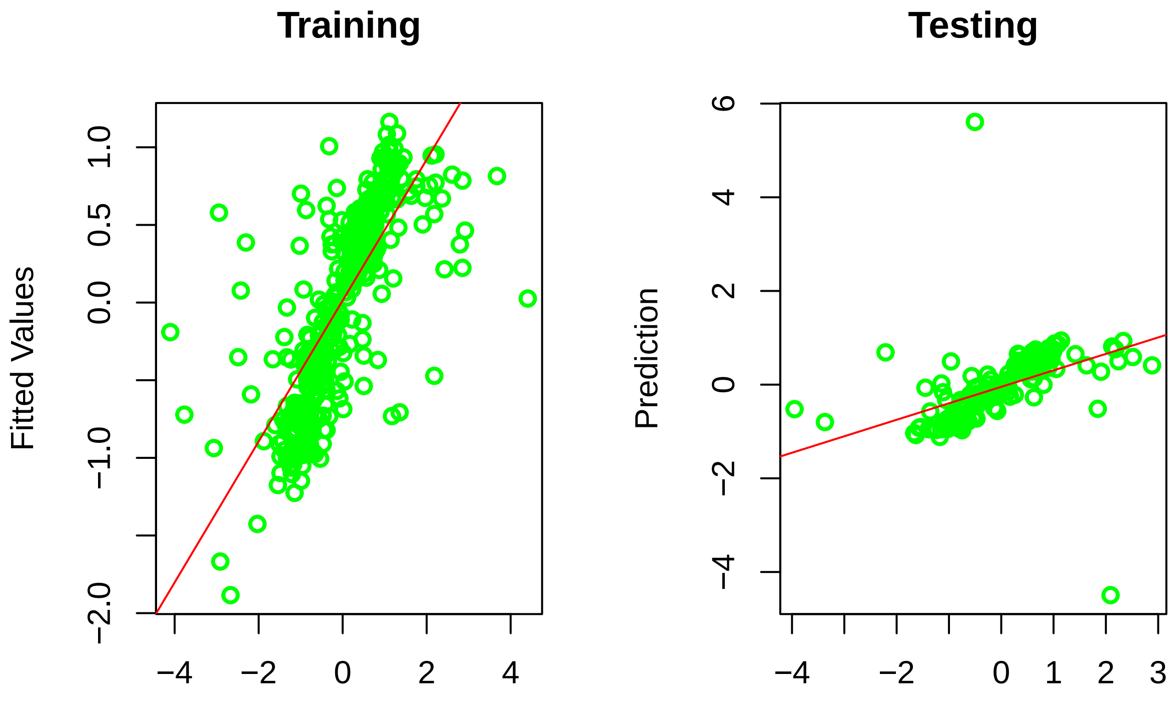
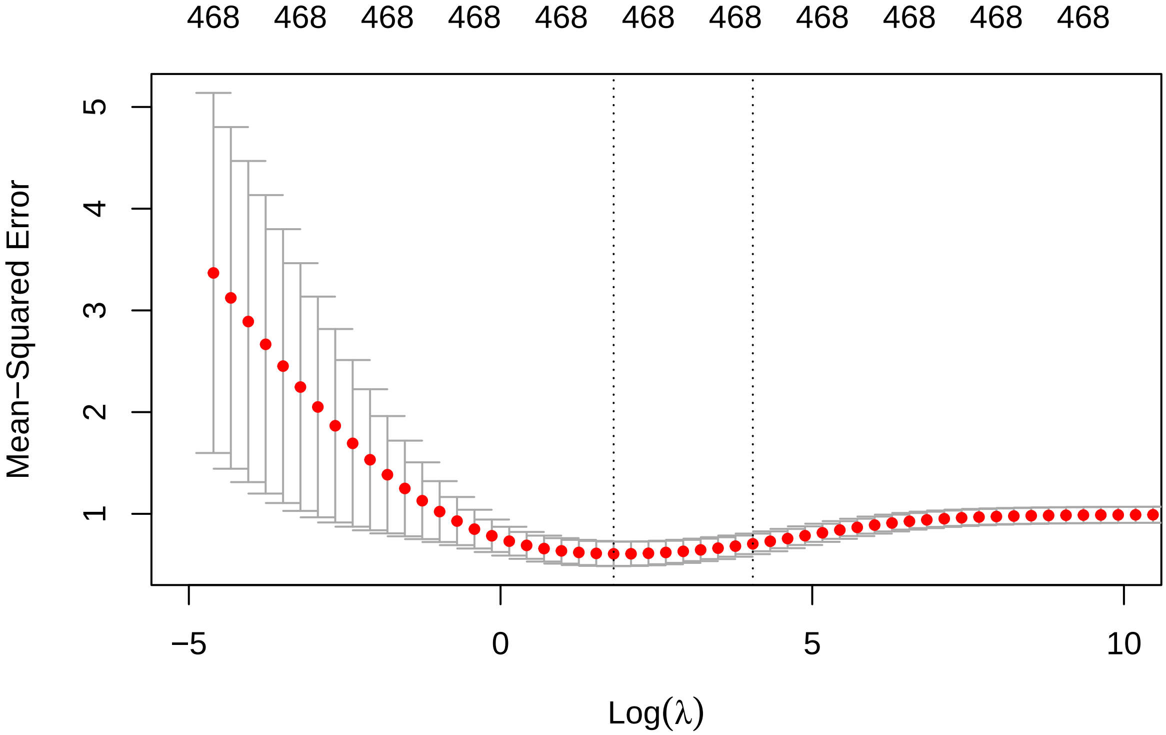


