Statistics 159 Project 2 Report

Aoyi Shan, Yukun He

10/28/2016

Abstract

In this project, we explore the difference between various muitiple regression methods and conclude on what approach should be used in order to fit our data better.

Introduction

In this project, our objective is to apply model selection methods introduced in Chapter 6, Linear Model Selection and Regularization, from the book "An Introduction to Statistical Learning" by Gareth James, Deniela Witten, Trevor Hastie and Robert Tibshirani. In our analysis, we compare the fit of models generated by 5 different regression methods, regression model obtained by Ordinary Least Squares, Ridge regression, Lasso regression, Principal Components regression and Partial Least Squares regression. To evaluate how well each model fits the data, we perform 10-fold cross validation in the model construction process and select the model with minimum cross-validation errors in terms of the running parameter. The requirement for this project can be found at https://github.com/ucb-stat159/stat159-fall-2016/blob/master/projects/proj02/proj02-predictive-modeling.pdf.

Data

The data we used in this analysis can be downloaded from http://www-bcf.usc.edu/~gareth/ISL/Credit.csv, which is provided by the textbook we refer to throughout the project, An Introduction to Statistical Learning.

After downloading the original data, in order to have comparable scales, we first standardize the data by centering around means and dividing by their respective standard deviation. Then, since we will use cross validation to improve the accuracy of our fit, we divide the data into training set and test set. We have 400 observations in total, so we randomly select 300 entries to be in the train set and the remaining 100 in the test set.

Methods

Least Squares Method

OLS estimators are considered as our base case in this project. OLS estimators are obtained by minimizing the sum of squares $RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}(\beta_j x_{ij}))^2$ where $\beta_0, ..., \beta_p$ are our coefficient estimates. This is the most common regression method, and it works the best when we have homoskedastic and uncorrelated errors within the data.

Shrinkage Method

Similar to the OLS method, shrinkage method also fits a model containg all independent variables but with a focus on constrains the coefficient estimates toward 0. By shrinking the coefficient estimates, the variance in estimators is significantly reduced compared to that of OLS.

Ridge Regression

Instead of minimizing RSS, Ridge regression minimizes $RSS + \lambda \sum bj^2$. This regression abandons the requirement of an unbiased estimator in order to obtain a more precise prediction intervals. The addition term is a shrinkage penalty and since it's a sum of coefficient estimators, it will be small when βs are all close to 0. Here, λ is called a tuning parameter and it controls how much RSS and shrinkage panelty affects the regression model. When $\lambda = 0$, ridge regression is the same as the least squares models. Becuase each λ corresponds to a different set of coefficient estimates, we usually try a wide range of λs and pick the best tuning parameter with the smallest MSE.

One of the main feature of ridge regression is the bias-variance trade-off. When $\lambda = 0$, which is the least square model, the variance is high the estimators are unbiased. As lambda increases, variance decreases significantly but bias only sightly increase. Since $MSE = variance + squaredbias$ along with the relative effect of the change, when λ is smaller than 10, the variance drops rapidly, with very little increase in bias, so MSE decreases.

Lasso Regression

The major difference between Lasso and Ridge is that Lasso method aims to minimize $RSS + \lambda \sum |\beta_j|$. This is to compensate for the fact that ridge regression with always generate a model with all predictors since it doesn't involve a process of variable reduction. Therefore, lasso improves on this by allowing some of the coefficient estimates to be exactly 0 if the tunning parameter is sufficiently large. Therefore, this variable selection process makes the model much easier to interpret with a reduced amount of predictors.

Comparison between Ridge and Lasso

In general, lasso is expected to outperform ridge when we have a small amount of predictors having significant coefficients, and the rest close to 0. Ridge regression will perform between when the response is affected by many regressors with equal-sized coefficients.

Dimension Reduction Methods

Dimension reduction method involves a transformation of variables before we fit in the least squares model. Instead of estimating $p + 1$ coefficients $\beta_0, ..., \beta_p$ when we have p regressors, it transform the data to $Z1, ..., Z_m, M < p$ where Z_m represent linear combinations of the original predictors, so that we only need to estimate $M + 1$ coefficients $\theta_0, \theta_1, ..., \theta_M$.

Principal Components Regression

In principal components regression, we first perform privipal components analysis on the original data which constructs M principal components, $Z_1, ..., Z_m$. Then, we use these components as the predictors and fit the least squares model to obtain coefficient estimates. The key assumption we hold in this regression model is that the directions in which $X_1, ..., X_p$ show the most variation are the directions that are associated with Y. If the assumption holds, then fitting a least squares model to $Z_1, ..., Z_m$ will lead to better results than fitting a least squares model to $X_1, ..., X_p$, since most or all of the information in the data that relates to the

response is contained in $Z_1, ..., Z_m$. With fewer predictors, we can also reduce the risk of overfitting. PCR performs the best when the first few pricipal components are sufficient to capture most of the variation in the predictors and their relationship with the response.

