HW 2: Simple Regression Analysis

Bryan Alcorn & Todd Vogel

10/6/2016

Contents

Abstract

The following reports looks into data on credit cards across multiple qualitative and quantitative variables. We utilize 5 types of linear modelling, including OLS, Lasso, Ridge, PLSR, and PCR to develop the most effective predictive model for credit card balance based on the existing predictors. Then, by comparing the MSE of each model between the actual and predicted response we determine the model with the most predictive power (that with the lowest MSE). In the end, we found that one model proved to be the best predictor of credit card balance.

Introduction

This project looks to determine the individual effects (coefficients) of credit predictors on the response variable, balance. By focusing on these attributes and applying various models we can determine the relationships between variables amd, thus, effectively predict the response. We will apply 4 linear models (Lasso, Ridge, PLSR, and PCR) and see how they compare to OLS (simple linear regression). If the models prove to be effective they can be applied to future credit card data to estimate balance numbers.

Data

The data utilized in this project is exclusively a Credit dataset with 11 variables. Of these 11 variables 4 are qualitative (`Gender`, `Student`, `Married`, and `Ethnicity`) and 7 and quantitative (`Income`, `Limit`, `Rating`, `Cards`, `Age`, `Education`, and `Balance`). The response variable here is `Balance` (dependent) and the rest are predictors. Each data point, or row, represents and individual consumer (of which there are 400). However, the data was trained on a training data set of 300 points and tested on a test data set of 100 points taken randomly from the original Credit.csv dataframe. To make the data more usable the qualitative variables were converted into indicator values and then each variable was mean centered and standardized so they would have comparable scales.

Methods

This project involved the use and implementation of 5 different linear models. The shrinkage and dimension-reduction modelling methods used are found in Chapter 6 of the book **Linear Model Selection and Regularization**. The traditional ordinary least squares model was found in Chapter 3 of that book.

Ordinary Least Squares

Here we determined the basic linear relationship between variables (`Balance` is response, and all others are predictors). In order to find these relationships we had to fit the variables to the following model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

To do so we utilized the `lm()` function to solve for the coefficient values β_0, β_1, β_2, to β_n. Now, once we had determined a model for the training data, we applied it to the test data to determine a predicted response. From there we calculated the mean squared error (a measure of predicted power), by using the function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{di})^2$$

This subtracts each predicted response from the actual response and squares it then sums up each of the points and takes the average.

Ridge Regression

Ridge Regression is the first of the two shrinkage modelling methods we used. When using shrinkage methods the goal is to penalize certain parameters that should have a less significant effect on the model. To do so we use the tuning parameter, λ, times $\sum_{j=1}^{p}\beta_j^2$ to yeild the shrinkage penalty. We then determine the coefficients that minimize the following equation:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}) + \lambda\sum_{j=1}^{p}\beta_j^2$$

To find these coefficients we used the function `cv.glmnet()` to determine through cross validation which λ value minimizes the above function (when λ was a given sequence of numbers under `grid`, and `alpha` was set to 0). After the coefficients were calculated we again calculated the MSE to determine the predictive power of our model.

Lasso Regression

Lasso regression is the second and final shrinkage modelling method used in this project. Although it is very similar to ridge regression there are a few key differences: the shrinkage penalty is now λ, times $\sum_{j=1}^{p} \beta_j^2$ (β is not squared), and lasso allows for the removal of certain variables (not just dampening their effect). To determine the coefficients we must look for the βs that minimize:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

Again, to find the coefficient we used `cv.glmnet()` and set λ to `grid`. However, now, `alpha` was set to 1. Finally, we calculated MSE to determine the predictive power of our model.

Principle Components Regression

PCR is the first of two dimension reduction modelling methods we used. This method labors under the assumption that a subset of all of the predictor variables account for the vast majority of the variance. These more significant variables are referred to as principle components (M). PCR works by setting M equal to some reduced number of variables and running cross validation on, the model with the lowest cross validation error is selected.

To develop a model through PCR we used the `pcr()` function and set `validation`= CV. We then found the model in which PRESS was larger to avoid overfitting. Finally, we calculated MSE to determine the predictive power of our model.

Partial Least Squares Regression

PLSR is the second and final dimension reduction modelling method used in this project. PLSR is very similar to PCR in that it looks at a subset of predictors, fits a linear model to those M variables, and determines the best dimension reduced model. However, unlike PCR, PLSR is a supervised alternative. This means that PLSR uses the response, Y, to determine if new features are good approximations and whether they are related to response.

To develop a model through PLSR we used the `plsr()` function and, like with PCR, set `validation`= CV. Again, we found the model in which PRESS was larger. Finally, we calculated MSE to determine the predictive power of our model.