Figure 3: Cross-validation for Optimal Lambda Plot for LASSO

In the case of ridge regression, the MSE for the training set is 0.427 and for the testing set is 0.992. As is the case for ridge regression, all the 468 predictors were selected by the fitted model which makes the interpretation more difficult. The difference between the MSE for the training and testing set is smaller, which indicates that the overfitting problem has been addressed to an extent (Filzmoser P and Nordhausen K, 2021). Further, based on the test set MSE of the linear model and ridge regression model, we can say the latter has a better prediction performance. However, outliers might still have had an impact on the parameter estimation. This can also be deduced from the similar plot of the fitted and predicted model for ridge regression.



Ridge Regression



Cross−validation for Optimal Lambda Plot for Ridge Regression

Finally, the lasso regression. The lasso unlike ridge regression performs variable selection and we can expect the same in our simulation study. After fitting the model on the training set 100 times, I receive the following statistics. The lasso regression does indeed perform variable selection. The fitted model identifies only a few predictors as relevant (shown below) and sets the rest of the predictors exactly to zero, thus, lasso generates a sparse model. Also, the difference between the MSE for the training set, 0.534, and the testing set, 0.815, is further reduced. Clearly, overfitting is of the least concern in lasso among the three models. Similarly, the lasso is also found to have the least prediction error or equivalently the best prediction performance.