Partial Least Squares

In comparison to PCR regression, which uses unsupervised method to dentify the principal components, partial least squares method takes the advantage of a supervised learning process. It uses the response variable Y to identify new vectors that are not only similar to existing regressors, but also select the ones that are related to the response. Therefore, as indicated in the textbook, the PLS approach attempts to find directions that help explain both the response and the predictors.

Analysis

Exploratory Data Analysis (EDA)

The first step of conducting analysis is to understand the data by conducting exploratory data analysis. To conduct the EDA, we obtained descriptive statistics and summaries of all variables by using the R package "FactoMineR". For the quantitative variables, we wrote a function called output_quantitative_stats() to get minimum, maximum, range, median, first and third quartiles, IQR, Mean and Sd of all the quantitative variables including "Income", "Limit", "Rating", "Cards", "Age", "Education", and "Balance". Similarly, we wrote a function called output_qualitative_stats() to generate a table with both the frequency and the relative frequency of the qualitative variables including "Gender", "Student", "Married", and "Ethnicity". To understand the data better, we also want to generate some plots to visualize the data. We wrote the functions histogram_generator() and boxplot_generator() to generate histograms and boxplots of the quantitative variables and condition_boxplot_generator() to generate conditional boxplots between "Balance" and the qualitative variables. To study the association between "Balance" and the rest of predictors, we also obtained the correlation matrix of all quantitative variables using function cor(), the scatterplot matrix using function pairs(), the anova between "Balance" and all the qualitaive variables using function aov(), and the PCA plot for the quantitative variables using function PCA().

Pre-modeling Data Processing

The second step before conducting analysis is to process the raw data into the data to be used in the further analysis. We used function model.matrix() to transform each categorical variable into dummy variables and function scale() to mean-center and standardize the data. We saved the processed data as scaled-credit.csv and we used the processed data for the model building process.

Regression Models

The third step is to build and evaluate a model on the processed data of size 400. We used sample() function to get a training set of size 300 to build the model and a test set of size 100 to assess the performance. For reproducibility purpose, we used function set.seed() before taking the sample.

After getting the training set and test set, we fitted a multiple linear regression model via Ordinary Least Squares (OLS) using the lm() function and set this as a benchmark. In addition to the OLS model, we used other four regression methods including Ridge regression (RR), Lasso regression (LR), Principal Components regression (PCR) and Partial Least Squares regression (PLSR) to fit the data set. To cooperate on the model building process and to make the workflow reproducible and efficient, we created a new feature branch for each of the four regression approaches. Each team member worked on different regression models on different

branches and then merged them into the master branch. We have four feature branches named 'ridge', 'lasso', 'pcr' and 'plsr' and each member contributed to two branches.

For ridge regression method and lasso regression method, we used R package "glmnet" and we used cv.glmnet() function to conduct ten-fold cross-validation on the train set. When we applied the cv.glmnet() function, we set alpha = 0 for ridge regression and alpha = 1 for lasso regression and we set intercept = FALSE and standardize = FALSE for both regressions since we already mean-centered and standardized all the variables. After conducting ten-fold cross-validation on the train set, we got the best lambda, which is the smallest lambda for each regression. We then used the best fitted lambda on the test set using function predict() and calculated the test MSE. We will use the test MSE to compare the performance of all the models later. Last, we used function glmnet() to refit the model on the full data set using the best lambda chosen by cross-validation and got the "official" coefficient estimates by function coef().

For pcr regression method and plsr regression method, we used R package "pls". We used function pcr() with the argument validation = "CV" to perform 10-fold cross-validation for pcr regression and we used function plsr() with the argument validation = "CV" to perform 10-fold cross-validation for plsr regression. After conducting cross-validation on the train set, we got the best m that minimizes the validation error and we plotted the cross-validation errors using the function validationplot() with argument val.type = "MSEP". We then used the best fitted m on the test set using function predict() and calculated the test MSE. Finally, we used function pcr() and plsr() for each regression method to refit the model on the full data set using the best m chosen by cross-validation and got the coefficient estimates.

After we finished the model building process, we saved all the outputs and graphs for further comparison and analysis and merged the feature branches with the "master" branch. To compare five regression methods and choose the best one, we used R packages "Matrix" and "xtable" to get a table of regression coefficients for all five methods with each column per regression method. We also included another table with the test MSE values for all regression methods. To visualize and compare the coefficients, we generated a plot of trend lines of the coefficients of five regression methods.