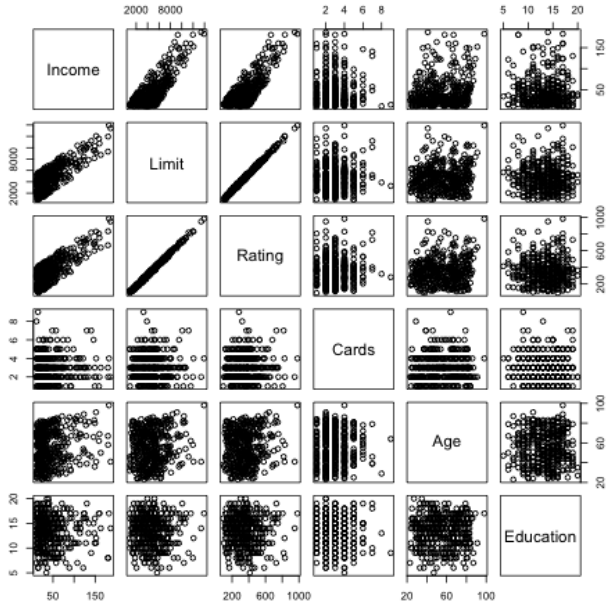
Analysis

In the Results section of the report, we determined the mean squared error values for each of our 5 models. For OLS, Ridge, Lasso, PCR, and PLSR the MSE's were 0.047, 0.045, 0.047, 0.415, and 0.308 respectively. Because our data was mean centered and standardized, all of these values exist on the same scale and can thus be prepared. Mean squared error values represent the average sum of the errors between actual and predicted response values. Therefore, the smaller the MSE value the smaller the error, and the better the predictive model. Given this, 0.045 is our smallest MSE value meaning Ridge is our best predictive model.

Lets take a look at some of the results of our analysis and see what some of the data looks like

Scatterplot between all the variables

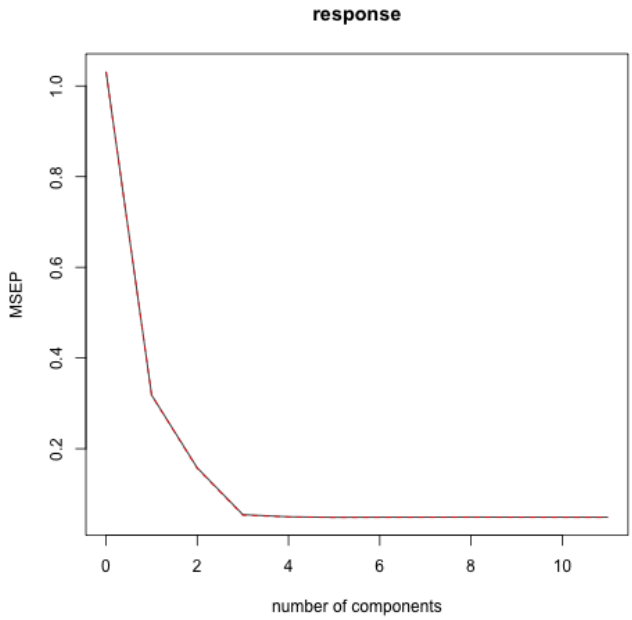
Scatterplot Matrix



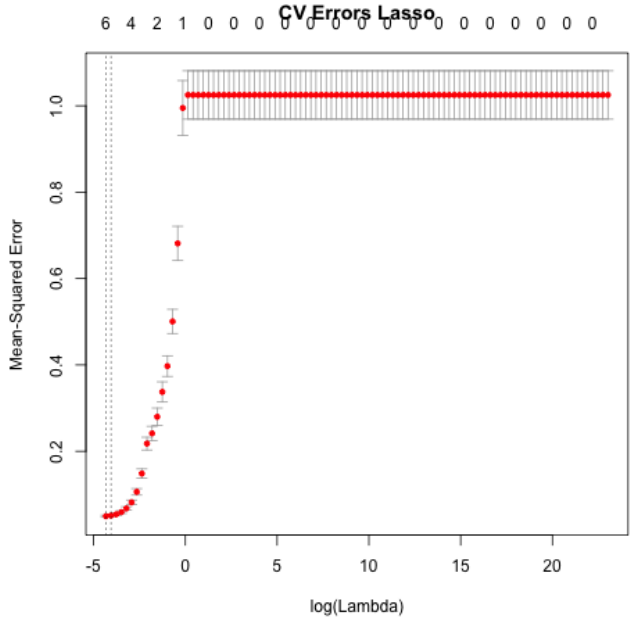
Correlation | Matrix

```
##                Income       Limit      Rating       Cards         Age
## Income     1.00000000  0.79208834  0.79137763 -0.01827261 0.175338403
## Limit      0.79208834  1.00000000  0.99687974  0.01023133 0.100887922
## Rating     0.79137763  0.99687974  1.00000000  0.05323903 0.103164996
## Cards     -0.01827261  0.01023133  0.05323903  1.00000000 0.042948288
## Age        0.17533840  0.10088792  0.10316500  0.04294829 1.000000000
## Education -0.02769198 -0.02354853 -0.03013563 -0.05108422 0.003619285
##               Education
## Income    -0.027691982
## Limit     -0.023548534
## Rating    -0.030135627
## Cards     -0.051084217
## Age        0.003619285
## Education  1.000000000
```