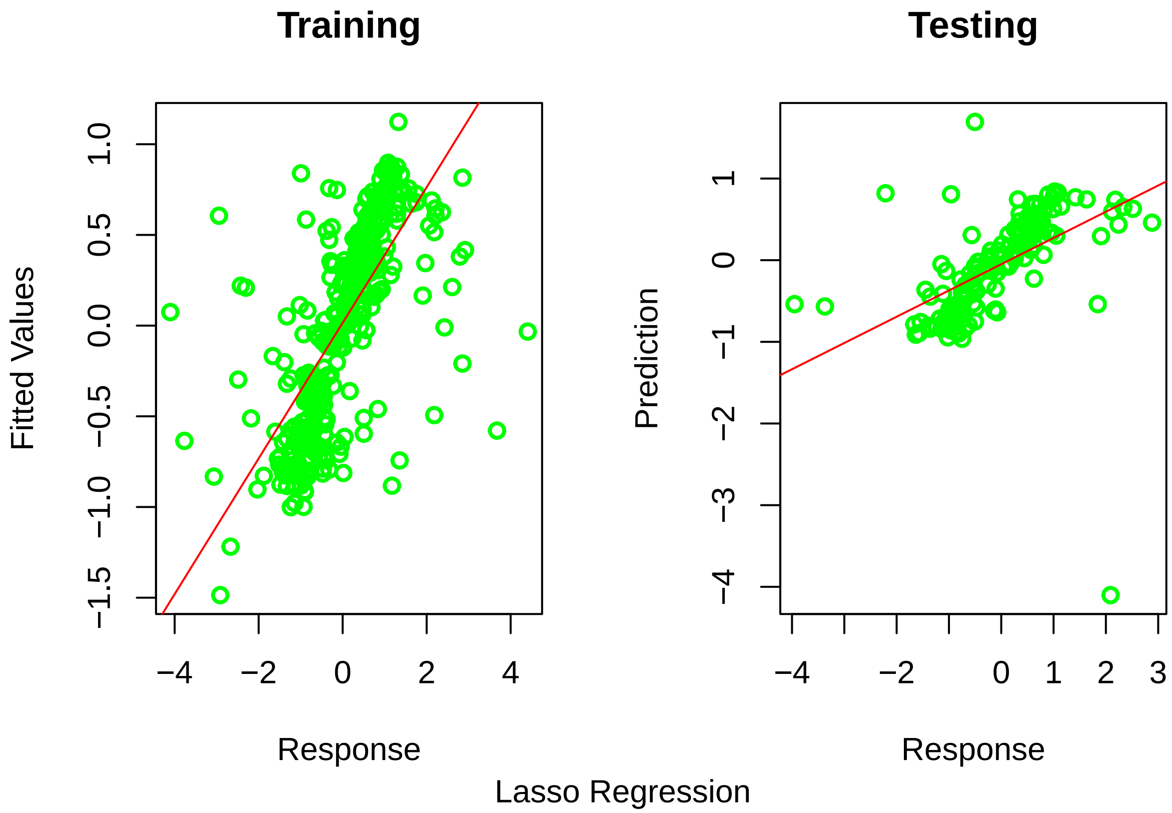


Table 1: Non-zero Predictors

	x
X38	0.0185275
X40	-0.0526812
X52	0.0109591
X53	0.0717730
X74	0.0738882
X86	0.3021361
X107	0.1329097
X228	0.0701494
X298	-0.0051368
X404	0.0833719
X442	0.0183406
X458	-0.0084899
X461	-0.0000591

The following matrix provides a review of the prediction performance of the three models.

Table 2: MSE Summary Matrix

	train_MSE	test_MSE
OLS	0.034	65.737
Ridge	0.427	0.992

	train_MSE	test_MSE
Lasso	0.534	0.815

But it must be pointed out that the results of the ridge regression and lasso are not that dissimilar. Both seem to have tackled overfitting and have good predictive power, and the lasso only seems to perform better by a slight margin. The only advantage of the lasso over the ridge is its ability to identify a small number of important predictors. Also, despite performing shrinkage over a large number of coefficients, ridge regression achieved a high predictive power, which depicts the efficiency of the method for prediction on correlated datasets. Hence, if the interpretability of the model is not of utmost significance, the simulation suggests both the lasso and ridge as efficient models in a high-dimensional setting. This analysis derived from the simulation stands in corroboration with the theory as well (Tibshirani,1996).

4 Empirical Application

For empirical application of the ridge and lasso regression, I use the high-dimensional lung cancer genomic dataset from the Chemores Cohort Study. The data titled 'Real.2' has a sample size (n) of 123 and there are 944 predictors (p) in all. More information on about the dataset can be found at Array Express, Dazard J. E. et al Github and website.

In such a high-dimensional setting (as ($p \gg n$)), the ordinary least square (OLS) linear regression model breaks down and we obtain infinitely many solutions. Therefore, due to a lack of unique solution in the linear regression model, we shift our focus to the alternatives, ridge regression and lasso. We proceed in the same manner as in the simulation study, however, with a few changes.

1. The training-testing split here is 60:40, instead of 75:25.
2. While in the simulation study, the predictions were based on 100 repetitions of a model that used the lambda sequence as defined by us and the optimal value of lambda derived using 10-fold cross-validation. In the empirical application, the predictions are made using a model that has been 10-fold cross-validated. The optimal value of lambda is calculated using the 10-fold cross-validation (CV) here as well.
3. The fitted model here has an intercept term too.