Results

OLS Regression

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.00124	0.01226	-0.10155	0.91919
Income	-0.59637	0.02016	-29.58094	0.00000
Limit	0.95449	0.18629	5.12379	0.00000
Rating	0.38309	0.18728	2.04550	0.04171
Cards	0.05100	0.01436	3.55029	0.00045
Age	-0.02592	0.01276	-2.03223	0.04305
Education	-0.01469	0.01242	-1.18310	0.23775
GenderFemale	-0.02197	0.01231	-1.78433	0.07542
StudentYes	0.27607	0.01273	21.68359	0.00000
MarriedYes	-0.01706	0.01266	-1.34767	0.17882
EthnicityAsian	0.00548	0.01512	0.36270	0.71710
EthnicityCaucasian	-0.00474	0.01494	-0.31721	0.75131

Table 1: OLS Coefficients

Since OLS would be the base case in this project, we also use 10-fold cross validation in building the OLS model by first using the train set to fit the model, calculate MSE in the test set and then fit the model we select to the entire data set. From the OLS regression output, we noticed some coefficients come with a big p-value, which means that they are not statistically significant. Therefore, we can conclude that the constant

term, Education, Gender, Marital Status and Ethinicity don't belong to this regression. Also we noticed that among the statistically significant regressors, some has very small coefficients, so the main factors influencing Balance are Income, Limit and Rating.

Ridge Regression

	Estimate
(Intercept)	0.00000
Income	-0.56871
Limit	0.71866
Rating	0.59306
Cards	0.04425
Age	-0.02538
Education	-0.00588
GenderFemale	-0.01068
StudentYes	0.27318
MarriedYes	-0.01103
EthnicityAsian	0.01638
EthnicityCaucasian	0.01101

Table 2: Ridge Coefficients

With Ridge regression, the λ we found that results in the smallest validation error is $\lambda = 0.01$. As we have a relatively small λ, we expect to see that the estimation with ridge is very similar to that of OLS but a little bit smaller due to the shrinkage effect.

Lasso Regression

	Estimate
(Intercept)	0.00000
Income	-0.55166
Limit	0.92505
Rating	0.36787
Cards	0.04500
Age	-0.01666
Education	0.00000
GenderFemale	0.00000
StudentYes	0.26681
MarriedYes	0.00000
EthnicityAsian	0.00000
EthnicityCaucasian	0.00000

Table 3: Lasso Coefficients

Lasso improves on Ridge because it adds the incentive to render statistically insignificant estimates to 0 by performing both variable selection and yields model that involves only a subset of the variables. In this case, the λ we found that results in the smallest validation error is $\lambda = 0.01$. We can see that a significant improvement is that we have a number of regressors with a coefficient of 0 which makes the intepretation much easier.

PCR Regression

	Estimate
Income	-0.59887
Limit	0.67141
Rating	0.67064
Cards	0.04043
Age	-0.02327
Education	-0.00600
GenderFemale	-0.01165
StudentYes	0.27636
MarriedYes	-0.01116
EthnicityAsian	0.01741
EthnicityCaucasian	0.01119

Table 4: PCR Coefficients

With PCR, we mainly focus on dimension reduction by unsupervised learning. By comparing validation errors for different Ms, we decide that the best M to use here is 10. Since we only have 11 predictors in this multiple regression, it is not a huge change, so that explains why PCR is very close to OLS. Therefore, we can say that the dimension of predictors for this regression almost cannot be reduced, there doesn't exist major principal components dominating the change in the response variable, so PCR doesn't help to improve that much on the OLS estimation.

PLS Regression

	Estimate
Income	-0.59801
Limit	0.67458
Rating	0.66699
Cards	0.04808
Age	-0.02592
Education	-0.00908
GenderFemale	-0.01671
StudentYes	0.27394
MarriedYes	-0.00800
EthnicityAsian	0.02257
EthnicityCaucasian	0.01101

Table 5: PLS Coefficients

Similarly, PLS regression is also trying to reduce the dimension of predictors, but in an supervised way. By comparing validation errors for different Ms, we decide that the best M to use here is 4. Since we originally have 11 predictors in this multiple regression, this is a huge improvement, so this method successfully find the major principal components dominating the change in Y, and therefore reduce the risk of overfitting and therefore obtain a better fit.