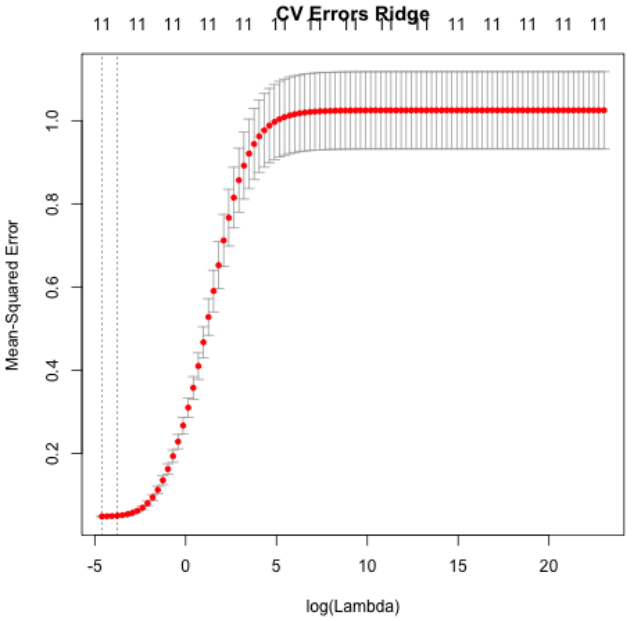
MSE Errors PLSR



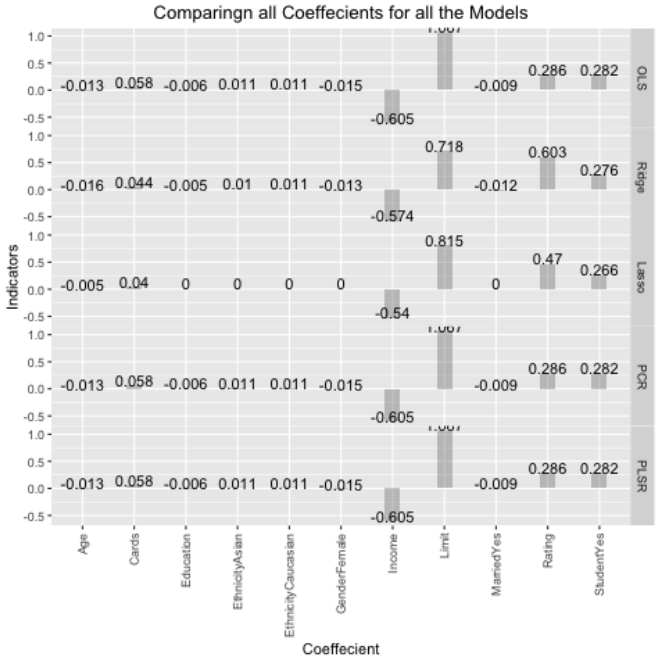
MSE Errors for Lasso



MSE Errors for Ridge



Comparing all Indicators accross models



Takeaways:

Strongest Predictors 1. Limit and rating seem to have a strong linear relationship 2. Also linear are limit and income and rating and income.

PCR vs PLSR, deciding which model is best * As the nunmber of components for PCR increase the

MSE decreases * For PLSR, however, there is a quick drop off after about 4 components, the best will be somewhere after that

Lasso vs Ridge, deciding which model is best * Lambda values for these two models are best before 0

Results

After forming the aforementioned regression models we found 12 coefficients that represent the best fit for each model. The resulting predictive function looks like:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_9 X_9 + \beta_{10} X_{10} + \beta_{11} X_{11}$$

Table 1: Information about Model Coefficients

	Variables	OLS	Ridge	Lasso	PCR	PLSR
1	Intercept	-0.003	0.000	0.000	0.000	0.000
2	Income	-0.605	-0.574	-0.540	-0.605	-0.605
3	Limit	1.067	0.718	0.815	1.067	1.067
4	Rating	0.286	0.603	0.470	0.286	0.286
5	Cards	0.058	0.044	0.040	0.058	0.058
6	Age	-0.013	-0.016	-0.005	-0.013	-0.013
7	Education	-0.006	-0.005	0.000	-0.006	-0.006
8	GenderFemale	-0.015	-0.013	0.000	-0.015	-0.015
9	StudentYes	0.282	0.276	0.266	0.282	0.282
10	MarriedYes	-0.009	-0.012	0.000	-0.009	-0.009
11	EthnicityAsian	0.011	0.010	0.000	0.011	0.011
12	EthnicityCaucasian	0.011	0.011	0.000	0.011	0.011

This table presents the fit between the prediction variables and the response variable (`Balance`) determined from the `cv.glmnet()`, `pcr()`, and `plsr()` functions, for each of our 5 models (OLS, Ridge, Lasso, PCR, PLSR).