Taking these changes into consideration, I run the regressions as before. The results attained are as follows.

The ridge regression keeps all the predictions in the fitted model. However, as the value of the tuning parameter increases, the coefficients shrink further to zero. The optimal value of lambda is found to be 132.194. Shown below is the 10-fold cross-validated plot of λ for ridge regression. The description for the plot is similar to that for lasso that I have presented in the simulation study.

The MSE on training set is found to be 3.538 and the prediction error is 3.827. Overfitting seems to be correctly addressed as the error rate on both of the sets are in the same order of magnitude.

The lasso regression on the other hand performs variable selection. Shown below is the coefficient path plot for the lasso on the entire data of Real.2. Each coloured line represents a predictor in the model and tracks the change in its value for different values of λ. As the value of λ decreases (right to left), more and more coefficients enter the model. Also, the coefficient size increases with decrease in λ. The axis above the plot shows the number of predictors in the model for different values of λ. The vertical dashed line represents the final lasso model that has been selected using the optimal value of lambda derived from the 10-fold CV. The optimal value of lambda is 0.498. Based on this value and the using the entire dataset, we find that only 5 predictors and an intercept are included in the generated sparse model.

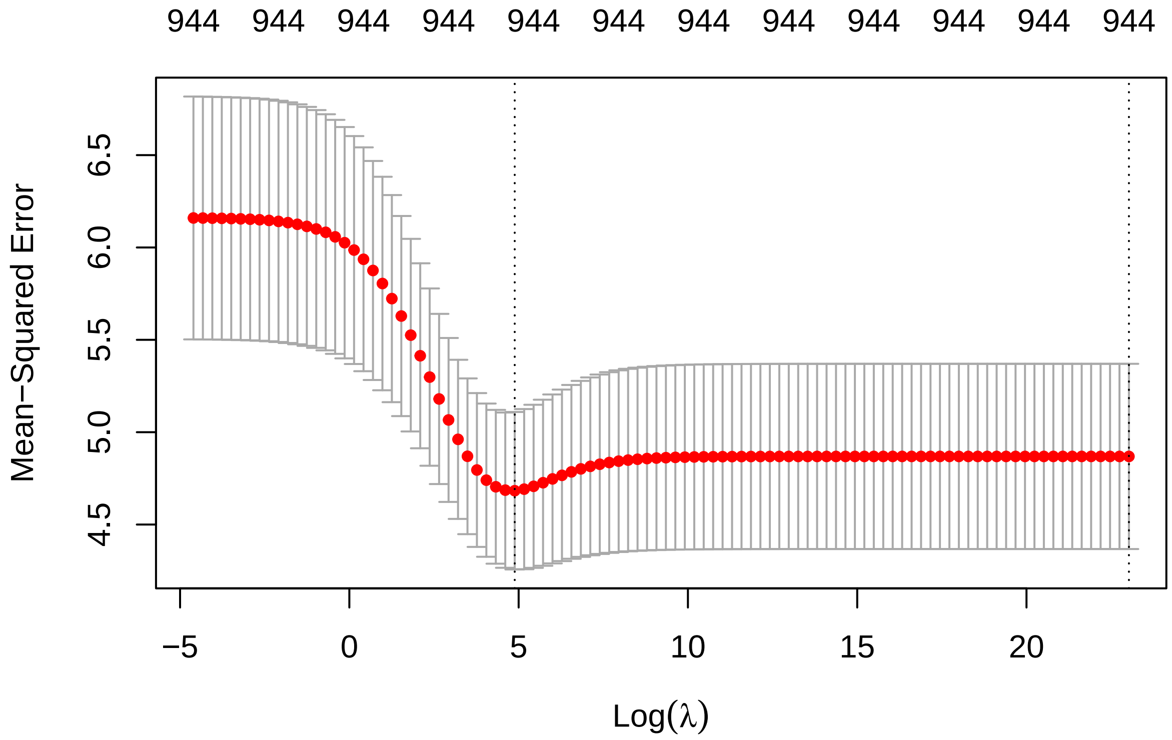
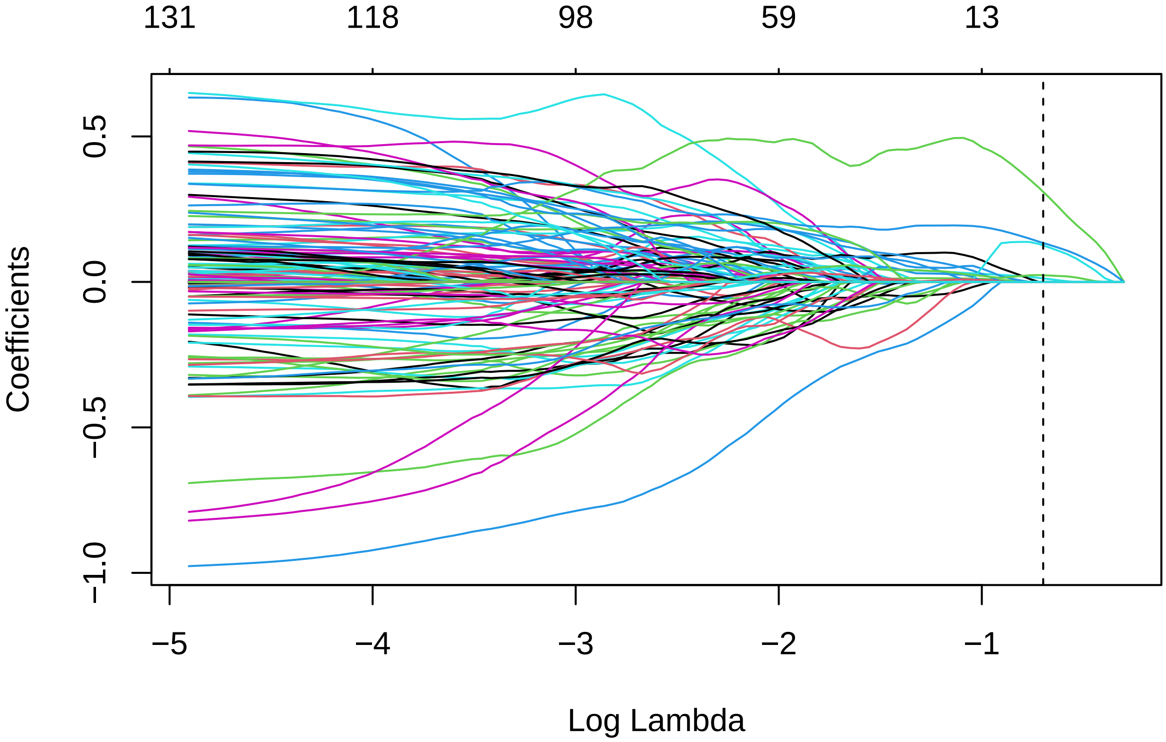


Figure 4: Cross validation for optimal lambda plot for Ridge Regression



Coefficients shrinkage path

Table 3: Non-zero Predictors

	x
(Intercept)	-0.6071818
hsa.miR.34c.5p	0.0121677
hsa.miR.564	0.3098580
hsa.miR.423.5p	0.1255423
ebv.miR.BART12	0.1339490
hsa.miR.628.3p	0.0228978

Finally, we split the data into train and test set and generate the 10-fold CV fitted model using the train set. The optimal value of lambda in this case is 0.498. The error rates are thereby calculated using the fitted model. While, the MSE on training set is 3.652, the MSE on test set is found to be 3.687.

Both the ridge regression and lasso, perform well in predicting the model as seen by their very similar prediction error rate and their ability to address overfitting. This result is clearly in line with our result from the simulation study and the existing literature. Therefore, the only added advantage that can be deemed in favour of lasso is its ease of interpretability. However, while considering the sparse model, we should keep the issues with generating a sparse model using lasso in a high-dimensional setting, as discussed before in mind.