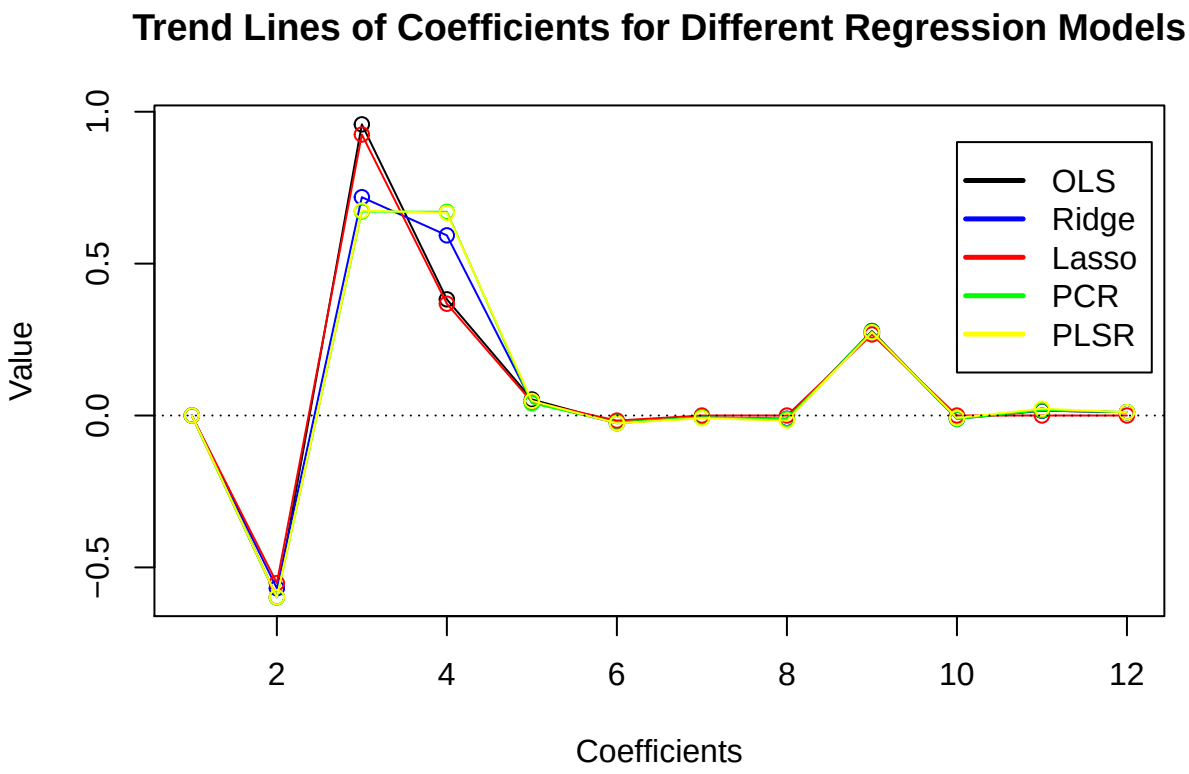
Combined All Coefficients Together

	ols	ridge	lasso	pcr	plsr
(Intercept)	0.00000	0.00000	0.00000	0.00000	0.00000
Income	-0.59817	-0.56871	-0.55166	-0.59887	-0.59801
Limit	0.95844	0.71866	0.92505	0.67141	0.67458
Rating	0.38248	0.59306	0.36787	0.67064	0.66699
Cards	0.05286	0.04425	0.04500	0.04043	0.04808
Age	-0.02303	-0.02538	-0.01666	-0.02327	-0.02592
Education	-0.00747	-0.00588	0.00000	-0.00600	-0.00908
GenderFemale	-0.01159	-0.01068	0.00000	-0.01165	-0.01671
StudentYes	0.27815	0.27318	0.26681	0.27636	0.27394
MarriedYes	-0.00905	-0.01103	0.00000	-0.01116	-0.00800
EthnicityAsian	0.01595	0.01638	0.00000	0.01741	0.02257
EthnicityCaucasian	0.01101	0.01101	0.00000	0.01119	0.01101

Table 6: Regression Coefficients for 5 Regression Methods

	MSE
ols	0.05179
ridge	0.05259
lasso	0.05154
pcr	0.05200
plsr	0.05192

Table 7: MSE of 5 Regression Methods



Conclusion

Combining the coefficients from all five regression models, we notice that only the coefficients for Limit, Rating and StudentYes varies and a number of regressors have coefficients close to 0. As we explained above, each model has its own feature and can lead to the optimal regression model under different circumstances. Here, for the data set we work on in this project, both our analysis and MSE shows that Partial Least Squares Regression model fits the best and leads to the most accurate prediction.