After finding the 5 predictive functions, we found the MSE's for each model:

Table 2: Information about Mean Squared Errors

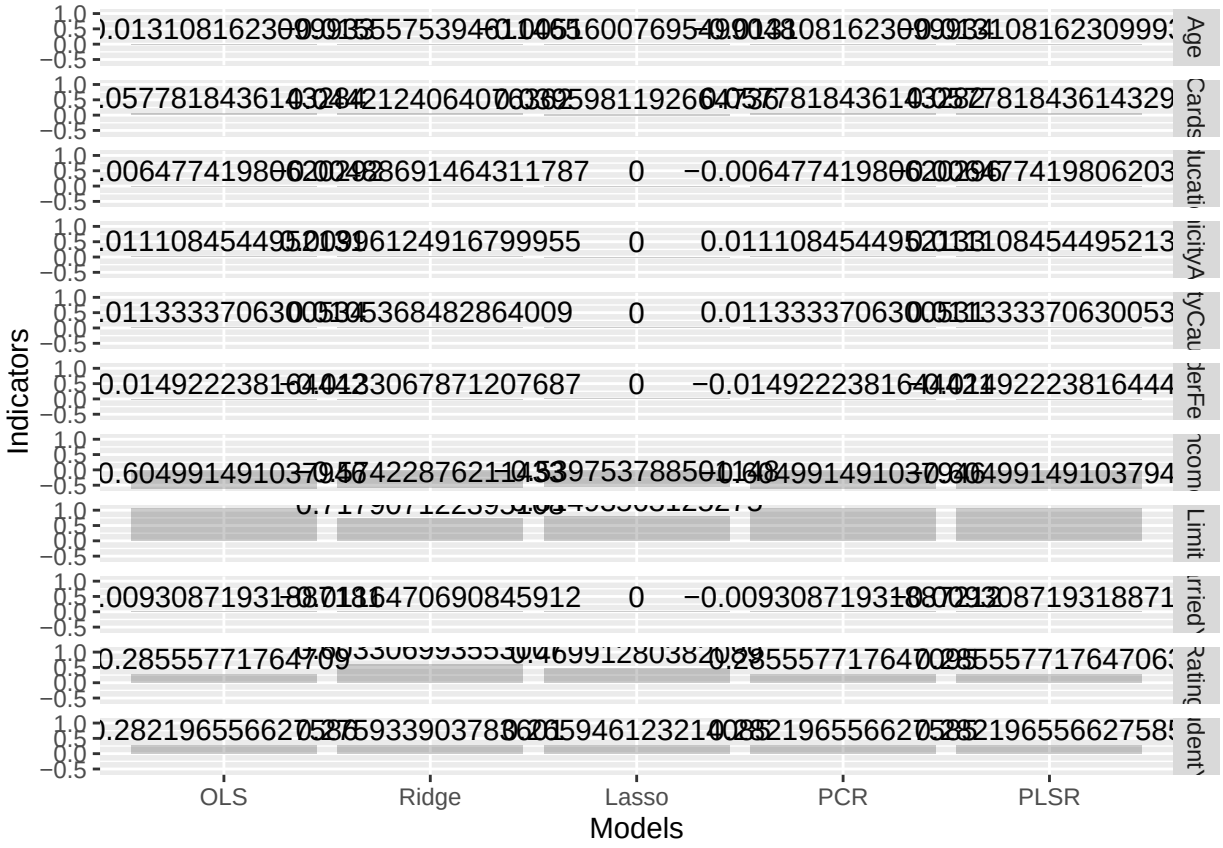
	Model	MSE
1	OLS	0.047
2	Ridge	0.045
3	Lasso	0.047
4	PCR	0.047
5	PLSR	0.047

Analysis of these numbers will reveal which model has the most predictive power.

Comparing Coefficients

```
## Warning: package 'reshape2' was built under R version 3.2.5
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```



Conclusion

With the analysis it is now clear that the Ridge regression provides the best predictive model of the Credit data. Now we can accurately apply new data to the model in order to predict an individuals credit balance. However, this is not to say that the model could not be improved. Far more modelling techniques could have been implemented and compared to the 5 we did including deep neural network and random forest models. Still, we have effectively created a strong predictive model for the data at hand.