5 Conclusion

As data becomes more complex, the dependence on least squares linear regression needs to be reduced. This is primarily because of the inability of the least squares model to perform efficiently and generate reliable results. In a high-dimension setting, which has now become prevalent than ever, such a model simply breaks down and is unable to provide a unique solution. Ridge regression and LASSO have come up as the plausible alternative shrinkage techniques to develop models in a high-dimensional data. While, prediction accuracy has been the primary aim of ridge regression, the lasso in addition to that also performs variable selection. This is helpful as it generates a sparse model with only relevant variables out of a large number of variables in the dataset whose significance in predicting the response variable is not known beforehand. Therefore, lasso brings forth an ease of interpretability of the model unlike the ridge regression wherein all the predictors are taken as relevant to the response variable and involved in the model.

The simulation study and empirical application were performed on the R statistical software, with the primary use of glmnet() package. 10-fold CV was used to calculate optimal value of lambda as prescribed in the literature. The results from both the analysis were in accordance to the theory. The ridge regression and lasso, both were found to efficient in predicting the model, however, the interpretability of the former is a big drawback for it. On the other hand, although lasso produced sparse, easily interpretable models, but a major concern that came up with it was the possibility of the lasso to not include all relevant non-zero predictors in the model when performed on a high-dimensional data.

As more focus is being given to the two techniques, alternatives like elastic net regression in shrinkage technique and Principal Components Regression (PCR) in dimension reduction methods have gained relevance as well. Finally, this study can be extended to examine the performance of aforementioned alternatives in linear and logistic regression models for instance.

6 References

*Algamal, Z. Y., Lee M. H. (2015). Applying Penalized Binary Logistic Regression with Correlation Based Elastic Net for Variables Selection. Journal of Modern Applied Statistical Methods. 14(1), 168-179. DOI: 10.22237/jmasm/1430453640

*Altelbany, S. (2021). Evaluation of Ridge, Elastic Net and Lasso Regression Methods in Precedence of Multicollinearity Problem: A Simulation Study. Journal of Applied Economics and Business Studies, 5(1), 131-142. https://doi.org/10.34260/jaebs.517

*Chan, J. Y.-L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z.-W., & Chen, Y.-L. (2022). Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review. Mathematics, 10(8), 1283. MDPI AG. Retrieved from http://dx.doi.org/10.3390/math10081283

*Dazard, J.-E., Rao, J. S., LeBlanc, M., Choe, M., & Duong, T. Bump hunting by patient rule induction method. PRIMsrc. Retrieved from http://www.primsrc.com/

*Dazard, J.-E. Jedazard/PRIMSRC: Bump hunting by patient rule induction method for survival, regression and classification in a multivariate setting and in high-dimensional data. GitHub. Retrieved from https://github.com/jedazard/PRIMsrc

*Embl-Ebi. Array express. EBI. http://www.ebi.ac.uk/arrayexpress/

*Filzmoser, P, Nordhausen, K. (2021). Robust linear regression for high-dimensional data: An overview. WIREs Comput Stat. 13:e1524. https://doi.org/10.1002/wics.1524

*Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67. DOI: 10.1080/00401706.1970.10488634

*James, G., Witten, D., Hastie, T. J., & Tibshirani, R. J. (2021). An introduction to statistical learning: With applications in R. Springer. https://doi.org/10.1007/978-1-4614-7138-7

*Ranstam, J., Cook J. A. (2018). LASSO regression. British Journal of Surgery, 105(10), Page 1348, https://doi.org/10.1002/bjs.10895

*Sirimongkolkasem, T., Drikvandi, R. (2019). On Regularisation Methods for Analysis of High Dimensional Data. Ann. Data. Sci. 6, 737–763. https://doi.org/10.1007/s40745-019-00209-4

*Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1), 267–288. http://www.jstor.org/stable/2346178

*Zhou, D. X. (2013). On grouping effect of elastic net. Statistics and Probability Letters, 83(9), 2108-2112. https://econpapers.repec.org/scripts/redir.pf?u=https%3A%2F%2Fdoi.org%2F10.1016%252Fj.spl.2013.05.014;h=repec:eee:stapro:v:83:y:2013:i:9:p:2108-2112